D-NIX 5.3
Systemadministration

Version C 91-05-05

089-9821-10

Diab Data AB
Box 2029
183 02 TÄBY
☎ 08-638 94 00

DIAB DATA

1. Översikt

2. Systemhandhavande

3. Laddningsprogrammet

4. Filer och hjälpprogram

5. Systemvård

6. Användaromgivningen

7. Felhantering

8. Terminalhantering

9. Skrivarhantering

11. Säkerhetskopiering

12. Cron

13. UUCP Nätverkssystem

15. Yttre enheter

1

1. Översikt

1. Översikt

1.1. Handboksöversikt

Denna handbok vänder sig till systemadministratören av datorsystemet.

Kapitel 2 beskriver start och stopp av maskinen. Här beskrivs även hur ett filsystem skapas, hur disketter formateras m m.

Kapitel 3 beskriver de program som kan användas från laddningsprogrammet. Med dessa kan ett flertal systemparametrar sättas och filsystemet kan kontrolleras.

Kapitel 4 beskriveri kortfattat intressanta filer och hjälpprogram. De är användbara om du vill ändra systemparametrar och mycket mer.

Kapitel 5 beskriver noggrant hur du kontrollerar filsystemet, samt beskrivning av programmet *mkcfig* som du använder om du vill ändra på konfigurationsparametrar i operativsystemet.

Kapitel 6 beskriver hur användarens miljö kan byggas upp. Här beskrivs vilka variabler och parametrar som behövs för att användaren skall få en effektiv miljö att arbeta i.

Kapitel 7 beskriver kortfattat felkoder, felmeddelanden och en del hjälpmedel för att lösa enkla fel på systemet.

Kapitel 8 bekriver hur terminalportarna ställs in och vilka parametrar som skall användas. Kapitlet innehåller även en beskrivning av hur portarna skall sättas upp för att användas med modem.

Kapitel 9 beskriver ingående hur skrivarsystemet skall sättas upp. Kapitlet beskriver alla hjälpprogram och ger även tips om hur utskrifter kan ske över nätverk m.m.

Kapitel 11 går igenom hur säkerhetskopieringen bör ske i ett system. Det är mycket viktigt att rutinerna för säkerhetskopiering läggs upp på ett tillfredställande sätt redan från början.

Kapitel 12 beskriver *cron* och hur programmet kan användas för att starta program vid bestämda tidpunkter.

Kapitel 13 beskriver *uucp* och dess hjälpprogram. Med dessa kan ett effektivt kommunikationssystem skapas mellan olika datorer. Kommunikationen kan ske över Ethernet, modem och mycket mer.

Kapitel 15 går igenom hur yttre enheter ska anslutas till datorn. Här finns dessutom tabeller över enhetsnamn och enhetsnummer samt andra detaljer som skiljer mellan olika datorversioner.

Övriga kapitel är reserverade för tillägg till handboken. I den mån de blir aktuella kommer de att distribueras till alla system.

1.1.1 Teckenuppsättningar

I denna handbok har vi valt att följa den internationella teckenrepresentationen vid beskrivningen av tecken. Det innebär att om man använder sig av ett svenskt tangentbord med svensk 7-bitars teckenuppsättning måste vissa tecken ersättas med andra. I listan nedan visas tecknen som används i handboken både i normal text och i datortexten i exemplen.

Tecken:		Ersätts med svenska tecknen:	
}	}	å	
]	]	Å	
{	{	ä	
[	[	Ä	
|	|	ö	
\	\	Ö	
~	~	ü	
^	^	Ü	
`	`	é	(ASCII 96 decimalt=grav accent)*)
@	@	É	
$	$	¤	
#	#	§	

*) Grav accent används i vissa fall i shellprocedurer och är lätt att förväxla med vanlig apostrof, i normal text (') och i datortext (`).

2

2. Systemhandhavande

2. Systemhandhavande

I vissa datortyper hanteras start och stopp av datorn annorlunda än i denna beskrivning. Information om detta medföljer då systemet.

2.1. Automatisk start

Systemstarten kan ske i två olika lägen, automatisk och manuell, beroende på läget hos nyckelbrytaren på frontpanelen. Normalt startas systemet i automatiskt läge. Skillnaden mellan de två lägena är att i det manuella läget startar operatören steg för steg maskinen, medan i det automatiska läget utför systemet hela systemstarten.

Starten i AUTO-läget sker till den bootnivå (0,1,2 eller 3) som finns inlagd i ett NVRAM (ett fast programmerbart minne) i systemet. Denna bootnivå kan ändras, liksom definitionen av den enhet från vilken start skall ske. Se kapitel 3, kommandot */sas/bootpar*.

Starten förutsätter att systemet fullständig stängts av på normalt sätt eller tagits ner fullständigt med kommandot */etc/shutdown* **-k**.

- Systemet är inkopplat, med nyckeln i läget OFF och huvudbrytaren frånslagen.
- Slå till huvudbrytaren på systemenheten.

Normalt startas systemet genom att vrida nyckeln från läget OFF till AUTO, varefter användaren loggar in med namn och lösenord. Automatisk start sker även om nyckeln är i AUTO då huvudbrytaren slås på.

- Vänta! Systemet laddas från Winchesterenheten.
- På frontpanelen tänds alla fyra lamporna. Efter en kort stund släcks lampan ERROR.

När systemet laddas in skrivs meddelanden ut på huvudkonsolen, vilket kan se ut på följande sätt när allt är klart:

```
OS: INFO: D-NIX X.X Virtual Ver X.X -
                                System xxxk User xxxxk Swap x.xM
Diab Data AB MP D-NIX X.X Y.Y Virtual
Wed Apr 3 15:16:45 MET 1991
CRON started
LP scheduler started

Console login: _
```

Skulle det uppstå något problem vid den automatiska systemstarten kan det bero på något av följande:

- Huvudkonsolen är inte korrekt ansluten eller inställd. Som standard används överföringshastighet 9600 baud. 8 databitar, ingen paritet och 1 stoppbit krävs. Eventuellt har kommunikationsparametrarna för huvudkonsolen ändrats i filen **/etc/gettydefs**. I detta fall kan alltid systemstart ske i manuellt läge upp till ett användarläge med

standardparametrarna enligt ovan, varefter filen **/etc/gettydefs** kan ändras.

- Systemet har inte stängts av korrekt vid tidigare systemstart. Vid ny start utför operativsystemet en kontroll av filsystemet för att rätta till eventuella felaktigheter. Olika meddelanden visas då på huvudkonsolen vid systemstarten. Se kommandot *fsck*.
- Systemprogrammen på Winchesterenheten är felaktiga. Starta systemet i Manuellt läge.

Om lampan ERROR inte slocknar efter en stund, betyder det kanske att automatiska diagnostikprogrammet har hittat fel i maskinvaran. En servicetekniker bör tillkallas.

När nyckeln lämnas i AUTO-läget efter en systemavstängning, och huvudströmbrytaren ej slagits av, påbörjas automatiskt ny start ca 10 se-- kunder efter att systemet helt stängts av. Att avstängningen är klar visas med ledtexten:

```
OS INFO: System halted
```

När nyckeln på datorn ställs i läge AUTO, skyddas systemet och inloggning krävs alltid. Detta förutsätter dock att startparametern 'boot level' i NVRAM är bootnivå 3. Nyckeln kan i detta läge tas ut.

2.2. Manuell start

2.2.1 Allmänt

Normalt startas systemet helt automatiskt. Det finns emellertid tillfällen då man behöver starta systemet manuellt t ex

- vid formatering av minnesmedium.
- vid ändring av programladdningsparametrar (bootstrap),
- vid problem med automatisk uppstart.

Vid manuell start kan programladdningen styras stegvis. Starten sker på följande sätt och förutsätter att systemet är helt avstängt. Manuell start kan även ske efter fullständig nedtagning av systemet med kommandot */etc/shutdown* **-k** utan att spänningen slagits av.

Observera! Huvudkonsolen skall vara inkopplad med standard kommunikationsparametrar.

Manuell start förutsätter att nyckeln på datorn står i läget OFF. För att starta vrids nyckeln först till RESET och därefter direkt tillbaka till OFF. Spänningen slås då på automatiskt och starten sker i manuellt läge.

Manuell start sker genom fyra bootnivåer:

Bootnivå 0: Bootstrap PROM styr inladdningen av ett laddningsprogram från skivminne eller flexskiva. En kort text **M680x0 prom** skrivs ut på huvudkonsolen och operatören svarar med **RETURN** eller namnet på det minnesmedium där laddningsprogrammet finns lagrat. Efter inladdning anger operatören namnet på den enhet där filsystemet finns.

Bootnivå 1: Laddningsprogrammet (bootstrapsystemet) är i drift. Detta är ett enkelt skivoperativsystem och vissa, korta program kan användas på denna nivå. Prompten på denna nivå är **>**. Med kommandot *dnix* fortsätter startsekvensen till nästa nivå genom laddning och start av operativsystemet (D-NIX).

Bootnivå 2: Enanvändarläge. Super-user är ensam i systemet. Prompten är **#**. Kommandot *telinit 2* ger start till den sista nivån.

Bootnivå 3: Fleranvändarmod. Systemet kräver inloggning. Prompten efter inloggning är **$** för normalanvändare och **#** för super-user. Exakt vad som sker vid övergång till bootnivå 3 (fleranvändarmod) styrs av innehållet i filen **/etc/inittab**.

2.2.2 Bootnivå 0

Direkt efter RESET eller spänningstillslag utförs en hårdvarutest och kontroll av att erforderlig minneskapacitet finns tillgänglig. När detta är genomfört svarar systemet

```
M680XX prom      (XX = maskinberoende)
```

och avvaktar operatörens kommando vid huvudterminalen.

Operatören skall nu ange den enhet varifrån laddningsprogrammet skall läsas in.

Om operatören svarar med enbart RETURN tas laddningsprogrammet från den standardenhet som finns angiven i datorns NVRAM. Normalt värde är det interna winchesterminnet. I detta fall läses laddningsprogrammet in och startas direkt i bootnivå 1 enligt nästa avsnitt.

Operatören kan istället uttryckligen ange enheten varifrån programmet skall läsas in. Enhetsnamnet avslutas med RETURN.

Enhetsnamnet på bootnivå 0 är samma som i laddningsprogrammet. Första delen av namnet anger enhetstypen och kanalnumret inom parentesen är ibland men inte alltid lika med motsvarande sekundära enhetsnummer (minor). Tabellen nedan listar standardnamnen.

Datormodell	BOOT-disk	BOOT-diskette
DS90-11	si(0,0)	mf(64,0)
DS90-10,20,21,30,31,DS101	si(16,0)	mf(8,0)
DIAB1320, DIAB1420	si(16,0)	mf(10,0)
DS90-41	si(16,0)	mf(0,0)
DIAB1130	si(0,0)	mf(0,0)
DIAB2420, DIAB2430	si(16,0)	mf(40,0)

För olika datortyper listas gällande enhetsnamn i kapitel 15.9.

Efter att enhetsnamnet eller enbart RETURN givits, läses laddningsprogrammet in från den angivna enheten och startas. Programmet frågar först efter den enhet som skall användas som "root"-filsystem. Denna fråga ges ej om enbart RETURN angivits.

```
System boot
Root device "xx(nn,nn)" ? _
```

Här anger operatören samma eller annan enhet enligt listan ovan, varefter laddningsprogrammet startar. Om laddningsprogrammet lästs in från disketten men winchesterenheten angivits som root-device bör nu disketten tas ut.

2.2.3 Bootnivå 1 Laddningsprogrammet

Laddningsprogrammet (Stand-Alone-Systemet) är i drift. Det är ett enkelt operativsystem och vissa, enkla program kan köras. Prompten > visar att kommandon kan ges.

I det fall där systemet inte har stängts av korrekt ges en varningstext om att filsystemet inte är korrekt **Warning, file system not clean**. Filsystemet skall, om detta inträffar, korrigeras med kommandot *fsck*, som utför test av filsystemet på det fasta skivminnet i massminnesenheten. Kommandot *fsck* får endast användas i laddningsprogrammet (bootnivå 1), om den enhet som är root-filsystem ska testas. Kommandot för test av rootfilsystemet ges på följande sätt:

```
>/etc/fsck -rr enhetsnamn
```

där enhetsnamn är samma som på bootnivå 0, t ex si(X,0) eller mf(X,0) med olika värden för X beroende på datormodell enligt föregående sida.

I biblioteket **/sas** på den boot-diskett som levereras med systemet finns några program som endast kan användas på denna nivå, bl a de nedan nämnda. Där finns även vissa testprogram som måste köras på denna nivå. Namnet **/sas** är en förkortning av "stand alone system", vilket innebär att endast ett program i taget kan köras och med en enda användare.

Exempel:

/sas/bootpar	Ändrar laddningsparametrar i NVRAM. Se även */etc/bootpar*.
/etc/fsck	Används på denna nivå för att testa och korrigera rootfilsystem.

Kapitel 3 innehåller en detaljerad beskrivning över laddningsprogrammet.

Om systemet är i bootnivå 0 eller 1 och ingen skrivning/läsning sker på skivminne eller diskett, kan systemet återstartas genom att nyckeln förs till läget RESET och tillbaka till OFF eller AUTO. Med samma villkor kan även spänningen slås av med huvudbrytaren på datorsystemet. Efter att operativsystemet startats enligt nedan, måste däremot en korrekt avstängning ske.

Operativsystemet laddas in och startas med kommandot *dnix*.

```
>dnix
```

Alternativt kan en valfri sträng anges efter kommandot, varvid standard systemparametrar används istället för de som angivits av kommandot */etc/mkcfig*. Operativsystemet startas då med *dnix xxx*. Den angivna strängen kan bestå av vilka tecken som helst.

Vid systemstart testar operativsystemet minnet och skriver sedan ett systemmeddelande med revisionsnummer och den interna minneskapaciteten i datorn.

```
OS:INFO: D-NIX X.X Virtual Ver X.X -
         System xxxk  User xxxxk  Swap x.xM
```

Därefter startar en initieringsprocess från filen **/etc/init** som initierar operativsystemet. Denna initieringsprocess *init* är process 1 och stannar i systemet som 'urfadern' till alla övriga processer.

2.2.4 Bootnivå 2 Enanvändarläge (systemnivå s)

I bootnivå 2 är operativsystemet i enanvändarläge. Enbart huvudterminalen kan användas och användaren är privilegierad (super-user) och behöver alltså inte logga in. Prompten # visar att kommandon kan ges och att alla kommandon kan användas. Kommandona i filerna **/etc/profile** och **/.profile** utförs innan prompten visas. Man kan även logga in som normalanvändare med kommandot *su* utan att lämna enanvändarnivån. Med *su* behöver super-usern inte känna till användarens lösenord utan kan logga in direkt. I bootnivå 2 kan alltså flera processer startas men med endast en användare i systemet.

I enanvändarläge används alltid enheten **/dev/syscon** som huvudkonsol och terminalparametrarna sätts upp enligt den andra gruppen av terminalparametrar för **console** i filen **/etc/gettydefs**. Filen **/etc/inittab** används inte i enanvändarläget. Detta läge kallas för systemnivå **s**.

Övergång till fleranvändarläge sker normalt med kommandot:

```
telinit 2
```

Om systemet tidigare varit i fleranvändarläge, kan enanvändarläget avslutas genom utloggning (CTRL-D) som svar på prompten #. I detta fall ges en fråga på huvudterminalen efter en ny systemnivå (run-level):

```
ENTER RUN-LEVEL (0-6,s or S):_              (Svara: 2)
```

Denna fråga ska besvaras med **2**, för att starta i fleranvändarmod.

Kommandorutinen i enanvändarläge avslutas och systemet övergår till fleranvändarläge.

Notera att en återgång till enanvändarnivån kan ske från den normala fleranvändarnivån (bootnivå 3) med kommandot */etc/shutdown*.

Observera! Med enbart BOOT-disketten som root-filsystem, kan fleranvändarläge inte användas, eftersom alla systemfiler som behövs för detta inte ryms på den BOOT-diskett som medföljer systemet.

2.2.5 Bootnivå 3 Fleranvändarläge

Vid övergång till fleranvändarläge läser *init*-processen filen **/etc/inittab** och startar upp de kommandon som anges där. Kommandon exekveras bl a för att rensa bort gamla temporärfiler och starta vissa systemprocesser, t ex nätverksprocesser. Här startas bland annat en kommandoavkodare som utför kommandon i filerna **/etc/brc** och **/etc/rc**. En login process (*/etc/getty*) startas för varje ansluten terminal som skall vara aktiv.

Alla numeriska systemnivåer, 0..6, inom operativsystemet är fleranvändarlägen, men som standard används systemnivå 2, enligt de definitioner som ligger i filen **/etc/inittab**. Systemnivå 0 är reserverad. I **/etc/inittab** får inga processer vara aktiva på systemnivå 0.

I filerna **/etc/brc** och **/etc/rc** finns systemkommandon som alltid utförs vid övergång till fleranvändarläge. Bl a ger dessa ett systemmeddelande till huvudterminalen, t ex:

```
Diab Data AB  D-NIX 5.X Y.Y Virtual
```

Bootnivå 3 är den nivå där systemet är avsett att normalt arbeta dvs som ett fleranvändarsystem. Fleranvändarläget kräver inloggning.

Normalt sker automatisk start direkt till denna bootnivå som normalt anges i NVRAM (parametern 'boot level'). Vid automatisk start kan nyckeln tas ut i läge AUTO, d v s rakt upp, så att systemet alltid startar i denna bootnivå.

I samband med inloggningen kan ytterligare en systemkonfiguration utföras. Före inloggning sätts först baudrate m m, systemets **nodnamn** visas på terminalen, följt av en systemtext ur filen **/etc/issue.** Frågetexten vid inloggning tas ur filen **/etc/gettydefs** och under inmatning av användarnamnet kan vissa terminalparametrar ändras automatiskt.

Efter godkänd inloggning bearbetas först automatiskt kommandon i **/etc/profile** följt av kommandon i filen **.profile** i användarens eget bibliotek. Via **/etc/profile** visas bl a eventuellt systemmeddelande från filen **/etc/motd.** Därefter visas prompten **$** eller **#** och systemet väntar på operatörens kommandon. Vid inloggning som super-user (root) visas ej **/etc/motd.**

Det är även möjligt att i förväg bestämma att en användare automatiskt enbart skall ha tillgång till ett enda program. Detta anges i så fall antingen som ett exec-kommando sist i filen **.profile** eller i filen **/etc/passwd** som sista parameter. Detta program startas då direkt istället för att någon prompt ges.

Under bootnivå 2 och 3 refereras skivminne, diskettenhet och kassettstreamer med enhetsnamnen som definierats i biblioteket **/dev** i rootfilsystemet. Motsvarande primära och sekundära enhetsnummer (major och minor) är länkade till enhetsnamnen med kommandot mknod.

Enhetsnumren är olika för olika datortyper. Vanligen har dock följande massminnesenheter samma namn i de flesta datorsystemen:

Första internal skivminnet	/dev/si0	
Diskettenheten	/dev/mf0	(3.5 tum eller 5 1/4 tum)
2: diskettenhet	/dev/mf1	(3.5 tum eller 5 1/4 tum)
1/4 tums streamer	/dev/st0	

En lista över enhets nummer och namn finns i kapitel 15.9.

2.3. Konfiguration av systemet

I detta avsnitt ges en kortfattad information om de filer och kommandon som kan behöva användas för att anpassa systemet till användarens behov. Ett litet antal kommandon beskrivs med utgångspunkt från hur användaren ser dem. En sammanfattande beskrivning ges i ett delavsnitt över vilka faktorer som bör beaktas vid inkoppling av en ny terminal eller skrivare.

Alla exempel förutsätter att den systemansvarige befinner sig i rootbiblioteket och är inloggad som super-user (root) om inget annat anges.

Observera! Innan du börjar ändra i känsliga systemfiler bör du försäkra dig om följande:

- Lär dig editorn *dmacs* ordentligt och se till att shellparametern **TERM** motsvarar den använda terminalen.
- Gör kopior av systemfilerna och gör ändringarna i dessa. Kopiera dessutom originalfilen till en backupfil. Efter verifiering att den rätta ändringen gjorts, kopieras den nya filen till originalfilen. Vid kopiering används kommandot *cp* enligt nedan, varvid filägare och accesstillstånd normalt återställs till originalfilens. Kommandona *chown* eller *chmod* kan annars behöva användas för att säkerställa att systemfilerna verkligen har rätta ägaren och åtkomsttillstånden. För ytterligare information om kommandon, se **Referenshandboken**.

Exempel:

Vid ändring av **/etc/inittab** görs först en kopia som arbetsfil och en säkerhetskopia görs av originalet.

```
cp /etc/inittab /etc/inittab.ny
cp /etc/inittab /etc/inittab.org
```

Backup-kopia av originalet.

Gör ändringar i **/etc/inittab.ny** med *dmacs* och avsluta *dmacs* med **ESC Z** för att spara ändringarna.

```
dmacs /etc/inittab.ny
```

Ändringarna kan verifieras med:

```
cat /etc/inittab.ny
```

Den ändrade filen kopieras till en ny originalfil:.

```
cp /etc/inittab.ny /etc/inittab
```

2.3.1 Skapa och ändra systemfiler med dmacs

Med editorn *dmacs* kan en fils innehåll ändras under förutsättning att användaren har läs- och skrivprivilegier till filen eller är super-user.

Exempel på ändring av filen **/etc/motd** (message of today), den fil vars innehåll presenteras för alla som loggar in på systemet. Editorn *dmacs* beskrivs i **Användarhandboken**.

Definiera parametern **TERM**, om den inte redan definierats, för den använda terminaltypen. Skriv t ex för 24-raders Facit Twist följande. Inga mellanslag får användas kring likhetstecknet.

```
TERM=twist
export TERM
```

Skriv:

```
dmacs /etc/motd
```

På skärmen visas nu innehållet i **/etc/motd**. T ex:

```
Välkommen till D-NIX
86-03-10 Vi har nu fått igång den nya laserskrivaren.
Den kan användas som skrivare lp2.
Synpunkter kan lämnas med mail till sysop.
```

Vi vill nu ändra skrivarens namn till laser. Placera markören på raden som ska ändras och förflytta markören till första positionen i ordet lp2.

- Tryck på **CTRL** och **D** samtidigt. Upprepa tryckningen tills hela ordet är borttaget.
- Skriv in ordet laser.
- Tryck på **ESC** följt av **Z**. Filen sparas och samtidigt avslutas editorn och prompten **#** skrivs ut igen.

Filen är nu ändrad, vilket kontrolleras med kommandot *cat*.

För att avsluta *dmacs* utan att ändra filen ges **CTRL-X CTRL-C svar Y**.

2.3.2 Lägga till användare

I systemet finns det vid systemstart bara en användare definierad, super-user. Men var och en av de personer som ska vara användare av systemet måste definieras. I operativsystemet definieras en användare med ett kommando *mkuser*. Kommandot är uppbyggt som ett interaktivt program där ett antal frågor ska besvaras.

För att *mkuser* skall "prata" svenska måste shell-variabeln **LANGUAGE** sättas innan kommandot ges:

```
LANGUAGE=swedish
export LANGUAGE
/etc/mkuser
```

För att ta bort en användare används kommandot */etc/rmuser*.

Dialogen i mkuser:

Användarnamn (Username)

Det namn som den nye användaren kommer att använda som sitt loginnamn. Om namnet redan är upptaget upprepas frågan. Undvik att ge användarnamn med enbart stora bokstäver och undvik tecknen _, $, &, /, # och @. Dessa kombinationer behandlas speciellt av startrutinen för inloggning (*/etc/getty*).

Användarid (User-id)

Den nummeridentifikation som användaren ska använda, ett heltal större än ett. För vanliga användare rekommenderas ett nummer större än 100. Om numret redan är upptaget upprepas frågan. Första lediga nummer visas inom parentes() och fås om enbart **RETURN** används.

Lösenord (Password)

Ange det lösenord som användaren vill använda vid inloggningen.

Gruppnamn (Group-name)

Namnet på den grupp användaren ska tillhöra. Det är möjligt att dela användarna i en eller flera grupper och för varje användare ange vilken grupp den tillhör. För varje grupp kan ett lösenord definieras. Om användaren inte skall tillhöra någon specifik grupp, ges enbart **RETURN**, varvid gruppnamnet sätts till övriga (other).

Grupp-id (Group-id)

Gruppens nummeridentifikation, ett heltal större än ett. Systemet skriver ut grupp-id om ett redan befintligt gruppnamn angivits. Om numret redan är upptaget upprepas frågan. Första lediga nummer fås om enbart **RETURN** används.

Extra användarbeskrivning (Extra user-description)

Godtycklig sträng, får inte innehålla kolon (:).

Startbibliotek (Login-directory)

Anger vilket bibliotek som ska vara användarens hembibliotek dvs det bibliotek som användaren hamnar i efter inloggningen. Existerar det redan måste svaret verifieras och ingen egen **.profile** skapas. Om bara **RETURN** ges blir startbiblioteket **/usr/användarnamn** (visas inom paranteser), där användarnamn är detsamma som ovan.

Startprogram (Start-program)

Här skall anges vilket startprogram som skall användas, om inget anges används ett standard startprogram, */bin/sh*. Existerar inte det angivna programmet måste svaret verifieras.

Terminal-typ (vt100)

Här anges vilken typ av terminal som användaren kommer att använda. Kommandon för att definiera shellvariabeln **TERM** till denna typ kommer automatiskt att läggas in i filen **.profile** som skapas i användarens hembibliotek. En standardterminal ges inom parentes. Om den används, tryck **RETURN**.

När alla frågor är besvarade kommer systemet att lägga upp användaren, vilket innebär att filerna **/etc/passwd** och **/etc/group** uppdateras. Dessutom skapas ett hembibliotek och en uppstartsfil **.profile** som placeras i användarens hembibliotek. Uppstartsfilen exekveras varje gång som användaren loggar in. I filen finns information om vilken terminaltyp som används (**TERM**) och den kan också innehålla funktioner på en del tangenter. I filen **/etc/passwd** finns följande information om varje användare: login-namn, krypterat lösenord, användaridentitet, gruppidentitet, kommentarfält, hembibliotek och eventuellt startbibliotek. Tillstånd och tidsgränser för ändring av lösenordet kan senare manuellt läggas till. I filen **/etc/group** sparas följande information: gruppnamn, lösenord, gruppidentifikation samt vilka personer som tillhör varje grupp.

Exempel på mkuser

```
# /etc/mkuser

Användar-namn: oskar
Användar-id (114):
Lösenord: (Lösenord)
Repetera lösenord: (lösenord)
Grupp-namn (other): other
Grupp-id: 1
Extra användar-beskrivning: Oskar Andersson
Start-bibliotek (usr/oskar):
Start-program (/bin/sh):
Terminal-typ (vt100):
```

Användaren är därmed definierad i systemet.

2.3.3 Tillägg/ändring av lösenord - passwd

Varje användare har ett lösenord kopplat till sitt hembibliotek för att skydda det från obehörig användning. Eftersom systemet levereras med endast en användare upplagd, super-user, som inte har något lösenord upplagt, bör den systemansvarige lägga upp ett sådant redan vid första kontakten med systemet. Uppläggningen sker med kommandot *passwd*.

När en användare definieras är det möjligt för den systemansvarige att lägga upp ett lösenord för användaren. Men även användaren själv måste lätt kunna ändra sitt lösenord om det skulle visa sig att andra använ-

dare fått tillgång till lösenordet. Ändring av lösenordet utförs även den med kommandot *passwd*.

Ett lösenord ska vara minst sex tecken långt och bestå av minst två bokstäver samt ett numeriskt tecken eller specialtecken. Används ett lösen ord med fler än åtta tecken kommer endast de första åtta att vara av betydelse. Systemet frågar efter det nya lösenordet två gånger för att försäkra sig om att det är korrekt stavat. Skulle systemet inte acceptera lösenordet beror det på att det inte stavats exakt lika vid de två tillfällena. Gör ett nytt försök! Tänk på att stora och små bokstäver tolkas som olika tecken.

Tillägg/Ändring av lösenord

```
# passwd

passwd: INFO: Changing password for: kalle
Old password:
New password:
Re-enter new password:
```

Observera! Lösenordet skrivs inte ut på skärmen.

En användare kan tvingas ange ett nytt lösenord efter en maximal tid i veckor och kan även hindras att själv ändra sitt lösenord. Detta styrs av koder i fältet med det krypterade lösenordet i **/etc/passwd** och kan endast definieras av den systemansvarige genom direkt editering i filen. Se **/etc/passwd** i kapitel 4 för detaljer.

För användare som ska kunna logga in till systemet via en extern modemport, finns möjlighet att definiera ett sekundärt lösenord (dial-up password). Se **/etc/d_passwd** och **/etc/dialups** i kapitel 4 samt beskrivningen av kommandot *login* i **Referenshandboken**.

2.3.4 Vilka användare är upplagda i systemet

När en eller flera användare har lagts upp i systemet kan det vara lämpligt att kontrollera att de är upplagda på ett riktigt sätt. Enklast kontrolleras detta genom att informationen i filen **/etc/passwd** skrivs ut. För detta ändamål kan kommandot *cat* användas, kommandot skriver ut filens innehåll på skärmen. På grund av begränsad radlängd i handboken delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

- Skriv *cat /etc/passwd*. Tryck **RETURN**.

På skärmen kan det se ut på följande sätt (observera att de 13 tecknen i första fältet för root representerar en tänkbar kryptering av lösenordet för root):

```
root:sk9XbaSmp0xKm:0:3:0000-Admin(0000):/:
daemon:*noway*:1:12:0000-Admin(0000):/:
bin:*noway*:2:2:0000-Admin(0000):/bin:
sys:*noway*:3:3:0000-Admin(0000):/usr/src:
adm:*noway*:4:4:0000-Admin(0000):/usr/adm:
uucp:*noway*:5:1:0000-uucp(0000):/usr/lib/uucp:
nuucp::6:1:0000-uucp(0000):/usr/spool/uucppublic:
          /usr/lib/uucp/uucico
rje:*noway:68:8:0000-rje(0000):/usr/rje:
shqer:*noway:69:8:0000-rje(0000):/usr/rje:
```

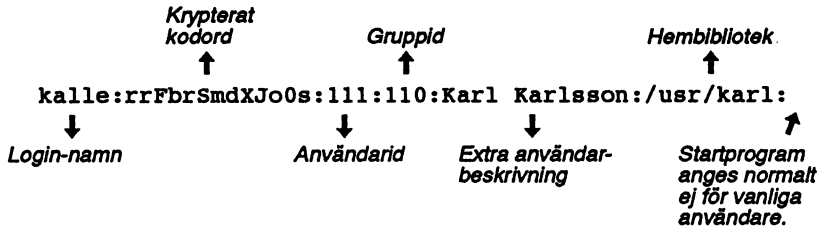
```
lp:*noway:71:2:0000-lp(0000):/usr/spool/lp:
jag:3Uuv4MJme0Khk:110:110:Sven Svensson:/usr/sven:
kalle:rrFbrSmdXJo0s:111:110:Karl Karlsson:/usr/karl:
```

Varje rad består av ett antal olika fält åtskilda med ':'. Vilka fält som finns i filen password visas i nedanstående figur.

Om all information angående användaren/användarna är korrekt är det bara att fortsätta med nästa kommando, skulle filen däremot innehålla något fel eller om något fält ska ändras kan detta göras med editorn *dmacs*. Bl a kan speciella koder läggas in som tidsbegränsar användandet av ett lösenord.

2.3.5 Ta bort användare - rmuser

När en användare inte längre skall finnas kvar i systemet används kommandot *rmuser*, filerna **/etc/passwd** och **/etc/group** uppdateras och an-



vändarens hembibliotek tas bort. Hembiblioteket kan bara tas bort om det är tomt, detta kan kringgås om optionen **-r** anges, för vidare information se beskrivning av *rmuser* i **Referenshandboken.**

För att *rmuser* skall skriva på svenska måste följande shell-variabel sättas

```
LANGUAGE=swedish
export LANGUAGE
```

Borttagning av användare

```
/etc/rmuser
Användar-namn:
```

2.4. Hantering av massminnen

2.4.1 Formatering av disketter på fleranvändarnivå - format

Formatering av disketter sker med programmet **/etc/format**. Det kan användas i både en- och fleranvändarmod när man är inloggad i systemet. Programmet startas enligt nedan visade exempel, där enhetsnamnet /dev/mf0 används för standardenheten. Om systemet har flera diskettstationer används vanligen /dev/mf1 för den andra.

Formatering av 720 kbyte disketter (3.5 eller 5 1/4 tum)

```
/etc/format  /dev/mf0
```

Formatering av 1.2 Mbyte high density disketter (5 1/4 tum)

```
/etc/format  -r 15 /dev/mf0
```

Formatering av 1.44 Mbyte disketter (3.5 tum)

```
/etc/format  -r 18 /dev/mf0
```

Om inga tillvalsparametrar ges gäller 720 kbyte som standard. Annars anges vanligen enbart antal sektorer/spår med optionen -r enligt ovan.

Val av parametrar kan göras interaktivt om programmet startas utan att något enhetsnamn anges. Nedan är ett exempel på interaktiv användning, där standardvärdena (för 720 kbyte) visas på skärmen. Dialogen kan se ut på följande sätt:

```
/etc/format
Disk format program
Enter device: /dev/mf0
Format device mf, unit 0 - OK? y

Selected mini floppy parameters
Number heads            2
Number cylinders       80
Number sectors/track    9
Sector size           512
Interleave factor       1
Parameters OK? y
Formatting disk...
Disk formatted
```

På frågan **Enter device:** anger du den enhet som skall formateras, i detta fall **/dev/mf0**. Svarar du med **y** på frågan **Parameters OK?** startas formateringen. Svarar du med **n** kan dessförinnan parametrarna ändras interaktivt.

För att formatera en high density 1.2 Mbytes diskett måste antal sektorer/spår ändras till **15** genom att ange det ändrade parametervärdet vid frågan **Number sectors/track**

För mer information om formatet hos disketter se avsnitt 15 i denna handbok.

2.4.2 Initiering av filsystem på diskett - mkfs

Innan en tom diskett eller ett tomt skivminne kan användas som ett filsystem måste ett filsystem skapas (initieras) på enheten med */etc/mkfs*. Samtidigt utförs en test av hela disketten/skivminnet. Initieringen förutsätter att formatering är gjord. Den interna winchesterenheten är alltid färdiginitierad vid leverans och innehåller alltid ett grundpaket av operativsystemet.

Initieringen utförs med kommandot */etc/mkfs*. Disketten rensas helt och ett nytt filsystem skapas med ett tomt root-bibliotek. Det är först efter det att ett filsystem är initierat som det är möjligt att lägga till, dvs använda kommandot *mount*, och lagra filer i filsystemet.

Initiering av filsystem på diskett

```
/etc/mkfs /dev/mf0
```

På skärmen kommer information att skrivas ut under initieringen. När initieringen är slutförd skrivs prompten # ut.

Volymstorleken känns av automatiskt av *mkfs*, men kan liksom blockstorleken (standardvärdet = 2 kb) väljas manuellt. För detaljer, se kommandot *mkfs* i **Referenshandboken**.

Observera! Disketter som enbart skall användas för att lagra backupfiler med kommandot *tar* behöver inte initieras. De måste däremot vara formaterade.

2.4.3 Lägga till/Ta bort filsystem - mount, umount

Normalt finns alla systemprogram och systemfiler på winchesterenheten men i en del fall måste andra medier kunna läsas och skrivas t ex en diskett. Det nya filsystemet kopplas till ett bibliotek i gamla filsystemet med kommandot *mount*. Biblioteket måste existera. Bibliotekets privilegier kommer att ändras till det nya filsystemets privilegier i och med att *mount* har utförts. Som standard blir alla enheter med mountade filsystem skyddade mot direkt (rå) skrivning.

Super-userprivilegier krävs för att koppla in och bort ett filsystem.

Det nya filsystemet måste existera på den enhet eller fil som ska läggas till. Jämför kommandot *mkfs*.

Filsystemet kopplas bort från root-filsystemet med kommandot *umount*, varvid filsystemet uppdateras och kopplas bort korrekt.

Observera! *umount*-kommandot måste användas innan disketten tas ut.

Tillägg av ett filsystem som finns på diskett

Disketten sätts in i diskettenheten på systemet. Användaren måste ha super-userprivilegier och disketten måste vara initierad.

```
/etc/mount /dev/mf0 /mf0
```

/mf0 är i detta fall namnet på kopplingsbiblioteket. Detta är ett vanligt bibliotek och ska vara tomt. Det är via detta bibliotek som systemet har tillgång till nya filsystemet.

När all hantering med disketten är klar måste kommandot *umount* användas innan disketten får tas ut ur diskettenheten.

Tillägg av ett filsystem som finns på yttre skivminne

Ett eller flera yttre winchesterskivminne kan kopplas till root-filsystemet på samma sätt som en diskett men med andra mount-parametrar. Förutsättningen är att ett filsystem har skapats med *mkfs* på det externa skivminnet.

Normalt kopplas yttre skivminnen in automatiskt vid systemets uppstart via kommandon i **/etc/brc**. För detaljer hänvisas till kapitel 15 samt till dokumentation medföljande det yttre skivminnet.

Exempel på hur ett filsystem kopplas bort

Innan en diskett avlägsnas ur diskettenheten måste systemet stänga det externa filsystemet korrekt, vilket innebär att eventuella filbuffertar skrivs ner på disketten och filsystemet kopplas bort från root-filsystemet.

Samma sak gäller om externa större skivminnen används.

För att ta bort en "mountad" diskett ur root-filsystemet ges kommandot:

```
/etc/umount /dev/mf0
```

/dev/mf0 är den enhet där disketten är placerad. Efter att *umount* kommandot använts kan disketten tas ur diskettenheten.

2.4.4 Initiering av fast skivminne

Följande beskrivning förklarar vad som behöver göras vid initieringen av ett nytt internt skivminne som rootsystem från distributionsdisketterna (BOOT-disketten).

Observera! Vid leverans av ett system är det fasta interna skivminnet alltid färdiginstallerat med systemprogram och initieringen redan gjord.

Observera! Denna beskrivning förutsätter att skivminnet är formaterat från leverantören. Formatera aldrig ett skivminne utan att ta kontakt med din leverantör. Normalt förstörs inte formateringen om något fel skulle uppstå. Det är endast när läs/skrivhuvuden eller någon annan maskinvara gått sönder som en formatering kan behövas. I dessa fall krävs att en servicetekniker utför reparationen.

För att skapa ett nytt filsystem på det interna fasta skivminnet, ska systemet först startas i manuell mod upp till enanvändarmod (Bootnivå 2)

med BOOT-disketten som root-device. Disketten som ska användas är märkt med texten "BOOT" och datortypen enligt följande exempel.

Exempel på märkning av BOOT-disketten:

```
D-NIX X.X            xxx-xxxx-xx/ n(1)
Version X.xx
DIABxxxx / BOOT
mount format / 1.2 Mb
Copyright Diab Data AB
```

Se kapitel 2.2 hur uppstarten sker.

Bildskärmen bör efter uppstart se ut så här:

```
M680XX prom                          (XX = maskinberoende)
mf(x,0)                               (X = maskinberoende)
System boot
Root device "xx(nn,nn)" ? mf(x,0)     (X = maskinberoende)
> dnix
OS:INFO: D-NIX X.X Virtual Ver X.X -
         System xxxk  User xxxxk  Swap x.xM
#
```

Stegen i det följande är:

1. Skapa ett filsystem på det fasta skivminnet med */etc/mkfs*.

   Exempel:

   ```
   /etc/mkfs  /dev/si0
   ```

   mkfs-kommandot känner själv av volymstorleken och använder som standard blockstorlek 2 kbyte.

   Filsystemet på skivminnet är nu initierat men innehåller ännu inga filer. Om ett yttre skivminne initierats på detta sätt (med ett annat enhetsnamn som argument till *mkfs*) kan det nu direkt användas genom att kopplas in i root-filsystemet som ett underbibliotek med kommandot *mount*.

   Interna skivminnet används normalt som root-filsystem, varför systemfilerna nu måste kopieras över och vissa specialfiler skapas. Detta görs genom att programmet harddoit utförs för att överföra data från BOOT-disketten till skivminnet samt initiera de enhetstabeller som krävs för att skivminnet ska kunna användas som root-filsystem.

   Observera! Nu bör alltid den Revisionsinformation som medföljer systemet läsas. Där kan eventuella avvikelser från nedanstående rutin beskrivas.

2. Kopiera de viktigaste filerna från BOOT-disketten genom att utföra kommandot *harddoit*.

   ```
   /harddoit
   dd: 1+0 records in
   dd: 1+0 records out
   tar: block factor 1
   #
   ```

   Raderna under kommandot är utskrifter från *dd*- och *tar*-kommandona i harddoit.

   Programmet harddoit samt vissa program i biblioteket **/sas** på BOOT-disketten kopieras inte över eftersom de endast kan användas med BOOT-disketten som root-filsystem.

3. Stoppa systemet och starta upp igen manuellt till enanvändarmod, denna gång från det fasta skivminnet. Systemet måste stoppas med kommandot *telinit* **h**, eftersom det normala */etc/shutdown* saknas på BOOT-disketten.

```
telinit h
```

När texten **System halted** visas på terminalen ska systemet startas från det nyinitierade skivminnet. Notera att med de filer som nu finns kan inte systemet starta i fleranvändarmod. Nyckeln skall fortfarande vara i läget OFF och uppstarten ske i manuell mod upp till enanvändarmod, men nu med winchesterskivminnet som rootsystem genom att enbart **RETURN** ges som svar på alla frågor vid uppstarten och kommandot *dnix* ges från laddningsprogrammet tills prompten # erhålls. Bildskärmen bör då se ut så här:

```
M680XX prom                    (XX = maskinberoende)
                               <Ge enbart RETURN>
> dnix
OS:INFO: MP D-NIX X.X Virtual Ver X.X -
         System xxxk  User xxxxk  Swap x.xM
#
```

4. Läs in övriga systemfiler från tar-disketterna enligt den Release Notice som medföljer programvaran. Disketterna läses in i ordningsföljd enligt numreringen. Först skall disketten(disketterna) märkta UNIQUE läsas in, följda av de som är märkta COMMON. Eventuella UPDATE-disketter skall **ej** läsas in förrän grundversionen är installerad enligt nedan.

```
tar -xvf /dev/mf0          (för varje diskett)
```

Exempel på märkning av disketterna:

```
D-NIX X.X              xxx-xxxx-xx/ n(2)
Version X.xx
DIABxxxx / UNIQUE
tar format
Copyright Diab Data AB

D-NIX X.X              xxx-xxxx-xx/ n(4)
Version X.xx
DIABxxxx / COMMON
tar format
Copyright Diab Data AB
```

där /n är /1, /2, etc.

Efter att alla tar-disketter är inlästa skall kommandot */etc/rinstall* ges (enligt Release Notice). De viktigaste systemfilerna levereras nämligen med tecknet + i slutet av namnet för att vid uppdatering ge möjlighet att välja de gamla systemfilerna eller de nya samt att uppdatera det nya operativsystemet om systemparametrarna har ändrats från standard. (Se kommandot */etc/mkcfig* i kapitel 5). Detta görs med */etc/rinstall* genom att användaren svarar på frågor.

```
/etc/rinstall
```

Vid initiering av skivminnet skall svaret **y** (för ja) ges på alla frågor.

5. Efter det att grundversionens disketter lästs in och */etc/rinstall* utförts enligt anvisningarna i Release Notice kan eventuella UPDATE-disketter för systemet läsas in, en version i taget. */etc/rinstall* utförs för varje version enligt anvisningarna i respektive Release Notice.

6. Efter att alla disketter är inlästa och */etc/rinstall* utförts enligt Release Notice, kan systemet stoppas och startas upp igen i AUTO-mod samt anpassas till användarens miljö genom att man definierar nya användare och lösenord m m. För att göra detta vrids nyckeln först försiktigt till positionen AUTO. Därefter ges kommandot:

   ```
   /etc/shutdown -k
   ```

   När texten **System halted** visats på terminalen startar systemet automatiskt upp till fleranvändarmod. Om systemet hinner stoppas och meddelandet **M680X0 prom** hinner visas på skärmen innan nyckeln vridits till läget AUTO, kan nyckeln behöva vridas först till RESET och därefter direkt till AUTO för att initiera den automatiska uppstarten.

7. Nu bör ett total backup tas av grundsystemet. Kopiera hela skivminnet till ett streamer-band eller till disketter.

8. Därefter laddas eventuella optionsprogram in och eventuellt användarens tidigare filer från en backup, varefter en ny totalbackup tas med systemet färdigt för användning.

Exempel på en *harddoit* fil:

Nedan är ett exempel på *harddoit*:

Den långa raden med *tar* visas nedan uppdelad på grund av begränsad radlängd i handboken. Tecknet \ i slutet av raden anger att raden är uppdelad. På bildskärmen kan den se annorlunda ut.

```
# harddoit boot routine

MNT=/mnt
DEV=/dev/si0

dd if=/dev/mf0 of=$DEV count=1

if [ $# = 1 ]
then
     exit 0
fi

if /etc/mount $DEV $MNT
then

     # Copy everything
     tar cfS - bin usr etc dev tmp sas mf0 .profile \
          dnix boot | (cd $MNT; tar xf -)

     # Make an empty /etc/mnttab
     >$MNT/etc/mnttab

     #
     /etc/umount $DEV

else
     echo "harddoit: ERROR: Failed to mount $DEV"
fi
```

2.5. Systemavstängning

2.5.1 Avstängningsprocedurer

När systemet stängs av är det viktigt att det görs på rätt sätt. Alla användare måste få möjlighet att logga ut från systemet och alla filer och filsystem måste stängas.

Det finns tre normala sätt att förändra systemets konfiguration:

1. Fullständig avstängning, varvid även spänningen slås av. Detta sker helt automatiskt i de flesta system genom att vrida nyckeln till OFF. Om kommandot enligt punkt 2 skall användas får inte nyckeln vridas till OFF förrän efter att den fullständiga avstängningen avslutats. Efter att operativsystemet fullständigt stängts av visas texten **OS INFO: System halted** på huvudkonsolen.
2. Avstängning och omstart av hela systemet kan alltid ske genom kommandot */etc/shutdown* **-k**. vilket beskrivs i nästa avsnitt. Inom 10 sekunder efter att operativsystemet avslutats, dvs efter att texten **OS INFO: System halted** skrivits ut, kan nyckeln vridas till OFF för att initiera manuell uppstart.
3. Nedgång från fleranvändarmod till enanvändarmod, vilket görs med kommandot */etc/shutdown*.

2.5.2 Normal avstängning

Fullständig avstängning från operativsystemnivå

Detta sköts automatiskt i de flesta system då nyckeln på frontpanelen vrids från AUTO till OFF, omsystemet är i en- eller fleranvändarmod. För system utan automatisk avstängning används istället kommandot *shutdown -k* enligt ovan.

- Vrid nyckeln från AUTO till OFF.
- Nu tar det en stund medan systemet stängs. Filsystemet är avstängt när följande avslutande text visas på huvudkonsolen. Före denna kommer andra systemmeddelanden under avstängningsproceduren.

```
OS INFO:  System halted
```

- 10 sekunder efter att denna text visats stängs automatiskt spänningen av om nyckeln fortfarande är i läget OFF.
- Efter att texten **System halted** visats är det möjligt att vrida nyckeln till RESET och tillbaka till OFF, vilket ger en manuell uppstart. Om nyckeln däremot vrids till AUTO sker en automatisk återstart.

Avstängningen sköts genom att systemet exekverar kommandot *powerfail*. Därvid sker avstängningen automatiskt enligt ungefär samma mönster som med det manuella kommandot *shutdown -k* enligt beskrivningen i nästa avsnitt.

Fullständig avstängning från laddningsprogrammet

Avstängning av systemet från bootnivå 0 (boot) och bootnivå 1 (laddningsprogrammet) kan ske direkt genom att stänga av spänningen med huvudbrytaren, under förutsättning att inget program håller på att skriva eller läsa i skivminnet. I dessa lägen kan även systemet omstartas direkt genom att vrida nyckeln till läget RESET och tillbaka till OFF. Om så önskas kan istället automatisk uppstart erhållas genom att direkt efter RESET lämna nyckeln i läget AUTO. I början av kapitlet beskrivs hur manuell uppstart sker via bootnivåerna 0, 1, 2 och 3, där 2 och 3 motsvarar en- och fleranvändarnivåerna.

2.5.3 /etc/shutdown

När systemet stängs av är det viktigt att det görs på rätt sätt. Alla användare måste få möjlighet att logga ut från systemet och alla filer och filsystem måste stängas.

Detta kan utföras med kommandot */etc/shutdown -k*. Samma funktioner utförs automatiskt med systemprogrammet /etc/powerfail, vid normal avstängning med hjälp av nyckeln.

Även då systemet enbart skall tas ner till enanvändarmod skall kommandot */etc/shutdown* användas istället för *telinit* **s**, eftersom shutdown även avslutar sådana processer som inte *telinit* kan avsluta på ett riktigt sätt.

shutdown är ett systemprogram. Dess primära funktion är att avsluta alla processer som körs, på ett ordnat och riktigt sätt. Utan optioner avslutas alla processer och systemet går ner i enanvändarnivå utan att helt stängas. Om optionen **-k** ges till kommandot, kopplas däremot systemet helt ner. En avstängning med shutdown tar normalt mindre än 5 minuter om inga användare finns inloggade.

shutdown går igenom följande steg: Först uppmanas alla inloggade användare att logga ut, via ett utsänt meddelande. De får fem minuter på sig att själva avsluta, annars loggas de ut automatiskt. Alla mountade filsystem uppdateras innan systemet stängs av. Detta måste göras innan systemet laddas igen, för att hålla filsystemet intakt.

Som parameter till */etc/shutdown* kan anges en tid i minuter istället för de normala 5 minuter som användarna får för att avsluta. Dessutom kan ett meddelande anges som parameter, vilket visas för användarna tillsammans med uppmaningen från shutdown att avsluta.

shutdown kräver super-userprivilegier och att operatören loggat in som 'root' på huvudkonsolen.

Kommandot används på följande sätt för att helt stänga av systemet:

- Skriv */etc/shutdown* **-k**
- Om inga är inloggade avslutas alla processer direkt på ett ordnat sätt.

- Med inloggade användare får var och en ett meddelande varje minut om att systemet tas ned inom en viss tid. Efter fem minuter tas systemet ned på ett ordnat sätt, varvid alla inloggade automatiskt loggas ut.

Vid fullständing avstängning med */etc/shutdown* **-k** måste operatören vänta tills systemet skrivit följande på konsolterminalen, innan nyckelbrytaren får vridas till läget OFF och/eller huvudströmbrytaren slås av.

```
OS:INFO: System halted
```

Därefter kan systemet startas upp manuellt via laddningsprogrammet och enanvändarmod upp till fleranvändarmod.

Kommandot används på följande sätt för att gå ner till enanvändarmod:

- Skriv */etc/shutdown*
- Om inga är inloggade avslutas alla processer direkt på ett ordnat sätt.
- Med inloggade användare får var och en ett meddelande varje minut om att systemet tas ned inom en viss tid. Efter fem minuter tas systemet ned på ett ordnat sätt, varvid alla inloggade automatiskt loggas ut.

Efter att alla processer utom *init* avslutats går systemet in i enanvändarmod, vilket är systemnivå **s**. Därefter kan enbart huvudkonsolen användas. Denna är alltid samma som den fysiska huvudkonsolen, eftersom */etc/shutdown* endast kan startas från den denna.

2.5.4 /etc/powerfail

/etc/powerfail är ett systemprogram som automatiskt anropas då nyckeln vrids till läget OFF. Programmet stänger av datorn helt tills spänningen automatiskt slås av. Alla processer avslutas och filsystemet uppdateras på samma sätt som med kommandot */etc/shutdown*, men snabbare och mer direkt. Användare som är inloggade får fortfarande max 5 minuter på sig att logga ut, annars stoppas systemet direkt.

På vissa datorer stängs ej spänningen av automatiskt.

3

3. Laddningsprogrammet

3. Laddningsprogrammet

3.1. Systemstart av laddningsprogrammet

För att använda de kommandon som beskrivs i detta kapitel måste systemet startas på nytt i manuellt läge. Vid systemstart läses först ett laddningsprogram in.

Laddningsprogrammet är ett litet operativsystem som enbart används för vissa speciella maskinberoende kommandon.

I biblioteket **/sas** finns några kommandon som enbart får användas under laddningsprogrammet.

Även vid automatisk systemstart läses laddningsprogrammet in, men då utförs automatiskt ett kommando som direkt laddar det stora operativsystemet (D-NIX) från skivminnet. Operativsystemet startas sedan först upp i enanvändarläge och går sedan automatiskt över i fleranvändarläge.

Vid manuell systemstart passeras följande nivåer. Dessa passeras automatiskt vid automatisk uppstart, upp till den bootnivå som angivit med kommandot */sas/bootpar* enligt följande kapitel.

Bootnivå 0: Texten **M680x0** visas på huvudkonsolen. Här väntar det program som ligger i hårdvaran på att operatören skall ange från vilken enhet som laddningprogrammet skall läsas in.

Bootnivå 1: Nu är laddningsprogrammet aktivt och de kommandon som beskrivs i detta kapitel kan användas. Tecknet > skrivs ut som prompt.

Bootnivå 2: Nu har operativsystemet startats. Denna nivå är operativsystemets enanvändarnivå.

Bootnivå 3: Detta är fleranvändarnivån, vilket är den nivå som automatisk uppstart normalt sker till. Inom denna finns i princip flera systemnivåer (run-levels), men normalt används bara systemnivå 2.

3.2. /sas/bootpar

Kommando för ändring av startparametrar i NVRAM. Kommandot finns i två versioner. Det ena, */sas/bootpar*, kan endast användas under laddningssystemet och inte under D-NIX. Därför måste systemet köras upp manuellt till laddningsprogrammet innan */sas/bootpar* kan användas. Det finns även ett motsvarande kommando */etc/bootpar* som används under under D-NIX, men skrivning till NVRAM kan då bara göras av en användare med super-user privilegier.

bootpar är interaktivt. Datorn ställer alltså frågor och operatören svarar. Svar med enbart **RETURN** gör att de gamla värdena som visas inom parentes behålles.

Exempel:

```
> /sas/bootpar
Console baudrate (9600)      Anger baudrate vid manuell
                             uppstart i boot-läge och i
                             laddningsprogrammet.
                             9600 Baud är standard.
Boot level       (3)         AUTO start-nivå (3 = fleranvän-
                             darmod)
Boot device      (M,XX)      Enhet varifrån systemet skall startas.
                             Värdena är maskinberoende enligt
                             tabellerna i kapitel 15. Vanligast är
                             (5,16) för interna skivminnet
                             M=primära enhetsnumret (major)
                             XX=kanalnumret, samma som anges vid
                             manuell uppstart.
Root device      (255,255)   Reserverad parameter.
Swap device      (255,255)   Reserverad parameter.
Pipe device      (255,255)   Reserverad parameter.
Mirror device    (255,255)   Reserverad parameter.
CPU clock freq. (25000000)   Ges i Hz och är maskinberoende enligt
                             tabellerna i kapitel 15.
Satisfied ?                  Svara y(es) om rätt.
Write to NVRAM ?             Svara y(es) om data skall
                             sparas.
```

3.3. /etc/fsck

Detta kommando ska endast användas av systemadministratören och endast på skivminnen som inte är "mountade" i systemet. Kommandot skall användas under laddningsprogrammet, i bootnivå 1 efter manuell systemstart, om det interna winchesterskivminnet skall testas.

Filsystem på skivminnen, både disketter och winchesterminne, kan testas med kommandot *fsck*. Normalt förstörs ingen data.

Om systemet upptäcker fel i filstrukturen, rapporteras detta. Med kommandot *fsck* är det möjligt att försöka reparera den filstruktur som är förstörd.

Test av root-filsystemet från laddningsprogrammet (bootnivå 1)

```
>/etc/fsck -rr si(X,Y)

 Checking si(X,Y)
 File system:                Volume:
 Volume size =   82Mb,
 Block size = 2048,13312 I-nodes (13312 cont.)

 Phase 1 - Check I-nodes
 Phase 2 - Check Pathnames
 Phase 3 - Check Unreferenced files
 Phase 4 - Check Link Counts
 Phase 5 - Check Bitmap

 Total number of files:    662
 Total number of blocks: 41472,  37136 - free blocks
```

Tillvalet **-rr** ska alltid anges i laddningsprogrammet.

Enhetsnamnet, exempelvis si(16,0), ges i samma form som vid manuell start från enheten enligt kapitel 2. Namnet beror av datortypen enligt tabellerna i kapitel 15.

Om kommandot *fsck* hittar filer eller bibliotek som inte är knutna till filsystemet, länkas dessa vid reparationen till biblioteket */lost+found*. I detta bibliotek finns det alltså möjligheter att återfinna data som gått förlorad vid reparationen.

Test av ett inte mountat filsystem

Ett inte mountat filsystem, t ex på en diskett, kan testas från operativsystemets en- eller fleranvändarnivå, men då ska istället enhetsnamnet anges som argument. För test av ett filsystem på diskett ges följande kommando:

```
/etc/fsck /dev/mf0
```

Se i övrigt beskrivning av *fsck* i kapitel 5.

4

4. Filer och hjälpprogram

4. Filer och hjälpprogram

4.1. Allmänt

Nedan beskrivs enheter och filer som innehåller parametrar eller kommandon för systemstart eller inloggning och som den systemansvarige kan vilja ändra på. Nyttiga hjälpprogram (med kursiv stil nedan) beskrivs kortfattat.

BootPROM	Litet laddningsprogram i en fast minneskrets. Kan ej ändras.
/etc/bcheckrc	Kommandofil som bearbetas vid systemstart.
/dev/bclock	Batteriuppbackad klock-krets.
/etc/bootpar	Kommando för ändring av **/dev/nvram**.
/etc/brc	Kommandofil som bearbetas vid start av D-NIX i fleranvändarmod. Där mountas eventuella extra skivminnen.
/dev/console	Systemets huvudkonsol. Skall alltid vara samma som **/dev/systty**.
/bin/date -B	Kommando för att ställa batteriklockan.
/dev	Bibliotek med fysiska enheter.
/etc/dialups	Lista över terminalportar där sekundärt lösenord krävs.
/etc/d_passwd	Lista med sekundära lösenord för inloggning via de terminalportar som listats i **/etc/dialups**.
/etc/getty	Kommando som sätter upp kommunikationsparametrar och startar inloggning. Exekveras automatiskt via */etc/inittab*.
/etc/gettydefs	Tabell med baudrate och andra kommunikationsparametrar för terminaler vid inloggning.
/etc/group	Lista över användargrupper med grupplösenord m m.
/etc/init	Systemprocess som är ursprunget till alla andra i operativsystemet.
/etc/inittab	Tabell med kommandon som ska göras vid systemstart och vid övergång mellan systemnivåer, t ex mellan enanvändarmod och fleranvändarmod.
/etc/issue	Fil med systemmeddelande som visas **före** inloggning, före login-frågan.

/bin/login	Kommando för inloggning, som automatiskt exekveras av */etc/getty*.
/bin/lp	Program för att ta emot utskriftsbegäran som skall köas och skrivas ut med spoolersystemet **lp**. (Standard)
/bin/lpr	Program för att ta emot utskriftsbegäran som skall köas och skrivas ut med spoolersystemet **lpr**.
.mail.rc	Fil med användarens egna alias-namn för kommandot **mail**.
/usr/lib/mail/mail.rc	Fil med globala alias-namn för kommandot **mail**.
/etc/mknod	Kommando för att lägga till nya fysiska enheter.
/etc/mkuser	Kommando för att lägga till nya användare.
/etc/motd	Aktuellt systemmeddelande. Visas **efter** inloggning.
/dev/nvram	Ändringsbara startparametrar i minneskrets.
/bin/passwd	Kommando för ändring av lösenord.
/etc/passwd	Lista över användare med lösenord m m.
/etc/powerfail	Kommandofil som bearbetas då nyckelbrytaren på frontpanelen vrids till OFF, varefter spänningen automatiskt slås av.
/etc/profile	Gemensam kommandofil som automatiskt exekveras före *.profile* efter inloggning.
.profile	Användarens kommandofil som exekveras efter inloggning.
/etc/rc	Kommandofil som bearbetas vid övergång till fleranvändarmod.
/etc/rmuser	Kommando för att ta bort en användare.
/etc/setspeed	Kommando för att ändra överföringshastighet och andra kommunikationsparametrar för en icke aktiv port.
/etc/shutdown	Kommando som kan ges av operatören för att stänga av systemet eller övergå till enanvändarmod.
/bin/stty	Program för temporär ändring av kommunikationsparametrar efter inloggning.
/dev/syscon	Den virtuella systemkonsolen, dvs den som gäller i enanvändarmod.

/dev/systty	Systemets huvudkonsol. Skall alltid vara samma som **/dev/console**.
/bin/telinit	Kommando för att byta systemnivå enligt filen **/etc/inittab**.
/etc/termcap	Parameterfil med terminalstyrtecken, kompatibelt med tidigare versioner.
/usr/lib/terminfo	Bibliotek med parameterfiler, som innehåller styrtecken för olika terminaltyper enligt shellvariabeln **TERM**. Dessa filer är kompatibla med filen **/etc/termcap** vid leverans av systemet.
/etc/timezone	Sträng med tidsskillnad för definition av lokal tid.

Observera! Samtliga kommandon skall ges med gemena (små) bokstäver!

För mer information om kommandon, se **Referenshandboken**.

4.2. Beskrivning

4.2.1 BootPROM

Ett litet program som laddar in ett startprogram. Startprogrammet kan antingen vara på den enhet **Boot device** som anges i NVRAM eller, i manuell mod, på den enhet som operatören anger från tangentbordet. Programmet BootPROM ligger i en fast minneskrets i systemet och kopieras över till datorns arbetsminne innan första systemstart. Här ligger även vissa testrutiner för maskinvaran som utförs innan startprogrammet laddas.

4.2.2 /etc/bcheckrc

Denna kommandofil exekveras automatiskt av systemet endast första gången övergång sker till fleranvändarmod efter spänningstillslag och inladdning, styrt av */etc/inittab*.

I standard UNIX ligger här kommandon för att testa filsystemet och för att sätta systemklockan. Detta system testar automatiskt anslutet filsystem och sätter systemklockan från batteriklockan vid start, varför detta inte behöver göras här.

Exempel på **/etc/bcheckrc**:

```
# ***** This file has those commands necessary to
# check the file system, date, and anything else that
# should be done before mounting the file systems.
#
# Warning: Since the root filesystem may be corrupted,
# do not create files or use here documents, etc. (ie.
# `date`) until check is complete!
#
# Since D-NIX has automatic file checking and date
# setting this file is not used.
#
```

4.2.3 /dev/bclock

Batteriuppbackad klockkrets, som sitter på datorkortet. Vid systemstart laddas systemets klocka automatiskt med datum och tid från batteriklockan. Batteriklockan kan läsas med *cat* **/dev/bclock** och innehåller datum och tid i GMT. För ändring av batteriklockan används kommandot *date* **-B**.

4.2.4 /etc/bootpar eller /sas/bootpar

Kommando för ändring av startparametrar i **/dev/nvram**. */etc/bootpar* kan användas på operativsystemnivå, men skrivning till NVRAM kan bara göras av en användare med superuser-privilegier. Normalt bör emellertid ändring av dessa parametrar göras i laddningsprogrammet med */sas/bootpar*.

bootpar är interaktivt. Datorn ställer alltså frågor och operatören svarar. Svar med enbart **RETURN** gör att de gamla värdena som visas inom parentes behålles.

Se */sas/bootpar* i kapitel 3 för detaljer och exempel!

4.2.5 /etc/brc

Denna kommandofil exekveras automatiskt då systemet startas upp till enanvändarmod första gången efter spänningstillslag och inladdning, styrt av */etc/inittab*. Bl a rensas mounttabellerna i systemet och eventuella extra skivminnen mountas. Exempel på **/etc/brc**:

```
:
# *****    This command file's function is to initiate
#          the system after boot
# Set root disk as mounted
/etc/devnm / | /etc/setmnt
# Mount all other disks
# mntchk/dev/smd0/smd0
# Set systemname
if [ "`uname -n`" != '(empty)' ]
then
     uname -n >/etc/systemid
     chmod 644 /etc/systemid
fi
# Write boot record in /usr/adm/sa/sa??
/bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`" \
                                        >/dev/null 2>$1
# Update Version number in /etc/motd and /etc/issue
if [ -x /etc/ed ]
then
     set `uname -a`
     case $1 in
          MP) shift;;
     esac
     VERS="$1 Version $3 $4"
     for FILE in /etc/issue /etc/motd
     do
          if [ -w $FILE ]
          then
          ed - $FILE <<++ >/dev/null 2>$1
/Version/c
               $VERS
.
w
q
++
          fi
     done
fi
```

4.2.6 /dev/console och /dev/systty

Dessa enhetsnamn är synonyma, skall alltid vara länkade till samma fysiska serieport och anger var systemets huvudkonsol finns. Jämför även **/dev/syscon**, som är virtuell systemkonsol i enanvändarmod.

4.2.7 /bin/date -B

Kommando för ändring av batteriklockan. Systemklockan ändras samtidigt. Systemklockan sätts automatiskt vid systemstart, men kan även ändras med kommandot *date* utan optionen **-B**.

Om batteriklockan går för fort eller sakta justeras systemklockan automatiskt med en justeringskonstant. Konstanten beräknas på nytt varje gång systemklockan sätts med *date* och nollställs då batteriklockan ställs med *date* **-B**. Därför skall batteriklockan bara ställas en gång och

därefter justeras eventuell avvikelse med *date* utan optionen **-B**. Ju längre tid det går mellan första inställningen (med **-B**) och sista (utan **-B**) desto bättre blir justeringskonstanten. Se i övrigt kommandot *date* i **Referenshandboken**.

Omställning för **sommartid** måste göras i filen **/etc/timezone**, som laddas till shellparametern **TZ** vid inloggning. Se beskrivningen längre fram i kapitlet.

Exempel 1:

Ändring av batteriklockan.

Nytt värde 1986-09-24 10.30.25 ges som:

```
date -B 8609241030.25
```

Exempel 2:

Ändring av systemklockan, varvid ny justeringskonstant beräknas, utgående från sista gången batteriklockan ställdes.

Nytt värde 1986-09-24 11.30.25 ges med lokal tid som:

```
date 8609241130.25
```

4.2.8 /dev

Detta bibliotek innehåller inte vanliga filer utan innehåller specialfiler, som används för åtkomst till fysiska enheter i systemet. Se kapitel 15.

4.2.9 /etc/dialups

Lista över terminalportar där extra lösenord skall krävas (dialup password). Vid inloggning till någon av dessa terminalportar kräver systemet, förutom det normala lösenordet, att användaren skall ange ett extra lösenord som skall finnas definierat i filen **/etc/d_passwd**. Finns inte användaren definierad i **/etc/d_passwd** avbryts inloggningen. Se i övrigt beskrivningen av *login* i **Referenshandboken**.

Exempel på **/etc/dialups** fil:

```
/dev/tty03
/dev/pk00
/dev/modem1
```

4.2.10 /etc/d_passwd

Lista över användare som får logga in via de terminalportar som listats i filen **/etc/dialups**. Här anges det extra lösenord som skall användas vid inloggning via dessa terminalportar. **/etc/d_passwd** är en fil med samma format på fälten som filen **/etc/passwd**. Endast de fyra första fälten krävs, inklusive avslutande kolon. Användarnamnet måste stämma med de numeriska ID-numren enligt filen **/etc/passwd**.

Kommandot *passwd* med **-f** optionen används för att ändra lösenordet i **/etc/d_passwd**.

Observera! Inga åldringskoder får finnas i fältet med lösenordet. Jämför **/etc/passwd**.

Exempel på **/etc/d_passwd** fil:

```
jag:5Xu5K8hdfK7as:110:110:Sven Svensson
kalle:kJie3dKd7csHk:111:110:Kalle Karlsson
```

4.2.11 /etc/getty

Detta kommando startas automatiskt av *init*, enligt raderna i **/etc/inittab**, för att initiera anslutna terminaler och starta inloggning. Se **/etc/inittab** för exempel.

Baudrate och andra kommunikationsparametrarna som skall användas tas från tabellerna i **/etc/gettydefs**, där även en textsträng anges, som visas som inloggningsfråga. Där kan även ytterligare textsträngar anges, avskilda med kommatecken. Dessa är då frågetexter som kommandot *login* kommer att använda efter att */etc/getty* startat login-processen.

/etc/getty kan prova olika kommunikationsparametrar genom att tabellerna i **/etc/gettydefs** kan länkas ihop. Byte till nästa tabell sker då användaren försöker mata in sitt namn, om systemet upptäcker överföringsfel eller ett break-tecken.

/etc/getty startas normalt upp via kommandot *nice* för att vanliga terminaler skall ha lägre prioritet än huvudterminalen. Huvudterminalen bör ha högre prioritet, för att underlätta för systemoperatören vid systemunderhåll eller felsökning.

Efter utloggning återstartas */etc/getty* av *init* igen för ny inloggning.

Kommandosträngen för att starta */etc/getty* är:

```
/etc/getty [-h] [-m] [-t timeout] [-l] [-R1] [-M] [-r]
           [-u period] line [speed [ type [ ld ]]]
```

eller för testning av en fil formatterad som **/etc/gettydefs**:

```
/etc/getty -c filnamn
```

där:

-h Anger att */etc/getty* inte ska göra 'hangup' före aktiveringen av linjen. Hangup innebär att utsignalen DTR i seriella kontakten släpps för att eventuella modem säkert skall kopplas ner från föregående inloggning.

-m Anger att porten som *getty* startas mot redan är öppen, t.ex. när *getty* startas av *mgetty*.

-t Anger att **/etc/getty** ska vänta maximalt *timeout* sekunder på en inloggning. Därefter ska linjen kopplas ner. Detta säkerställer nedkoppling av modemlinjer om någon ringer in, men inte loggar in.

-l Hindrar **getty** från att automatiskt transformera mellan versaler och gemena, om login-namnet avgivits med enbart versaler.

-R1 och -M Används med D-CALL systemet. Se dokumentation till D-CALL.

-r Hindrar *getty* från att skriva ut något meddelande, förrän minst ett tecken lästs från linjen. Detta tillval kan användas om två datorer kopplas ihop direkt via två terminalportar, dvs då två getty-processer står mot varandra.

-u *period* Anger att *getty* skall vänta om någon annan använder linjen och att en låsfil skall skapas då förbindelsen etableras, dvs då bärvågsindikering erhålles. Även kommandona *cu*, *kermit* och *uucp* använder samma låsfil. Om linjen är upptagen, dvs en giltig låsfil finns, väntar *getty* och testar låsfilen regelbundet tills den blir ogiltig eller tas bort. Argumentet *period* anger periodtiden i sekunder, normalt 60 sekunder. En låsfil innehåller ägarens process-id och är giltig enbart om ägarprocessen är aktiv i systemet. Låsfilens namn bildas enligt **/usr/spool/locks/LCK..ttyXX** från terminalportens namn.

line Detta är den fysiska enhet dit terminalen är kopplad. Här anges en enhet i biblioteket **/dev**, t ex tty02.

speed Detta är en pekare som visar på en "rad" i filen **/etc/gettydefs**. Där anges vilka terminalparametrar som ska provas först och hur loginfrågan ska se ut och vilka parametrar som ska provas sedan om det behövs. Om denna parameter utelämnas, används den första raden i filen **/etc/gettydefs**. Första raden används även om ingen rad motsvarande *speed* finns i **/etc/gettydefs.**

type Används ej. I vissa system anges här om speciella terminaldrivrutiner skall användas.

ld Används ej. I vissa system anges här om speciella kommunikationsprocedurer skall användas.

-c För att testa formatet på filen **/etc/gettydefs**, eller en ny ändrad fil som ska användas, kan */etc/getty* ges från tangentbordet

som ett kommando med flaggan **-c**. Då testas tabellerna och rapporterar eventuella fel. Med korrekta tabeller skrivs parametrarna ut.

Inloggningen sker i följande ordning när */etc/getty* startas:

- *getty* väntar alltid tills bärvåg (DCD) indikerats.
- Om optionen **-u** givits, väntar *getty* tills linjen är ledig och skapar då en låsfil.
- Terminalparametrarna sätts enligt **/etc/gettydefs**.
- Om optionen **-r** givits, väntar *getty* tills minst ett tecken lästs från linjen.
- En system-identifikation skrivs ut. Denna är nodnamnet som kan ses med kommandot *uname -n* och skapas med */etc/mkcfig*.
- Texten i filen **/etc/issue** skrivs ut.
- Inloggningsfrågan i **/etc/gettydefs** visas.
- Användaren matar in sitt användarnamn på terminalen. Under tiden kan */etc/getty* ställa om vissa terminalparametrar.
- Efter att användarnamnet matats in, men innan *login* anropas, ställer */etc/getty* om terminalparametrarna enligt den andra parametergruppen för raden i **/etc/gettydefs**.
- Processen *login* startas med användarnamnet som parameter.
- *login* läser och testar inloggningskoden (password) samt sätter upp vissa shellvariabler innan den shell eller annan process startas som angivits i */etc/passwd*.

Övriga parametrar som kan ändras av **/etc/getty** beror av de tecken som mottages vid inmatning av användarnamnet och är följande:

Om namnet avslutas med koden **RETURN** (0D Hex) istället för det normala New-Line (0A Hex) sätts terminalparametrarna så att **RETURN** alltid konverteras till New-Line vid inläsning från tangentbordet.

Om namnet inte innehåller någon liten bokstav, sätts parametrarna så att alla stora bokstäver i fortsättningen översätts till små vid inläsning från terminalen. Denna automatiska översättning hindras med tillval **-l**.

Under inmatning av login-namn och lösenord används # och @ som tecken för **erase** och **kill**, om dessa inte omdefinierats i **/etc/gettydefs**.

4.2.12 /etc/gettydefs

Denna fil innehåller tabeller för inloggning med kommandot */etc/getty*, för att definiera terminalparametrar. En parameter till kommandot */etc/getty* är **speed**, vilken används som pekare till en "tabellrad" i filen **/etc/gettydefs**.

Varje tabellrad följs av en tom rad i filen, även sista tabellraden. En "tabellrad" kan sträcka sig över flera textrader.

Om */etc/getty* startas utan parametern **speed** eller om **speed** ej återfinns i filen, används första tabellraden i **/etc/gettydefs**.

De olika fälten är begränsade av tecknet '#' och är:

```
speed # param1 # param2 #Login prompt # next
```

eller

```
speed # param1 # param2 #Login,Lopt,Popt,Dopt # next
```

speed Motsvarar parametern *speed* i */etc/getty*.

param1 Terminalparametrar i ett format som liknar parametrarna till kommandot stty, men med stora bokstäver och alltid bokstaven B före baudrate-talet samt bokstaven V före namn på eventuella kontrolltecken. */etc/getty* sätter upp dessa innan systemmeddelanden visas och användarnamnet läses in. Dessa parametrar används tills login-kommandot ska startas. För parametrarna, se kommandot *stty* i **Referenshandboken**. Inga minustecken får finnas i stty-parametrarna i **/etc/gettydefs**.

param2 Dessa terminalparametrar sätts upp efter inmatning av namnet, men innan loginkommandot startas.

Login,Lopt,Popt,Dopt Detta fält kan innehålla upp till fyra olika frågesträngar, separerade av kommatecken. Härigenom kan valfritt språk användas vid inloggningsrutinerna. Notera att eventuella mellanslag i dessa strängar visas som de är i detta fält. Den första strängen används av */etc/getty* för första frågan efter användarens namn. De tre följande strängarna sänds som tillvalen **-L**, **-P** och **-D**, i denna ordning, till kommandot *login* tillsammans med det inlästa användarnamnet. *login* använder därefter dessa frågesträngar för att fråga efter lösenord och eventuellt nytt användarnamn. Se kommandot *login* för detaljer.

next Detta är en pekare till **speed** i en annan tabell i **/etc/gettydefs** och är första parametern i nästa tabell som */etc/getty* ska prova vid inmatningsfel.

De terminalparametrar som ges i andra parametergruppen (*param2*) i tabellraden med **console** som **speed**-parameter används även för huvudkonsolen i enanvändar-läge.

Nedan visas ett exempel på tabeller i filen **/etc/gettydefs**. På grund av begränsad radlängd i handboken delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
console# B9600 HUPCL IXON CLOCAL # B9600 ICRNL IXON\
         IXANY IXOFF OPOST ONLCR TAB3 ISIG ICANON ECHO\
         ECHOE ECHOK CLOCAL
                              #Console Login: #console

1200# B1200 HUPCL IXON # B1200 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #1200

300# B300 HUPCL IXON # B300 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE EXHOK
                                         #login: #300

9600# B9600 HUPCL IXON # B9600 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #9600

2400# B2400 HUPCL IXON # B2400 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #2400

4800# B4800 HUPCL IXON # B4800 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #4800

19200# B19200 HUPCL IXON # B19200 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #19200

38400# B38400 HUPCL IXON # B38400 ICRNL IXON IXANY IXOFF\
      OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                        #login: #38400

net# HUPCL IXON # ICRNL IXON IXANY IXOFF OPOST ONLCR\
     TAB3 ISIG ICANON ECHO ECHOE ECHOK
                                  #Network Login: #net
```

Notera att raden med **net#**, vilken används om ett nätverk är anslutet, inte får innehålla någon baudrate.

Exempel på en rad i **/etc/gettydefs** med frågesträngar för *login*-kommandot på svenska och med **erase** definierat som CTRL-H. Notera att mellanslag kan ingå i frågesträngarna, lämpligt särskilt som sista tecken för att separera frågan från användarens svar.

```
sw96# B9600 HUPCL PARENB CS7 IXON VERASE '^H'
    # B9600 SANE IXANY TAB3 ECHOE VERASE '^H'
    #Namn: ,Namn: ,Lösen: ,Modemkod: #sw96
```

4.2.13 /etc/mgetty

mgetty startas automatiskt av *init* enligt raderna i **/etc/inittab**, för att starta övervakning av ett antal terminaler. *mgetty* används framför allt i stora system med många terminallinjer; systemets resurser behöver då inte belastas med väntande getty-processer för alla terminalportar.

Vilka terminaler som skall övervakas läses in från filen **/etc/mgettytab** eller annan angiven fil (**tabfil** nedan); terminalparametrar läses från **/etc/gettydefs**. Då bärvåg erhålls på någon av de övervakade terminalportarna startas en *getty*-process på vanligt sätt.

/etc/mgetty startas vanligen med *nice* för att vanliga terminaler skall ha lägre prioritet än huvudterminalen. Alla terminaler som övervakas av en viss */etc/mgetty* får samma prioritet.

Kommandosträngen i **/etc/inittab** för att starta */etc/mgetty* är:

/etc/mgetty [-V] [-c] [-o max_öppna] [-t *sekunder*] [-f *tabfil*]

eller, för att visa version, eller för att testa tabfilen:

/etc/mgetty [-V] [-c] [-f tabfil]

där:

-V	Visa versionen av */etc/mgetty* och avsluta sedan programmet.
-c	Läs **tabfil** och skriv ut hur den tolkas av */etc/mgetty*. Om ingen tabfil anges med **-f** används standardvärdet **/etc/mgettytab**. Programmet avslutas då hela filen genomlästs.
-f tabfil	Denna parameter talar om namnet på tabfilen som */etc/mgetty* läser. Om parametern inte ges till */etc/mgetty* används standardvärdet **/etc/mgettytab**.
-o max_öppna	Denna parameter talar om hur många av de portar som finns i **tabfil** som skall övervakas samtidigt av *mgetty*. Detta tillval används bara för nätverksportar där det finns en hel grupp portar, men där man bara plockar upp och använder en ny av dessa i taget vid inloggning via nätverket. Kan även användas då det finns växlar. Om parametern inte ges kommer samtliga portar i **tabfil** att övervakas.
-t sekunder	Denna parameter talar om hur ofta tabfilen måste kontrolleras. Kontroll sker av ändring av filen. Om så är fallet läses den på nytt. Tiden ges i sekunder, standardvärdet är 60.

Tabfilen till */etc/mgetty* har samma utseende som **/etc/inittab**, d.v.s. formatet:

```
id:rstate:action:process # kommentar
```

Fältet id: ID-kod för raden. Här kan upp till 4 tecken användas. Alla rader måste ha olika ID-koder. De används internt av */etc/mgetty*. ID-koden måste vara unik, inte bara för denna fil, utan för **/etc/inittab** och alla *mgetty*-tabfiler som används.

Fältet rstate: Här ges systemnivån: 0,...,6 eller a,b eller c. Detta fält används dock inte då */etc/mgetty* startas på en viss nivå i **/etc/inittab**.

Fältet action: Här tolkas 'once' och 'respawn' som att porten skall övervakas. Alla övriga strängar fungerar som 'off', d.v.s. respektive kommando aktiveras inte.

Fältet process: Här ges det kommando som skall exekveras då bärvåg erhålls på den port som ges. En enda typ av kommando får förekomma, nämligen *getty*. Om *nice* förekommer före *getty* kommer detta att skalas bort. Alla parametrar till *getty* kan användas, utom **-h** för hangup, som skalas bort och utförs av */etc/mgetty* själv. Om den terminalport som ges i kommandot inte finns, kommer kommandot inte att utföras.

/etc/mgetty utför följande:

- Vid systemstart läses tabfilen för vilka portar som skall övervakas.
- Vid systemstart läses också filen **/etc/gettydefs**. Detta görs för att porten skall kunna öppnas och sättas upp på ett riktigt sätt. *mgetty* gör alltid hangup på porten om inte **-h** angivits.
- Om bärvåg indikeras på en port startas en *getty* för den porten. *getty* startas med specialtillvalet **-m** för att indikera att porten redan är öppnad, och med **-h** för att hangup inte skall göras.
- Då en *getty* avslutas börjar portövervakning igen, enligt ovan.
- Status på tabfilen kontrolleras regelbundet, dels då en *getty* som varit aktiv avslutas, dels regelbundet en gång i minuten (eller kontrollerat av den tid som givits med **-t** sekunder-parametern). Om status ändrats, d.v.s. något har ändrats i filen, läses den på nytt. Om en port sätts till 'off', men är aktiv för tillfället, kommer den inte att startas om då *getty* på porten avslutas. Den kommer däremot inte att stoppas av *mgetty*.

Exempel på utseende av tabfilen

En tabfil till */etc/mgetty* som används för att starta fjärrinloggning kan se ut så här (filen heter **/etc/rlogintab**) :

```
r00:2:respawn:nice -16 /etc/getty -A rld00 rnet
r01:2:respawn:nice -16 /etc/getty -A rld01 rnet
...
r31:2:respawn:nice -16 /etc/getty -A rld31 rnet
```

Observera att dessa rader tagits helt oförändrade från **/etc/inittab**, där de tidigare varit placerade. Detta är meningen med att ha samma utseende av filerna **/etc/inittab** och **/etc/mgettytab**. Rader kan flyttas helt oförändrade mellan dessa. Vissa fält ignoreras dock av *mgetty*, enligt ovan.

Eftersom det står *nice* -16 före kommandona, bör den rad i **/etc/inittab** som startar *mgetty* ha ett utseende liknande:

```
mgt1:2:respawn:nice -16 /etc/mgetty -o 4 -f /etc/rlogintab
```

-o 4-tillvalet används för att bara ha 4 av de givna enheterna öppna vid varje tillfälle.

Exempel på hur utskrift av tabfilen kan kontrolleras

En tabfil till */etc/mgetty* ges nedan; filen har namnet **/etc/mterm**.

```
ut2:2:off:nice -16 /etc/getty -h -t 60 -r tty_uucp 1200
at3:2:respawn:nice -16 /etc/getty -t 60 -u 10 tty12 9600
t3:2:respawn:nice -16 /etc/getty -h -t 13 -r tty03 1200
t4:2:respawn:nice -16 /etc/getty -t 99 tty04 2400
```

En utskrift för att kontrollera tabfilen görs med:

```
$ /etc/mgetty -c -f /etc/mterm
ID ut2:    Command is IGNORED.
ID at3:
           Command='/etc/getty -t 60 -u 10 -m -h tty12 9600'
           Device='/dev/tty12'
           Speed='9600', do HANGUP
ID t3:
           Command='/etc/getty -t 13 -r -m -h tty03 1200'
           Device='/dev/tty03'
           Speed='1200'
ID t4:
           Command='/etc/getty -t 99 -m -h tty04 2400'
           Device='/dev/tty04'
           Speed='2400', do HANGUP
```

Felmeddelanden

/etc/mgetty kan generera en del felmeddelanden. Dessa skrivs alltid på systemkonsolen.

Följande felmeddelanden skrivs på konsolen, och */etc/mgetty* avslutas:

```
mgetty: Input file '/etc/rlogintab' not found.
```

tabfilen med det givna namnet finns inte i systemet. Det är troligen felstavat, eller också har filen tagits bort.

```
mgetty: Can't open '/etc/rlogintab' for input read.
```

tabfilen med det givna namnet finns, men mgetty har inte privilegier att läsa den. Troligen har mgetty startats av en vanlig användare och inte av root.

```
mgetty: Input file '/etc/rlogintab' has no data.
```

tabfilen finns, den är läsbar, men saknar data.

```
mgetty: Can't allocate more memory.
```

Det finns inte mer minne internt för att göra 'malloc'.

```
mgetty: Can't fork to start getty.
```

Det går inte att starta en ny process. Kanske finns så många processer i systemet att den övre gränsen nåtts.

```
mgetty: EXEC of getty failed.
```

getty-kommandot som gavs i tabfilen kan **ej** startas. Kanske har ett annat program än */etc/getty* använts, och har tagits bort, eller är kanske inte exekverbart.

Ett meddelande kan ges medan *mgetty* kör. Detta skrivs också på systemkonsolen:

```
mgetty: Can't open device '/dev/rld01'.
```

mgetty har inte rättighet att öppna enheten, eller lyckas av någon annan orsak inte med detta.

4.2.14 /etc/group

Lista över användargrupper. Nedan ges ett exempel där de olika fälten förklaras.

Exempel:

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
mail::6:root
rje::8:rje,shqer
daemon::12:root,daemon
vi::110:jag,kalle
```

Filen innehåller en rad per grupp. Raden har fyra fält, separerade med tecknet **:**.

Fält 1:

Gruppnamn. Detta används i kommandot *newgrp* när en användare vill byta grupp efter inloggningen. Gruppen 'other' med numret 1 är en standardgrupp till vilken alla användare hör, om ingen grupp anges då användaren skapas med kommandot */etc/mkuser*.

Fält 2:

Krypterad version av lösenordet för gruppen. Lösenordet används endast i kommandot *newgrp* när en användare byter grupp efter inloggningen, om användaren inte ingår i den nya gruppen (se nedan). Ett lösenord för en användargrupp kan definieras med kommandot *passwd* eller när gruppen skapas med */etc/mkuser*.

Fält 3:

Gruppnummer.

Fält 4:

Kommentar, där de användare listas som ingår i gruppen. Kommandot */etc/mkuser* skriver automatiskt in användarnamnen då de skapas. Namnen skiljs åt med kommatecken.

4.2.15 /etc/init

Processen *init* är den första process som laddas in och startas då operativsystemet startas upp. *init* stannar sedan kvar i systemet som ursprunget till alla andra processer. Dess huvudsakliga funktion är att starta upp andra processer enligt informationen i systemfilen **/etc/inittab**. T ex startar den upp kommandot */etc/getty* mot varje terminalanslutning för inloggning av användare i fleranvändarmod. Den kan även starta initieringsprocesser såväl som allmänna systemprocesser som ska arbeta i bakgrunden, t ex nätverksprogram eller speciella filhanterare. *init* kan återstarta processer, som t ex inloggning efter att en användare loggat ut ur systemet.

I filen **/etc/inittab** anges ett kommando på varje rad och varje rad innehåller en siffra (i 2:a fältet) som anger på vilken systemnivå som detta kommando skall aktiveras. Flera systemnivåer kan anges på varje rad, varigenom olika systemkonfigurationer kan skapas, beroende på vald systemnivå (run-level). Systemnivå 2 är standard för fleranvändarmod. Möjliga systemnivåer är 0 .. 6 samt s, där s är enanvändarmod. För att byta systemnivå kan kommandot *telinit* användas. Nivå 0 är reserverad och inga processer får definieras på nivå 0 i **/etc/inittab**. Nivå 0 används nämligen av */etc/shutdown* medan olika optionsprogram stängs av på ett kontrollerat sätt.

Det 3:e fältet på varje rad markerar ett villkor för hur processen ska arbeta. Detta och övriga fält på raderna beskrivs närmare nedan för **/etc/inittab**.

Vid manuell start av systemet upp till enanvändarmod, läses inte filen **/etc/inittab** utan systemet startar direkt i enanvändarmod utan den fråga som beskrivs nedan. På bootnivå 0 och i laddningsprogrammet an-

vänds alltid den fysiska huvudkonsolen **/dev/console** och standardparametrarna 9600 Baud, 8 databitar, ingen paritet, oberoende av konfigurationsfilerna i systemet. I enanvändarmod används däremot **/dev/syscon** och parametrarna i det andra fältet (*param2*) i raden för console i **/etc/gettydefs**.

Vid automatisk systemstart upp till fleranvändarmod startas *init* direkt på den systemnivå som markerats med villkoret initdefault i **/etc/inittab**. Om inget initdefault-villkor finns eller om ingen **/etc/inittab** finns. skriver *init* ut en fråga på den virtuella systemkonsolen (**/dev/syscon**) och ber operatören ange en systemnivå i form av ett tecken.

```
INIT: ENTER RUN-LEVEL(0-6,h or S): _
```

Detta kan vara en siffra i området 0..6 eller bokstaven **s**. Systemnivå 2 är den normala fleranvändarnivån, medan bokstaven **s** anger enanvändarmod (single-user-mode) och bokstaven **h** anger att systemet skall stoppas (halt).

/dev/syscon kan vid denna fråga vara länkad till en annan terminal än systemets huvudkonsol. Ett meddelande ges då till systemets huvudkonsol **/dev/systty** (**/dev/console**) var den virtuella systemkonsolen är länkad.

Om operatören sitter vid systemets fysiska huvudkonsol och frågan efter systemnivå inte kommer, kan det bero på att **/dev/syscon** är länkad till en annan terminal. Operatören kan då skriva tecknet DEL på systemets huvudkonsol, varvid *init* automatiskt länkar om **/dev/syscon** dit i samband med frågan om systemnivå.

Övergång från enanvändarmod till fleranvändarmod görs antingen genom kommandot *telinit* **2** eller genom att avsluta med CTRL-D (End-of-file). Notera att CTRL-D kan inte användas om systemet startats upp manuellt och aldrig varit uppe i fleranvändarmod. Vid avslutning med CTRL-D kommer *init* att ge en ny fråga enligt ovan efter systemnivå, varvid **2** anges för att gå till normal fleranvändarmod.

Endast första gången systemet övergår till fleranvändarmod efter start, startas de processer som i **/etc/inittab** markerats med villkoret **boot** eller **bootwait** (under förutsättning att systemnivån som angivits för dem är rätt). Dessa startas alltid före andra processer oberoende av positionen i filen **/etc/inittab**. Variabeln **TZ** sätts alltid av *init*, från **/etc/timezone**, innan någon process startas.

Efter att *init* startat alla processer enligt **/etc/inittab** väntar *init* på en av följande händelser:

- Någon process som skall återstartas dör. T ex om en användare loggar ut, ska *init* åter starta inloggning på den berörda terminalen. Vid snabbt upprepade återstarter (t ex vid programfel) väntar *init* extra lång tid (5 minuter) var 10:e gång.

- En signal att systemet ska stängas av, dvs. en powerfail-signal (SIGPWR). Då exekverar *init* de processer som i **/etc/inittab** markerats med villkoret *powerfail*, fortfarande beroende på systemnivån som markerats på raden i **/etc/inittab**.

- En signal kommer från t ex kommandot *telinit* med en ny systemnivå (eller samma). Då går *init* åter genom **/etc/inittab** och vidtar åtgärder enligt den givna systemnivån.

Om någon tidigare startad process, t ex inloggning på en terminal, skall avslutas vid en ändring av systemnivån, sänder *init* först en varning (SIGTERM) till processen och efter 10 sekunder avslutas processen med signalen SIGKILL.

Notera att endast processer som startats av *init* kan avslutas. Om någon process startat upp andra processer i bakgrunden och sedan själv avslutas, känner *init* inte till dessa och kan inte avsluta dem. Jämför kommandot */etc/shutdown* och filen **/etc/rc**.

4.2.16 /etc/inittab

Denna fil innehåller information om alla processer som skall startas upp automatiskt av systemet av *init*-processen vid övergång till fleranvändarmod eller vid val av en annan systemnivå inom fleranvändarmoden.

Varje process som ska startas beskrivs med fyra fält på en rad. Liksom i shell kan en rad vara längre än 80 tecken. Tecknet \ i slutet av textraden anger att informationen fortsätter på nästa textrad. Max 512 tecken tillåts i en processbeskrivning (en rad).

Om **/etc/inittab** ändras kan systemet tvingas läsa filen igen genom kommandot *telinit* **q**.

De olika fälten är begränsade av tecknet **:** och är:

```
id:rstate:action:process # kommentar
```

Fältet *id*:

ID-kod för raden. Upp till fyra tecken. Alla rader måste ha olika ID-koder. De används intern i *init*.

Fältet *rstate*:

En lista över de systemnivåer där denna process ska vara aktiv. *rstate* kan vara en eller flera siffror 0..6 som representerar systemnivåer i fleranvändarmod. Om fältet *rstate* är tomt betyder det att processen är aktiv på alla nivåer. Alternativt kan *rstate* anges som a, b och/eller c. Processer som markerats så startas enbart då motsvarande nivå-kod ges via kommandot *telinit*, utan att den verkliga systemnivån ändras. Nivå 0 får ej anges, då den i D-NIX representerar en nivå där inga processer i **/etc/inittab** är aktiva. Nivå 2 är normala fleranvändarnivån.

Fältet *action*:

Villkor som anger hur processen ska behandlas av *init*. Villkoren anges som en textsträng. Möjliga villkor är enligt nedan. Om inget annan sägs stoppas respektive process om systemnivån ändras så att *rstate* ej längre överensstämmer med systemnivån.

boot Endast starta processen vid första start till fleranvändarmod enligt givna nivån. *rstate* bör vara tomt.

bootwait Samma som 'boot' men vänta tills processen är klar innan nästa process startas enligt **/etc/inittab**.

initdefault Detta är enbart en markering som visar att den systemnivå som givits i motsvarande *rstate*-fält skall gälla vid automatisk systemstart till fleranvändarmod. Om ingen rad med initdefault finns frågar *init* efter systemnivå enligt beskrivningen för */etc/init* ovan.

sysinit Innan *init* skriver eller läser på systemkonsolen, t ex för en fråga om ny systemnivå, exekveras processer som markerats med sysinit, om nivån i *rstate* är rätt. *init* väntar alltid tills denna process är klar.

off Motsvarande process ska ej vara aktiv. Startas ej och stoppas om den redan är aktiv.

once Starta processen då systemnivån blir den nivå som givits i fältet *rstate*. Om processen redan är igång görs ingenting om systemnivån är rätt.

wait Samma som once men vänta tills processen är klar innan nästa process startas enligt **/etc/inittab**.

respawn Detta villkor används normalt för att starta inloggning på terminalerna i systemet och innebär att processen (vanligen */etc/getty*) startas utan att *init* väntar på att avslutas. Dessutom återstartas processen efter att den avslutats, styrt av nivålistan i *rstate*. I filerna **/etc/utmp** och **/etc/wtmp** sparar *init* information om de processer som startats och avslutats.

ondemand Samma som respawn men används bara med nivå a, b eller c för aktivering med kommandot *telinit*. Dessa processer stoppas ej automatiskt av *init* vid byte av systemnivå. De stoppas bara om motsvarande rad i **/etc/inittab** markeras med off eller tas bort, eller om systemet går ner i enanvändarmod.

powerfail Denna process exekveras då signalen SIGPWR indikerar att spänningen ska stängas av. Denna signal ges bl a då nyckelbryta-

ren på frontpanelen vrids till läget OFF.

powerwait Samma som powerfail, men vänta tills processen är klar innan nästa process kan startas enligt **/etc/inittab**.

Fältet *process*:

Detta är det kommando som ska exekveras och kan skrivas i standard shell-format. Vid exekveringen skapas ett kommando från strängen *process* med följande format:

```
sh -c 'exec process'
```

Strängen *process* kan innehålla tillval och argument till kommandon.

Fältet *kommentar*:

Kommentarer kan läggas in i slutet av varje rad, efter tecknet # enligt normal shell-standard. Kommentarer som avslutar rader med */etc/getty*-kommandon, visas med kommandot *who* som förklarande text tillsammans med användarnamnet, och kan till exempel innehålla upplysningar om var respektive terminal är placerad.

Notera att bootnivå 3 motsvarar den standardiserade systemnivån 2 i **/etc/inittab**, dvs fleranvändarnivån och att filen **/etc/inittab** inte alls används vid start direkt till enanvändarmod (bootnivå 2).

Systemnivå 0..6 är alla på systemets fleranvändarnivå. I systemet är nivå 0 reserverad och skall vara utan aktiva processer. Några systemnivåer kan som i exemplet konfigureras så att endast huvudkonsolen är aktiv, där den systemansvarige kan ha ensam tillgång till systemet. Denna nivå kan väljas med kommandot *telinit* **3**, med **/etc/inittab** enligt exemplet.

Notera att alla vanliga terminaler normalt startas med lägre prioritet än huvudkonsolen, kommandot *nice -16* i **/etc/inittab**. Dessutom sänks automatiskt prioriteten 16 steg för alla vanliga användare som loggar in på huvudkonsolen.

Exempel på tabeller i /etc/inittab:

På grund av begränsad radlängd i handboken delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
is:2:initdefault:
il::sysinit:ln /dev/systty /dev/syscon 1>/dev/null 2>&1
bl::bootwait:/etc/bcheckrc </dev/console >/dev/console
                                    2>&1 #bootlog
bc::bootwait:/etc/brc 1>/dev/console 2>&1 #bootrun command
sl::wait:ln /dev/systty /dev/syscon 1>/dev/null 2>&1
err:s123456:respawn:/usr/lib/errdemon -c /usr/adm/errfile
                                    /usr/adm/errmessages #error demon
tc:s0123456:off:/etc/tc #tc demon
load:s0123456:respawn:/usr/bin/load #load daemon
rc:2:wait:/etc/rc 1>/dev/console 2>&1 #run com
pf::powerfail:/etc/powerfail #power fail routines
net:2:off:nice -8 /usr/lib/net/netman -F
ra:2:off:nice -8 /usr/lib/net/raccess -F
db3:2:off:nice -12 /usr/bin/dblm /mimerdbl /usr/mimer/sysdb/sysdb
db4:2:off:nice -12 /usr/bin/dblm4 /mimer4dbl
                                    /usr/mimer4/sysdb/sys42
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
```

```
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:off:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
t8:2:off:nice -16 /etc/getty tty08 9600
t9:2:off:nice -16 /etc/getty tty09 9600
t10:2:off:nice -16 /etc/getty tty10 9600
t11:2:off:nice -16 /etc/getty tty11 9600
```

4.2.17 /etc/issue

Detta är en textfil med lämplig systemidentifikation eller systemmeddelande, som automatiskt visas av */etc/getty* före inloggning, men efter raden med systemets nodnamn.

Efter denna text visas den inloggningsfråga, som definierats i **/etc/gettydefs**, innan användaren börjar mata in sitt användarnamn.

Se även **/etc/motd** som visas efter inloggning.

4.2.18 /bin/login

Detta är kommandot som sköter själva inloggningen, med test av lösenordet, password, och uppsättning av de shellvariabler som är standard, t ex **HOME**, **PATH** m fl.).

Kommandot anropas av */etc/getty* men kan även användas direkt i formen *exec login* för att tvinga fram en utloggning och direkt återinloggning som en annan användare utan att terminalparametrarna ställs om med */etc/getty*. Denna form kan enbart användas från den primära shell som startades vid inloggning och inte från shell som startats med kommandot *sh*.

4.2.19 /bin/lp

Program för att ta emot utskriftsbegäran till kön i skrivarsystemet (**lp**-spooler-systemet). I *lp*-kommandot anges med ett tillval på vilken logisk skrivare som utskriften skall skrivas ut. Alternativt kan en skrivargrupp anges, varvid utskriften sker på första lediga skrivare i gruppen. Det som ska skrivas kan även riktas till *lp*-kommandot med en pipe, varefter det läggs i skrivarkön.

Utskrifter i skrivarkön tas om hand av processen *lpsched* som alltid aktiveras så snart någon utskrift finns i kön. *lpsched* tar utskrifterna ur kön och startar de kommandofiler som hör till respektive logisk skrivare och sänder utskriften till den fysiska skrivare som definierats för den logiska skrivaren. lpsched aktiveras även då systemet startas upp för att tömma eventuellt kvarvarande skrivarkö, som finns kvar sedan föregående avstängning.

Exempel 1:

Skriv till skrivaren **main**, statuslistan som ges av *lpstat*:

```
lpstat -t | lp -dmain
```

Exempel 2:

Skriv filen **myfile** till skrivaren **myfilter**:

```
lp -dmyfilter myfile
```

Se även spooler-systemet *lpr* enligt nedan. Endast ett av dessa system (*lp* eller *lpr*) skall vara aktiverade i systemet. Båda levereras med operativsystemet och **lp**-systemet är normalt aktiverat.

Alla konfigurationsdata och status för *lp*-systemet finns i biblioteket **/usr/spool/lp**.

Övriga kommandon i *lp*-systemet är:

lpstat	Listar status för begärda utskrifter eller med optionen **-t**, för *lp*-systemet som helhet.
cancel	Stoppar och tar bort en köad eller pågående utskrift. Som parameter anges ett könummer, enligt listan från *lpstat*, eller ett skrivarnamn.
disable	Hindrar temporärt alla utskrifter från köerna till en viss skrivare. Kommandot kan även ta bort hela kön av utskrifter till en skrivare.
enable	Möjliggör att utskrifter kan göras till en viss skrivare.
/usr/lib/lpmove	Flyttar köade utskrifter från en skrivare till en annan.
/usr/lib/lpshut	Stoppar alla utskrifter temporärt genom att stoppa processen *lpsched*.
/usr/lib/reject	Hindrar *lp* att sända utskrifter till en viss skrivarkö.
/usr/lib/accept	Tillåter *lp* att skriva till en skrivarkö.
/usr/lib/lpadmin	Detta är ett kommando för att skapa eller ändra konfigurationen för *lp*-systemet. Kopplingar införs mellan logiska skrivare, fysiska skrivare och de styrprogram som ska användas.
/usr/spool/lp/bin/lppg	Är ett filterprogram som kan användas i styrprogram för logiska skrivare för att sidvis styra utskrifterna med interaktiva kommandon via *lpsubmit*.
lpsubmit	Är ett interaktivt kommando för att styra utskrifter med *lppg*.

Hanteringen av **lp**-systemet beskrivs med praktiska råd i kapitel 9 i denna handbok och detaljer finns i **Referenshandboken** för de olika kommandona.

4.2.20 /bin/lpr

Ett alternativt printer-spoolersystem finns även med i grundpaketet. Kommandot *lpr* kan användas att sända utskrifter till "logiska skrivare" i systemet, jämför *lp*-kommandot. Det är bakgrundsprocessen */usr/lib/lpd* som sköter utskriften från den kö som bildas med *lpr*. Notera att endast en av dessa båda system skall vara aktiva i systemet. Se kapitel 9.

Med *lpr* finns ett antal tillval. Ett tillval (**-t**) är vilken logisk skrivare som utskriften ska sändas till. Utan angivande av logisk skrivare riktas utskriften till **/dev/lp**. Det som ska skrivas kan även riktas till *lpr*-kommandot med en pipe, varefter det läggs i skrivarkön.

Exempel 1:

Skriv filen **myfile** till skrivaren **pr1**:

```
lpr -tpr1 myfile
```

Kopplingen mellan "logisk skrivare" och fysisk skrivare görs i filen **/usr/lib/lpdpar/lpdtable**. Denna fil innehåller en rad för varje skrivare och de villkorssträngar som kan anges i optionen **-t** till *lpr* för att välja skrivare.

Kommandot *queue* används för att lista vilka utskrifter som ligger i kö för att skrivas ut med *lpr*-spoolersystemet. Med *queue* kan icke påbörjade utskrifter tas bort ur kön. Utskriftsordningen styrs av en utskriftsprioritet som kan ändras med kommandot *queue*.

Hanteringen av *lpr*-systemet beskrivs med praktiska råd i kapitel 9 och detaljer finns i **Referenshandboken** för de olika kommandona.

4.2.21 .mail.rc

Filen **.mail.rc** kan finnas i användarens inloggningsbibliotek och innehåller då en lista över alias-namn för kommandot *mail*. Detta är en lista med förkortningar som av *mail* översätts till de verkliga användarnamnen, eventuellt på andra maskiner i en nätverk. *mail* söker först i denna lokala fil, därefter söker den i motsvarande globala alias-fil som heter **/usr/lib/mail/mail.rc**.

Varje rad i filen innehåller två fält med alias-namn och den verkliga mail-adressen. Dessutom kan kommentarer fritt läggas in efter dessa två fält.

Exempel:

Ett meddelande ska sändas med kommandot:

```
mail karl
```

till Kalle Karlsson på en maskin som heter com2 genom nätverket via en annan maskin som heter com1. Kalle Karlsson har användarnamnet kalle. (Denna fil används inte om man använder d-post)

Raden i **.mail.rc** ska då vara:

```
karl        com1!com2!kalle        Kalle Karlsson
```

4.2.22 /usr/lib/mail/mail.rc

Denna fil innehåller globala alias-namn för kommandot *mail*. Kommandot *mail* söker först i användarens egen fil **$HOME/.mail.rc** och sedan i den globala **/usr/lib/mail/mail.rc** efter det namn som angivits som parameter till *mail*.

Formatet hos filen är samma som för filen **.mail.rc** i föregående stycke.

4.2.23 /etc/mknod

Kommando för att lägga till nya enhetsnamn i biblioteket **/dev**. Detta förutsätter att motsvarande drivrutin redan finns i operativsystemet. */etc/mknod* gör ett nytt enhetsnamn och anger vilken drivrutin och vilken fysisk kanal som skall användas för enheten. Vilka enhetsnummer som normalt används ses i kapitel 15.

Exempel:

Definiera en ny terminalenhet med namnet **tty06** och kanal nummer 6. Systemets drivrutin för terminaler med seriell anslutning har nummer 1. I kommandot *mknod* anges även att enheten arbetar teckenvis genom tecknet **c** för 'character oriented'. För inloggning krävs även ändring i **/etc/inittab**, dvs att terminalen aktiveras.

```
/etc/mknod /dev/tty06 c 1 6
```

4.2.24 /etc/mkuser

Kommando med vilket en privilegierad användare (super-user) kan lägga till en ny användare i systemet. Programmet är interaktivt.

4.2.25 /etc/motd

Detta är en textfil, vars innehåll visas för varje användare efter inloggning på systemet. Den systemansvarige kan här lägga in lämpliga aktuella meddelanden. Textfilen visas endast om ett meddelande finns i filen. Jämför **/etc/issue** som visas före inloggning.

/etc/motd visas normalt inte vid inloggning som super-user med namnet root.

4.2.26 /dev/nvram

Starttabeller i en fast men programmerbar minneskrets, ett sk NVRAM, non-volative RAM. Anger bl a varifrån systemet skall laddas vid start. Värdena i NVRAM kan ändras med */sas/bootpar* eller */etc/bootpar*. Se */sas/bootpar* i kapitel 3 för detaljer.

4.2.27 /bin/passwd

Kommando med vilket varje användare kan ändra sitt lösenord för inloggning.

En privilegierad användare (super-user) kan även ändra andra användares lösenord med **/etc/passwd**.

Vid inmatning av lösenord syns inga ekon på bildskärmen. Lösenordet måste ges två gånger för att minska risken för misstag. Ett lösenord ska omfatta minst sex tecken och får inte vara lika med användarnamnet. Se **Referenshandboken** för mer detaljer.

Om de extra tecknen för tidsbegränsning av lösenordet finns i fält 2 i filen **/etc/passwd**, kan användaren hindras ändra lösenordet enligt de villkor som beskrivs nedan för fält 2 i **/etc/passwd**.

Kommandot *passwd* kan, med optionen **-f**, användas även för att ändra användarens lösenord i filen **/etc/d_passwd** om denna används i systemet för sekundär inloggning. Även grupplösenord i filen **/etc/group** kan ändras med optionen **-f**. Bara super-user får ändra **/etc/group**.

4.2.28 /etc/passwd

Lista över användare. Nedan ges ett exempel, varefter de olika fälten förklaras. Nya användare läggs till med kommandot */etc/mkuser*.

En användare kan även läggas till manuellt genom att editerar in en ny rad i **/etc/passwd** och i **/etc/group**. För användare som skall kunna skapa nya filer måste ett bibliotek skapas, som ägs av användaren. Då skall även filen **.profile** kopieras till det nya biblioteket. Allt detta görs automatiskt av kommandot */etc/mkuser*.

Exempel på /etc/passwd fil:

På grund av begränsad radlängd i manualen delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
root:xAsp9seK0nqrd:0:3:0000-Admin(0000):/:
daemon:*noway*:1:12:0000-Admin(0000):/:
bin:*noway*:2:2:0000-Admin(0000):/bin:
sys:*noway*:3:3:0000-Admin(0000):/usr/src:
adm:*noway*:4:4:0000-Admin(0000):/usr/adm:
uucp:*noway*:5:1:0000-uucp(0000):/usr/lib/uucp:
nuucp::6:1:0000-uucp(0000):/usr/spool/uucppublic:
                           /usr/lib/uucp/uucico
rje:*noway:68:8:0000-rje(0000):/usr/rje:
shqer:*noway:69:8:0000-rje(0000):/usr/rje:
lp:*noway:71:2:0000-lp(0000):/usr/spool/lp:
jag:3Uuv4MJme0Khk:110:110:Sven Svensson:/usr/sven:
kalle:rrFbrSmdXJo0s:111:110:Karl Karlsson:/usr/karl:
```

Här har ännu inget lösenord definierats för **root**, vilket bör ske så snart systemet installerats. Ett riktigt lösenord bör samtidigt definieras för användare **lp**, systemadministratör för *lp* skrivarsystem.

Filen innehåller en rad per användare. Raden har sju fält, separerade med kolon **:**. Endast femte fältet med kommentaren får innehålla mellanslag.

Fält 1:

Användarnamnet. Detta är det namn som anges vid inloggning. Små bokstäver måste användas.

Fält 2:

Krypterad version av lösenordet, password. Inte ens super-user kan lista ut verkliga lösenordet. Ett tomt fält anger att inget lösenord

krävs vid inloggning. Strängen ***noway** är en omöjlig kod och förhindrar helt inloggning.

Det krypterade lösenordet består av 13 tecken valda ur ett 64-teckens alfabete (., /, 0-9, A-Z, a-z). I position 14 kan finnas ett kommatecken **,** följt av två tecken och en sträng. Kommat och de två tecknen anger att lösenordet skall ha en begränsad giltighet i tiden och editeras in av superuser om användaren ska tvingas ändra sitt lösenord regelbundet.

Första tecknet anger maximala tiden i veckor som samma lösenord får användas. En användare som loggar in efter att maximala tiden har gått tvingas ange ett nytt lösenord. Nästa tecken anger minsta antal veckor som ett lösenord ska användas innan det får ändras. De två tecknen anger antal veckor i området 0..63, som motsvarar ordningsnumret i 64-teckens alfabetet, dvs / = 1 vecka och z = 63 veckor. Om de två tecknen är .. tvingas användaren ge nytt lösenord vid nästa inloggning. Om de två tecknen är ./ kan endast superuser ändra lösenordet.

Sista strängen lagras automatisk och anger när det aktuella lösenordet definierades.

Fält 3:

Användarnummer. Unikt nummer (ett heltal större än eller lika med noll) för varje användare. **root** *måste* ha användarnummer 0, vilket anger att **root** är privilegierad (super-user). Vanliga användare bör ha nummer större än 99.

Fält 4:

Användargruppnummer (ett heltal större än eller lika med noll). Vid normal inloggning blir en användare automatiskt inloggad i den grupp som anges här. Grupp 1 används normalt för standardguppen **other** och vanliga användargrupper bör ha nummer större än 99. Efter inloggning kan användaren byta grupp med kommandot *newgrp*, förutsatt att han känner till gruppens lösenord, eller är listad som gruppmedlem i **/etc/group**. Man behöver inte ange gruppens lösenord vid vanlig inloggning. Jämför filen **/etc/group**.

Fält 5:

Kommentarfält, där förklarande text kan anges. T ex användarens namn eller andra data. Vissa teststrängar som anges här kan användas av kommandot *login* som speciella parametrar. Se *login* i **Referenshandboken**.

Fält 6:

Inloggningsbibliotek, där användaren automatiskt hamnar vid inloggning och dit *cd*-kommandot utan parametrar leder om detta inte ändrats efter inloggning genom omdefiniering av shellvariabeln **HOME**.

Fält 7:

Här anges vilken process som skall startas när användaren loggar in. Här kan anges en annan kommandoavkodare eller en vanlig process. Tomt fält anger att standardkommandoavkodaren shell skall användas, */bin/sh*.

Om en vanlig process anges, kommer användaren endast att kunna utföra den processen och aldrig komma åt systemkommandona om inte den startade processen kan skapa nya processer. Då den angivna processen avslutas loggas användaren automatiskt ut.

Standard IN, UT och FEL-kanaler finns definierade även utan kommandoavkodare. Parametern **TZ**, som anger tidszonen för lokal tid, är också definierad.

Ett annat sätt att automatiskt starta ett program vid inloggning är att lägga ett kommando i filen **.profile**, varvid användaren efter avslutning av detta kommando har tillgång till den vanliga kommandoavkodaren. Om användaren automatiskt skall loggas ut kan kommandot startas med *exec* i **.profile**.

Om detta fält är enbart * så försöker loginprocessen initiera en sekundär inloggning efter att ha bytt rootfilsystem till inloggningsbiblioteket. Detta kräver att inloggningsbiblioteket innehåller alla de systemfiler som krävs, t ex */bin/login* och **/etc/passwd**, eftersom inga filer ovanför det nya root-biblioteket kan användas.

4.2.29 /etc/powerfail

Detta är en kommandofil som automatiskt exekveras då nyckelbrytaren på frontpanelen vrids till läget OFF(MANUAL). Alla processer avslutas då och filsystemet stängs på ett kontrollerat sätt. Systemmeddelanden ges till enheten **/dev/syscon**.

Den exekveras även om en power-fail-insignal detekteras från en ansluten spänningsövervakningskrets.

/etc/powerfail stänger systemet på ungefär samma sätt som kommandot */etc/shutdown*.

Efter en fullständig avstängning av operativsystemet stänger bootprogrammet av spänningen, men enbart om frontpanelens nyckel är i läget OFF. I vissa system sker istället en manuell omstart i detta fall.

4.2.30 /etc/profile

Detta är en gemensam kommandofil som automatiskt bearbetas av kommandoavkodaren efter inloggning till systemet, samt då systemet övergår till enanvändarmod. Den bearbetas före användarens egen **.profile**. Normalt ska **/etc/profile** sätta parametern **TZ** för lokal tid ur filen **/etc/timezone** och visa eventuellt systemmeddelande ut filen **/etc/motd**.

I exemplet nedan läggs även systemets nodnamn in i prompten. Nodnamnet kan ändras med kommandot */etc/mkcfig*. Dessutom visas en lista över aktuella nyhetsmeddelanden ur biblioteket **/usr/news** om

motsvarande kommando finns i systemet. Detta kommando ingår inte i grundsystemet.

Exempel på /etc/profile:

På grund av begränsad radlängd i handboken delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
:
# Set umask
umask 022

# Set timezone
TZ=`cat /etc/timezone`

# Get System name
if [ -f /etc/systemid ]
then
     PROMT=`cat /etc/systemid`
fi

# Do special setups for ordinary users
if [ $LOGNAME != "adm" -a $LOGNAME != "nuucp" -a \
                                $LOGNAME != "sys" ]
then
     if [ $LOGNAME != "root" ]
     then
          # Show message of today
          if [ -f /etc/motd ]
          then
               cat /etc/motd
          fi

          # Set prompt
          PS1=${PROMT}'$ '

          # Check news
          if [ -x /usr/bin/news ]
          then
               news -n
          fi
     else
          # Set prompt for root
          PS1=${PROMT}'# '
     fi
else
     # Set prompt for special users
     PS1=${PROMT}'$ '
fi

# Check mail
if [ -x /bin/mail ]
then
     mail -e && echo "You have mail"
fi

# Export global variables
export TZ PATH PS1
```

4.2.31 .profile

En kommandofil som ska ligga i varje användares inloggningsbibliotek. Den utförs automatiskt av kommandoavkodaren shell när användaren loggar in på systemet och används ofta för att definiera lämpliga styrtecken för den använda terminalen med *stty* och för att definiera kommandovariabler. Kommandovariabler som ska kunna användas av andra program måste "exporteras" med kommandot *export*.

Se även filen **/etc/profile**. Kommandona i den gemensamma filen **/etc/profile** exekveras före **.profile** vid inloggning. Då systemet går in i enanvändarmod, används **/.profile**.

Notera att de terminalstyrtecken som initieras med **.profile** endast gäller så länge användaren är inloggad. Om man byter användarnamn med kommandot *su*, exekveras emellertid inte den nya användarens **.profile** om inte bindestreck (-) ges som tillval till kommandot (*su* -). Dock ändras alltid PATH då *su*-kommandot ges.

I **.profile** kan kommandon anges så att ett visst program startas automatiskt när användaren loggar in. Om detta program startas med *exec* sker automatiskt utloggning då det startade programmet avslutas. Se även **/etc/passwd**.

I **.profile** kan en mängd olika användarspecifika shell-variabler definieras.

Bl a kan terminaltypen sättas, t.ex. **vt100** för de program som använder shellvariabeln TERM och motsvarande hjälpfiler i biblioteket **/usr/lib/terminfo** eller i filen **/etc/termcap**. I dessa filer finns information om styrtecken för olika terminaler, som till exempel markörstyrtecken, styrning av videoattribut m m. **terminfo** innehåller definitionerna i kompilerad form, medan **/etc/termcap** är i textformat.

I **.profile** kan TERM t ex definieras enligt nedan för vt100-kompatibla terminaler.

```
TERM=vt100
export TERM
```

Användaren kan även lägga in egna kommandon i **.profile**, förutsatt att att användaren har läs- och skrivtillstånd i filen. Normalt bör varje användare ha en egen kopia av **.profile** och inte bara en länk till systemets **.profile**.

Även standard shellvariabler, som definierats av loginprocessen kan ändras här, till exempel:

HOME=/usr/adam/arbetsfiler betyder att hem-biblioteket ändras från inloggningsbiblioteket som definierades i **/etc/passwd**.

PATH=:/usr/adam/bin:$PATH betyder att kommandon som anges utan biblioteksnamn söks först i aktuellt bibliotek, sedan i biblioteket **/usr/adam/bin** och först därefter i de bibliotek som är systemstandard enligt den definition på PATH som sattes upp vid inloggningen. Första tomma fältet före : anger att det aktuella biblioteket skall sökas först. Som standard söks i **/bin** och i **/usr/bin** om PATH inte omdefinieras.

Exempel på en **.profile**-fil:

```
stty -tabs cr0 ff0 nl0 echoe erase '^H' kill '^X'
TERM=vt100
echo 'Terminaltyp (TERM) är satt till ' $TERM
PATH=$PATH:/usr/ucb
export TERM PATH
```

4.2.32 /etc/rc

Kommandofil, som automatiskt bearbetas när systemet övergår till fleranvändarnivå (systemnivå 2, dvs. bootnivå 3). Den bearbetas även om man med kommandot *telinit* **2** ändrar systemnivån till 2. Det är ett kommando i **/etc/inittab** som startar **/etc/rc** på systemnivå 2.

I **/etc/rc** finns t ex kommandon för att rensa bort gamla temporärfiler och för att starta upp speciella processer för initiering av systemet, t ex processer för hantering av printerspoolern.

Generellt bör annars processer som skall finnas permanent i systemet startas med ett eget kommando i **/etc/inittab**. Detta gäller till exempel nätverksprogram eller speciella filhanterare. Vissa program måste däremot startas i **/etc/rc**, nämligen program som skapar egna processer och själva avslutas medan de nya processerna lever kvar i systemet. Denna typ av program bör även stoppas med hjälp av särskilda kommandon i shell-kommandofilen */etc/shutdown* och helst även i */etc/powerfail.*

Standard IN, UT eller FEL-kanaler är riktade till enheten **/dev/syscon** under exekvering av kommandona i **/etc/rc**. Denna länkas normalt till den fysiska huvudkonsolen **/dev/console** genom ett kommando i **/etc/inittab** som exekveras före **/etc/rc**. Se kapitlet om *shell* i **Användarhandboken** om standard IN/UT/FEL-kanaler.

Nedan finns ett exempel på hur filen **/etc/rc** kan se ut, men exakta utseendet beror på vilka extra tillval systemet är utrustat med. Rader med tecknet # i början är kommentarrader och innehåller ibland exempel på hur vissa tillval startas. Om dessa ska aktiveras tas då bara kommentartecknet (#) bort.

Exempel på filen /etc/rc:

På grund av begränsad radlängd i handboken delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
:
# Spooler system
if [ -x /usr/bin/lp ]
then
      SPOOLER=lp
else
      SPOOLER=lpr
fi
# Set timezone
TZ=`cat /etc/timezone`
export TZ
# Run commands only if run level is 2
set `who -r`
if [ $7 = 2 ]
then
      # Print signon on console
      echo "Diab Data AB "`uname -svr`" Virtual"
      date
      # Clean up temporary libraries
      (cd /tmp; for FILE in *; do rm -rf /tmp/¤FILE; done )
      (cd /usr/tmp; for FILE in *; do rm -rf /usr/tmp/¤FILE; done)
      # Remove logins disabled file
      rm -f /etc/nologin
      # Move old cronlog and create a new one
      if [ -f /usr/lib/cron/log ]
      then
            mv /usr/lib/cron/log /usr/lib/cron/OLDlog
                                 > /usr/lib/cron/log
      fi
      # Remove all uucp lock files
      rm -f /usr/spool/uucp/LCK* /usr/spool/locks/LCK*
      # Remove accounting lock file
      rm -f /usr/adm/acct/nite/lock*
      # Move old sulog file
      mv /usr/adm/sulog /usr/adm/OLDsulog >/dev/null 2>&1
      # Start cron
      if [ -x /etc/cron ]
```

```
        then
                rm -f /usr/lib/cron/FIFO
                n=`ps -lax | fgrep cron | wc -l`
                if [ $n = 0 ]
                then
                        nice -16 /etc/cron
                        echo "CRON started"
                fi
        fi
        if [ "$SPOOLER" = lpr ]
        then
                # Remove lock files
                rm -f /usr/spool/lpd/lock /usr/spool/lpd/lpdctl
                rm -f /usr/spool/lock/* /usr/spool/lpd/ERRLOG
                nice -20 /usr/lib/lpd &
                echo "LPD spooler started"
        else
                # Start lpscheduler
                n=`ps -lax | fgrep lpsched | wc -l`
                if [ $n = 0 ]
                then
                        rm -f /usr/spool/lp/SCHEDLOCK
                        nice -20 /usr/lib/lpsched
                        echo "LP scheduler started"
                fi
        fi
        # Start accounting
        # /bin/su - adm -c /usr/lib/acct/startup >/dev/null 2>$1
fi
```

4.2.33 /etc/rmuser

Kommando med vilket en privilegierad användare (super-user) kan ta bort användare i systemet. Inloggningsbiblioteket kan tas bort automatiskt, liksom gruppdefinitionen.

4.2.34 /etc/setspeed

/etc/setspeed sätter hastigheten eller andra parametrar för olika enheter. Vanligen används setspeed till skrivare tillsammans med skrivarsystemet lpr. Man kan blanda hastigheter och enheter i uttrycket, som t ex:

```
/etc/setspeed -s2400 /dev/lp -s1200 /dev/letter &
```

Detta sätter hastigheten för **/dev/lp** till 2400 baud och hastigheten för **/dev/letter** till 1200 baud.

/etc/setspeed kan även ändra övriga kommunikationsparametrar, som då ges på samma format som till kommandot *stty*. Detta sker om optionen **-h** anges. Notera att i detta fall skall **<** anges framför enhetsnamnet. Exempelvis om skrivaren kräver 8-bits data och ingen paritet förutom 2400 baud:

```
/etc/setspeed -h 2400 -parity -istrip </dev/lp &
```

/etc/setspeed ligger kvar som en "demon". När den tas bort , t ex vid utloggning om enheten är en terminal, upphör de uppsatta hastigheterna att gälla.

Vid fel i setspeed p g a felaktiga parametrar startar ej programmet.

Om */etc/setspeed* lägges in i **/etc/rc** för att automatiskt startas då systemet startas, kan det ligga aktivt hela tiden. I **/etc/rc** skall då kommando-raden med */etc/setspeed* avslutas med tecknet **&**, för att tillåta

/etc/rc att fortsätta bearbetas även vid fel från */etc/setspeed* om t ex en angiven skrivare ej är ansluten.

Tillval

-in	Sätter upp inhastigheten till n.
-on	Sätter upp uthastigheten till n.
-sn	Sätter upp in- och uthastigheterna till n, där n väljs ur följande hastigheter: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, exta, extb.
-h parametrar < enhet	Med detta tillval kan alla kommunikationsparametrar anges, exakt som för kommandot *stty*, vilket nämligen anropas av setspeed. Mellanslag krävs efter **-h**.

4.2.35 /etc/shutdown

Detta kommando kan användas då operatören vill gå ner i enanvändarmod eller vid manuell initiering av en fullständing avstängning av systemet. Endast användare med namnet **root** som är inloggade på systemets fysiska huvudkonsol (**/dev/console**) tillåts ge kommandot */etc/shutdown*.

Meddelanden sänds till alla inloggande som får en viss maximal tid på sig att avsluta. Därefter avbryts alla processer i systemet. Speciella processer för t ex terminalkoncentratorer eller nätverk stängs av på ett riktigt sätt innan filsystemet stängs av och systemet tas ned.

Om kommandot ges med optionen **-k** stängs systemet helt, tills texten **System halted** visas på huvudkonsolen.

Kommandot */etc/shutdown* beskrivs i detalj i kapitel 2.

4.2.36 /bin/stty

/bin/stty är ett kommando som används för att visa eller ändra kommunikationsparametrar för aktiva terminaler. Se **Referenshandboken** där kommandot och dess parametrar beskrivs i detalj. För återställande av förstörda terminalparametrar används kommandot enligt nedan från den berörda terminalen. Detta kan vara användbart om något program avslutats felaktigt, vilket kan visas sig genom att tecken skrivna på tangentbordet inte ekas på bildskärmen.

CTRL-J ges både före och efter kommandot.

```
CTRL-J /bin/stty sane CTRL-J
```

Detta kommando återställer en terminal till en rimlig terminaluppsättning. Baudrate påverkas ej av parametern *sane*.

4.2.37 /dev/syscon

Detta enhetsnamn anger den virtuella systemkonsolen och skall i normala fall vara länkad till samma serieport som systemets fysiska huvud-

konsol. D v s **/dev/syscon**, **/dev/console** och **/dev/systty** är alltså normalt samma serieport.

Den virtuella systemkonsolen används som terminal i enanvändarmod och kan temporärt länkas till en annan terminal. Vid övergång från fleranvändarmod till enanvändarmod, länkas alltid **/dev/syscon** till den terminal varifrån kommandot *telinit* **s** gavs. I systemet blir detta emellertid alltid huvudkonsolen (**/dev/console**), eftersom övergången alltid skall ske med kommandot */etc/shutdown* och alltid från huvudkonsolen, för att säkerställa att olika optionsprogram blir korrekt avslutade. Vid återgång till normal fleranvändarmod (systemnivå 2 i filen **/etc/inittab**) länkas den virtuella systemkonsolen till den fysiska huvudkonsolen genom ett kommando i **/etc/inittab**.

Terminalparametrarna för **/dev/syscon** i enanvändarmod sätts enligt den andra parametergruppen (*param2*) i tabellen console i **/etc/gettydefs**.

4.2.38 /dev/systty

Detta enhetsnamn är synonymt med **/dev/console** och skall alltid vara länkade till samma fysiska serieport och anger var systemets huvudkonsol finns.

4.2.39 /bin/telinit

Detta är ett kommando med vilket operatören kan byta systemnivå och tvinga *init* att gå genom **/etc/inittab** på nytt med den nya nivån. Därvid kan en annan kombination av processer aktiveras. *telinit* länkar automatiskt **/dev/syscon** till den terminal från vilken *telinit* givits, men om inte enanvändarmod väljs återställs normalt länkningen av ett kommando i **/etc/inittab**. */bin/telinit* är länkat till */etc/init*.

Den nya systemnivån anges som parameter till kommandot. Parametern kan vara:

0..6	Byt till ny systemnivå. Standard är 2.
q	Gå genom **/etc/inittab** igen, men bara om filen ändrats.
a, b, c	Aktivera endast de processer som i **/etc/inittab** markerats med systemnivå a, b eller c. Detta kommando ändrar *inte* systemnivån och påverkar inte övriga processer direkt.
s	Byt till enanvändarmod. OBS! Använd normalt istället kommandot */etc/shutdown*.
h	Stänger av systemet direkt. Alla processer stoppas och filsystemet stängs. Obs! Använd normalt istället nyckelbrytaren eller kommandot */etc/shutdown -k*. Med *telinit h* får processerna mycket kort tid för att avslutas (10 sek) innan systemet stoppar.

Med **q** byts inte systemnivån utan *init* tvingas gå genom filen **/etc/inittab** igen, men endast om den ändrats. Med **s** kan byte ske till enanvändarmod, men detta bör i normala fall göras med kommandot */etc/shutdown* för att vara säker på att alla övriga processer blir korrekt avslutade.

Ett exempel kan ses i beskrivningen av **/etc/inittab** nedan.

För att lista nuvarande systemnivå kan kommandot *who -r* användas.

4.2.40 /etc/termcap

Detta är en parameterfil med terminalparametrar. Vid leverans är **/etc/termcap** kompatibel med de kompilerade filerna i biblioteket **/usr/lib/terminfo**. Vilken terminaltyp som används bestäms av shellvariabeln **TERM**. Denna definieras vanligen i användarens **.profile** fil.

/etc/termcap är en printbar textfil där parametrarna är uttryckta i klartext.

Vilka terminaltyper som finns definierade i **/etc/termcap** kan ses genom att lista anbart rubrikraderna med följande kommando. Glöm inte apostroferna kring tecknet |.

```
fgrep '|' /etc/termcap
```

I listan kan raden för adm3a t ex se ut som följer. Rubrikraderna består av fält, begränsade av tecknet |. Andra fältet (adm3a nedan) används normalt som terminaltyp i shell-variabeln **TERM**.

```
la|adm3a|3a|lsi adm3:\
```

Efter rubrikraden för varje terminaltyp följer ett antal rader med koder som gäller just den terminaltypen, exempelvis hur markören ska styras, hur många rader som finns på bildskärmen, koden för att rensa bildskärmen, koder för att styra videoattribut på skärmen m m.

och hela filen kan ses med:

```
cat /etc/termcap
```

4.2.41 /usr/lib/terminfo

Detta bibliotek innehåller parametrar för styrning av de terminaler som kan användas till systemet, t ex markörstyrning, bildskärmens storlek med mera. Informationen finns i kompilerad form och data för varje terminaltyp ligger i en egen fil. Biblioteket **/usr/lib/terminfo** är indelat i underbibliotek så att alla datafiler för alla terminaltyper som börjar med samma bokstav ligger i samma underbibliotek.

Exempel:

Parametrar för terminaltypen twist finns i filen **/usr/lib/terminfo/t/twist**.

Vilka terminaler som är inkluderade i **/usr/lib/terminfo** kan listas med *ls*-kommandot, då filernas namn anger terminaltypen.

```
ls /usr/lib/terminfo/*/*
```

Vid leverans är filen **/etc/termcap** kompatibel med filerna i **/usr/lib/terminfo**.

4.2.42 /etc/timezone

Denna fil innehåller en textsträng som läses in av **/etc/profile** och definierar shellparametern **TZ**. **TZ** är även definierad för processer som startas direkt av init-processen.

TZ innehåller en sträng på tre tecken som indikerar lokal tid i utskriften från kommandot date, samt tidsskilladen i timmar mellan GMT (Greenwich Mean Time) och lokal tid.

I exemplen nedan finns två varianter av teckensträngen, MET = Middle European Time (vintertid) och MDT = Middle european Daylight saving Time (sommartid).

Systemet arbetar alltid i GMT-tid, medan kommandon som *date*, *ls* **-l** med flera anger lokal tid, utgående från parametern **TZ**.

Vid omställning mellan vinteroch sommartid ska strängen i **/etc/timezone** ändras. För svensk vinter/sommartid skall **/etc/timezone** innehålla följande strängar:

Svensk normaltid (-1 timme från GMT):

```
MET-1
```

Svensk sommartid (-2 timmar från GMT):

```
MDT-2
```

Systemet kan göra automatisk omställning mellan sommar och vintertid enligt US Daylight Saving Time, enligt de tidpunkter på året som är standardiserade i USA. Detta aktiveras genom att **TZ** utökas med ytterligare en teckensträng, varvid den andra strängen används på sommaren och den första på vintern. **TZ** kan då t ex vara EST+6EDT.

5

5. Systemvård

5. Systemvård

5.1. Förebyggande underhåll

Med förebyggande underhåll minskar risken för allvarligare fel i ett datorsystem. Vid varje uppstart av datorn görs automatiskt vissa systemtester av maskinvaran.

Systemoperatören bör regelbundet göra backup av de ingående filsystemen och se till att datorns maskinvarumiljö är bra, att damm inte samlas kring och i datorn, och att värme och fuktighet och elektriska störningar inte äventyrar funktionen hos systemet.

Kommandot *fsck* ska endast användas av systemadministratören och endast på skivminnen som inte är tillgängliga i systemet. Kommandot skall användas under laddningsprogrammet, i bootnivå 1 efter manuell uppstart, om interna winchesterskivminnet skall testas.

Skivminne, både disketter och winchesterminne, kan testas med kommandot *fsck*. Normalt förstörs inga data.

Om systemet upptäcker fel i filstrukturen rapporteras detta. Med kommandot *fsck* är det möjligt att försöka reparera den filstruktur som är förstörd.

Om kommandot *fsck* hittar filer eller bibliotek som inte är knutna till filsystemet, länkas dessa vid reparationen till biblioteket **/lost+found**. I detta bibliotek finns det alltså möjligheter att återfinna data som gått förlorade vid reparationen.

Nedan är ett exempel på hur *fsck* kan användas för att testa root-filsystemet på den interna winchesterenheten. Systemet stoppas först och startas upp i manuell mod till laddningsprogrammet. Uppstarten kan här ske från det interna skivminnet.

Se i övrigt kommandot *fsck* i **Referenshandboken**.

Test av root-filsystemet från laddningsprogrammet

```
>/etc/fsck -rr si(X,Y)

Checking si(X,Y)
File system:  root        Volume: si0
Volume size =   82Mb,
Block size = 2048,13312 I-nodes (13312 cont.)

Phase 1 - Check I-nodes
Phase 2 - Check Pathnames
Phase 3 - Check Unreferenced files
Phase 4 - Check Link Counts
Phase 5 - Check Bitmap

Total number of files:    662
Total number of blocks: 41472,  37136 - free blocks
```

Tillvalet **-rr** ska alltid anges i laddningsprogrammet. Parametern si(X,Y) anger den enhet som filsystemet befinner sig på. Denna ska anges i samma form som vid manuell uppstart från enheten enligt kapitel 2.

Ett icke mountat filsystem, t.ex. på en diskett eller yttre skivminne, kan testas från enanvändarmod eller fleranvändarmod, men då ska istället enhetsnamnet anges som det definierats i **/dev**. För test av ett filsystem på diskett ges till exempel följande kommando:

```
/etc/fsck /dev/mf0
```

I kapitel 15 listas de olika enhetsnamn som används i systemet.

5.2. Test av filsystem - fsck

Testprogrammet *fsck* för D-NIX filsystem är interaktivt och både testar och reparerar filsystemet. *fsck* använder redundant information i systemfilerna för att utföra flera olika tester av att filsystemet är korrekt. Om något inte stämmer skrivs det ut och operatören kan välja att låta det vara eller beordra *fsck* att försöka rätta felet.

Permanenta fel i filstrukturen kan uppstå om systemet avbryts medan uppdatering av filstrukturen sker eller om fysiska fel uppstår på det externa minnet. *fsck* kan ofta reparera filsystemet genom att utnyttja information om i vilken ordning operativsystemet skriver ner de olika delarna av strukturen till det fysiska filsystemet.

Nedan beskrivs hur filsystemet normalt uppdateras av operativsystemet, tänkbara orsaker till fel och hur *fsck* eventuellt kan reparera felen. De frågor som programmet kan ställa till operatören behandlas och vad dessa innebär.

Senare i detta kapitel finns en lista över de olika felmeddelandena och möjliga svar operatören kan ge. Där beskrivs vad varje felmeddelande betyder och eventuella följdfel som kan uppträda.

5.2.1 Allmänt

Innan operativsystemet laddas in och startas görs normalt en test av att filsystemet är korrekt med *fsck*. Detta hindrar systemet från att skriva ner data till ett filsystem vars struktur redan är felaktig, med oberäkneliga resultat. Om fel upptäcks måste dessa rättas innan filsystemet används.

I det följande beskrivs hur D-NIX filsystem är uppbyggt, hur det kontinuerligt uppdateras och hur filsystemet kan bli felaktigt. Därefter beskrivs hur *fsck* kan försöka att korrigera filsystemet.

5.2.2 Uppdatering av filsystemet

Varje dag kan hundratals filer skapas, ändras och tas bort. Varje gång en fil ändras uppdaterar operativsystemet flera delar i den struktur som utgör ett filsystem. Dessa uppdateringar måste alla skrivas ner till skivminnet för att filsystemet ska vara korrekt. För att förstå vad som kan ha hänt då ett permanent fel uppstått i filsystemet är det viktigt att gå genom i vilken ordning de olika uppdateringarna sker. Med användning av denna kunskap kan kommandot fsck försöka göra rättelser i filstrukturen.

Operativsystemet innehåller minnesbuffertar med block som ska skrivas till filsystemet. Då något skrivs till filsystemet skrivs först till dessa minnesbuffertar varefter dessa ställs i kö för att skrivas till det fysiska filsystemet, t.ex. ett skivminne. Fysisk skrivning sker inte förrän motsvarande buffertutrymme behövs till annat ändamål, om systemrutinen flush anropas eller när kommandot *sync* ges. Alla block i minnes-

buffertarna skrivs dessutom ner regelbundet, var 30:e eller 60:e sekund (se kommandot */etc/mkcfig*).

Det finns fyra delar som uppdateras i filstrukturen. Dessa är inoder, indexblock, datablock (bibliotek och filer) och bitmapblock. Förutom filstrukturen finns några sektorer på skivminnet som innehåller speciella data.

Inoder

En i-nod innehåller den egentliga informationen om en fil. Informationen i en i-nod är typen av fil (bibliotek, data eller specialfil), antalet bibliotekspekare som är länkade till filen, en lista över alla block som tillhör filen, storleken av filen och ägare samt modifieringstider.

I-noden skrivs till filsystemet när filen som motsvarar i-noden "stängs" efter att modifieringar gjorts. Mer exakt ställs de i kö för fysisk skrivning enligt ovan.

Indexblock

Det finns tre typer av indexblock - enkla, dubbla och tredubbla. Ett enkelt indexblock innehåller en lista med pekare till de datablock som ingår i filen. Ett dubbelt indexblock innehåller en lista med pekare till enkla indexblock, som i sin tur listar datablock. Ett tredubbelt indexblock innehåller en lista med pekare till dubbla indexblock, som i sin tur har pekare till enkla indexblock med pekare till datablock.

Indexblock skrivs till filsystemet närhelst de ändrats och markerats klara av operativsystemet. Mer exakt ställs de i en kö enligt ovan för att skrivas ner till det fysiska minnet.

Datablock

Ett datablock innehåller enbart filens datainformation eller bibliotekets poster med ingående filer. Varje post i ett bibliotek består av ett filnamn och numret på filens i-nod.

Datablock skrivs till filsystemet närhelst de ändrats och markerats klara av operativsystemet, men enligt ovan skrivs de inte fysiskt till det externa minnet direkt utan ställs i kö.

Bitmap blocks

Bitmappen innehåller information om alla block på ett fysiskt filsystem som inte används till i-noder, indexblock, datablock eller bitmapblock.

Bitmapblock skrivs till filsystemet närhelst de ändrats och markerats klara av operativsystemet, men enligt ovan skrivs de inte fysiskt till det externa minnet direkt utan ställs i kö.

Det finns även en fil som kallas badspot. Där finns alla block som har fysiska fel på filsystemet och som alltså aldrig skall användas. Denna fil skrivs normalt enbart vid initiering av filsystemet.

Valbar integritetsnivå på filsystemet

Operativsystemet kan använda buffertpoolen på tre olika sätt. Detta påverkar filsystemets integritet vid eventuell systemkrasch. Det påverkar även systemets prestanda.

Normalt används buffertpoolen för att erhålla bästa prestanda, dvs skivminnesblock som används ofta skrivs inte ner till skivminnet förrän systemanropet *sync* eller *flush* utförs. Detta gäller när integritetsnivå 0 satts upp, vilket är standard.

På nivå 1 skrivs alla biblioteksblock och inod-block i buffertpoolen ner till skivminnet vid varje ändring. Detta ger en sänkning av prestanda i huvudsak vid skapande och borttagande av filer.

På nivå 2 skrivs även alla index- och datablock i buffertpoolen ner till skivminnet vid varje uppdatering. Detta sänker prestanda vid alla skrivningar.

Integritetsnivån är en systemparameter som kan ändras med kommandot */etc/mkcfig*.

Bitmappen på skivminnet anses som överflödig information och skrivs enbart ner på normalt sätt oberoende av integritetsnivån, eftersom prestanda skulle påverkas alltför mycket om den skulle skrivas oftare. Därför är det viktigt att programmet */etc/fsck* alltid utförs efter en systemkrasch, även om en hög integritetsnivå angivits.

Bootsektorer på skivminnet

Ett skivminne innehåller några sektorer med fysisk information. Dessa nämns ibland vid allvarligare felmeddelanden.

Sektor 0 på skivminnet kallas bootsektor och kan innehålla information om skivminnet och filsystemet som gör att datorn kan startas (bootas) från skivminnet. Där anges bl a hur datorn hittar laddningsprogrammet på skivminnet.

Sektor 1 kallas VDS pekarblock och innehåller bl a en pekare till VDS-blocket (se nedan) samt den flagga som visar om filsystemet uppdaterats korrekt vid avstängning.

Sektor 2 kallas Init Record och innehåller viss formateringsinformation för skivminnet, som behövs för att systemet skall hantera skivminnet korrekt.

En ytterligare sektor (som vanligen är sektor 4) kallas VDS-block, eller Volume Descriptor Block. Där finns pekare till olika delar av filsystemet, som t ex var inod-listan startar, var bitmapblocken finns m m. Dessutom finns här plats för ett valfritt namn på filsystemet, vilket kan lagras och läsas med kommandot */etc/labelit*.

Biblioteket /..systemfiles

Detta bibliotek är knutet till det skivminne det skapats på och innehåller länkar som gör det möjligt att komma åt viss fysisk information om filsystemet. Biblioteket innehåller följande filer:

badspots	Lista med dåliga block på skivminnet som upptäckts efter senaste formateringen. Bl a */etc/badblk* lagrar information här.
bitmap	Information som visar vilka delar av skivminnet som utnyttjas. Genom denna hittar systemet snabbt ledig plats.
inodelist	Här finns filsystemets inoder. Filen är delvis kontinuerlig för snabbast möjliga åtkomst.
volumedescr	Detta är VDS-blocket, som beskrivs ovan.

5.2.3 Fel i filsystem

Ett filsystem kan bli felaktigt på flera sätt. De vanligaste uppträder om datorsystemet stängs av på felaktigt sätt eller vid hårdvarufel i dator eller skivminnesenheterna.

Felaktig avstängning och uppstart

Filsystemet kan bli felaktigt om en avstängning inte sker på rätt sätt, t.ex. om *sync*-rutinen inte anropas innan datorn stoppas eller innan ett tillgängligt filsystem görs fysiskt skrivskyddat eller innan ett tillgängligt filsystem slås av eller tas bort (ex. en diskett).

Filsystem kan även bli ytterligare förstörda om något program tillåts skriva till ett filsystem som redan är felaktigt. Detta förstör normalt helt möjligheten att reparera filsystemet.

Maskinvarufel

Om någon del av maskinvaran, dator eller skivminne, är felaktig kan svåra fel uppstå. Vissa kan vara korrigerbara eller lokala, som när ett fysiskt block på en skiva blir oanvändbart. Andra fel kan vara allvarligare som t.ex. om styrenheten till skivminnet är trasigt.

5.2.4 Att upptäcka och korrigera fel i filsystem

Ett passivt filsystem, d v s ett filsystem som inte är tillgängligt och utan att något är skrivet till filsystemet som inte ännu fysiskt överförts, kan testas för att upptäcka eventuella avvikelser från den givna strukturen genom att jämföra överflödig information i filsystemet. Överflödig information läses från filsystemet eller beräknas utifrån andra kända data. Det är viktigt att inga processer tillåts skriva till filsystemet under testen på grund av att *fsck* går genom filerna flera gånger.

Om en felaktighet hittas rapporterar *fsck* detta till operatören som får välja om rättelse ska ske eller ej. Det är även möjligt att välja att alltid rätta utan frågor, genom att ange vissa tillval till *fsck*-kommandot.

I detta delkapitel visas hur felaktigheter kan hittas och eventuella rättelser utföras för i-noder, indexblock, datablock med biblioteksposter och bitmapblock. Dessa rättelser kan utföras av *fsck*.

I-noder

Listan över i-noder är det första som testas, varvid i-nod nummer 1 är den första. I-nod 0 finns inte. Varje i-nod testas vad gäller formatet, typen, antal länkar, block med dubbla referenser, dåliga block och storleken på i-noden.

Format och typ

Varje i-nod innehåller en typbeteckning (mode). Denna mod beskriver typen och status för i-noden. I-noder finns av fem olika typer:

- Vanlig fil
- Bibliotek

- Specialfil, blocktyp
- Specialfil, teckentyp
- FIFO

Om en i-nod inte är en av dessa anses den vara felaktig. I-noder kan befinna sig i en av tre stadier - oanvänd, använd eller ingendera. I sista fallet är det fel. En i-nod kan få detta felaktiga status om felaktiga data skrivits till i-nod-listan genom t.ex. ett hårdvarufel. I detta fall kan *fsck* enbart radera i-noden (filen).

Antal länkar

I varje i-nod finns information om antal länkar, dvs antal pekare med olika filnamn i ett eller flera bibliotek som pekar på samma i-nod. *fsck* kontrollerar att länkantalet stämmer med verkliga antalet pekare som hittas i bibliotekshiearkin, med start från root-biblioteket och nedåt i hela filsystemet.

Om det lagrade länkantalet är skilt från noll och verkliga antalet pekare är noll, betyder det att inget filnamn finns för i-noden. Om det lagrade länkantalet är skilt från noll och inte samma som antalet pekare, betyder det att en biblioteksposts med pekare har lagts till eller tagits bort utan att i-noden uppdaterats.

Om det lagrade länkantalet är skilt från noll och men inga pekare finns, kan *fsck* länka den gömda i-noden till ett temporärt filnamn i systembiblioteket **/lost+found**. Om länkantalet och antal pekare är olika, men skilda från noll kan *fsck* korrigera det lagrade länkantalet i i-noden.

Block med dubbla referenser

I varje i-nod finns en lista med pekare till datablock eller en lista med pekare till indexblock. Dessa pekar på alla block i filsystemet som tillhör i-noden. *fsck* jämför dessa pekare med en lista över redan upptagna block. Denna lista skapas och utökas eftersom i-noderna gås igenom. Om ett blocknummer redan tillhör en annan i-nod, markeras den som dubbelrefererad. Om dubbelrefererade block hittas söker *fsck* upp den andra i-noden som blocket tillhör och försöker undersöka vilken av i-noderna som är felaktig. Vanligen anses den senast ändrade i-noden som den rätta och den äldsta raderas då. Denna situation kan uppstå bl a om bitmapblocket inte blivit korrekt uppdaterat så att ett block som markerats ledigt även tillhör en fil. När filer med dubbelt refererade block påträffas bör operatören kontrollera om blocken verkligen innehåller data som hör till filen och i annat fall manuellt ta bort motsvarande area ut filen.

En stor mängd dubbelrefererade block i en i-nod kan bero på att ett indexblock inte skrivits ner till det fysiska filsystemet. *fsck* frågar då operatören om båda i-noderna ska raderas.

Dåliga block

I varje i-nod listas direkt eller via indexblock, alla block som tillhör i-noden. *fsck* testar om blocknumret ligger inom de min- och maxgränser

som gäller för filsystemet. Om blocknumret är utanför gränserna anses det som ett felaktigt blocknummer. Om det finns ett stort antal felaktiga blocknummer i en i-nod kan det bero på att ett indexblock inte skrivits ner till det fysiska filsystemet. *fsck* frågar då operatören om i-noden ska raderas.

Kontroll av filstorleken

Varje i-nod har ett 32-bits (4 bytes) fält som anger antal databytes i den fil (eller bibliotek) som är associerad med i-noden. Denna storlek kontrolleras om den är rimlig. T.ex. ska biblioteksfilers storlek vara en multipel av 16 och antalet blockpekare ska stämma med filens storlek.

En i-nod för en biblioteksfil i systemet har en viss bit satt i typbeteckningen (mode). Storleken av en sådan fil måste vara en multipel av 16 eftersom varje biblioteksspost är 16 bytes (2 bytes med i-nodnummer och 14 bytes för fil- eller underbiblioteksnamnet).

fsck skriver bara ut en varningstext vid sådana fel i en biblioteksfil eftersom det inte finns tillräcklig information för att kunna korrigera felet.

En grov test av filstorleken erhålls genom att räkna ut hur många block som krävs för att rymma filen och jämföra med antalet block som tillhör i-noden. Antalet bytes divideras med blockstorleken och detta avrundas uppåt. Dessutom adderas ett block för varje indexblock som hör till i-noden. Om det verkliga antalet block skiljer sig från detta beräknade antal, varnar *fsck* för detta.

Indexblock

Indexblock tillhör alltid en i-nod och är egentligen enbart en utökning av listan över datablock som tillhör i-noden.

Fel i indexblock kan vara dubbla referenser till datablock eller felaktiga blocknummer enligt ovan. Därutöver kan indexblocket självt felaktigt anses även tillhöra en annan i-nod.

Datablock

De två möjliga typerna av datablock är vanliga datablock och biblioteksblock. Vanliga datablock innehåller de data som lagrats i filen. Biblioteksblock innehåller biblioteksposter.

fsck kan inte testa vanliga datablock.

Varje biblioteksblock testas vad gäller formatet och storleken. Varje post i biblioteksblocket innehåller filnamn och i-nodnummer till motsvarande fil. I-nodnumret testas och måste peka på en befintlig korrekt i-nod. Inodnumret ska vara inom min/max-gränserna för filsystemet och filerna "." och ".." ska finnas korrekt definierade och peka på i-noder av typen 'bibliotek'. Biblioteksblocket ska själv vara refererat i det överordnade biblioteket i filsystemet.

Om, i en post i ett bibliotek, i-nodnumret pekar på en i-nod som inte finns definierad, tar *fsck* bort denna post. Detta kan bl a uppstå om datablocket som innehöll biblioteksposten skrivits ner till filsystemet medan motsvarande i-nod inte skrivits ner.

Om, i en post i ett bibliotek, i-nodnumret pekar utanför i-nodlistan, kan *fsck* ta bort denna post. Detta kan uppstå om felaktiga data skrivits till bibliotekets datablock.

Första biblioteksposten måste vara i-nodnumret för ".". Dess värde ska vara i-nodnumret till biblioteket egen i-nod.

Andra biblioteksposten måste vara i-nodnumret för "..". Dess värde ska vara i-nodnumret för det bibliotek som utgör förälder-biblioteket (eller till bibliotekets egen i-nod om biblioteket är root-biblioteket).

Om någon av dessa i-nodnummer är felaktiga kan *fsck* ersätta dem med korrekta värden.

fsck testar allmänt filsystemets hierarki och hur filer och bibliotek är sammanlänkade. Om bibliotek eller filer hittas som inte refereras i något bibliotek i filsystemet, kan *fsck* länka in dessa i under systembiblioteket **/lost+found**. Detta kan bli följden om i-noder skrivs ner till filsystemet trots att motsvarande biblioteks-datablock inte skrivs ner.

Biblioteket **/lost+found** skapas när filsystemet initieras av kommandot *mkfs*. Detta bibliotek har en fast storlek och kan därför bli fullt vid testning av filsystemet med *fsck*. I detta fall bör filerna i **/lost+found** flyttas eller tas bort innan *fsck* återstartas.

5.3. Fsck felmeddelanden

fsck arbetar i flera faser. Efter initiering, går *fsck* genom filsystemet flera gånger för att rensa filsystemet, testa block och storlekar, filnamn och sökvägar, korsreferenser (länkar), antal länkar och bitmappen (vilken kan byggas upp på nytt).

När ett fel upptäcks rapporterar *fsck* detta till operatören, om detta inte kopplats borts genom en option till kommandot. I de fall operatören ska få välja åtgärd skrivs en fråga ut och operatören skall vanligen svara med **y** (för ja eller yes) eller **n** (för nej). Detta kapitel förklarar de olika meddelandena, möjliga svar och de fel som orsakar felmeddelandet.

Felen som beskrivs i resten av kapitlet är ordnade i den ordning de hittas av *fsck*, genom de olika faser som *fsck* genomlöper.

fsck kan enbart testa filsystem som inte är tillgängliga (se *mount*-kommandot). För att testa rootfilsystemet måste därför datorn startas upp till laddningsprogrammet utan att operativsystemet startas. Alternativt kan systemet startas med en diskett som root-filsystem om det fasta skivminnet skall testas.

Flera tillval kan anges till *fsck*-kommandot. Dessa beskrivs i **Referenshandboken**. Vid exekvering i laddningsprogrammet måste tillvalet **-rr** användas, varvid inga frågor ges men *fsck* försöker göra sitt bästa för att reparera filsystemet.

Faserna är:

- Initiering
- Phase 1: Check i-nodes
- Phase 2: Check pathnames
- Phase 3: Check unreferences files
- Phase 4: Check link counts
- Phase 4b: Fix system files
- Phase 5: Check bit map
- Phase 6: Replace bit map
- Cleanup message

5.3.1 Initiering

Innan en test av filsystemet kan göras, måste vissa tabeller skapas och initieras samt vissa systemfiler öppnas. Denna sektion behandlar meddelanden och fel då detta utförs. Fel kan uppkomma i kommandoraden till *fsck*, på grund av begränsat minnesutrymme, då filer öppnas, på grund av fel i filstatus eller storlek av systemfiler och då temporära filer önskas.

Invalid option -X

X är en ej tillåten option till *fsck*. Tillåtna tillval är **-V -y -n -s -t** och **-rr**. Vid detta fel avbryts *fsck*. Se beskrivningen av kommandot *fsck* i **Referenshandboken** för mer detaljer.

File name must follow option -t

Optionen **-t** måste åtföljas av ett filnamn på en fil som ska användas som temporär arbetsfil. Vid detta fel avbryts *fsck*. Se beskrivningen av kommandot *fsck* i **Referenshandboken** för mer detaljer.

Cannot get memory

fsck misslyckades i sin begäran till operativsystemet efter virtuellt minne till sina tabeller. Detta kan innebära ett allvarligt problem, som kan kräva insats av servicepersonal. Vid detta fel avbryts *fsck*.

Cannot open /etc/checklist

Standardfilen med en lista över de filsystem som ska testas av *fsck* kan inte öppnas för läsning. *fsck* avbryts vid detta fel. Operatören bör kontrollera att rätt åtkomstmoder angivits för filen så att *fsck* har tillstånd att läsa den.

Can not create FFF

fsck kan inte skapa temporärfilen FFF. Programmet ignorerar då det filsystem som håller på att testas och fortsätter med nästa filsystem. Kontrollera åtkomsttillstånden för filen FFF och/eller biblioteket där filen ska skapas.

Read error at block b (n) - CONTINUE?

Blocket med nummer b (block storlek n) kunde inte läsas.

YES	Fortsätt, blocket som inte kunde läsas fylls med nollor.
NO	*fsck* avbryts.

Write error at block b (n) - CONTINUE?

Blocket med nummer b (block storlek n) kunde inte skrivas till det fysiska skivminnet.

YES	Fortsätt, ignorera felet.
NO	*fsck* avbryts.

Search for VDS-block?

Pekaren till VDS-blocket (volym descriptor block) är förstörd. VDS-blocket innehåller viktig strukturinformation om filsystemet utan vilket filsystemet inte kan tolkas. Om detta block inte hittas avbryts *fsck*.

YES	*fsck* försöker hitta VDS-blocket. Om det hittas får operatören en fråga (med info om blocknumret) om det accepteras eller ej.
NO	*fsck* avbryts.

FATAL ERROR: I-node for i-nodelist is corrupted - try to do a fixup?

I-noden som innehåller pekare till i-nodlistan är förstörd. Om denna inte kan repareras kommer *fsck* att försöka använda den del av i-nod listan som ligger direkt i följd i filsystemet. I detta fall kan vissa filer förloras.

YES	*fsck* försöker återskapa den förstörda i-noden.
NO	Ingen reparation görs, endast den ursprungliga delen av i-nodlistan som ligger direkt i följd i filsystemet används.

5.3.2 Phase 1: Check I-nodes

I denna fas testas i-nodlistan. Fel som kan uppstå är då tabeller för antalet länkar skapas av *fsck*, då i-noder utan eller med fel antal länkar hittas, då dubbelreferenser till block hittas, om filstorleken är fel eller i-nodens format är fel.

Invalid index block in inode list - fix?

I-noden som innehåller pekare till block för i-nodlistan innehåller felaktiga indexblock. Detta kan betyda att i-nodlistan delvis är förstörd.

YES	I-noden repareras.
NO	Ingen åtgärd.

5.3.3 Phase 2: Check Pathnames

I denna fas tas biblioteksreferenser bort som pekar på felaktiga i-noder, vilka ändrats i föregående fas. Här listas fel i i-noden för root-biblioteket eller fel i biblioteksreferenser, t.ex. i-nodnummer som är för stort eller felaktigt eller om blocknummer är dubblerade eller felaktiga.

Root inode not directory (FIX)

I-noden för root-biblioteket (vanligen i-nod nummer 2) är inte av typen bibliotek.

Möjliga åtgärder anges av operatören:

YES	Ändra root-bibliotekets i-nod så att typen blir 'bibliotek'. Om i-nodens datablock verkligen inte är innehåller biblioteksposter, kommer då ett stort antal fel att genereras.

NO	*fsck* avslutas direkt.

Dups/bad in root inode (CONTINUE)

I föregående faser har dubbelrefererade block eller felaktiga blocknummer hittats i i-noden för root-biblioteket (vanligen i-nod nummer 2) i filsystemet.

Möjliga åtgärder anges av operatören:

YES	Ignorera felet i i-noden för root-biblioteket och försök fortsätta testa filsystemet. Om i-noden för root-biblioteket är fel blir det nu många följdfel.
NO	*fsck* avslutas direkt.

Unable to read directory - continue?

En i-nod pekar på ett bibliotek som inte kan läsas.

YES	*fsck* fortsätter och ignorerar biblioteket.
NO	*fsck* avslutas direkt.

I - node is out of range - remove?

En biblioteksposten har ett i-nodnummer som är större än i-nodlistan eller noll, vilket är fel.

YES	Biblioteksposten tas bort.
NO	Felet ignoreras.

Null file name - remove?

En biblioteksposten har ett i-nodnummer men inget filnamn.

YES	Biblioteksposten tas bort.
NO	Felet ignoreras.

Invalid '.'-link - fix?

Posten "." i en biblioteksposten pekar inte till biblioteket självt.

YES	Posten ändras att peka på biblioteket självt.
NO	Felet ignoreras.

Invalid '..'-link - fix?

Posten ".." i en biblioteksposten pekar inte till förälder-biblioteket.

YES	Posten ändras att peka på förälderbiblioteket.
NO	Felet ignoreras.

Multiple LINKS to this DIR - remove?

En biblioteksposten pekar på ett bibliotek som redan har en annan förälder.

YES	Posten tar bort.
NO	Felet ignoreras.

Invalid directory size - fix?

Storleken av ett bibliotek är inte en multipel av 16 bytes.

YES	Storleken justeras.
NO	Felet ignoreras.

Unknown file type - remove?

Ordet mode i i-noden, som ska ange filtypen, är felaktigt.

YES	I-noden tas bort.
NO	Felet ignoreras.

Blocks not aligned - remove?

Indexpekare som pekar på tillhörande block är inte korrekt enligt de blockgränser som gäller i filsystemet.

YES	I-noden tas bort.
NO	Felet ignoreras.

Blocks out of range - remove?

Indexpekare som pekar på tillhörande block pekar utanför fysiska arean för filsystemet.

YES	I-noden tas bort.
NO	Felet ignoreras.

Invalid size of file - remove?

Filstorleken stämmer inte mot antalet tillhörande block.

YES	I-noden tas bort.
NO	Felet ignoreras.

Invalid index pointers - remove?

Indexpekaren som pekar på tillhörande block är felaktig.

YES	I-noden tas bort.
NO	Felet ignoreras.

Blocks claimed by other inode - remove?

Block som tillhör denna i-nod tillhör även en annan i-nod.

YES	I-noden tas bort.
NO	Felet ignoreras.

I-node not allocated - remove?

En biblioteksspost pekar på en i-nod som är markerad ledig.

YES	Posten tas bort.
NO	Felet ignoreras.

Unable to read inode - remove?

En biblioteksspost pekar på en i-nod som inte kan läsas.

YES	Posten tas bort.
NO	Felet ignoreras.

I-node has been deleted - remove?

En biblioteksspost pekar på en i-nod som har raderats av *fsck*.

YES	Posten tas bort.
NO	Felet ignoreras.

'.'-link missing - fix?

Biblioteket innehåller ingen länk till ".".

YES	Lägg till denna post till biblioteket.
NO	Felet ignoreras.

'..'-link missing - fix?

Biblioteket innehåller ingen länk till "..".

YES	Lägg till denna post till biblioteket.
NO	Felet ignoreras.

5.3.4 Phase 3: Check Unreferenced Files

I denna fas testas biblioteksstrukturen ytterligare och fel listas om orefererade bibliotek hittas eller om om referensen till ett föräldrabibliotek är felaktigt.

Father directory: I-node= n - reconnect?

Biblioteket har ett förälderbibliotek med i-noden n och typen 'bibliotek', men det finns ingen referens till biblioteket från detta förälderbibliotek.

YES	Den saknade referensen läggs in i förälderbiblioteket med ett temporärt filnamn.
NO	Förälderbiblioteket ändras inte, men operatören kan välja att lägga en referens i systembiblioteket **/lost+found** istället.

Connect to dir '/lost+found'?

Filen eller biblioteket har inget förälderbibliotek eller biblioteket lades inte in i förälderbiblioteket enligt ovan. **/lost+found** är ett bibliotek med fast storlek, varför det kan bli fullt. I detta fall måste filerna i **/lost+found** gås igenom och flyttas eller tas bort innan *fsck* startas igen.

YES	Filen eller biblioteket läggs in i biblioteket **/lost+found**.
NO	Ingen åtgärd.

Remove file?

Filen eller biblioteket har inget förälderbibliotek och lades inte in i **/lost+found** enligt ovan.

YES	Filen raderas och i-noden nollställs.

NO Ingen åtgärd.

5.3.5 Phase 4: Check Link Counts

I denna fas genomgås länkningsinformationen efter att eventuella korrigeringar enligt fas 2 och 3 gjorts. Felen listade nedan är från orefererade i-noder där inte en referens skapats i föregående fas, fel i länkantalet för filer, bibliotek eller specialfiler eller felaktiga block i filer eller bibliotek.

Invalid link count, I-node= n Is x, should be y - adjust?

I-nodens länkantal stämmer ej överens med antal referenser till i-noden.

YES Länkantalet rättas till.

NO Ingen åtgärd.

I-node = n, is ok but has no father - remove?

I-noden är en korrekt i-nod, men ingen referens till den har hittats och den har inte lagts in i **/lost+found** i en tidigare fas.

YES I-noden och därmed filen raderas.

NO Ingen åtgärd.

I-node is not referenced - remove?

I-noden är en korrekt i-nod, som saknar referenser. Dessutom är i-nodens länkantal noll.

YES I-noden och därmed filen raderas.

NO Ingen åtgärd.

5.3.6 Phase 4b: Fix system files

I-node for I-nodelist was in error and corrected **Update i-node?**

I-noden med pekare till resten av i-nodlistan var felaktig men *fsck* bör kunna återställa den.

YES I-noden uppdateras.

NO Ingen åtgärd.

Fix up cont. part of I-nodelist?

I-noden med pekare till resten av i-nodlistan var felaktig. *fsck* kan endast återställa den del av listan som lagrats i direkt på varandra följande block.

YES I-noden uppdateras.

NO Ingen åtgärd.

I-node for bitmap was in error - fixup?

I-noden som pekar ut bitmapblocken är felaktig.

YES I-noden korrigeras.

NO Ingen åtgärd.

I-node for root directory was in error - fixup?

I-noden som pekar ut root-biblioteket var felaktig.

YES	I-noden korrigeras.
NO	Ingen åtgärd.

I-node for badspot file was in error - fixup?

I-noden som pekar ut filen **badspot** med en bitmap över fysiskt felaktiga block i filsystemet, var felaktig.

YES	I-noden korrigeras.
NO	Ingen åtgärd.

5.3.7 Phase 5: Check bit-map

I denna fas testas bitmappen. Fel kan uppstå på grund av felaktiga block i bitmapfilen, av att block som används eller är oanvända markerats felaktigt i bitmappen eller av att totala antalet block i bitmappen inte stämmer med filsystemet fysiska storlek.

Dessa tester utförs inte om *fsck* opionen **-rr** angivits, varvid den fysiska bitmappen helt ignoreras och återskapas av *fsck*.

5.3.8 Phase 6: Replace bit map

fsck har nu skapat en intern bitmap efter genomgång av och eventuell reparation av hela filsystemet. Om denna beräknade bitmap inte överensstämmer med den fysiska bitmappen kan den rättade bitmappen skrivas ner till filsystemet.

Replace bit map?

Bitmappen som *fsck* beräknat i minnet stämmer ej med bitmappen på det fysiska skivminnet.

YES	Bitmappen på skivminnet uppdateras.
NO	Ingen åtgärd.

5.3.9 Cleanup

*******FILE SYSTEM WAS MODIFIED*******

Detta är ett meddelande som *fsck* ger då korrigering gjorts av filsystemet.

5.4. Konfiguration D-NIX - mkcfig

Det finns möjlighet att ändra vissa parametrar i operativsystemet för optimering av systemet till den aktuella tillämpningen. Dessa parametrar lagras i ett parameterblock i systemfilen och läses in vid uppstart av systemet.

Detta behöver emellertid normalt inte göras annat än i undantagsfall, till exempel om ovanligt många samtidiga processer behövs eller vid ovanligt intensiv användning av stora och många filer. Om behov uppstår, till exempel på grund av felmeddelanden från systemet, bör först programmen kontrolleras. Onödigt stora parametervärden (särskilt för filbuffertarean) tar plats i primärminnet, medan för små värden kan göra systemet långsammare i vissa fall. Om systemet har ett stort primärminne, bör detta i första hand utnyttjas genom att använda en stor area för filbuffertar, om snabb exekvering är viktig.

Kommandot *mkcfig* är interaktivt, men kan även startas med tillval och parametrar för direkt ändring av systemparametrarna. Då *mkcfig* startas interaktivt, visas alla systemparametrarna enligt exemplet nedan, varefter operatören eventuellt ändrar och skriver tillbaka blocket. Därefter kan systemet startas om, efter normal avstängning, t.ex. med */etc/shutdown* **-k**, för att de nya parametrarna skall gälla.

Endast super-user kan använda *mkcfig* och bör befinna sig i root-biblioteket, där normalt operativsystemfilen finns. Gör gärna en backup-kopia av filen innan den ändras. Vid behov kan systemet startas upp med standardparametrar istället för med de som *mkcfig* skapat. I laddningsprogrammet startas då systemet med kommandot: "*/dnix xxxx*".

Kommandot startas interaktivt i biblioteket "/" med

```
/etc/mkcfig dnix
```

där **dnix** är det normalt använda namnet för operativsystemfilen. Lista root-biblioteket för att hitta rätt namn. Texten som *mkcfig* skriver är på svenska eller engelska, beroende på om variabeln **LANGUAGE** sätts till *swedish* eller *english* innan *mkcfig* startas. Exempel:

```
LANGUAGE=swedish
export LANGUAGE
/etc/mkcfig dnix
```

Se **Referenshandboken**, där det även beskrivs hur *mkcfig* används som direktkommando med tillval.

Observera att de verkliga parametervärdena varierar mellan olika datortyper. Använd kommandot *sar* med olika tillval (om detta finns i systemet) för att kartlägga prestanda innan några ändringar införs.

I filen **/usr/adm/dnix.par** sparas automatiskt parametrarna vid första systemstart efter att de ändrats med *mkcfig*.

5.4.1 Konfigureringsparametrar för /dnix

Nedan är ett exempel på en utskrift från *mkcfig*. Fler parametrar kan tillkomma i senare versioner.

```
Max antal aktiva processer...................:  300
Max antal laddade filer......................:  300
Max antal noder..............................: 1000
Filbuffertstorlek i Kb.......................: 700
Max antal öppna filer........................:  400
Max antal fillås.............................:  128
Max antal 'tillgängliga' filsystem...........:    4
Maxstorlek för 'page swap area' i Mb. .......:   48
Max antal meddelande köer....................:   20
Max antal semaforer..........................:   20
Max antal semafor-justering-vid-exit.........:   64
Max antal delbara minnes segment.............:   20
Max antal invalda minnes segment.............:   64
Antal sekunder mellan filsystem uppdatering..:   60
Antal tecken i tty input buffert.............:  255
Minuter innan systemstopp vid strömavbrott...:    0
Antal kontrollerande terminal noder..........:  128
Tillåt ej skrivning på monterade enheter.....:    1
Säkerhetsnivå på filsystemet.................:  0
Maximal storlek på minnespool i kb...........:  128
Maximal storlek på timeout-tabell............:  256
Nod namn.....................................:  'ma'

Vill du ändra konfigureringsparametrar?
   ("j" = ja, "a" = avbryt, övrigt = fortsätt) ..: ej uppdatering
```

Svara med **n** för att avbryta utan ändringar, med **j** för att ändra. Då ger programmet frågor interaktivt. Möjliga värden anges i frågan. Svara bara **RETURN** om de gamla värden ska behållas.
Svara med **RETURN** eller **n** efter att ha gjort ändringar, varvid en ny fråga kommer om parametrarna ska skrivas till operativsystemsfilen, som angivits i *mkcfig*-kommandot.

```
Vill du verkligen ändra parametrarna för dnix?
   ("j" = ja, "n" = nej) ...:___
```

Svara med **j** för att uppdatera filen, med **n** för att avsluta *utan* uppdatering.

Exempel:

Ändra 'Max antal noder' till 1500.

```
Vill du ändra konfigureringsparametrar?
("j" = ja, "a" = avbryt, övrigt = nej) ...:j

Max antal aktiva processer...................:  300
nytt värde (mellan 10 och 2048)? ___          (RETURN)
Max antal laddade filer......................:  300
nytt värde (mellan 10 och 2046)? ___          (RETURN)
Max antal noder..............................:  1000
nytt värde (mellan 50 och 8192)? 1500 RETURN
```

Svara **RETURN** på övriga frågor. Efter alla frågor visas igen alla parametrar, liksom i början samt samma slutfråga.

```
Vill du ändra konfigureringsparametrar?
    ("j" = ja, "a" = avbryt, övrigt = nej) ...:RETURN

Vill du verkligen ändra parametrarna för dnix?
    ("j" = ja, "n" = nej) ...:j
```

Nu skrivs de nya parametrarna ut till filen **dnix** och *mkcfig* avslutas.

Kommandot ps

Kommandot *ps -axl* används för att visa en ögonblicksbild över alla processer som är aktiva i systemet. I exemplet nedan betyder xxx att ytterligare fält finns, men inte visas här. I fältet MEM_SIZE är vänstra värdet kodareans storlek medan det högra värdet anger dataareans storlek i kbytes.

```
ps -axl

UID  TTY      PIP  PPIP STATE PRI MEM_SIZE  xxx   CMD
root -        48   1    ---:S  20  40+72    xxx   cron
lp   -        54   1    ---:S  24  48+80    xxx   lpsched
root console  1492 1    ---:S   4  40+64    xxx   getty
root tty38    1894 1    ---:S   4  40+64    xxx   getty
root -        2234 1    ---:S  20  64+208   xxx   dblm
kj   tty28    2559 1    ---:S  20  40+56    xxx   -sh
```

Kommandot sar

Kommandot *sar* finns enbart i utbyggnadspaketet, men refereras nedan eftersom det kan användas för att kartlägga systemets prestanda. Tillvalen och namnen på parametrarna motsvarande *mkcfigs* parametrar anges. För att samla in och visa värden var 5:e sekund, 99 gånger ges följande kommando med lämpligt tillval:

```
sar -tillval  5  99
```

5.4.2 Parameterbeskrivning

Max antal aktiva processer

Detta är antal TCB som reserveras i systemet, vardera ca 356 bytes. Aktiva processer är de som listas med kommandot *ps - axl*. Om denna parameter ändras bör även flera andra parametrar justeras i motsvarande grad. Om felmeddelandet **End-of-TCB's** kommer, kan det bero på att ett större värde behövs. Kontrollera emellertid först att inte något program felaktigt har skapat för många processer. Normala värden är i området 64 - 400. Med *sar -v* visas denna parameter under namnet **proc**.

Max antal laddade filer

Detta är max antal processer som använder olika kod-areor i minnet. Styrblocket för varje kodarea tar ca 100 bytes. Om flera processer laddas från samma programfil, delar de kodarean i minnet och denna räknas enbart som en laddad fil. Detta kan ses t.ex. för shell (-sh) i listan från *ps -axl*. Normala värden är i området 64 - 400. Med *sar -v* visas denna parameter under namnet **text**.

Max antal noder

Detta är totala antal noder i systemet, som ligger i en pool och delas av alla processer. Noder används enligt tabellen nedan och varje nod kräver ca 132 bytes. Om felmeddelandet **End-of-Nodes** kommer kan parametervärdet behöva ökas. Kontrollera först att det inte är något programfel. Fler noder kan behövas bl a om antal filbuffertar ökas genom att parametern för filbuffertstorleken ökas. (Se nedan). Ett lämpligt värde är 5 * 'antalet aktiva processer'.

Normala värden är i området 256 - 2048. Med *sar -v* visas denna parameter under namnet **nodes**.

- En nod för varje öppen fil eller fysisk enhet.
- En eller flera noder för varje I/O-beställning till en buffrad extern enhet, t.ex. ett skivminne. Flera noder behövs då read-ahead används.
- En nod för varje NoWait-beställning från en process. Detta är inte så vanligt bland normala kommandon.
- En nod för varje I/O-beställning till en I/O-hanterare, vilken i sin tur behöver noder för sina I/O-beställningar.

Filbuffertstorlek i 1k-block

Detta är systemets total area i kbytes för filbuffertar. Varje buffert kräver 1 kbyte + ett 56 bytes styrblock. Med stor buffertarea behålls fler skivsektorer i buffertarean och färre fysiska skivaccesser krävs, vilket kan minska exekveringstiden för vissa program, särskilt när temporära filer används. Å andra sidan är mer primärminne låst som buffertarea, vilket kan orsaka högre swap-frekvens om många eller stora program körs. Process-swapping sker alltid via fysiska skivaccesser och använder inte filbuffertar. Normala värden är i området 64 - 700 eller upp till 2000 för skivminnesintensiva system med stort minne. Med *sar -b* visar parametrarna **%rcache** och **%wcache** träffprocenten för filbuffertarna (cache hit rate). %rcache (läs-cache) bör vara minst 80 (%) i de flesta system, helst över 90%. Med program som gör starkt slumpmässiga läsningar och skrivningar kan emellertid den optimala träffprocenten vara lägre.

- Öka eller minska filbuffertparametern för optimering av systemets prestanda!

Max antal öppna filer

Detta är max antal samtidigt öppna filer, globalt i systemet. Varje filstyrblock är 156 bytes. Denna parameter kan möjligen behöva ökas vid tillämpningar med ovanligt många samtidigt öppna filer. Felmeddelandet **File-Table-Overflow** ges om parametern inte räcker till. Ett värde motsvarande 3 * 'antal aktiva processer' är lämpligt, men det optimala värdet beror starkt på vilka typer av program som används. Normala värden är i området 96 - 400. Med *sar -v* visas denna parameter under namnet **file**.

Max antal fillås

Systemparameter, som anger hur många samtidiga dataareor i systemet som totalt kan hållas låsta. Varje lås kräver 32 bytes. En eller flera areor i en eller flera filer kan låsas mot åtkomst från andra användare, medan uppdatering sker från en process. Ett normalt värde är 128, men med stora databasprogram kan detta behöva ökas. Med *sar -v* visas denna parameter under namnet **lock**.

Max antal tillgängliga filsystem

Detta är max totalt antal samtidigt 'tillgängliga' fysiska filesystem inklusive root-filsystemet, d v s normalt antalet skivminnen. Varje mountstyrblock kräver 172 bytes. Obs! Dessa resurser används inte när filhanterare mountas eller när fysiska enheter accessas direkt (utan att vara tillgängliga). Ett normalt värde är 4.

Maxstorlek för 'page swap area' i MB

Denna parameter bestämmer max antal poster i minnestabellerna i systemet och ges som den maximala sammanlagda logiska (virtuella) minnesarean i Mbytes för samtliga aktiva processer. Tabellerna upptar ett utrymme på ca 3584 bytes/Mbyte logiskt minne. Felmeddelandet **End-of-Swap area** ges då virtuella minnet är fullt. Detta meddelande kan dock även ges när det inte finns mer plats på skivminnet för "swapping", i detta fall vanligen tillsammans med meddelandet **End-of-Device.**

Den verkligt utnyttjade virtuella arean kan grovt uppskattas genom att addera MEM_SIZE för alla processer enligt kommandot *ps -axl*. Både text- och dataareorna räknas, visade i kbyte. Ett lämpligt värde för denna parameter erhålls genom att lägga till 50% extra utöver det uppmätta värsta fallets värde.

I system där flera kopior av stora program används, kan mycket stort virtuellt minne specificeras. Detta är möjligt eftersom det fysiska minnesbehovet och swap-frekvensen reduceras när delade textareor (kod) används och eftersom normalt enbart de aktiva delarna av programfilerna finns i minnet samtidigt. Jämför parametern "max antal laddade filer" ovan.

Normala värden är i området 8 - 48 Mbytes eller upp till 128 i stora system. Med *sar -v* visas denna parameter under namnet **swap**.

Notera att denna parameter inte har något att göra med swaparean på skivminnet. Virtuell minnesteknik och swapping beskrivs i nästa delkapitel.

Max antal meddelande köer

Systemparameter, som maximerar antal meddelandeköer (message queues) i systemet. Varje kö tar 58 bytes. Dessa fungerar som ett slags 'pipes' mellan processer internt i systemet. Dessa används sällan i vanliga program. Ett normalt värde är 20. Med *sar -v* visas denna parameter under namnet **messq**. Se även beskrivningen för kommandot *ipcs*.

Max antal semaforer

Systemparameter, som maximerar antal semaforer i systemet. Varje semafor tar 42 bytes. Dessa används sällan av vanliga program. Ett normalt värde är 20. Med *sar -v* visas denna parameter under namnet **sem**. Se även beskrivningen för kommandot *ipcs*.

Max antal semafor-justering-vid-exit

Systemparameter, som anger max antal systemanrop med begäran att systemet automatiskt skall återställa semaforer om en process avbryts

utan att återställa på normalt sätt. Varje styrblock tar 10 bytes. Detta används sällan av vanliga program. Ett normalt värde är 64. Med kommandot *sar -v* visas denna parameter under namnet **semadj**. Se även beskrivningen för kommandot *ipcs*.

Max antal delbara minnessegment

Systemparameter som anger max antal globala namngivna fria minnessegment som kan delas av flera processer, dvs. där gemensamma data för flera processer kan lagras. Ett styrblock för ett segment kräver 66 bytes. Dessa segment allokeras som separata areor i minnet men kan mappas in som delar av olika programs minnesareor. Minnessegmenten kan swappas till skivminnet av systemet. Ett normalt värde är 20. Med *sar -v* visas denna parameter under namnet **shm**. Se även beskrivningen för kommandot *ipcs*.

Max antal invalda minnessegment

Systemparameter som anger hur många inmappningar som samtidigt kan förekomma i systemet mellan processer och fria minnessegment. Varje inmappning tar 14 bytes. En process kan mappa in ett eller flera segment i sitt logiska minne för access. Ett normalt värde är 64. Med *sar -v* visas denna parameter under namnet **shmat**. Se även beskrivningen för kommandot *ipcs*.

Antal sekunder mellan filbuffertuppdateringar

Detta anger tiden i sekunder mellan de automatiska uppdateringarna av skivminnena från filbuffertarna, normalt 60 sekunder. (Jämför kommandot *sync*).

Storlek på terminalinputbuffert i bytes

Detta är storleken på input-bufferten för type-ahead som används av terminalrutinen. Den kan behöva ökas om kommunikationslinjer används där mer än 255 byte sänds per block. Obs! Den verkliga buffertarean allokeras dynamiskt från systemets minnespool. Se parametern "Max storlek på minnespool" nedan.

Minuter innan systemstopp vid strömavbrott

När en kraftavbrottssignal kommer från ett anslutet UPS-system med batteribackup, väntar systemet maximalt denna tid innan signalen SIGPWD sänds till initprocessen för att stänga av systemet (om inte kraften återvänder). Avstängningen sker med /etc/powerfail. Denna parameter skall vara skild från noll enbart om ett UPS-system med batteribackup är anslutet.

Antal kontrollerande terminalnoder

Detta anger maximala antalet terminalenheter som kan vara kontrollerande terminaler samtidigt aktiva i systemet. Varje kontrollerande terminalnod kräver 8 bytes. Om värdet ändras bör även minnespoolens storlek ändras enligt nedan. Antal kontrollerande terminalnoder kan vara större än antalet fysiska enheter när nätverk eller fönsterhanterare används. Varje fönster i en fönsterhanterare fungerar normalt som

en kontrollerande terminal såväl som loginterminalen från vilken fönsterhanteraren startats. Ett normalt värde är 128.

Tillåt ej skrivning på monterade enheter

Om denna parameter är 1 (vilket är standard) hindras rå skrivning direkt till de enheter, där ett filsystem mountats. Skrivning kan då bara ske filstrukturerat. Sätts värdet 0 så kan alla som har skrivtillstånd till enheten skriva till den i rå mod, oberoende av om ett filsystem är mountat där eller inte.

Filsystemets säkerhetsnivå

Denna parameter är oftast 0 och tillåter full filbuffring. När parametern är 1 blir alla filbuffertar med bibliotek och i-noddata omedelbart "flushade" efter varje ändring. Om den är 2 kommer även buffertpoolerna som innehåller indexpekare eller användardata att omedelbart bli "flushade" vid varje skrivning.

Maximal storlek på minnespool i kb

Detta är den totala storleken på en minnespool som utnyttjas av systemet. Som exempel allokeras alla tty-buffertar från denna pool såväl som buffertar för NFS. Med många terminalanslutningar eller stora tty-buffertstorlekar angivna, bör detta värde ökas. Ett normalt värde är i området 100 - 128 kbytes. Värdet bör vara minst "ttybuffertstorlek" * "antal kontrollerande terminaler" enligt ovan, konverterat till kbytes plus omkring 60 kbytes extra. Värdet kan ökas om felmeddelandet **End-of-system memory pool** visas.

Maximal storlek på timeout-tabell

Detta är maximala antalet timeout-block i systemet. Varje block är 24 bytes. De används när fysiska enheter accessas men även när systemanropen alarm() och nap() utförs från processer. Dock ökar tabellbehovet endast lite när antal terminalportar ökas, eftersom bara ett block extra krävs för varje terminalkoncentratorkort. Ett normalt värde är 256, men detta bör ökas om systemmeddelandet **Overflow in callout table** visas.

Nodnamn

Detta är nodnamnet som identifierar systemet i ett nätverk. Namnet är en sträng med maximalt 8 tecken. För att ta bort nodnamnet i interaktiv mod, anges ett minustecken. Apostroferna kring strängen, som visas av *mkcfig*, ingår inte. Nodnamnet kan läsas med kommandot *uname -n* och lagras normalt av systemet vid uppstart till filen /etc/systemid. Observera att det är lämpligt att begränsa nodnamnets längd till 6 tecken eftersom namnet används av flera system och vissa andra system kräver att max 6 tecken ingår.

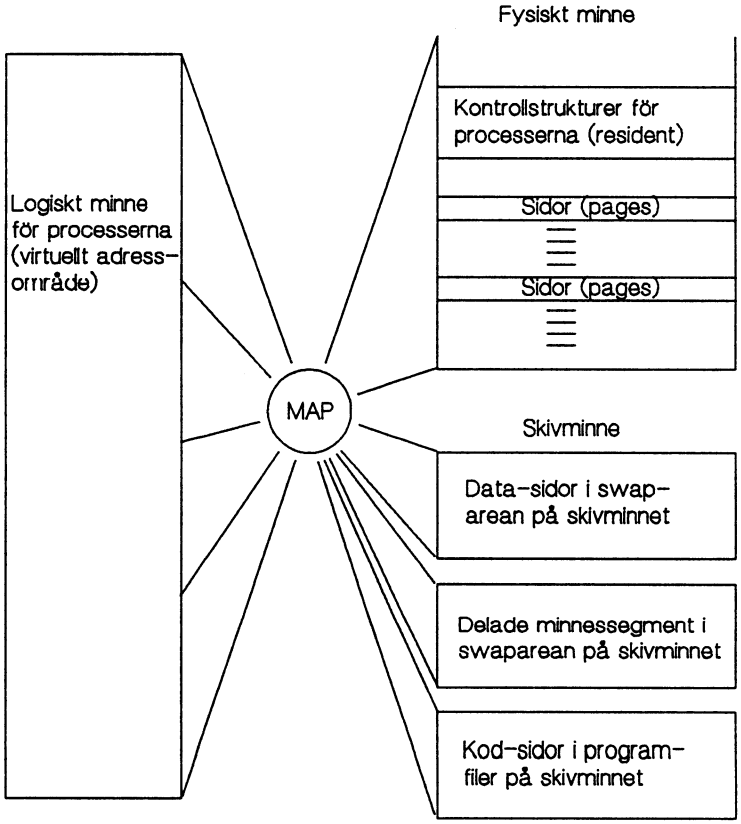
5.4.3 Virtuell minneshantering och swapping

Virtuellt minne och mappning till fysiska minnet.

Varje process exekverar i ett logiskt adressområde (virtuellt minne) oberoende av övriga processer och oberoende av den verkliga fördelningen av det fysiska minnet. Det logiska minnet kan vara 0 - 2 Gbyte per proces (eller med en annan övre gräns beroende på systemet). Inom denna area ligger programkoden (text) och dataarean samt stacken allokerade enligt programmet.

Hårdvaran mappar de logiska adresserna i sidor (delar om vanligtvis 8 kbytes) till det fysiska minnet, som normalt är begränsat till ett antal Mbytes. Varje sida (page) kan ligga var som helst i minnet, swappat ut på skivminnet eller fortfarande ligga kvar i programfilen på skivminnet. Hårdvaran känner automatiskt av sidfel (page faults), dvs när ett program försöker accessa en adress som inte finns i det fysiska minnet. Vid page-faults letar systemet reda på en ledig sida i minnet och laddar eventuellt data eller programkod från skivminnet, innan processen tillåts fortsätta exekvera.

Detta betyder att även mycket stora processer och många processer kan köras i ett system trots ett begränsat fysiskt minne. Dock måste alltid en liten area med datastrukturer stanna resident i minnet för varje aktiv process.



Swapping på skivminnet

Sidor (pages) som swappats ut på skivminnet skrivs direkt till ett ledigt skivminnesblock (bitmappen på skivminnet används för att hitta ett fritt block). Enbart dataareor och eventuellt delade minnessegment skrivs verkligen till skivan, inte textarean (programkoden) vilken finns tillgänglig i programfilen och aldrig ändras. Endast sidor som ändrats skrivs till skivan och sidor som sällan uppdateras swappas först ut.

Swapping startar när 75% av det fysiska minnet är upptaget av de residenta strukturerna och processernas sidor. Vid 75% är swap-frekvensen låg, men den ökar när minnet utnyttjas mer. Dessutom används swapfunktionen för att ladda program, varvid automatiskt enbart de delar av programmen som är aktiva verkligen laddas.

Med *sar -w* visar parametrarna **swpin/s** och **swpot/s** swap-frekvensen för skrivning respektive läsning av sidor till/från skivan.

Fysiska minnesbehovet

Trots att den virtuella minnestekniken reducerar minnesbehovet krävs det plats i det fysiska minnet för vissa strukturer som alltid stannar residenta i minnet och inte kan swappas ut. Dessa bör beaktas när behovet av fysiskt minne beräknas.

Storleken av vissa av dessa strukturer kan ändras med *mkcfig*, men andra beror direkt av de processer som körs i systemet.

- Ett minimum av 1-1.5 Mbyte upptas av operativsystemet och dess strukturer, när normala *mkcfig*-parametrar används.
- Minnesbehovet ökar när mer resurser anges med *mkcfig*. Särskilt ökade filbuffertar och större minnestabeller för virtuellt minne (swap area) konsumerar resident fysiskt minne.
- För varje process som verkligen körs är en datastruktur på ca 24 kbytes resident, även för processer som helt är utswappade. Med många processer kan dessa strukturer uppta en stor minnesarea, även när processerna delar programkod. Exempel: 200 aktiva processer => 4.8 Mbyte minne som ockuperas av processernas strukturer.
- Det är möjligt att låsa en process i det fysiska minnet, varvid swapping av processens sidor hindras och processen permanent upptar en del av det fysiska minnet. Detta används inte av normala processer, men kan krävas för vissa realtidsfunktioner där t.ex. svarsstiden för programavbrott måste vara kort.
- Vid uppstart av systemet visas på konsolterminalen hur mycket systemminne respektive användarminne (user) som finns. Systemminnet, med alla strukturer, måste ligga i fysiskt kontinuerligt minne vilket ger en absolut övre gräns som är datorberoende, men som för små system kan vara ner till några få Mbyte även om totala minnet är mycket större.

5.4.4 Rekommenderade parametrar

mkcfig finns med både engelsk och svensk hjälptext. Nedan följer rekommenderade konfigurationsparametrar för de olika datorerna med de engelska texterna.

Öka eller minska filbuffertparametern (*) tills bästa prestanda erhålls för det aktuella systemet.

Konfigureringsparametrar för DS90-10/11

```
Max number of active processes...............:  64
Max number of loaded core files..............:  64
Max number of nodes..........................:  256
Size of file buffers in Kb...................:  64 (*)
Max number of open files.....................:  96
Max number of file locks.....................:  128
Max number of mounted file systems...........:  4
Max size of page swap area in Mb.............:  8
Max number of message queues.................:  20
Max number of semaphores.....................:  20
Max number of semaphore adjust operations....:  64
Max number of shared memory segments.........:  20
Max number of attached memory segments.......:  64
Number of seconds between file system sync...:  60
Number of characters in tty input buffer.....:  255
Minutes before shutdown at powerfail.........:  0
Number of controlling terminal nodes.........:  32
Disable raw write on mounted device..........:  1
File system security level...................:  0
Max size of memory pool in kb................:  100
Size of callout table .......................:  256
Node name....................................:  ''
```

Konfigureringsparametrar för DS90-20/21//30/31/41, DS101 och DIAB2xxx

```
                                              Min  Normal
Max number of active processes...............:  128  300
Max number of loaded core files..............:  128  300
Max number of nodes..........................:  512  1000
Size of file buffers in Kb...................:  256  700-2000 (*)
Max number of open files.....................:  200  400
Max number of file locks.....................:  128
Max number of mounted file systems...........:  4
Max size of page swap area in Mb.............:  32   48 - 128
Max number of message queues.................:  20
Max number of semaphores.....................:  20
Max number of semaphore adjust operations....:  64
Max number of shared memory segments.........:  20
Max number of attached memory segments.......:  64
Number of seconds between file system sync...:  60
Number of characters in tty input buffer.....:  255
Minutes before shutdown at powerfail.........:  0
Number of controlling terminal nodes.........:  128
Disable raw write on mounted device..........:  1
File system security level...................:  0
Max size of memory pool in kb................:  100  128
Size of callout table .......................:  256
Node name....................................:  ''
```

Min: Med bara 4 MB minne i ett litet system.
Normal: Ett system med 20-40 användare och 16 MB minne.

Konfigureringsparametrar för DS90-30S, DIAB1x20

```
Max number of active processes...............:  128
Max number of loaded core files..............:  128
Max number of nodes..........................:  512
Size of file buffers in Kb...................:  256 (*)
Max number of open files.....................:  200
Max number of file locks.....................:  128
Max number of mounted file systems...........:  4
Max size of page swap area in Mb.............:  32
Max number of message queues.................:  20
Max number of semaphores.....................:  20
Max number of semaphore adjust operations....:  64
Max number of shared memory segments.........:  20
Max number of attached memory segments.......:  64
Number of seconds between file system sync...:  60
Number of characters in tty input buffer.....:  255
Minutes before shutdown at powerfail.........:  0
Number of controlling terminal nodes.........:  128
Disable raw write on mounted device..........:  1
File system security level...................:  0
Max size of memory pool in kb................:  128
Size of callout table .......................:  256
Node name....................................:  ''
```

6

6. Användaromgivningen

6. Användaromgivningen

6.1. Användaromgivningen - (User Environment)

6.1.1 Inloggning

Då datorsystemet är startat kan användare logga in från alla aktiverade terminaler. På dessa terminaler har då systemet skrivit ut texten **Login:** eller motsvarande. Om terminalen har varit avslagen kan användaren trycka på **RETURN** för att få denna frågetext. Texten kan föregås av en systemidentifikation innan **Login:** skrivs ut. Därefter anges användarnamnet, varefter systemet frågar efter lösenord. Tecknen ekas inte till bildskärmen medan lösenordet skrivs. Medan användarnamn och lösenord skrivs kan inte de normala tangenterna för att backa och sudda tecken användas. Tecknet **#** kan användas för att ta bort föregående tecken men det är inte säkert att det syns på bildskärmen.

6.1.2 Efter godkänd inloggning

Efter inloggning startas en kommandoavkodare (shell), som först skriver ut ett systemmeddelande, som systemoperatören lagrat i filen **/etc/motd**. Detta görs med kommandon i **/etc/profile**. Därefter bearbetas de kommandon som användaren har i sin fil **.profile**. Först efter detta kan användaren börja ge kommandon från tangentbordet. I vissa fall kan i filen **.profile** finnas kommandon så att ett tillämpningsprogram automatiskt startas då användaren loggar in.

Användaren befinner sig i sitt inloggningsbibliotek efter inloggning, om inte kommandon i **.profile** givits för att byta bibliotek.

Användaren tillhör en grupp användare enligt de definitioner som finns i filen **/etc/passwd** och **/etc/group**. Detta ger honom/henne möjlighet att läsa och skriva filer som inte är egna, men som tillhör gruppen, beroende på vilken behörighet som gäller för de olika filerna.

Användaren kan byta till en annan grupp med kommandot *newgrp* om han känner till lösenordet för den nya gruppen eller är medlem i den nya gruppen.

6.1.3 Begränsat shell (restricted shell)

I filen **/etc/passwd** eller i användarens **.profile** kan definieras att användaren enbart skall få använda vissa kommandon och hindras att skriva till filer utanför det bibliotek där han befinner sig efter inloggning. Då startas en begränsad kommandoavkodare, som inför följande begränsningar:

- Användaren kan inte byta bibliotek.
- Användaren kan inte ändra variabeln **PATH**.

- Användaren kan inte komma åt filer utanför det aktuella biblioteket genom att _/_ inte får förekomma i filnamn.
- Användaren får inte använda omdirigering av utmatning från kommandon med > eller >>.

Om däremot en vanlig kommandoavkodare (shell) kan startas, d.v.s. finns i de bibliotek som anges i **PATH**, så finns inga begränsningar där. Likaså kan de kommandon som startas komma åt filer fritt utan begränsningar, men inte genom standard input/output från begränsat shell.

6.1.4 Shell-variabler

I kommandoavkodaren kan olika variabler definieras som kan påverka hur kommandon utförs. Alla dessa variabler kallas användarens omgivning (environment). Dessa kan ses med kommandot *set*, givet utan parametrar.

Där ingår bland annat följande variabler som automatiskt definieras vid inloggning, men användaren kan själv ändra eller lägga till variabler.

LOGNAME	användarnamnen för användaren
HOME	användarens inloggningbibliotek
PATH	**:/bin:/usr/bin** om den inte omdefinieras i **.profile**
SHELL	namnet på kommandoavkodaren angiven i **/etc/passwd**
MAIL	den fil där meddelanden kommer med *mail*
TZ	tidszonen vid tidsangivelser
PS1	**$**, den prompt som skrivs ut på bildskärmen

Speciellt för användaren root (super-user) innehåller **PATH** även **/etc**.

Shellvariabeln **TERM** definieras vanligen i **.profile** och skall ange namnet på den terminaltyp som används. Den används inte av de enklaste kommandona i systemet, men är nödvändig för alla kommandon som behöver information om hur bildskärmen skall hanteras, t.ex. ordbehandlare, som måste veta vilka styrtecken som skall användas.

Då ett kommando skrivs in på tangentbordet kan vissa tecken användas för att ta bort tecken eller hela raden, och vissa tecken används för att avsluta inmatning. Vilka dessa är definieras med kommmandot *stty* i filen **.profile** och gäller alltså först efter en godkänd inloggning. Dessa är följande, och är normalt definierade enligt nedan.

```
Backspace (Backa och sudda föregående tecken)    CTRL-H
Delete (Sudda hela raden)                        CTRL-X
End-Of-File (Logga ut från systemet eller        CTRL-D
             avsluta inmatning)
```

Notera här att eventuell bakåtpil på terminalen ofta ger helt annan kod till datorn och inte kan användas vid inmatning av ett kommando.

6.1.5 Skrivarsystem lp eller lpr

Två olika skrivarsystem levereras med systemet, och systemoperatören aktiverar enbart ett av dessa. Vid utskrift till skrivare kan alltså antingen kommandot *lp* eller kommandot *lpr* användas.

För att underlätta utskrifter finns ett kommando som heter *print*, vilket är en liten shell-procedur, som delar in det som skrivs ut i sidor, lägger till sidrubriker, och som är anpassat till det aktiva skrivarsystemet. Kommandot *print* anropar kommandot *pr* för formatering, och kan därför inte användas då texten skall formateras på annat sätt.

Se kapitel 9 för en fullständig beskrivning av skrivarsystemen.

6.1.6 Filåtkomstskydd, umask

Varje fil har nio åtkomstskydd. Dessa är lästillstånd, skrivtillstånd och exekveringstillstånd för var och en av: ägaren, gruppen och alla övriga. Exekveringstillstånd för en fil innebär att filen kan användas som ett kommando. Exekveringstillstånd för ett bibliotek innebär tillstånd att gå in i ett bibliotek med kommandot *cd*, samt möjliggör åtkomst av filerna i biblioteket enligt de övriga åtkomsttillstånden.

Vilka åtkomstskydd en fil får när den skapas med vanliga kommandon (t.ex. *cat* eller *cp*) avgörs av en bitmask, som användaren kan lista/ändra med kommandot *umask*, se **Referenshandboken**.

6.2. .profile

Filen **.profile** skall ligga i användarens inloggningsbibliotek, och innehåller kommandon som utförs direkt efter inloggning. Bl.a. finns normalt kommandot *stty* för att definiera lämpliga styrkoder för kommunikationen mellan dator och terminal. shell-variablen **TERM** definieras alltid här och användaren kan definiera egna variabler. Dessa skall alltid markeras med *export* för att kunna användas av andra kommandon.

Exempel på filen .profile:

```
stty -tabs cr0 ff0 nl0 echoe erase '^H' kill '^X'
TERM=twist
echo 'Terminaltypen (TERM) är : ' $TERM
PATH=$PATH:/usr/kalle/bin
export TERM PATH
```

I **.profile** kan även tillämpningsprogram startas automatiskt, t.ex. programmet *menu* (se D-MENU) som beskrivs i följande avsnitt.

Genom att på sista raden i **.profile** starta ett program med kommandot *exec* kommer utloggning att ske automatiskt då programmet avslutas. Här kan t.ex. en begränsad shell startas efter att shellparametrarna **PATH** och **HOME** ändrats.

6.3. D-MENU

Principen med D-MENU är att användaren vid inloggning till systemet möts av en meny. Menyn kan antingen vara en av systemets standardmenyer eller en av användarens egna menyer.

Utifrån denna meny kan användaren starta olika program i systemet, endera från menyn som visades vid inloggning eller från någon annan meny som kan startas från den första menyn.

Alla som startar D-MENU behörighetskontrolleras dels av menysystemets egen behörighetskontroll, dels av operativsystemets normala behörighetskontroller. En kontroll av att användaren är tillåten att, via den använda terminalen, få tillgång till systemet görs också.

Eventuella fel som inträffar när D-MENU används detekteras och presenteras i klarspråk för användaren. Det gäller både fel i menyhanteraren och fel som uppstår i program och systemanrop som startas från menyhanteraren.

Menysystemet innehåller även en menyeditor. Menyeditorn underlättar för användaren att skapa och uppdatera menyer. Både menyhanteraren och menyeditorn är språkoberoende. Detta gäller alla texter i systemet, felmeddelanden, dialogtexter och hjälpfiler.

D-MENU är en allmän användarresurs som binder ihop användaren med alla andra resurser i systemet på ett för användaren enkelt sätt. Användaren har möjligheter att bygga upp menyer och grupper av menyer inom olika användningsområden. D-MENU består av två program:

/usr/bin/menu	Program för menyhantering
/usr/bin/menuedit	Program för menyeditering

Följande bibliotek används av D-MENU:

/usr/lib/menues	Systemmenyer, hjälptexter och procedurer.
/etc	Tabellfiler för behörighet samt skrivartabeller.
/usr/USERID/menues	Användarens egna menyer och hjälptexter.

D-MENU innehåller dessutom ett behörighetskontrollsystem som antingen tillåter eller omöjliggör för användaren att utföra ett val beroende på användarens definierade behörighetsnycklar.

Observera! Det är viktigt att Systemdaministratören ensam ansvarar för följande administrativa uppgifter:

- installation av mjukvara
- skapa användare till menysystemet
- tilldela menyanvändarna behörighetsnycklar
- initiering av användarnas environment-variabler

- skapa och uppdatera systemmenyer.

Den person som fungerar som menyadministratör måste vara priviligierad som super-user eller kunna logga in som root.

Terminaler som används för D-MENU måste ha en minimistorlek på skärmen som är 24 linjer med 80 kolumner, mjukvarukontrollerad skärmrensning och markörplacering. Terminaltypen måste också finnas definierad i operativsystemets termcap-fil.

Observera! Systemdaministratören behöver ej sätta upp **/usr/lib/menues/bin** i användarnas **PATH** variabel. D-MENU sköter detta själv vid start.

För detaljer om D-MENU se **D-MENU Användarhandbok**.

7

7. Felhantering

7. Felhantering

7.1. Förenklad felsökning

Systemprogrammet *ds90test* skall alltid köras från enanvändarläge (bootnivå 2) i operativsystemet. Programmet består av fyra olika delar som testar olika delar av maskinvaran. Delarna är:

- Terminaltest
- Minnestest
- Winchesterenhet
- Diskettenhet

För att byta från fleranvändarläge till enanvändarläge och starta *ds90test* ger du följande kommandon:

```
# /etc/shutdown
# ds90test
```

Programmet frågar nu efter vilken terminal som används, vt100 eller adm3a.

Nu visas en testmeny enligt nedan på bildskärmen med frågor om vilka tester som skall utföras. Operatören anger först om felmeddelanden från en av testerna ska visas på skärmen genom att svara **T**, **M**, **D** eller **F**. Svara med **on** eller **off** för de olika testerna. Om du svarar on kommer programmet att fråga hur många gånger (count) testen skall utföras.

```
DS90 SYSTEM TEST        STATUS:       MONITOR:(T/M/D/F):__
*******************************************************
TEST            ON/OFF COUNT      ERROR  COUNT  MESSAGE
                       PERFORMING ERROR  COUNT  MESSAGE

Terminal test   __     __         Write    x
                        x         Read     x
                                  Alarm    x

Memory test     __     __         Byte     x
                        x         Bit      x

Disk test       __     __         Write    x
                        x         Read     x
                                  Compare  x

Floppy test     __     __         Write    x
                        x         Read     x
```

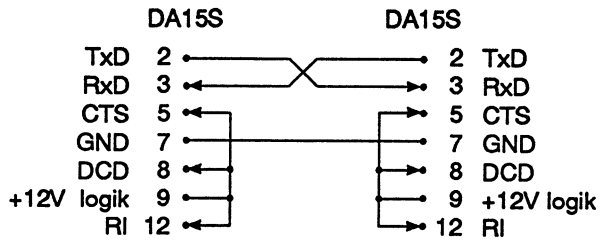
När du har svarat på alla frågorna kommer testerna att starta. På sista raden i menyn visas hela tiden hur långt testerna har hunnit. Efter varje test kommer menyn att uppdateras och antalet upptäckta fel kommer att visas på bildskärmen.

Observera! Om ett upptäckt fel är av sådan art att det inte kan avhjälpas med den information som ges av *ds90test* bör en kvalificerad servicetekniker anlitas.

7.1.1 Terminaltest

När testen utförs skall terminalportarna vara korskopplade. Portarna **/dev/tty02** och **/dev/tty03** skall kopplas samman med en testkabel, som skall ha följande utseende:

Testkabel för ds90test



Testet utförs genom att ett 30 byte långt bitmönster skrivs till utporten **/dev/tty02**. Mönstret läses sedan från inporten **/dev/tty03**. Portarna används i RAW-mod och inte i ECHO-mod. Olika mönster används och riktningen kastas om för att prova alla funktioner både hos in- och utportarna. Om det mottagna bitmönstret inte överenstämmer med det sända, kommer siffran som markerar lästa (read) att räknas upp. Om ingenting kan läsas från inporten (time-out) kommer siffran som markerar **Alarm** att räknas upp med ett.

Observera! Enbart serieportarna med enhetsnamnen /dev/tty02 och /dev/tty03 kan testas med ds90test. Dessa länkas till fysiska portar med sina sekundära enhetsnummer (minor), se kapitel 15.

7.1.2 Minnestest

Minnestesten begär 256 kbyte minne av operativsystemet och skriver ett bitmönster till denna minnesarea. Mönstret är 0 för det första 256 byte blocket, 1 för nästa o s v. Om några fel upptäcks när mönstret läses kommer positionen **Byte** i menyn att räknas upp.

Dessutom görs en test på bitnivå i minnesarean genom att läsa och skriva alla bitar. Om några fel upptäcks kommer positionen **Bit** att räknas upp.

Om flera minnestester skall utföras kommer minnesarean först att frigöras. Därefter kommer en nya area att begäras från systemet för varje test.

7.1.3 Winchesterenhet

Denna test skapar en 2 Mbyte temporärfil i biblioteket /tmp. Filen fylls med samma bitmönster som användes vid minnestesten. Därefter kommer testen att utföra 2048 slumpmässiga positioneringar och läsa och jämföra bitmönstren för aktuell byte. Skulle något fel upptäckas kommer positionen **Read** att räknas upp med ett. När testen är avslutad tas temporärfilen bort.

Observera! Denna test använder filhanteraren och förstör därför inga data på winchesterenheten.

7.1.4 Diskettenhet

Denna test skriver en 256 kbyte buffert till enheten /dev/mf0. Det bitmönster som används är detsamma som användes av minnestesten. Därefter kommer testen att utföra 2048 slumpmässiga positioneringar och läsa och jämföra bitmönstren för aktuell byte. Skulle något fel upptäckas kommer positionen **Read** att räknas upp med ett.

Observera! Denna test skriver fysiskt till disketten, vilket innebär att data som tidigare lagrats på disketten kommar att ***förstöras***.

7.2. Felkoder

Nedan följer en lista över de felkoder som returneras av operativsystemet vid anrop från programmen.

EPERM	-1	/* Not owner */
ENOENT	-2	/* No such file or directory */
ESRCH	-3	/* No such process */
EINTR	-4	/* Interrupted system call */
EIO	-5	/* I/O-error */
ENXIO	-6	/* No such device or address */
E2BIG	-7	/* Arg list to long */
ENOEXEC	-8	/* Exec format error */
EBADF	-9	/* Bad file number */
ECHILD	-10	/* No children */
EAGAIN	-11	/* No more processes */
ENOMEM	-12	/* Not enough core */
EACCES	-13	/* Permission denied */
EFAULT	-14	/* Bad address */
ENOTBLK	-15	/* Block device required */
EBUSY	-16	/* Mount device busy */
EEXIST	-17	/* File exists */
EXDEV	-18	/* Cross-device link */
ENODEV	-19	/* No such device */
ENOTDIR	-20	/* Not a directory */
EISDIR	-21	/* Is a directory */
EINVAL	-22	/* Invalid argument */
ENFILE	-23	/* File table overflow */
EMFILE	-24	/* Too many open files */
ENOTTY	-25	/* Not a typewriter */
ETXTBSY	-26	/* Text file busy */
EFBIG	-27	/* File too large */
ENOSPC	-28	/* No space left on device */
ESPIPE	-29	/* Illegal seek */
EROFS	-30	/* Read-only file system */
EMLINK	-31	/* Too many links */
EPIPE	-32	/* Broken pipe */
EDOM	-33	/* Argument too large */
ERANGE	-34	/* Result too large */
ENOMSG	-35	/* No such message of desired type */
EIDRM	-36	/* Identifier removed */
ECHRNG	-37	/* Channel number out of range */
EL2NSYNC	-38	/* Level 2 not syncronized */
EL3HLT	-39	/* Level 3 halted */
EL3RST	-40	/* Level 3 reset */
EMSGSIZE	-40	/* Message too long (bsd) */
ELNRNG	-41	/* Link number out of range */
EUNATCH	-42	/* Protocol driver not attached */
ENOCSI	-43	/* No CSI structure available */
EL2HLT	-44	/* Level 2 halted */
EDEADLK	-45	/* Record locking deadlock */
ENDLCK	-46	/* No lock structures available */

```
ENOSTR       -60     /* Device not a stream */
ENODATA      -61     /* No data (for no-delay I/O) */
ETIME        -62     /* Timer expired */
ENOSR        -63     /* Out of streams resources */
ENONET       -64     /* Machine is not on the network */
ENOPKG       -65     /* Package not installed */
EREMOTE      -66     /* The object is remote */
ENOLINK      -67     /* The link has been severed */
EADV         -68     /* Advertise error */
ESRMNT       -69     /* Srmount error */
ECOMM        -70     /* Communication error on send */
EPROTO       -71     /* Protocol error */
EMULTIHOP    -74     /* Multihop attempted */
ELBIN        -75     /* Inode is remote (not really error) */
EDOTDOT      -76     /* Cross mount point (not really error) */
EBADMSG      -77     /* Trying to read unreadable message */
             -78     /* Reserved */
             -79     /* Reserved */
ENOTUNIQ     -80     /* Given log name not unique */
EBADFD       -81     /* File descriptor invalid for this operation */
EREMCHG      -82     /* Remote address changed */
ELIBACC      -83     /* Can't access a needed shared library */
ELIBBAD      -84     /* Accessing a corrupted shared library */
ELIBSCN      -85     /* Library section in a.out corrupted */
ELIBMAX      -86     /* Attempting to link in to many libraries */
ELIBEXEC     -87     /* Attempting to exec a shared library */

EBADLU       -9      /* Bad lu  (=EBADF)*/
EUCLEAN      -28     /* Structure needs cleaning SYSV (=ENOSPC)*/

ENONODES     -119    /* No more nodesv */
EREQAB       -120    /* Request aborted */
EBADFC       -121    /* Illegal function code */
ESWAP        -122    /* Reffered process is swapped */
ECANCL       -123    /* Reffered request is canceled */
EDRVNF       -124    /* Device driver not found */
ETIMOUT      -125    /* Timeout on driver */
ELOCKOK      -126    /* Locking is OK - no open node needed */
ENOLOCK      -127    /* No locks for this process any more */
EWAIT        -128    /* Request waiting for termination */

ENAMETOOLONG -155    /* Filename too long */
ENOTEMPTY    -158    /* Directory not empty */
ENOSYS       -163    /* Function not implemented */
```

7.3. Reparation av dåliga sektorer - badblk

Ibland uppstår fysiska fel på en sektor av massminnet. Denna sektor måste då ersättas med en sektor utan några fel. Detta kan göras med hjälpprogrammet *badblk* utan att skivminnet behöver formateras om.

badblk placerar numren på alla dåliga sektorer i filen **/..systemfiles-/badspots**. Samtidigt ersätts den dåliga sektorn med en ny, så länge det finns reservsektorer tillgängliga. Om det är möjligt kommer även data från den dåliga sektorn att kopieras till den nya sektorn.

Filsystemet får inte vara mountat och det måste vara rent. Om det är den interna winchesterenheten som skall repareras måste starten ske manuellt upp till enanvändarmod och med en BOOT-diskett som root-enhet.

Utför alltid */etc/fsck* före och efter *badblk* för att garantera att filsystemet är rent.

För ytterligare detaljer om *badblk*, se **Referenshandboken**.

7.4. Systemmeddelanden

Systemmeddelanden med information om händelser eller fel ges av operativsystemet direkt och kommer då bara till huvudkonsolen. Om logfunktionen är aktiverad, samlas systemmeddelanden även i en fil. Se kommandot *errdemon* i **Referenshandboken**.

Dessa meddelanden är oftast i klartext, på engelska och mycket kortfattade.

Vissa meddelanden är relaterade till maskinvaran, som t ex skivminnen eller datorns interna minne. Då ges ofta adresser och enhetsnummer (major/minor) i meddelandet. Vid felmeddelanden bör denna information antecknas.

Vissa meddelanden inleds med texten PANIC. Detta innebär att en allvarlig felfunktion inträffat och systemet stannar helt. Om nyckelbrytaren är i AUTO-läget startar systemet upp igen, varvid den automatiska skivminnestesten vanligen måste korrigera filsystemet.

Nedan ges en lista med några typiska felmeddelanden och mycket kortfattade förklaringar.

```
OS: PANIC: Trap in system mode - SYSTEM HALTED!
Trap=nn, ssp=nn, sr=nn, ssr=nn, pc=nn, fad=nn
d0-d7 nn nn nn nn nn nn nn nn
a0-a7 nn nn nn nn nn nn nn nn
```

Detta felmeddelande ges om systemet tvärstannar på grund av ett fel som operativsystemet inte kan åtgärda utan att riskera att förstöra information. Normalt kan systemet återstartas.

```
End of nodes

Too many active processes

System file table is full (nn)

End of swap area
```

Dessa felmeddelanden kan bero på ett felaktigt användarprogram, som begär för mycket resurser av systemet. Vid applikationer med höga krav på dessa parametrar kan kommandot *mkcfig* användas för att ändra systemet. Se **Referenshandboken**.

```
OS: ERROR: Parity error syndrome nnn pad nnn in user mode
```

eller

```
OS: ERROR: Parity error syndrome nn ssr nn psr nn &pad nn pad nn
```

eller

```
OS: Parity error, VME/Ethernet/CPU, adr nnn, syndr nnn
```

Detta felmeddelande ges om ett paritetsfel upptäcks i minnet när en användarprocess körs. Om däremot paritetsfel upptäcks i systemmod stoppas systemet direkt (OS: PANIC:). Anteckna värdena i felmeddelandet och stäng av systemet för teknisk service.

```
Too many parameters at exec
```

Detta meddelande ges om ett kommando givits med alltför många parametrar.

```
Using default configuration parameters
```

Detta ges om någon av systemparametrarna som startas med kommandot */etc/mkcfig* är felaktiga. Då har standardparametrar använts istället.

```
Error nnn at parameter copy, cmd=nnn
```

Detta felmeddelande beror vanligen på att biblioteket **/tmp** på root-filsystemet ej finns eller inte har åtkomsttillstånd för alla.

```
Unable to lock memory
```

Detta kan bero på felaktiga systemparametrar enligt /etc/mkcfig eller att interna minnet behöver utökas.

Nedanstående felmeddelanden beror på yttre anslutna enheter, som skivminnen, streamer, diskettenheter m m. I meddelandet anges oftast enhetsnumret (major/minor) för identifikation. Se kapitel 15.

```
Write protect
```

Skrivskyddad. Skrivskyddet på t ex en diskett är aktiverat.

```
Device busy
```

Kan t ex erhållas vid *umount* av en diskett eller ett skivminne om någon fil fortfarande är öppen.

```
Drive off line
```

Kan t ex betyda att en diskett inte är insatt i diskettenheten.

```
No space on dev mm,nn
End of media
```

Det finns inte plats på den yttre enheten.

```
Timeout
```

Den yttre enheten svarar inte inom uppsatt tid.

```
Error on dev mm/nn bn=nn cmd=nn sts=nn
```

Vid felmeddelande av denna typ skall de givna värdena på blocknummer(bn), commando(cmd) och status(sts) antecknas. Detta kan t ex bero på fel i något enstaka block på skivminnesenheten och repareras normalt vid en ny uppstart då kommandot *fsck* automatiskt exekveras. I annat fall används kommandot *badblk* för att ersätta det felaktiga blocket. **mm/nn** anger primärt och sekundärt enhetsnummer enligt kapitel 15.

```
Bad init record
Bad VDS pointer
Bitmap integrety error on dec mm/nn
```

Fel har uppstått på skivminnet. Åtgärd krävs av en servicetekniker. Systemet skall omedelbart stängas av. VDS-pekaren och bitmappen kan i bästa fall repareras med */etc/fsck* från BOOT-disketten.

8

8. Terminalhantering

8. Terminalhantering

8.1. Parametrar för terminaler, modem och skrivare

I detta avsnitt ges en kort sammanställning över vilka olika faktorer som bör beaktas vid inkoppling av terminaler, modem eller skrivare till de normala seriekanalerna i systemet, på datorkortet samt via de interna terminalexpansionskorten. Dessa kanaler hanteras alla av samma systemdrivprogram i operativsystemet.

För ytterligare terminaler och skrivare som ansluts via VME-kort används andra drivrutiner i systemet. För dessa finns separata beskrivningar.

8.1.1 Anslutning av utrustning till seriekanalerna.

Vid anslutning av olika typer av terminaler, modem och skrivare till systemet används seriellt gränssnitt enligt V.24/V.28(RS232C) med asynkron kommunikation.

En av anslutningarna är alltid huvudkonsol (**/dev/console**) och en annan alltid systemskrivare (**/dev/lp**), medan övriga kan definieras efter användarens önskemål. Som konsol i enanvändarläge används enheten **/dev/syscon**, vilken normalt är länkad till **/dev/console**.

Då en yttre utrustning ansluts, anpassas vanligen denna till datorn genom att kommunikationshastighet (baudrate) och övriga parametrar ställs in lokalt via byglingar eller lokala kommandon. I tabellen nedan finns de parametrar som vanligen används av systemet.

	Terminal	**Skrivare**	
Baudrate	9600 baud	9600 baud	
Gränssnitt	V.24	V.24	
Ordlängd	8 databitar	8 databitar	
Paritet	Ingen paritet	Ingen paritet	
Antal stoppbitar	1 stoppbit	1 stoppbit	
Buffer-full-signal	XON/XOFF	XON/XOFF	Not 1.
Terminaltyp	vt100		Not 2.

1. Systemskrivaren (**/dev/lp**) underhåller även statisk buffert-fullsignalering från skrivaren genom att en aktivt hög nivå krävs från skrivaren på datorns CTS-ingång för att data ska sändas till skrivaren.
2. Terminaltypen (**TERM**) sätts normalt först vid inloggning genom kommandon i användarens **.profile**.

8.1.2 Terminal/modem med inloggningsmöjlighet.

I fleranvändarläge (normalt vid automatisk systemstart) sätts alltid baudrate och övriga kommunikationsparametrar för inloggningsterminaler vid inloggning enligt filerna **/etc/inittab** och **/etc/gettydefs**. Medan användaren försöker skriva sitt användarnamn under inloggning försöker systemet anpassa terminalparametrarna till inmatade tecken. Efter inloggning ändras normalt vissa av parametrarna med ett *stty*-kommando i filen **.profile**.

Terminalparametrarna kan även temporärt ändras efter inloggning med kommandot *stty*, men om kommunikationshastighet och liknande ändras, krävs att terminalen lokalt ställs om efter inloggning och efter att kommandot stty givits. Se *stty* i **Referenshandboken**.

Efter inloggning exekveras automatiskt kommandona i filen **/etc/profile** samt i filen **.profile** i användarens inloggningsbibliotek. Filen **.profile** skapas automatiskt av */etc/mkuser* då en ny användare definieras, om ett nytt bibliotek anges som inloggningsbibliotek. I **.profile** definieras normalt terminaltyp och vissa styrtecken för terminalen ändras.

Vid manuell systemstart används inte filen **/etc/inittab** förrän då övergång till fleranvändarläge görs (Bootnivå 3). Huvudkonsolens baudrate och terminaltyp kan emellertid anges som parametrar i en krets i systemet. Dessa definieras med kommandot */sas/bootpar*.

Vid manuell systemstart, då systemet startas i enanvändarläge (bootnivå 2) utförs alltid kommandona i **/etc/profile** och **/.profile** trots att ingen inloggning sker. Som terminalparametrar i enanvändarläge används de som anges i andra delen av tabellraden för console i **/etc/gettydefs**.

Terminaltypen bestäms av shell-parametern **TERM**, vilken definieras i **.profile**. **TERM** är definierad som en sträng och är en parameter som används av vissa program (bl a editorn *dmacs*) tillsammans med information i biblioteket **/usr/lib/terminfo** eller i filen **/etc/termcap**. Där definieras vilka styrtecken som terminalen kräver och andra terminalberoende parametrar, t ex markörstyrtecken och antal rader på skärmen. Dessa parametrar används av olika program med bildskärmshantering, t ex editorn *dmacs*. Filen **/etc/termcap** är en äldre typ av parameterfil, som är i textformat och kan visas på bildskärmen med kommandot *cat /etc/termcap*. Biblioteket terminfo innehåller en fil för varje terminal i kompilerad form, men vid leveransen är alltid dessa båda kompatibla. Det finns även program som använder egna filer med koder istället för dessa standardfiler.

Det aktuella värdet på **TERM** kan ses med kommandot:

```
echo $TERM
```

För att se vilka olika terminaltyper som finns i systemet, dvs vilka namn som parametern **TERM** kan sättas till, kan följande kommando ges för att visa titelraderna i parameterfilen **/etc/termcap**:

```
fgrep '|' /etc/termcap
```

Därvid visas de rader ur filen där terminaltypnamnen står. Raderna innehåller flera textsträngar, begränsade av tecknet **|**. Den andra text-

strängen, **twist** i exemplet nedan, anger det namn som **TERM** skall ha för den terminal som beskrivs. Exempel på rad som listas med ovanstående kommando (terminaltyp twist):

```
dl|twist|vt-100|pt100|pt-100|dec vt100:\
```

Alternativt kan terminaltyperna definierade i biblioteket **/usr/lib/-terminfo** listas med följande kommando. Där ligger varje terminaldefinition i en egen kompilerad fil, med filnamnet lika med terminaltypen. För snabb åtkomst ligger dessa filer i underbibliotek med enbokstavsnamn, som är lika med första bokstaven i terminaltypen. Vid leverans är terminalparametrarna samma i **/etc/termcap** och **/usr/lib/terminfo**.

```
ls /usr/lib/terminfo/*/*
```

De vanligaste terminaltyperna är adm3a, vt100, twist (24 raders Facit Twist), twi72 (72 raders Facit Twist), vt220 och vt240.

8.1.3 Skrivare/modem utan inloggningsmöjlighet.

För en skrivare kan alla kommunikationsparametrar ändras. Detta kan antingen göras genom ett *stty*-kommando i kommandofilen för den logiska skrivaren eller automatiskt vid systemstart, genom att kommandot */etc/setspeed* lägges in i filen **/etc/rc**.

Flera skrivare kan kopplas till systemet via de seriella portarna och texter sänds till skrivarna via ett programsystem som kallas spooler. Systemet levereras med två olika spoolersystem, *lp*-systemet och *lpr*-systemet. Endast ett av dessa skall vara aktivt i systemet.

Då modem skall kopplas till systemet via en serieport skall porten normalt modifieras enligt följande avsnitt, även om inloggning ej ska kunna ske.

8.2. Tillägg av terminal/modem-anslutningar

Systemet levereras normalt med seriella anslutningar för tre terminaler och en skrivare. Det är även möjligt att lägga till fler terminal-, skrivar- eller modemanslutningar genom att installera interna terminalexpansionskort med vardera fyra extra anslutningar eller via VME-kort (VME/TC1) som har vardera 10 anslutningar. VME/TC1 är lätt att installera i efterhand.

Information om terminaler från vilka inloggning ska kunna ske och vilken baudrate och övriga parametrar de använder finns lagrad i filerna **/etc/inittab** och **/etc/gettydefs**. I biblioteket /dev finns motsvarande fysiska enhet definierad.

Vid leverans är upp till 12 seriella anslutningar definierade för upp till två extra interna terminalexpansionskort, men dessa ska aktiveras i **/etc/inittab** vid installation av korten enligt Exempel 1.

Modemanslutningar med inloggningsmöjlighet skall även de aktiveras i /etc/inittab, men också i **/etc/gettydefs**. Se Exempel 2 och 4.

Observera! Efter aktivering av serieanslutningarna i **/etc/inittab**, måste kommandot *telinit q* ges för att systemet ska läsa av den ändrade filen. Alternativt kan man istället gå ner till enanvändarläge med kommandot */etc/shutdown* varefter **/etc/inittab** läses av systemet vid återgång till fleranvändarnivå med *telinit 2*.

Notera att skrivaranslutningar ej ska aktiveras i **/etc/inittab**. Ej heller modemanslutningar som enbart är till för utgående telefonförbindelser och som aldrig någonsin kommer att hantera inloggningar, se exempel 3.

Utskrift av innehållet i filen **/etc/inittab** görs med kommandot *cat*.

```
cat /etc/inittab
```

På grund av begränsad radlängd i manualen delas långa rader i listningarna nedan. På bildskärmen kan de se annorlunda ut.

```
**** Endast rader som berör terminalerna visas nedan ****
**** I filen finns flera rader före dessa ****

co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:off:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
t8:2:off:nice -16 /etc/getty tty08 9600
t9:2:off:nice -16 /etc/getty tty09 9600
t10:2:off:nice -16 /etc/getty tty10 9600
t11:2:off:nice -16 /etc/getty tty11 9600
```

I detta fall är vi endast intresserade av raderna med */etc/getty*. Med detta kommando startas inloggning på de angivna fysiska enheterna (console, tty02, tty03,...tty11). Som parameter anges, förutom den fysiska enheten, även en sträng som pekar till en tabell i filen **/etc/gettydefs**. Aktiva terminaler är markerade med texten **respawn**, vilken anger att systemet automatiskt startar inloggning på terminalen både vid första systemstart och när någon loggat ut.

Filen **/etc/gettydefs** kan också ses med kommandot *cat*:

```
console# B9600 HUPCL IXON CLOCAL # B9600 ICRNL IXON IXANY IXOFF\
OPOST ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK CLOCAL \
#Console Login: #console

1200# B1200 HUPCL IXON # B1200 ICRNL IXON  IXANY IXOFF OPOST \
     ONLCR TAB3 ISIGG ICANON ECHO ECHOE ECHOK #login: #1200

300# B300 HUPCL IXON # B300 ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #300

9600# B9600 HUPCL IXON # B9600 ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #9600

2400# B2400 HUPCL IXON # B2400 ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #2400

4800# B4800 HUPCL IXON # B4800 ICRNL IXON IXANY IXOFF OPOST\
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #4800

19200# B19200 HUPCL IXON # B19200 ICRNL IXON IXANY IXOFF OPOST\
      ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #19200

38400# B38400 HUPCL IXON # B38400 ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #38400

net# HUPCL IXON # ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE  ECHOK #Network Login: #net
```

Då vi jämför dessa två filer ser vi att endast huvudterminalen (console), tty02 och tty03 är aktiverade i **/etc/inittab**, medan färdiga rader finns även för tty04..tty11. Vi ser även att filen **/etc/gettydefs** anger att terminaler med parametern 9600 i **/etc/inittab** skall ha 9600 baud, samt att standarrdvärdenian 8 databitar, ingen paritet används. Se *stty* i **Referenshandboken** för parametrarnas betydelse, men notera att parameternamnen skiljer sig något från motsvarande i *stty*-kommandot. Parametrarna i **/etc/gettydefs** får inte innehålla minustecken.

Utskrift av de fysiska terminalanslutningarna i biblioteket **/dev** görs med kommandot *ls*.

```
ls /dev/tty?* /dev/console

/dev/console
/dev/tty02
/dev/tty03
/dev/tty04
/dev/tty05
/dev/tty06
/dev/tty07
/dev/tty08
/dev/tty09
/dev/tty10
/dev/tty11
```

Exempel 1:

Aktivering av terminalanslutning tty04.

Om ett extra terminalanslutningskort ansluts behöver endast filen **/etc/inittab** ändras för att aktivera de önskade anslutningarna, i detta fall enbart tty04.

Om **/etc/inittab** ej skulle innehålla motsvarande rad ska den läggas till, men i detta exempel finns den redan och skall endast aktiveras. Ändra i **/etc/inittab**, men gör först en kopia av filen.

```
dmacs /etc/inittab
```

På skärmen finns nu hela filen enligt föregående exempel. Raderna som berör terminalanslutningarna console och tty02..tty07 ser ut så här. console, tty02 och tty03 är redan aktiverade.

```
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:off:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
```

- Förflytta markören till raden med tty04 och position 6 med hjälp av piltangenterna, dvs till texten **off**. Om piltangenterna inte är korrekt uppsatta används kontrolltecken. (Ner: **CTRL-N**, Upp: **CTRL-P**, Fram: **CTRL-F**, Bak: **CTRL-B**).
- Ta bort texten med **off** med tre tryckningar på **CTRL-D**.
- Skriv in texten **respawn**. Jämför raden för tty03.

Den nya terminalens hastighet (baudrate) förutsätts här vara 9600 baud enligt standard. Om så ej är fallet kan även den sista parametern på raden ändras att peka på en annan tabell i filen **/etc/gettydefs**, exempelvis 2400. Textsträngen är egentligen bara en pekare till tabellen, men vanligen använder man baudrate som pekarsträng. Jämför innehållet i filen **/etc/gettydefs**, så att rätt textsträng används.

- Upprepa eventuellt detta förfarande på raderna för tty05, tty06 etc. om fler terminalanslutningar ska aktiveras.
- Om något blir fel avbryts *dmacs* med **CTRL-X C**.
- Tryck på **ESC Z** för att spara filen.

En enkel kontroll av att filen har uppdaterats kan göras med kommandot *cat*.

```
cat /etc/inittab
```

Hela filen visas på skärmen. Nedan visas endast samma rader som ovan, med terminalanslutning tty04 aktiverad.

```
co:123456:respawn:/etc/getty console console
t2:2:respawn:nice -16 /etc/getty tty02 9600
t3:2:respawn:nice -16 /etc/getty tty03 9600
t4:2:respawn:nice -16 /etc/getty tty04 9600
t5:2:off:nice -16 /etc/getty tty05 9600
```

```
t6:2:off:nice -16 /etc/getty tty06 9600
t7:2:off:nice -16 /etc/getty tty07 9600
```

Observera! För att systemet skall läsa den ändrade filen **/etc/inittab**, måste därefter följande kommando ges. Därefter kan den aktiverade terminalen användas som inloggningsterminal.

```
telinit q
```

Exempel 2:

Modemkoppling av seriekanal tty05 med inloggning.

I detta fall förutsätter vi att seriekanal tty05 skall kopplas till ett externt modem (2400 baud) med inloggningsmöjlighet.

Anslutningen tty05 aktiveras med editorn *dmacs* i filen **/etc/inittab** enligt Exempel 1 på föregående sidor, om detta inte redan gjorts. Dessutom ändras, i **/etc/inittab**, den parameter som står sist på raden för enheten tty05 till strängen **2400**, om modemet är ett 2400 bauds modem. Jämför med motsvarande rad i filen **/etc/gettydefs**.

Split speed med olika sändande och mottagande hastighet kan inte användas på några av tty-portarna. Använd i så fall ett modernt modem som klarar av att köra med olika hastigheter mot dator och linje.

I tabellfilen **/etc/gettydefs** får motsvarande rad inte innehålla parametern **CLOCAL** för att modemsignalen DCD skall kännas av. Denna parameter måste därför tas bort i **/etc/gettydefs** för modem som kräver signalering med DCD.

Observera! För att kunna utnyttja modemsignalering krävs att rätt kabel används. Se **Förberedelser för installation**.

Efter ändring kommer filen **/etc/inittab** att innehålla följande rad för enheten **/dev/tty05**. För resten av filen, se exemplet i början av kapitlet. Använd kommandot *cat /etc/inittab* för att se filen. De ändrade strängarna är markerade med fetstil nedan.

```
t5:2:respawn:nice -16 /etc/getty tty05 2400
```

Filen **/etc/gettydefs** ändras med *dmacs*:

```
dmacs /etc/gettydefs
```

Raden med 2400 skall se ut enligt nedan (uppdelad på flera rader i handboken):

```
2400# B2400 HUPCL IXON # B2400 ICRNL IXON IXANY IXOFF OPOST \
     ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #2400
```

Observera! För att systemet skall läsa den ändrade filen **/etc/inittab**, måste därefter följande kommando ges. Därefter kan en användare ringa in utifrån till modemkanalen och logga in som vanlig användare.

```
telinit q
```

Exempel 3:

Modemkoppling av seriekanal tty05 utan inloggning.

I detta fall förutsätter vi att seriekanal tty05 skall kopplas till ett externt modem (2400 baud) men utan inloggningsmöjlighet. Kommandot *cu* används då för att ringa ut via modemet.

I detta fall får porten inte vara aktiv i filen **/etc/inittab**. Motsvarande rad skall vara markerad **off** i **/etc/inittab**.

```
t5:2:off:nice -16 /etc/getty tty05 2400
```

Baudrate och modemsignalering sätts upp med parametrar till det kommando som används för att ringa ut, t ex *cu* eller *kermit*.

Observera! För att kunna utnyttja modemsignalering krävs att rätt kabel används.

Exempel 4:

Modemkoppling för inloggning och för att ringa ut.

I detta fall förutsätter vi att tidigare använda seriekanal, som det yttre modemet kopplats till, används nu också, men nu skall vi även ha möjlighet att ringa både ut och in på en och samma kanal - dock ej samtidigt.

Porten skall vara aktiverad enligt följande i filen **/etc/inittab**.

```
t5:2:respawn:nice -16 /etc/getty -u60 tty05 2400
```

För förklaring av tillval till *getty* se **Referenshandboken**.

Observera! Modemet får i detta fall inte ligga med bärvågsindikering (DCD) konstant hög.

I filen **/etc/gettydefs** skall denna rad finnas

```
2400# B2400 HUPCL IXON # B300 ICRNL IXON IXANY IXOFF OPOST \
      ONLCR TAB3 ISIG ICANON ECHO ECHOE ECHOK #login: #2400
```

För att detta nu skall aktiveras krävs att en eventuell tidigare gettyprocess mot seriekanalen dödas, eller, om det inte funnits någon mot seriekanalen tidigare, utför kommandot

```
telinit q
```

9

9. Skrivarhantering

9. Skrivarhantering

9.1. Utskrift till skrivare

9.1.1 Val av lp eller lpr skrivarsystem

Vid leverans är systemet genererat med en skrivarutgång på utgången märkt **printer**. Den enhet som arbetar mot skrivaren har beteckningen **/dev/lp**.

Kommandot *lp* skriver till skrivaren genom ett spoolersystem, men det finns även ett annat spoolersystem till vilket utskrifter sker med kommandot *lpr*.

Normalt används alltid endast ett av dessa båda system och *lp*-systemet är normalt aktiverat vid leveransen.

Innan *lp*-kommandot kan användas måste emellertid följande kommando ges för att tillåta utskrifter med *lp*-systemet. Logga in som **root** eller **lp** för att ge detta kommando:

```
/usr/lib/accept main
```

Om istället *lpr*-systemet skall användas måste filen **/etc/rc** ändras så att *lpr*-systemet startas istället för *lp*-systemet systemstart. Dessutom skall då även kommandot *print* ändras. Detta kommando är en shell-procedur som ligger i filen **/bin/print** och anropar *lp*, varför *lp*-kommandot i denna fil skall ändras till ett *lpr*-kommando.

Exempel:

Utskrift med *lp*-systemet:

```
lp /etc/motd
```

Nu skrivs innehållet i filen **/etc/motd** ut på systemets standard-skrivare, vilken normalt är ansluten till utgången **/dev/lp**.

Exempel:

Utskrift med *lpr*-systemet.

```
lpr /etc/motd
```

Det är även möjligt att aktivera flera skrivare i systemet. Dessa kan till exempel kopplas till terminalportar, vilka då inte ska vara aktiverade i filen **/etc/inittab**. Seriekanaler som skall användas för skrivare, kan med kommandot */etc/mknod* döpas om med enhetsnamn som inte börjar på ttyxx, t.ex. **/dev/laser** eller liknande.

I kommandot *lp* eller *lpr* kan som parameter anges vilken logisk skrivare som ska användas. Namnet på den logiska skrivaren är kopplat till en viss fysisk skrivare. Om ingen logisk skrivare anges i kommandot används den som definierats som standard i systemet, normalt */dev/lp*.

Vid utskrift till skrivare med kommandot *lp* behandlas texten först av ett program i en kommandofil (shell-script) innan den matas ut till skrivaren. Parametern logisk skrivare anger vilken kommandofil som ska användas.

Vid utskrift med kommandot *lpr*, kan utskriften föregås av användarrubriker och/eller initieringssekvenser som automatiskt sänds från olika filer, valda av parametern logisk skrivare.

9.1.2 Aktivering av lpr istället för lp

Om spoolersystemet *lpr* skall användas i stället för *lp*-systemet, måste filen **/etc/rc** och kommandofilen **/bin/print** ändras. Detta sker med kommandot *dmacs*.

```
dmacs /etc/rc
```

Här visas nu filen **/etc/rc** på skärmen. De rader som skall ändras är i början av filen och ser ut som följer med *lp* aktiv:

```
:
# Spooler system
if [ -x /usr/bin/lp ]
then
     SPOOLER=lp
else
     SPOOLER=lpr
fi
...
```

Raderna ändras sålunda att *lp* byts ut mot *lpr* och vice versa (tre ställen). Början av filen **/etc/rc** skall därefter innehålla följande text:

```
:
# Spooler system
if [ -x /usr/bin/lpr ]
then
     SPOOLER=lpr
else
     SPOOLER=lp
fi
...
```

Därefter bör kommandofilen **/bin/print** ändras så att den anropar *lpr* istället för *lp*. Detta görs också med *dmacs*.

Filen ser från början ut på följande sätt:

```
pr -l51 -b -w112 $* | lp
```

Efter ändringen skall den se ut på följande sätt:

```
pr -l51 -b -w112 $* | lpr
```

För att funktionen ska aktiveras via filen **/etc/rc** måste därefter systemet tas ner med */etc/shutdown*, och därefter upp igen med kommandot *telinit 2*.

9.1.3 Ändring av hastighet till skrivare

Som standard sker kommunikationen mot skrivare med 9600 baud. Vissa skrivare kan inte hantera denna överföringshastighet, varför hastigheten måste ställas om för skrivaren. Detta kan göras genom att lägga in lämpligt *stty*-kommando i kommandofilen för den använda logiska skrivaren. Då lpr-systemet används ändras istället hastigheten direkt för den fysiska enheten genom kommandot */etc/setspeed*, vilket kräver super user-privilegier. Detta kommando kan läggas till i filen **/etc/rc** för att automatiskt ställa om hastigheten då systemet startas. *setspeed* kan även ställa om andra parametrar som t.ex. antal databitar och paritet.

Observera! Om kommandofilen för den logiska printern innehåller *stty*-kommandon kan inte *setspeed*-kommandot användas. Detta är normalt fallet när ***lp***-systemet används.

Ändring av hastigheten vid användning av *lp*-systemet.

I **lp**-systemet styrs en logisk skrivare av en shellprocedur i vilken ett *stty*-kommando läggs in för anpassning till skrivarens överföringshastighet, t ex enligt följande exempel, som visar början av en sådan procedur. Se kapitel 9.3 för ett exempel på hur en ändring görs..

```
:
# lp interface
#
stty 19200 <&1              # Ny Baudrate (Tillägg)
...
...
...
```

Ändring av hastigheten vid användning av *lpr*-systemet.

Nedan följer en beskrivning hur en kommandorad lägges in i filen **/etc/rc** för att ändra skrivarutgångens överföringshastighet till 1200 baud när ***lpr***-systemet används.

Kommandoraden är:

```
nice -19 /etc/setspeed -h 1200 </dev/lp &
```

Filen **/etc/rc** innehåller kommandon som exekveras automatiskt vid uppstart av systemet innan någon inloggning sker. Utseendet på denna fil kan skilja sig något men instruktionerna som följer nedan är oberoende av det exakta utseendet.

Observera! Du måste vara säker på att den ändring du gör i **/etc/rc** är korrekt, annars kan systemets funktion äventyras. Om något går galet kan *dmacs* avbrytas med **CTRL-X C** och **y**. Se **Användarhandboken** för information om *dmacs*.

Starta *dmacs* från rootbiblioteket med kommandot nedan. Filens innehåll visas då, vilket kan vara enligt nedan. På grund av begränsad radlängd i manualen delas långa rader i listningen nedan. På bildskärmen kan de se annorlunda ut.

```
dmacs /etc/rc
:
# Spooler system
if [ -x /usr/bin/lp ]
then
     SPOOLER=lp
else
     SPOOLER=lpr
fi
```

**** Här följer ett antal rader i filen som ej visas i handboken. ****

**** Nära slutet av filen finns följande rader. ****

```
# Start cron
if [ -x /etc/cron ]
then
     rm -f /usr/lib/cron/FIFO
     n=`ps -lax | fgrep cron | wc -l`
     if [ $n = 0 ]
     then
          nice -16 /etc/cron
          echo "CRON started"
     fi
fi
if [ "$SPOOLER" = lpr ]
then
     # Remove lock files
     rm -f /usr/spool/lpd/lock /usr/spool/lpd/lpdctl
     rm -f /usr/spool/lock/* /usr/spool/lpd/ERRLOG
     nice -20 /usr/lib/lpd &
     echo "LPD spooler started"
else
     # Start lpscheduler
     n=`ps -lax | fgrep lpsched | wc -l`
     if [ $n = 0 ]
     then
          rm -f /usr/spool/lp/SCHEDLOCK
          nice -20 /usr/lib/lpsched
          echo "LP scheduler started"
     fi
fi
```

I filen ska raden med *setspeed* läggas till. Denna rad skall läggas in i filen **/etc/rc** före raderna med start av spoolersystemet.

Gå ner till raden med if ["$SPOOLER" = lpr] och ytterligare ett par rader ner till # Remove lock files med några **CTRL-N** (eller nedåtpil). Ge **RETURN** för att ge plats för en ny rad och gå upp till den nya raden med **CTRL-P** (eller uppåtpil). Skriv därefter dit det nya kommandot. Om allt ser rätt ut enligt nedan sparas filen sedan med **ESC Z**.

Innehållet bör därefter listas med *cat* eller *pg* för kontroll. Filen får följande utseende efter tillägget. Endast en del av filen, med den inlagda raden för *setspeed* visas nedan:

```
cat /etc/rc

...
```

```
...
# Start cron
if [ -x /etc/cron ]
then
     rm -f /usr/lib/cron/FIFO
     n=`ps -lax | fgrep cron | wc -l`
     if [ $n = 0 ]
     then
          nice -16 /etc/cron
          echo "CRON started"
     fi
fi
if [ "$SPOOLER" = lpr ]
then
     nice -19 /etc/setspeed -h 1200 </dev/lp &
     # Remove lock files
     ...
     ...
```

För att funktionen ska aktiveras via filen **/etc/rc** måste därefter systemet tas ner till enanvändarnivån med */etc/shutdown*, och därefter upp igen med kommandot *telinit 2*.

9.2. Lp skrivarhantering

Skrivarsystemet *lp* är en serie kommandon, som hanterar en eller flera skrivare. Texter som skall skrivas ut läggs på kö (spooling) till olika skrivare och de verkliga utskrifterna sker i bakgrunden. *lp* kan även skriva till skrivare som finns på andra datorer, anslutna via ett nätverk. Systemadministratören kan definiera om och anpassa systemfilerna till olika typer av skrivare och grupper av skrivare kan hanteras genom att systemet självt väljer en ledig skrivare om utskriften sänts till gruppen istället för till en speciell skrivare. Detta ger maximalt kapacitetsutnyttjande i systemet. Vid utskriften kan konverteringar göras automatiskt för att anpassa formatet till den fysiska skrivaren. Användaren kan med kommandon utföra följande:

- Sända en skrivbegäran till kön eller ta bort den.
- Hindra eller tillåta skrivkö till olika enheter.
- Starta och stoppa utskrifter från köerna.
- Ändra skrivarkonfigurationen.
- Lista status för *lp*-systemet.

Detta kapitel beskriver hur den som är *lp*-administratör kan använda de särskilda kommandon som inte vanliga användare har tillgång till för att modifiera *lp*-systemet för bästa prestanda. Användare **lp** finns definierad i systemet som *lp*-administratör. Ett lämpligt lösenord definieras av super-user innan inloggning som **lp** sker. I filen **/usr/spool/lp/.profile** görs lämpliga terminaldefinitioner. Efter inloggning som **lp** behöver inte hela sökvägen anges till *lp*-kommandona då **PATH** innehåller även **/usr/lib**.

I detta kapitel anges referenser till olika kommandon och filer.

För kommandon och filer som inte beskrivs i **Referenshandboken**, finns referenser med formen *namn*(N), vilka refererar till andra handböcker. I de fall N är en siffra eller en siffra följd av en bokstav, refereras till handboken för utvecklingssystemet.

9.2.1 Översikt över funktioner i lp-systemet

Definitioner

Vissa termer måste definieras innan en kort beskrivning ges av de olika kommandona i *lp*-systemet. *lp* är uppbyggt för att kunna anpassas till många olika konfigurationer av datorsystem. Ändringar införs med kommandot *lpadmin*.

lp skiljer på begreppet logisk skrivare och fysisk skrivare(device).

En fysisk skrivare är en fysisk enhet (device) eller en fil och representeras av ett fullständigt filnamn (pathname) i filsystemet.

En logisk skrivare (nedan ofta bara kallad 'skrivare') har ett logiskt namn och representerar en fysisk skrivare. Den logiska skrivaren kan vid olika tidpunkter associeras med olika fysiska skrivare. En klass av skrivare (class) är ett logiskt namn som anger en grupp logiska skrivare enligt en ordnad lista. Varje skrivarklass måste innehålla minst en skrivare. Varje logisk skrivare kan tillhöra en eller flera klasser eller ingen klass alls. En destination är antingen en logisk skrivare eller en klass av skrivare. En av destinationerna kan definieras som standardskrivare i systemet. Om inte användaren anger någon destination då kommandot *lp* används, kommer utdata att skrivas till denna standardskrivare. Om i *lp* anges en logisk skrivare sänds utskriften dit, men om en klass anges så väljer systemet den första lediga skrivaren inom den angivna klassen.

Då kommandot *lp* används skapar det en skrivbegäran, som består av de filer som ska skrivas ut samt tillvalsflaggorna från *lp*:s kommandorad. Ett gränssnittsprogram krävs för varje logisk skrivare. Bakgrundsprogrammet som hanterar utskrifterna heter *lpsched* och sänder skrivbegäran till olika destinationer genom att anropa olika gränssnittsprogram, vilka sköter den direkta fysiska utskriften och eventuell formatering och annan anpassning till skrivaren.

lp-konfigurationen i ett system består av fysiska enheter (device), destinationer och gränssnittsprogram.

Kommandon

Generella kommandon för alla användare:

Med *lp*-kommandot sänder användarna utskrifter från filer. Det skapar och sänder utskriftsbegäran till systemet och returnerar ett könummer (request-id). Formen på könumret (request-id) är

```
dest-seqno
```

och sänd till användaren. Det skrivs vanligen ut på terminalen. I könumret är **dest** den destination dit utskriftsbegäran sändes och **seqno** är ett unikt sekvensnummer i hela *lp*-systemet.

Med kommandot *cancel* kan utskriftsbegäran som sänts tas bort ur kön. Även utskrifter som redan börjat skrivas ut stoppas. Som parameter anges antingen en (eller flera) logiska skrivare direkt, för att stoppa de utskrifter som pågår på dessa skrivare eller ett eller flera könummer (request-id), för att ta bort specifika utskriftsbegäran.

Kommandot *disable* hindrar *lpsched* från att vidarebefordra utskrifter till en logisk skrivare.

Kommandot *enable* tillåter *lpsched* att vidarebefordra utskrifter till en logisk skrivare.

Kommandon som enbart *lp*-administratören får använda:

En person eller flera, som fungerar som *lp*-administratörer, kan utföra nedanstående speciella funktioner. Det krävs antingen att de loggar in till systemet som 'super-user' eller med användarnamnet **lp** för att de ska tillåtas utföra dessa funktioner. Alla filer i *lp*-systemet ägs av använ-

daren **lp** utom kommandona *lpadmin* och *lpsched*, vilka ägs av root. Följande kommandon kommer att beskrivas mer i detalj i följande kapitel:

lpadmin	Ändrar konfigurationen hos *lp*-systemet. I flera fall måste *lpsched* stängas av medan förändringarna sker.
lpsched	Hanterar köerna och sänder utskrifter till de olika gränssnittsprogrammen, vilka sköter den direkta utskriften till de fysiska skrivarna.
lpshut	Stoppar *lpsched*. All printning stoppas temporärt, men andra kommandon i *lp*-systemet kan fortfarande användas.
accept	Tillåter användaren att med *lp* sända utskrifter till angivna destinationer.
reject	Hindrar *lp* från att acceptera utskrifter till angivna destinationer.
lpmove	Flyttar köade utskrifter från en destination till en annan. Även hela kön till en destination kan flyttas till en annan. Detta kommando kräver att *lpsched* är stoppad.
lppg	Filterprogram (i **/usr/spool/lp/bin**) för sidvis utskrift, styrt via användarkommandot *lpsubmit*.
lpsubmit	Kommando som styr *lppg* och möjliggör interaktiv styrning av utskrifterna sida för sida.

9.2.2 Ändra lp-konfigurationen - lpadmin

Ändringar i konfigurationen hos *lp*-systemet gös med kommandot *lpadmin* och filerna skall aldrig ändras manuellt. *lpadmin* spärrar mot vissa ändringar om inte *lpsched* stoppas först. Vissa funktioner kan emellertid utföras med systemet igång. Detta anges då nedan.

Tillägg av nya destinationer

Följande information krävs för att *lpadmin* ska lägga till en ny logisk skrivare

1. Logiska skrivarnamnet (**-p** *skrivare*) är ett valfritt namn som måste följa reglerna nedan:

   Det får inte vara längre än 14 tecken.

   Det måste bestå av enbart alfanumeriska tecken samt understrykningstecken.

   Det får inte vara samma som en redan definierad destination (skrivare eller klass).

2. Den fysiska skrivaren (device) som associeras med den logiska skrivaren anges med (**-v** *device*). Detta är ett fullständigt filnamn (pathname) till en fysisk enhet eller en fil till vilken användare **lp** har skrivtillstånd. Den fysiska enheten kan vara en skrivarutgång eller en login-terminal. I vissa fall används denna ej (t.ex. vid anslutning via ett nätverk), och kan då definieras som **/dev/null**.
3. Skrivarens gränssnittsprogram. Detta kan specificeras på ett av tre sätt.

   Det kan väljas från en lista av modellgränssnitt som levereras med *lp*-systemet. Detta görs med (**-m** *model*).

   Det kan vara samma gränssnittsprogram som en annan redan definierad skrivare (**-e** *annan skrivare*).

   Det kan vara ett eget filter-program, som skall vara i en fil som ägs av användare lp. Det väljs då med (**-i** *filterprogram*) och kopieras automatiskt till lp-systemet (biblioteket **/usr/spool/lp/interface**) då skrivaren definieras med *lpadmin*.

Valfri information enligt nedan kan anges när en ny logisk skrivare skapas:

1. Optionen **-h** kan specificeras för att visa att den fysiska enheten är en skrivarutgång eller en fil. Detta antas annars som standard. Om enheten däremot är en login-terminalport, måste **-l** specificeras till *lpadmin*. Detta tvingar *lpsched* att automatiskt koppla bort inloggningsmöjligheten på den använda fysiska enheten varje gång *lpsched* startar. Detta rapporteras då status läses med kommandot *lpstat*:

   ```
   lpstat -pa
   printer a (login terminal) disabled Oct 31 11:15
   - disabled by scheduler: login terminal
   ```

   Detta sker för att inte någon ska logga in på den kanal där utskriften sker.
2. Den nya logiska skrivaren kan läggas in som medlem i en eller flera skrivarklasser. Detta sker med (**-c***class*). Om en klass som inte finns anges, skapas den. Namnet på klassen måste vara enligt samma regler som namnet på en logisk skrivare enligt ovan.

Exempel

Följande exempel kommer att ligga till grund för flera andra exempel i senare kapitel.

1. Skapa en logisk skrivare med namnet pr1, som associeras med enheten **/dev/printer** och välj modellprogrammet **dumb** som gränssnittsprogram.

   ```
   /usr/lib/lpadmin -ppr1 -v/dev/printer -mdumb
   ```
2. Skapa en logisk skrivare med namnet pr2, som associeras med enheten **/dev/tty22** och välj ett gränssnittsprogram som är en ändrad version av modellprogrammet **prx**. Döp gränssnittsprogrammet till

pnew. /dev/tty22 är normalt en inloggningsterminal i systemet. Kopiera först modellprogrammet och ändra innan *lpadmin* används.

```
cp /usr/spool/lp/model/prx pnew
dmacs pnew (Gör ändringarna!)
/usr/lib/lpadmin -ppr2 -v/dev/tty22 -ipnew -l
```

3. Skapa en logisk skrivare med namnet pr3, som associeras med enheten **/dev/tty23**. Skrivaren pr3 skall läggas in i en ny klass som ska heta *cl1* och använder samma gränssnittsprogram som skrivare pr2.

```
/usr/lib/lpadmin -ppr3 -v/dev/tty23 -epr2 -ccl1
```

Ändra en befintlig destination

Vid ändring av en befintlig destination används namnet för den logiska skrivaren (**-p***printer*) som referens. Följande kan ändras:

1. Den fysiska enheten (device) för skrivaren kan ändras med (**-v***device*). Om bara detta skall ändras kan det göras med *lpsched* igång.
2. Ett annan gränssnittsprogram för den logiska skrivaren kan väljas (**-m***model*), (**-e***annan skrivare*), (**-i***interface*).
3. Skrivaren kan specificeras som direktkopplad (**-h**) eller via en terminalport som normalt används för inloggning (**-l**).
4. Skrivaren kan läggas in i en ny eller redan befintlig klass (**-c***class*).
5. Skrivaren kan tas bort från en klass (**-r***class*). Då den sista medlemmen i en klass tas bort, försvinner även klassen. Ingen destination, som det finns en kö av utskrifter till, kan tas bort. I detta fall måste *lpmove* eller *cancel* först användas för att flytta eller ta bort kön.

Exempel

Dessa exempel är baserade på den *lp*-konfiguration som skapades i exemplet i föregående kapitel.

1. Lägg in skrivare pr2 till klassen *cl1*.

```
/usr/lib/lpadmin -ppr2 -ccl1
```

2. Byt skrivare pr2:s gränssnittsprogram till modellprogrammet **prx**, ändra dess fysiska enhet till **/dev/tty24** och lägg in den i den nya klassen, som heter *cl2*. Tillsammans med ovanstående kommer den då att tillhöra två klasser.

```
/usr/lib/lpadmin -ppr2 -mprx -v/dev/tty24 -ccl2
```

   Notera att skrivare pr2 och pr3 nu använder olika gränssnittsprogram, trots att pr3 ursprungligen skapades med samma gränssnittsprogram som pr2.

3. Specificera att skrivare pr2 nu är en direktkopplad skrivare.

```
/usr/lib/lpadmin -ppr2 -h
```

4. Lägg in även skrivare pr1 i klassen *cl2*.

```
/usr/lib/lpadmin -ppr1 -ccl2
```

   Medlemmarna i klassen *cl2* är nu pr2 och pr1, i denna ordning! Utskrifter som sänds till klassen *cl2* kommer att sändas till pr2 om

den är ledig; endast om pr2 är upptagen sänds den till nästa, d.v.s. pr1.

5. Ta bort skrivare pr2 och pr3 från klassen *cl1*.

```
/usr/lib/lpadmin -ppr2 -rcl1
/usr/lib/lpadmin -ppr3 -rcl1
```

Eftersom pr3 var den sista medlemmen i klass *cl1*, tas även klassen bort.

6. Lägg in pr3 i en ny klass som ska heta *cl3*.

```
/usr/lib/padmin -ppr3 -ccl3
```

Specificera standarddestination i systemet

Standard-destinationen i systemet kan ändras även medan *lpsched* är igång.

Exempel

1. Ange att klassen cl1 skall vara standarddestination, d.v.s. om kommandot *lp* ges utan att destinationen anges sänds utskriften till någon skrivare i klassen *cl1*.

```
/usr/lib/lpadmin -dcl1
```

2. Ta bort standarddestinationen.

```
/usr/lib/lpadmin -d
```

Ta bort destinationer i systemet

Klasser och logiska skrivare kan tas bort endast om det inte finns några köade utskrifter till dem. Köade utskrifter måste antingen stoppas helt med *cancel* eller flyttas till andra destinationer med *lpmove* innan destinationen kan tas bort. Om den borttagna destinationen är definierad som standard-destination så kommer ingen standard-destination att vara definierad efter borttagandet. Om en klass tas bort kvarstår dess tidigare medlemmar fortfarande som logiska skrivare. En klass tas även bort automatiskt om den sista medlemmet tas bort.

Exempel

1. Gör skrivare pr1 till systemets standard-destination.

```
/usr/lib/lpadmin -dpr1
```

Ta bort skrivare pr1:

```
/usr/lib/lpadmin -xpr1
```

Nu finns ingen standard-destination.

2. Ta bort den logiska skrivaren pr2.

```
/usr/lib/lpadmin -xpr2
```

Även klassen *cl2* togs därmed bort eftersom pr2 var dess enda medlem.

3. Ta bort klassen *cl3*.

```
/usr/lib/lpadmin -xcl3
```

Klassen *cl3* tas bort, men skrivaren pr3 finns fortfarande kvar som destination.

9.2.3 Gör en utskriftsbegäran med lp

Då *lp*-destinationer har skapats i systemet, kan användarna sända utskrifter med kommandot *lp*. Det könummer (request-id) som returneras kan användas för ett se om utskriften är klar eller för att stoppa utskriften.

lp-systemet bestämmer till vilken destination som utskriften skall ske med följande regler i ordning.

- Om användaren direkt anger destinationen **-d***dest* på *lp*-kommandoraden används denna (dest).
- Om shellvariabeln **LPDEST** är definierad sänds utskriften annars till den destination som den anger.
- Om en standard-destination är definierad i *lp*-systemet sänds utskriften annars dit.
- Om ingen av ovanstående finns, blir det ingen utskrift.

Exempel

1. Det finns minst fyra sätt att med kommandot *lp* sända en fil (t.ex. filen **/etc/passwd**) för utskrift till standarddestinationen i systemet.

```
lp /etc/passwd
lp </etc/passwd
cat /etc/passwd | lp
lp -c/etc/passwd
```

   De sista tre sätten gör att en kopia av filen skrivs ut, medan det första sättet skriver direkt från den angivna filen. Om därvid filen ändras under tiden mellan *lp*-kommandot och då den verkligen skrivs ut, kommer ändringarna att påverka utskriften.

2. Skriv ut två kopior av filen abc till den logiska skrivaren xyz och lägg in en titel "my file" på den extra sida med användarnamn och datum som skrivs före filen. Kommandot *pr* formaterar filen till sidor.

```
pr abc | lp -dxyz -n2 -t"my file"
```

3. Skriv ut filen xxx på en skrivare av typen Facit 4512 via den logiska skrivaren prp. Ange tillvalet **-17** till gränssnittsprogrammet för prp för att välja antal tecken per rad. Ge ett meddelande till användarens terminal när utskriften är klar.

```
lp -w -dprp -o17 xxx
```

   I ovanstående exempel är **-17** ett tillval som ges till gränssnittsprogrammet för Facit 4512 genom att *lp* skalar bort tecken **o** i **-o17**. Se beskrivningen av kommandot *lpadmin*.

9.2.4 Lista lp-status - lpstat

Kommandot *lpstat* listar statusinformation om utskriftsbegäran i köerna i *lp*-systemet, destinationer och status för utskriftshanteraren *lpsched*.

EXEMPEL

1. Lista status för alla köade utskrifter som denna användare sänt.

   ```
   lpstat
   ```

   Statusinformationen för varje utskriftsbegäran innehåller könummer (request-id), användarens loginnamn, totala antalet tecken att skriva samt datum och tid då utskriften begärdes.

2. Lista status för de logiska skrivarna p1 och p2.

   ```
   lpstat -pp1,p2
   ```

3. Lista status för hela *lp*-systemet.

   ```
   lpstat -t
   ```

Om angiven utskriftsbegäran har sänts till en annan dator i ett nätverk, startas automatiskt shell-proceduren */usr/spool/lp/remote/lpstat*, som begär information genom att exekvera lpstat på den dator till vilken utskriftsbegäran sänts.

9.2.5 Ta bort utskriftsbegäran - cancel

Utskriftsbegäran kan tas bort med kommandot *cancel*. Två olika typer av argument kan ges könummer (request-id) eller logiska skrivarnamn. Om könummer ges tas angivna utskriftsbegäran bort ur kön och om utskrift redan pågår stoppas den direkt. Om ett skrivarnamn anges stoppas den utskrift som just pågår på den skrivaren. Båda typerna av argument kan anges samtidigt.

Exempel

Stoppa den utskrift som pågår vid den logiska skrivaren xyx.

```
cancel xyz
```

Om den användare som stoppade en utskrift inte är samma som sände den till utskriftskön, sänds automatiskt ett meddelande via *mail* till ägaren av utskriftsbegäran. *lp*-systemet tillåter vem som helst att stoppa utskrifter för att undvika de dröjsmål som annars kan uppstå för att uppsöka en systemansvarig vid felaktigheter i utskriften.

Om angiven utskriftsbegäran har sänts till en annan dator i ett nätverk, startas automatiskt shell-proceduren */usr/spool/lp/remote/cancel*, som exekverar *cancel* på på den dator till vilken utskriftsbegäran sänts.

9.2.6 Tillåt/hindra sändning av utskrifter - accept och reject

lp tillåter inte utskrift till en nyskapad destination förrän lp-administratören kontrollerat att den är klar att ta emot utskrifter, t.ex. att gränssnittsprogrammet är rätt, att den fysiska enheten kopplats in etc. Med kommandot accept anges att det är tillåtet för lp att sända utskrifter till destinationen.

För att temporärt hindra *lp* från att sända utskrifter till en destination kan kommandot *reject* användas. Om t.ex. en skrivare har flyttats eller är trasig eller om det för tillfället finns för många utskrifter i kö till skrivare, kan detta vara lämplig. Notera att utskrifter som redan är i kö till den angivna skrivaren fortsätter att skrivas ut även efter *reject*. Kommandot *accept* används sedan för att på nytt tillåta utskriftsbegäran till skrivaren.

För att se om en destination är tillgänglig för *lp* kan kommandot *lpstat* ges med tillvalet **-a**.

Exempel

1. Gör så att *lp* vägrar ta emot utskrifter till destinationen xyz:

   ```
   /usr/lib/reject -r"printer xyz needs repair" xyz
   ```

   Alla användare som försöker skriva ut med *lp* till xyz kommer att erhålla följande felmeddelande.

   ```
   lp can not accept requests for destination "xyz"
   -- printer xyz needs repair
   ```

2. Tillåt *lp* att sända utskrifter till destinationen xyz.

   ```
   /usr/lib/accept xyz
   ```

9.2.7 Tillåt/hindra utskrifter - enable och disable

Kommandot *enable* gör att *lp*-systemet tillåter utskrifter till en specificerad logisk skrivare. Utskrifter tillåts enbart via gränssnittsprogram för tillåtna skrivare. Observera att det är möjligt att tillåta utskrifter till en skrivare (*enable*) samtidigt som man kan hindra att fler utskrifter köas till skrivaren (*reject*) med *lp*.

Kommandot *disable* är motsatsen till *enable* och hindrar all utskrift till skrivaren, oberoende av om *lp* tillåts sända utskrifter till skrivarens kö (*accept*) eller ej (*reject*). Skrivare kan behöva kopplas bort från *lp*-systemet av olika skäl, som fysiska fel på skrivaren, att papperet fastnat eller om man inte vill att skrivaren ska gå då ingen personal finns på kvällen. Om utskrifter pågår vid *disable*-kommandot kommer pågående utskrift att skrivas ut igen i sin helhet (från början), antingen direkt på en annan skrivare om utskriften var riktad till en klass, eller på samma skrivare när den senare tillåts (*enable*). Optionen **-c** till *disable* gör att eventuellt pågående utskrift stoppas helt (jämför kommandot *cancel*). Detta är lämpligt om skrivaren ska stängas av på grund av felaktiga utskrifter.

Exempel

Hindra utskrift till skrivare xyz, t.ex. på grund av att papperet fastnat. Andra raden nedan skrivs ut av kommandot *disable*.

```
disable -r"paper jam" xyz
printer "xyz" now disabled
```

Lista status för skrivare xyz.

```
lpstat -pxyz
printer "xyz" disabled since Jan 5 10:15 - paper jam
```

Tillåt skrivare xyz igen.

```
enable xyz
```

9.2.8 Flytta utskriftsbegäran mellan destinationer - lpmove

Ibland behöver köade utskriftsbegäran flyttas mellan destinationer. T.ex. då en skrivare kopplas bort från systemet kan det vara lämpligt att flytta över hela kön av utskrifter till en annan skrivare som fungerar. Detta görs med kommandot *lpmove*. Med *lpmove* kan även specifika utskriftsbegäran flyttas till andra destinationer. *lpmove* kan enbart användas då *lpsched* är stoppad och kan alltså inte flytta utskrifter som redan håller på att skrivas ut.

Om *lpmove* har flyttat bort en skrivarkö, hindras dessutom automatiskt *lp* från att sända fler utskrifter dit (jämför kommandot *reject*).

Exempel

1. Flytta hela kön med utskriftsbegäran till skrivare abc över till skrivaren xyz.

   ```
   /usr/lib/lpmove abc xyz
   ```

   Alla flyttade utskriftsbegäran kommer att byta namn från abc-nnn till xyz-nnn. Som en sidoeffekt kommer destinationen abc att inte längre tillåtas ta emot utskrifter. Ge kommandot */usr/lib/accept* för att kunna sända till skrivaren igen.

2. Flytta utskriftern prp-222 och abc-1200 till skrivaren xyz.

   ```
   /usr/lib/lpmove prp-222 abc-1200 xyz
   ```

   De två utskriftsbegäran byter därmed namn till xyz-543 och zyx1200. Numret i andra delen av namnet ändras om samma nummer redan finns i kön tidigare.

9.2.9 Stoppa och starta köhanteraren - lpshut och lpsched

lpsched (scheduler) är det program som hanterar utskriftsbegäran som sänds av *lp* till köerna, och startar det begärda gränssnittsprogrammet för den fysiska utskriften till skrivare eller andra enheter. Varje gång köhanteraren (*lpsched*) sänder en utskrift till ett gränssnittsprogram skrivs även ett meddelande i en log-fil,
/usr/spool/lp/log. Detta meddelande innehåller användarens logi-namn, det logiska skrivarnamnet och datum och tid då utskriften startades. Om en utskrift startats på nytt (exempelvis efter *disable/enable*, kan mer än ett meddelande för samma utskrift lagras. Eventuella fel-

meddelanden lagras också i log-filen. Varje gång processen *lpsched* startas döps log-filen om till **/usr/spool/lp/oldlog** och en ny log-fil skapas.

Inga utskrifter sker förrän *lpsched* har startats.

Kommandon nedan kan användas för att se om lpsched är igång eller ej. Den kallas p-scheduler i utskriften från lpstat.

```
lpstat -r
```

lpsched startas normalt via systemfilen **/etc/rc** då datorsystemet startas och är alltid aktivt tills hela systemet stängs av eller går ner i D-NIX enanvändarnivå. När *lpsched* skall startas vägrar den om filen **/usr/spool/lp/SCHEDLOCK** finns, vilket hindrar att flera köhanterare startas. Annars skapas denna fil då *lpsched* startas.

Ibland kan det vara nödvändigt att stoppa *lpsched* om *lp*-systemet skall ändras. Detta görs med kommandot

```
/usr/lib/lpshut
```

vilket direkt stoppar *lpsched*, varvid alla utskrifter stoppas. Alla utskrifter som finns i köerna stannar kvar och väntar, medan de utskrifter som redan pågår kommer att åter skrivas ut i sin helhet när *lpsched* åter startas.

För att återstarta ges kommandot

```
/usr/lib/lpsched
```

Kort efter detta kommando bör *lpstat*-kommandot ges för att se att *lpsched* verkligen kommit igång. Om inte kan det bero på att *lpsched* tidigare avslutats felaktigt utan att filen **/usr/spool/lp/SCHEDLOCK** tagits bort. Försök då igen med

```
rm -f/usr/spool/lp/SCHEDLOCK
/usr/lib/lpsched
```

Varefter *lpsched* bör komma igång korrekt.

All administration av *lp*-systemet måste ske via *lpadmin* kommandot. Att editera filer direkt medför att spool-deamonen ej startar. Detta händer t.ex. då man skapar en fil i **member** utan att göra detta via *lpadmin*.

9.2.10 Gränssnittsprogram för skrivare

Varje logisk skrivare i *lp*-systemet måste ha ett gränssnittsprogram som utför utskriften till den fysiska enheten. Ett gränssnittsprogram kan vara en shell-procedur eller ett annat exekverbart program, t.ex. ett C-program. I biblioteket **/usr/spool/lp/model** finns några modellprogram, som är shell-procedurer. Då *lpsched* startar upp en utskrift till en logisk skrivare PPP görs detta genom att gränssnittsprogrammet PPP startas med följande argument. Gränssnittsprogrammet måste ligga i biblioteket **/usr/spool/lp/interface** och startas automatiskt med följande argument.

```
interface/PPP id user title copies options file ...
```

där

id	är könumret (request-id), vilket returneras av *lp* då utskriftsbegäran görs.
user	är användarnamnet för den användare som gjorde utskriftsbegäran.
title	är en valfri titelsträng som angivits som argument till *lp*-kommandot.
copies	är antalet begärda kopior, vilket angivits som argument till *lp*.
options	är en lista, separerad av blanktecken med tillval för gränssnittsprogrammet. Dessa specificerades med -ooptions till *lp*.
file	är fullständiga filnamnet för filen som ska skrivas ut.
....	Flera filer kan specificeras i samma begäran från *lp*-kommandot.

Exempel

Följande exempel är utskrifter som gjorts av användare **smith**. Standarddestination har definierats som skrivare xyz. Varje exempel visar både *lp*-kommandot och motsvarande startkommando för gränssnittsprogrammet. Alla parametrar finns med, även de som är tomma strängar. I listan nedan är långa rader uppdelade på två för att få plats på sidan.

1.

```
lp /etc/passwd /etc/group
interface/xyz xyz-52 smith "" 1 "" /etc/passwd /etc/group
```

2.

```
pr /etc/passwd | lp -t"users" -n5
interface/xyz xyz-53 smith users 5 ""
     /usr/spool/lp/request/xyz/d0-53
```

3.

```
lp /etc/passwd -oa -ob
interface/xyz xyz-54 smith "" 1 "a b" /etc/passwd
```

När gränssnittsprogrammet startas kommer dess standard input från den tomma enheten **/dev/null** och både standard output och standard error output är riktat till den fysiska enhet som specificerats för skrivaren. För fysiska enheter tas även standard input från enheten om dess åtkomsttillstånd så tillåter. I de fall när enheten är en vanlig fil, lägges utskriften alltid till i slutet av filen.

Med dessa förutsättningar kan gränssnittsprogrammet formatera data på valfritt sätt och kan t.ex. sätta upp de rätta terminalparametrarna (jämför kommandot *stty*) överföringshastighet, antal databitar, paritet m m. Då en shell-procedur används kan följande enbart göras då enheten kan öppnas även för läsning:

```
stty mode ... <&1
```

Kommandot *stty* kräver nämligen att enheten anges som standard input.

När utskriften är färdig skall gränssnittsprogrammet avslutas med en utgångsstatus som indikerar om allt gått bra. De statuskoder som används av *lpsched* är följande:

Kod	Betydelse i lp-sched
0	Utskriften gick bra.
1 till 127	Fel uppstod vid utskriften, t.ex. alltför många icke-skrivbara tecken). De fel som returneras på detta sätt påverkar inte andra utskrifter i kön. *lpsched* sänder ett meddelande till användaren (mail) att det blev fel vid utskriften.
> 127	Dessa koder är reserverade för internt (utom 144) bruk i *lpsched* och får ej returneras av gränssnittsprogram (med undantag av 144).
144	Utgångsstatus 144 måste returneras av gränssnittsprogrammet om en signal mottages, som innebär att programmet avslutas i förtid.

Om fel som påverkar även andra utskrifter uppstår (t.ex. om ett begärt filterprogram inte finns) bör gränssnittsprogrammet koppla bort skrivaren med *disable*-kommandot så att utskrifterna i kön väntar. När en skrivare kopplats bort med *disable* kommer gränssnittsprogrammet automatiskt att avslutas genom signal 15 (jämför kommandot *kill*).

Ett antal modellprogram för olika skrivare levereras med systemet och kan användas för att skapa egna gränssnittsprogram. Dessa modellprogram, som finns i biblioteket **/usr/spool/lp/model**, har ett antal gemensamma tillval definierade för att styra utskrifterna.

När ett gränssnittsprogram skapats från en av dessa modeller, kan en eller flera av dessa tillval väljas vid utskriften med hjälp av *lp*- kommandots eget tillval **-o**. Flera tillval kan ges som en sträng med mellanslag, omgiven av apostrofer, t.ex. **-o'12 he'**. Exempelvis på utskrift med en standard titelsida:

```
lp -ddest -o'he' filnamn
```

Alla modeller har inte exakt samma uppsättning tillval beroende på att skrivarna har olika många finesser. Vissa tillval påverkar inte den fysiska skrivaren utan används för att sätta upp parametrar till den fysiska enheten eller avkodas internt i gränssnittsprogrammet.

Vill man veta exakt vilka tillval som gäller för en viss modell, är det lättast att skriva ut modellen på papper och titta under rubriken **Available options for printer NNNN 9999**.

Standardtillvalen till modellen för Facits P7010 laserskrivare är:

Tillval	Beskrivning
10	Väljer fonten "courier" med 10 tecken/tum.
12	Väljer fonten "prestige elite" med 12 tecken/tum
17	Väljer fonten "letter gothic" med 17 tecken/tum.
he	Gränssnittsprogrammet skriver ut en headersida innan övrig text skrivs ut.
ls	Sätter skrivaren i formatet Landscape.
pt	Sätter skrivaren i formatet Portrait.
nff	Ingen avslutande sidframmatning (formfeed) görs.
ntb	Tabuleringstecken expanderas inte.
raw	Ställer om enheten, där skrivaren är ansluten, till rå hantering.
sw	Tecknen konverteras till svensk ASCII standard vid utskriften.
us	Tecknen konverteras till amerikansk ASCII standard vid

utskriften.

? Skriver en hjälptext över gränssnittets tillgängliga tillval på den terminal användaren loggat in på.

Av ovan nämnda tillval är det några som alltid gäller(default) om man inte säger något annat, i detta exempel är det 10, pt och sw.

9.2.11 Att definiera fysiska enheter som lp-skrivare

Direktkopplade enheter

Som ett exempel på hur direktkopplade enheter definieras som skrivare i *lp*-systemet antar vi att **/dev/tty15** ska definieras som skrivare xyz. Följande urföres av den systemansvarige, inloggad som super-user:

1. Se till att *lp*-processen kan skriva till enheten och hindra andra processer att skriva dit.

```
chown lp /dev/tty15
chmod 600 /dev/tty15
```

2. Ändra **/etc/inittab** så att tty15 inte längre aktiveras som inloggningsterminal. Detta gör att processen /etc/getty inte tillåter någon att logga in på denna terminalingång. Ändra tredje fältet i raden för tty15 till strängen **off** enligt nedan.

```
t15:2:off:nice -16 /etc/getty tty15 1200
```

Ge sedan kommandot:

```
telinit Q
```

Om en */etc/getty*-process redan är igång för inloggning på tty15 tas den bort med kommandot *kill -9* med process-id som argument. Detta kan ses med kommandot *ps -xl*. När D-NIX-systemet startas på nytt kommer tty15 att initieras med standardparametrar, varför det är skrivarens gränssnittsprogram i *lp*-systemet som måste sätta upp rätt överföringshastighet (baud-rate) och andra moder enligt *stty*-kommandot.

3. Skapa den logiska skrivaren xyz i *lp*-systemet och använd modellprogrammet prx som gränssnittsprogram.

```
/usr/lib/lpadmin -pxyz -v/dev/tty15 -mprx
```

4. När xyz skapas kommer den inte att vara tillgänglig (disable) och *lp* kan inte sända utskrifter till den (reject). För att tillåta att utskrifter köas med *lp* till xyz, ges kommandot:

```
/usr/lib/accept xyz
```

Därefter tillåter *lp*-systemet att en kö med utskrifter får byggas upp till denna skrivare, för att skriva ut när skrivaren gjorts tillgänglig med *enable*-kommandot:

5. Innan kommandot *enable* ges för xyz kontrolleras att utskrifter verkligen kan göras till den fysiska skrivaren. Detta innebör t.ex. att papperets överkant är rätt justerat (top-of-form) och att skrivaren är påslagen och är on-line. Utskrifter till skrivaren möjliggörs med:

```
enable xyz
```

Nu kan utskrifter som sänts till den logiska skrivaren skrivas ut.

Inloggningsterminaler

Utskrift med *lp*-systemet kan även ske till en inloggningsterminal. Ska t.ex. en Facit 4512 skrivare kopplas till en terminalport och utskrift ska ske till denna med det logiska skrivarnamnet abc, görs följande:

1. Skapa den logiska skrivaren abc med modellgränssnittsprogrammet dumb:

   ```
   /usr/lib/lpadmin -pabc -v/dev/null -mdumb -l
   ```

   Vi anger här **/dev/null** som fysisk enhet eftersom den verkliga enheten kommer att specificeras varje gång abc görs tillgänglig med *enable*. Den verkliga enheten kan vara olika vid olika tillfällen. När abc skapas kommer den inte att vara tillgänglig och *lp* kommer inte heller att kunna sända utskrifter till den. För att tillåta att utskrifter köas till abc ges kommandot:

   ```
   /usr/lib/accept abc
   ```

   Därefter tillåts att utskrifter köas till abc för att senare skrivas ut när skrivaren gjorts tillgänglig. Detta bör emellertid inte ske förrän följande har utförts.

2. Utför inloggning på den önskade terminalingången, om detta inte redan gjorts.

3. Använd kommandot *tty* för att se vilken fysisk enhet terminalen är kopplad till, exempelvis **/dev/tty02**. Associera denna enhet med den logiska skrivaren abc med kommandot:

   ```
   /usr/lib/lpadmin -pabc -v/dev/tty02
   ```

   Obervera att kommandot *lpadmin* enbart kan användas av *lp*-administratören (användare **lp**) eller super-user. Om vanliga användare ska få skriva ut på detta sätt bör *lp*-administratören skapa ett program, ägt av **lp** eller **root**, som utför detta med tillståndsbiten 'set-user-id' satt.

4. Efter att skrivaren kopplats till terminalporten, eller till terminalens skrivarport kan utskrifter tillåtas med *enable*. Innan dess bör kontrolleras att utskrifter verkligen kan göras till den fysiska skrivaren. Detta innebör t.ex. att papperets överkant är rätt justerat (top-of-form) och att skrivaren är påslagen och är on-line. Utskrifter till skrivaren möjliggörs med:

   ```
   enable abc
   ```

   Nu kan utskrifter som sänts till den logiska skrivaren skrivas ut.

5. När alla utskrifter är klara till abc eller när terminalporten åter ska användas som login-terminal, hindras utskrifterna med:

   ```
   disable abc
   printer "abc" now disabled
   ```

   Om skrivaren abc är tillgänglig (enabled) när D-NIX-systemet startas eller då *lpsched* startas, kommer den automatiskt att kopplas bort (disable).

9.2.12 Sammanfattning

De administrativa rutinerna som *lp*-administratören behöver göra har beskrivits i detalj. Dessa funktioner är skapa och ändra skrivarkonfigurationer för *lp*, skapa/ändra/använda gränssnittsprogram för skrivare, tillåta/hindra köandet av utskrifter, tillåta/hindra utskrifter till skrivarna, flytta eller ta bort köade utskriftsbegäran, stoppa/starta köhanteraren *lpsched*. *lp*-systemet ger många fördelar gentemot vanliga program för skrivarhantering, bl a:

- Skrivare kan grupperas i klasser.
- *lp*-systemet kan konfigureras om för att anpassas till den omgivning den ska arbeta i.
- Administratören kan förse systemet med olika gränssnittsprogram för att formatera utskrifterna på valfritt sätt.
- Funktionerna i *lp*-systemet utförs med enkla kommandot och inte genom att editera filer manuellt.

9.3. Manuellt styrd utskriftshantering

Vid utskrift på skrivare kan man drabbas av olika problem som kan göra att utskrifterna måste stoppas och startas om, t.ex. om papperet fastnar eller om inställningarna var felaktiga. Detta kan vara både tidsödande och kostsamt om stora filer måste skrivas ut på nytt och förtryckta blanketter används.

Till skrivarhanteringen finns därför ett interaktivt hjälpkommando *lpsubmit*, med vilket operatören kan styra utskriften sida för sida, om ett särskilt filterprogram (*/usr/spool/lp/bin/lppg*) används i gränssnittsprogrammet.

Då *lppg* används, kommer utskriften till skrivaren inte att starta förrän användaren via *lpsubmit* givit ett startkommando. Användaren kan välja från vilken sida utskrift ska ske och stoppa/starta utskriften.

En modell för ett gränssnittsprogram, som använder *lppg* finns i filen **/usr/spool/lp/model/pager**. Nedan följer en beskrivning av hur man med ovanstående tre ingredienser konfiguerar en manuellt hanterad skrivare.

Se även *lpsubmit* i **Referenshandboken**.

9.3.1 Konfigurering

Att skapa en destination för manuellt styrd utskrift går till på samma sätt som när man skapar en vanlig destination men här följer ändå ett exempel.

Vi skall skapa en logisk skrivare med namnet prman, som skall knytas till enheten **/dev/tty07** och skapa ett gränssnittsprogram som är en ändrad version av modellen **pager**. Gränssnittsprogrammet döper vi till iprman och ändringarna som skall göras är dels att överföringshastigheten via enheten skall anpassas till skrivaren, dels att den sidindelning som normalt görs av *lppg* skall tas bort, eftersom sidindelning sker i våra användarprogram.

1. Vi börjar med att se till att enheten **/dev/tty07** inte kan användas för inloggning (med *getty*) genom att i filen **/etc/inittab** ändra raden med tty07 så att det står *off* i tredje fältet och inte *respawn*. Eventuell ändring görs med en editor, t.ex. *dmacs*. När **/etc/inittab** har lagrats tillbaka aktiveras ändringen med kommandot:

```
# telinit q
```

2. Ändra ägare och privilegier på enheten till följande.

```
# chown lp /dev/tty07
# chgrp bin /dev/tty07
# chmod 660 /dev/tty07
```

3. Logga in som lp-administratör eller anta dess skepnad med kommandot:

```
# su - lp
```

4. Vi byter bibliotek till modellbiblioteket

```
$ cd model
```

och kopierar modellen pager till en ny fil med namnet iprman:

```
$ cp pager iprman
```

Ändringarna editeras t.ex. med *dmacs*.

```
$ dmacs iprman
```

I listan nedan har följande ändringar gjorts:

Vi har lagt till raden med *stty*-kommandot.

Vi har tagit bort tillvalet **-l 72** till *lppg*.

Filen sparas med *dmacs*-kommandot **ESC Z**.

Nedan följer en lista över det slutgiltiga programmet iprman. I listan har de rader som ändrats markerats med fetstil.

```
:
# lp interface using lppg.
#

stty 19200 <&1              # Ny Baudrate (Tillägg)
copies=$4
shift 5
files="$*"
i=1

# trap cancel command
# -------------------

trap 'echo "\f\r\c"; exit 144' 15
while [ $i -le $copies ]
do
     for file in $files
     do
          /usr/spool/lp/bin/lppg "$file" 2>&1
     done # -l 72 har tagits bort
     i=`expr $i + 1`
done
exit 0
```

5. Nu skall vi skapa den logiska skrivaren, det görs med följande kommando:

```
$ /usr/lib/lpadmin -pprman -v/dev/tty07 -iiprman -h
```

Den nu skapade logiska skrivaren måste göras tillgänglig genom att man tillåter utskrifter till den och tillåter den att skriva ut till skrivaren.

```
$ accept prman
destination "prman" now accepting requests
$ enable prman
printer "prman" now enabled
```

9.3.2 Att använda lpsubmit

Med kommandot *lpsubmit* styr operatören utskriften interaktivt genom att operatörens kommandon sänds av lpsubmit via pipe-filer i **/tmp** till *lppg*. Detta förutsätter att *lppg* är aktivt, varför enbart pågående utskrifter kan styras. Med *lpsubmit* kan man utföra följande:

- Starta utskrift
- Stoppa utskrift

- Gå en sida framåt
- Gå en sida bakåt
- Gå till början av filen
- Avsluta men ta ej bort utskriftsbegäran
- Avsluta och ta bort utskriftsbegäran

Texterna som visas på terminalen av *lpsubmit* är normalt på engelska, men svenska texter kan väljas genom att innan start sätta shellvariabeln **LANGUAGE** med:

```
LANGUAGE=swedish ; export LANGUAGE.
```

För att kunna starta *lpsubmit* krävs att användaren har en utskrift aktiverad i utskriftskön. Som exempel kan en lista sändas till utskrift via det nya gränssnittsprogrammet med:

```
$ l /bin | pr -l72 -f | lp -dprman
request id is prman-45 (standard input)
```

Listan kommer inte att skrivas ut på skrivaren förrän användarens med ett *lpsubmit*-kommando startat utskriften. För att kontrollera denna utskrift manuellt startas *lpsubmit* med:

```
$ lpsubmit prman-45
```

När *lpsubmit* startats har bildskärmen följande utseende. När man kommit in i denna bild väljer man kommando genom att flytta sig med piltangenterna och trycka på **RETURN**.

```
Utskrift av: prman-45
Sida: 1 av: 9 Fil: 1 av: 1 Kopia: 1 av: 1
=========================================================
     * Starta utskrift     * Gå en sida bakåt
     * Stoppa utskrift     * Gå till början av filen
     * Skriv en sida       * Avsluta men ta ej bort filen
     * Gå en sida framåt   * Avsluta och ta bort filen
=========================================================
```

9.4. Utskrifter via nätverk

Skrivarhanteringssystemet *lp* kan konfigureras att via ett nätverk sända filer för utskrift till ett annat system. Nätverket kan t ex vara av typen Ethernet. Utskrift av en fil till en fjärrskrivare över ett nätverk går i stora drag till på följande sätt:

Utskriften görs som vanligt med *lp*-kommandot på den lokala datorn, men till en logisk skrivare som styrs av ett särskilt gränssnittsprogram. Detta hanterar automatiskt överföringen av data och aktiveringen av *lp*-systemet på destinationsdatorn. Sändningen av data och kommandon över nätverket utförs under användaridentiteten **lp**.

Med denna princip kommer utskriftsfiler att köas i bägge datorerna, varvid standardfunktioner för att omdirigera köer, ställa av skrivare med mera fungerar.

Utskriftsfiler från andra datorer får en "titelsida" för varje skrivarhanterarsystem som den passerar. Framsidan produceras av gränssnittsprogrammet för aktuell skrivare och kan lätt modifieras.

9.4.1 Konfigurering

Att få en utskrift att komma till rätt skrivare på ett annat system kräver en del förberedande arbete både på det lokala systemet och det andra systemet. Dels ska användare **lp** ha tillstånd att logga in via nätverket på destinationssystemet (filen **.netkey**), dels ska ett gränssnittsprogram väljas och aktiveras i den lokala skrivaren. Det finns ett modellprogram för nätverk (**rlp**) som anpassas till det använda nätverket och med eventuellt skrivarnamn på destinationsdatorn. Nedan beskrivs detta med ett exempel. Logga först in som användare **lp**.

Låt oss säga att vårt system har namnet sysa och systemet där skrivaren sitter har namnet sysb. Skrivaren på sysb har det logiska namnet laser.

På båda systemen ska en fil skapas som möjliggör automatlogin via nätverket för användare **lp**. Den heter **/usr/spool/lp/.netkey** och skall innehålla rader med tre fält som är separerade med minst ett blank- eller tabtecken. Första fältet skall innehålla systemnamn eller '*' för godtyckligt systemnamn, fält två skall innehålla namnet på användaren (i detta fall **lp**) och det tredje fältet skall innehålla ett lösenord. Det viktiga är att samma rad finns i bägge systemens **.netkey**-filer. För mer information kring filen **.netkey** konsultera din nätverkshandbok. Kontrollera om filen **.netkey** finns med hjälp av kommandot *ls -a*. Filerna skapas eller ändras på båda systemen så att de innehåller gemensamma rader.

Exempel på .netkey:

```
sysb lp    Spooling
*    lp    remote-printer
```

i exemplet ser vi att systemet sysb har ett eget lösenord, medan övriga system har ett gemensamt.

Vi ska använda det modellprogram som levereras med systemet (**rlp**), men kan behöva göra en ändring, beroende på vilket nätverk vi använder. Gå in i biblioteket **model** och gör eventuellt följande ändring. Ett exempel på **rlp** modellprogram listas i följande kapitel.

I programmet ska en av följande tre rader väljas genom att #-tecknet tas bort bara för den rad som ska gälla.

```
SERVER="remsh $HOST lp -t'uname -n'"
# SERVER="erx $HOST!lp -t'uname -n'"
# SERVER="rx $HOST!lp -t'uname -n'"
```

Observera att det är bara en av raderna som får sakna ett "#" tecken. Första och andra raden är via Ethernet, den tredje är via D-NET. Vi har här valt den första raden. På den aktuella raden skall nu även destinationen läggas till (logiska skrivaren i destinationsdatorn), i vårt fall kommer raden att bli enligt nedan. Filen sparas sedan.

```
SERVER="remsh $HOST!lp -dlaser -t'uname -n'"
```

Nu skall skrivarhanteringssystemet på sysa stängas av med kommandot:

```
$ lpshut
```

En ny logisk skrivare skapas på sysa, som använder modellprogrammet **rlp** som gränssnittsprogram, vilket sänder utskrifterna till sysb:s skrivare. Kommandot blir enligt våra förutsättningar:

```
$ /usr/lib/lpadmin -psysb -mrlp -v/dev/null -h
```

Kommandot *lpadmin* kräver att en enhet anges, varför man här anger enheten **/dev/null**, när utskrift skall ske via nätverk.

Den logiska skrivaren sysb är nu skapad och måste göras tillgänglig genom att man tillåter utskrifter till den samt tillåter den att sända utskrifterna vidare.

```
$ accept sysb
destination "sysb" now accepting requests
$ enable sysb
printer "sysb" now enabled
```

Glöm inte att återstarta skrivarhanteringssystemet, använd kommandot:

```
$ lpsched
```

Att från systemet sysa göra en utskrift till sysb och dess skrivare laser kan se ut som följer,

```
$ cat /usr/spool/lp/model/rlp | lp -dsysb -o'12 us'
```

I detta exempel sänds två tillval med via **-o**. Dessa tillval kommer att skickas vidare till sysb och skrivaren laser och tas om hand av det gränssnittsprogrammet.

9.4.2 Exempel på modellen /usr/spool/lp/model/rlp

```
:
#
# Remote spooling interface
# Change server macro according to your requirements
#
MYNAME=`basename $0`
ID=$1
USER=$2
HOST=$MYNAME
SERVER="remsh $HOST lp -t`uname -n`"
# SERVER="erx $HOST!lp -t`uname -n`"
# SERVER="rx $HOST!lp -t`uname -n`"
#
HOME=/usr/spool/lp
export HOME
x="======================================================"
(
echo "$x\n$x\n"
banner "$USER"
echo "\n"
user=`grep "^$USER:" /etc/passwd | line | cut -d: -f5`
if [ -n "$user" ]
then
     user="`echo $user | cut -d, -f1`"
     echo "User: $user\n"
else
     echo "\n"
fi
echo "Request id: $1 Printer: $MYNAME\n"
date
echo "\n"
if [ -n "$3" ]
then
     banner $3
fi
copies=$4
echo "\014\c"

LPOPT=""
for OPT in $5
do
     LPOPT="$LPOPT-o$OPT "
done
shift 5
files="$*"
i=1
while [ $i -le $copies ]
do
     for file in $files
     do
          cat <"$file" 2>&1
          echo "\014\c"
     done
     i=`expr $i + 1`
done
) | (
$SERVER $LPOPT
RVAL=$?
if [ $RVAL -ne 0 ]
then
     disable -r"Network problems(?)" $MYNAME
     exit $RVAL
fi
) | (read M; echo $ID $HOST $M >>log_$MYNAME)
```

9.5. Lpr skrivarhantering

Ett alternativt skrivar-system, *lpr*-systemet, finns med i grundpaketet. Kommandot *lpr* kan användas för att sända utskrifter till "logiska skrivare" i systemet (jämför *lp*-kommandot). Det är då bakgrundsprocessen */usr/lib/lpd* som sköter utskriften från den kö som bildas med *lpr*. Notera att endast en av dessa båda system (*lp* eller *lpr*) skall vara aktiva i systemet.

- I *lpr*-systemet väljer användaren med tillvalet **-t** till *lpr*-kommandot ett eller flera villkor som tolkas av systemet för att utskriften skall sändas via en logisk skrivare till en lämplig fysisk skrivare. En fysisk skrivare kan ha flera logiska namn med olika styrprogram för olika ändamål.
- Initieringsprogram kan automatiskt startas vid varje utskrift, olika till varje logisk skrivare.
- Inledande datafiler (header) samt avslutande datafiler (tailer) kan definieras för varje logisk skrivare.
- Filter-program kan automatiskt användas för att erhålla grafik eller standardiserad kodning av styrtecken för varje logisk skrivare.

Detta kapitel beskriver hur den som är *lpr*-administratör kan konfigurera de olika filerna i systemet för att modifiera *lpr*-systemet för bästa prestanda.

9.5.1 Översikt över funktioner i lpr-systemet

lpr-systemet är uppbyggt för att kunna anpassas till många olika konfigurationer av datorsystem.

lpr skiljer på begreppen logisk skrivare, villkor och fysisk skrivare (device). En fysisk skrivare är en fysisk enhet (device) och representeras av ett enhetsnamn i bilbioteket **/dev** i filsystemet. En logisk skrivare har ett logiskt namn och en fysisk skrivare kan ha flera olika logiska namn.

Olika villkorssträngar kan associeras med varje logiska skrivare, t.ex. kan de skrivare som klarar grafik markeras med *graphic* och de som klarar skönskrift markeras med *pretty*. Dessa villkor anges som alternativ till det logiska skrivarnamnet i tillvalet **-t** till *lpr*-kommandot, varvid systemet väljer den skrivare som för tillfället är ledig, men som klarar av det som användaren vill kunna göra.

Utan angivande av logisk skrivare riktas utskriften till **/dev/lp**. Det som ska skrivas kan även riktas till *lpr*-kommandot med en pipe, varefter det läggs i skrivarkön.

lpr-systemet konfigureras genom filer i biblioteket **/usr/lib/lpdpar**.

Exempel:

Skriv filen 'myfile' till skrivaren pr1:

```
lpr -tpr1 myfile
```

Filen lpdtable: Kopplingen mellan "logisk skrivare" och fysisk skrivare görs i filen lpdtable. Denna fil innehåller en rad för varje fysisk skrivare, med innehåll som kan vara enligt följande exempel.

```
lp myhead pr1
lp2 head2 pr2 pretty
```

där parametrarna är (med exempel ur andra raden):

lp2	**/dev/lp2** är den motsvarande fysiska skrivarporten.
head2	är det logiska skrivarnamnet. Det anger bl a att filen **/usr/lib/lpdpar/lpdhandler/head2** skrivs ut i början av varje dokument, om inte tillvalet **-h0** anges till *lpr* för att hindra detta. En fysisk skrivare kan återkomma på flera rader som därmed refereras med olika logiska skrivarnamn.
pr2 pretty	pr2 och pretty är villkorssträngar. Flera villkor kan ges för varje skrivare.

Den andra parametern anger den logiska skrivaren. För varje logisk skrivare kan filer i följande bibliotek skapas, med samma namn som den logiska skrivaren. Om motsvarande fil saknas används inte den funktionen.

Biblioteket devinit:	Initieringsprogram
Biblioteket lpdheader:	Inledande datafiler (header)
Biblioteket lpdtailer:	Avslutande datafiler (tailer)
Biblioteket lpdhandler:	Filterprogram för grafik m m

Kommandon

Kommandot *lpr* används för att sända utskrifter till skrivarkön och användaren anger då villkor för den skrivare som skall användas. Med *lpr* kan en eller flera filer sändas till utskrift. Om inga filer anges tas data från standard input. *lpr* skapar och sänder utskriftsbegäran till systemet. Varje begäran får ett könummer och tilldelas en utskriftsprioritet. Systemet försöker effektivisera genom att skriva ut korta filer före långa.

/usr/lib/lpd är den process som sköter utskriften från kön till de fysiska skrivarna när *lpr*-kommandot används.

Kommandot *queue* listar vilka utskrifter som ligger i kö för att skrivas ut med *lpr*-spoolersystemet. Med *queue* kan icke påbörjade utskrifter tas bort ur kön. Utskriftsordningen styrs av en utskriftsprioritet som kan ändras med kommandot *queue*.

Kommandot */etc/setspeed* används då baudrate eller andra parametrar behöver ändras från systemets standardvärden. Kommandot lägges då i filen **/etc/rc** enligt instruktioner i början av detta kapitel.

9.5.2 Ändra lpr-konfigurationen

Anmärkning! På systemet finns två olika spoolersystem, *lp*-systemet och *lpr*-systemet. Endast ett kan användas åt gången.

För att återstarta utskrift av eventuella filer i utskriftskön efter systemstart, måste filen **/etc/rc** ha kommandon enligt nedan. Sista raden startar spool-programmet. Dessutom måste biblioteket **/usr/spool/lpd** vara tillgängligt för att *lpr*-systemet skall kunna användas. Detta bibliotek måste vara skapat med läs/skriv- och exekveringstillstånd för samtliga användare.

```
rm -f /usr/spool/lpd/ERRLOG
rm -f /usr/spool/lpd/lock
rm -f /usr/spool/lpd/lpdctl
rm -f /usr/spool/lock/*
nice -20 /usr/lib/lpd &
```

Konfigurering skrivarval (Fil: lpdtable)

Filen **/usr/lib/lpdpar/lpdtable** innehåller en tabell över de skrivare som finns tillgängliga samt de inställningar som gäller för dem. Om filen inte finns tillgänglig används **/dev/lp** som skrivare.

Typsträngen (eller strängarna) angivna i **-t**-tillvalet för *lpr*kommandot jämförs med strängarna i denna tabellfil och vanligen väljs en skrivare för vilken alla dessa strängar matchar. Därtill kan den första raden i tabellfilen specificera strängval för vilket endast en sträng behövs för att ange en grupp av skrivare. Om flera skrivare uppfyller kraven vid valet, används den skrivare som är ledig för tillfället. En default typsträng kan defineras i shellvariabeln **PRTYPE**. Om så inte är fallet, väljs den skrivare som är markerad med valsträngen default.

Om endast **/dev/lp** används i systemet och ingen initiering, inga titelsidor, tailers eller filtreringsprogram används, är inte biblioteket **/usr/lib/lpdpar** nödvändigt.

Ett exempel på en tabellfil ges nedan.

Om en rad i tabellen börjar med en sträng, tolkas denna sträng (device) som den fysiska utskriftsporten och den andra strängen (tabellnamn) används för att välja initieringsprogram, titel- och tailerfiler och (**-f** tillval) filtreringsprogram. Följande strängar används som valsträngar. Tabellnamnsträngen används även som valsträng för skrivaren. Valsträngarna jämförs med typsträngen angiven i **-t**-tillvalet.

Om en rad börjar med ett mellanrum eller en TAB, tyds detta som en fortsättning på föregående rad, förutsatt att det inte är första raden på filen.

Om första raden i tabellfilen börjar med ett mellanrum eller en TAB, kommer alla strängar på denna raden att specificera valsträngar för vilket endast en strängs överensstämmande är nödvändig. Se valexempel 3 nedan.

Exempel på en tabellfil med följande fält:

Device, Tabellnamn, Valsträngar.

```
        rum1 rum2 rum3
lp      facit4542  english medium default rum1
lp1     facit4544  english graphic medium default rum3
lp2     diablo     swedish pretty rum1
lp3     graf2      english graphic rum2
```

Denna tabellen definerar **/dev/lp** till en Facit 4542 att vara default-skrivare med medium hastighet och ett engelskt tecken-set På samma sätt som **/dev/lp1** är en Facit 4544 med ett engelskt teckenset, grafisk presentation, medium hastighet och liksom en defaultskrivare. **/dev/lp2** är en diablo skrivare med ett svensk tecken-set. **/dev/lp3** är en andra skrivare för grafik. I detta tabellexempel, kan gruppvalsträngarna på första raden beskriva den fysiska platsen för skrivarna.

Den andra strängen, kallad tabellnamn ovan, anger filnamn för titel- och tailer-filer, initieringsprogram för utskriftsenheter och filtreringsprogram.

Om endast **/dev/lp** används i systemet och ingen initiering, inga titlar, tailers eller filtreringsprogram används, behövs inte biblioteket **/usr/lib/lpdpar**.

Valexempel 1:

Användaren specificerar tillvalet **-tenglish** för kommandot *lpr* och utskriften dirigeras till första fria enhet som har english som angivet tecken-set. I detta fall gäller det både för /dev/lp, /dev/lp1 och /dev/lp3.

Valexempel 2:

Skrivartypen kan också specifieras med en kombination av sättningar, såsom **-tenglish,graphic**, vilket i detta fallet skulle betyda /dev/lp1 eller /dev/lp3. Observera att komma (,) krävs mellan varje sträng i tillvalet.

Valexempel 3:

Om valet **-tgrapcic,rum1,rum2** är specificerat väljs vilken grafisk skrivare som helst antingen i rum1 *eller* rum2. I detta fallet kan endast /dev/lp3 användas. Detta är ett exempel där första raden i tabellfilen måste användas eftersom både rum1 och rum2 annars skulle krävas.

Valexempel 4:

Typen kan också specificeras av enhetsnamnet, d.v.s. **-tlp**, **-tlp1**, **-tlp2** eller **-tlp3**, i vilket fall som helst kan dock inga andra valsträngar kan specificeras.

Observera att strängarna inte översätts utan endast jämförs med strängar givna som typer.

Initieringsprogram för skrivare (Bibliotek: devinit)

Tabellnamn-strängen i tabellfilen är namnet på en fil i biblioteket **/usr/lib/lpdpar/devinit**. Denna fil exekveras som kommando med en

standard input och output dirigerad till skrivaren innan någon annan text skickas ut. Om denna fil inte finns, sker ingen initiering.

Titel-/tailerfiler till skrivare (Bibliotek: lpdheader och lpdtailer)

Tabellnamn-strängen i tabellfilen är namnet på en fil i biblioteket **/usr/lib/lpdpar/lpdheader**. Data i denna fil skrivs alltid ut före annan text. Titelfilen kan innehålla setup-strängar för skrivaren. Dessa behövs för att kontrollera att skrivaren alltid skriver likadant. Om ingen fil finns med detta namn, skrivs ingen titelsida ut.

Tabellnamn-strängen i tabellfilen är namnet på en fil i biblioteket **/usr/lib/lpdpar/lpdtailer**. Data i denna fil skrivs alltid ut efter all annan text. Om ingen fil med detta namn finns, skrivs ingen särskild slutsida ut.

Filtreringsprogram för skrivare (Bibliotek: lpdhandler)

Tabellnamn-strängen i tabellfilen är namnet på en fil i biblioteket **/usr/lib/lpdpar/lpdhandler**. Denna fil innehåller filtreringsprogram, som används för avkodning av speciella sekvenser vid data output av grafisk presentation och andra speciella utskrifter. Detta är möjligt bara om **-f** -tillvalet är given och om filen för filtreringsprogrammet existerar. Filtreringsprogrammen läser från standard input (data som skall skrivas ut) och skriver till standard output (blir automatiskt dirigerad till skrivaren) Speciella sekvenser i datainnehållet startar och stannar filtreringsprogrammet.

Den speciella sekvensen startar med två bytes med de oktala ASCII-värdena 0377. Sekvensen ser ut som nedan och åtföljs av alla data som skall till output genom filtret.

```
----------------------------------------
! 0377 ! 0377 ! typ ! format ! 000 ! 010 !
----------------------------------------
```

Fältet **typ** är en byte med 0 (ASCII 060 oktalt) för textmod och 1 (ASCII 061 oktalt) för binär mod. Fältet **format** är en sträng som sänds som kommandoradsargument till filterprogrammet. 'format' kan innehålla flera strängar separerade med mellanslag.

Om 'typ' är '0', sänds all text up till nästa speciella sekvens till filtret. För att avsluta filtreringsprogrammet kan en sekvens sändas med 'typ' som är '0', och 'format' lika med strängen 'text'.

Om 'typ' är '1' kommer en eller flera binära poster att följa den speciella sekvensen. Filtreringssprogrammet avslutas när en binär post som har sitt 'id'-fält lika med 1 är utskrivet.

Den binära posten har följande format:

```
-------------------------
! id ! size ! binary data !
-------------------------
```

Fältet 'id' har normalt värdet 0 (ASCII 000) om flera poster följer. Den sista binära posten skall ha 'id'-värdet 1 (ASCII 001) för att filtreringsprogrammet skall avslutas när binära data i posten har skrivits ut. Fältet 'size' är ett 16 bitar binärt värde som har den högsta byten först. Fältet

'binary data' är en sträng med dimensionen 'size' bytes, och denna sträng sänds genom filtreringsprogrammet till skrivaren.

Specificera standardskrivare i systemet

Standard-skrivare i systemet kan ändras genom att shellvariabeln **PRTYPE** definieras i användarens fil **.profile** eller i den allmänna systemfilen **/etc/profile**. Om denna variabel inte är definieras anses **/dev/lp** vara standardskrivare. **PRTYPE** kan innehålla en hel standardsträng med flera villkor om den innesluts mellan apostrofer. Denna används då som standard **-t** tillval för *lpr*-kommandot.

Villkorssträngen 'default' är reserverad och anger vilken av flera skrivare med samma villkor som är standardskrivare om varken **PRTYPE** eller tillvalet **-t** angivits.

9.5.3 Gör en utskriftsbegäran med lpr

Då *lpr*-filerna har skapats i systemet, kan användarna sända utskrifter med kommandot *lpr*. Varje utskriftsbegäran får ett könummer, som kan användas tillsammans med kommandot *queue* för ett se om utskriften är klar eller för att stoppa utskriften.

lpr-systemet bestämmer till vilken skrivare som utskriften skall ske med följande regler i ordning.

- Om användaren direkt anger fysiska enheten **-t***dest* på *lpr*-kommandoraden används denna (dest).
- Om användaren anger en logisk skrivare **-t***skrivare* används denna.
- Om användaren anger ett eller flera villkor, varav egentligen även det logiska skrivarnamnet behandlas som ett villkor, söker systemet efter den första hittade lediga skrivaren som uppfyller alla villkor.
- Om shellvariabeln **PRTYPE** är definierad sänds utskriften annars till den skrivare som uppfyller de villkor som angivits i **PRTYPE** (motsvarande **-t$PRTYPE**).
- Annars sänds utskriften till **/dev/lp**, som är standardskrivare om ingenting alls angivits. Detta innebär att *lpr*-kommandot fungerar även utan filerna i **/usr/lib/lpdpar**-biblioteket, men då enbart till **/dev/lp** och enbart om tillvalet **-t** inte används.
- Om villkor anges i tillvalet **-t** men dessa inte kan uppfyllas och filen lpdtable finns, blir det ingen utskrift.

Exempel 1:

```
lpr -tpretty mytext
```

Sänder filen mytext till spoolkön för utskrift till en skrivare med villkorssträngen pretty. Enligt exemplet ovan av filen **/usr/lib/lpdpar/lpdtable**, skrivs den ut på den fysiska skrivaren /dev/lp2.

Exempel 2:

```
lpr -tenglish,graphic,rum1,rum2 myfile
```

Filen sänds till spoolkön för en skrivare som är vald av både english och graphic, och antingen i rum1 eller rum2, förutsatt att exemplet på tabellfilen nedan är använt. För tabellfilen nedan, används den skrivare som är ansluten till /dev/lp3.

Example 3:

```
lpr -f -tgraphic myfile
```

Filen sänds till spoolkön för en skrivare som är vald genom strängen graphic. Om det finns ett filterprogram i biblioteket **/usr/lib/lpdpar/lpdhandler** för den aktuella skrivaren, kommer det att användas. I varje fall ersätts TAB:ar i textsträngarna till mellanrumssträngar före output (jämför *lpr*-kommandot i **Referenshandboken**). Med tabellfilen ovan används den skrivare som är ansluten antingen till **/dev/lp1** eller **/dev/lp3**.

9.5.4 Lista lpr-status - queue

Kommandot *queue* listar de filer som är köade i skrivarkön med kommandot *lpr*. Listan ser ut som i följande exempel:

```
DEVICE     JOBID PRI  UID   COMMENT
/dev/lp    227   40   root  Spooled ...
           228   40   bin   Spooled ...
```

Listan är sorterad i utskriftsordning (PRI) och listan kan innehålla flera fysiska enheter (device) i den ordning de listas i filen lpdtable. Optionen **-t** kan användas med *queue* för att lista utskrifter till en enhet eller en skrivare motsvarande *lpr*-tillvalet **-t**.

Utskriftsordningen kan ändras genom att ändra utskriftsprioriteten, med tillvalet **-p** till *queue*-kommandot. Värdet som anges är hur mycket prioriteten skall ändras. Super-user kan även ange negativa värden, vilket innebär att prioriteten höjs istället för sänks i förhållande till övriga köade utskrifter.

Exempel

Utskriftsprioriteten för utskrift 228 ska sänkas med 10.

```
queue -p 10 228
```

9.5.5 Ta bort utskriftsbegäran - queue -r

queue kan användas för att ta bort utskrifter som väntar i kön genom tillvalet **-r**. Det är däremot inte möjligt att stoppa en redan pågående utskrift och en vanlig användare, som inte är super-user, kan inte ta bort en annan användares utskrifter ur kön,

Exempel

En utskrift ska tas bort ur kön. Genom ett tidigare queue-kommando har vi sett att utskriften har könummer 228.

```
queue -r 228
```

9.5.6 Stoppa och starta köhanteraren - /usr/lib/lpd

/usr/lib/lpd är det program som hanterar de utskriftsbegäran som sänds av *lpr* till kön. */usr/lib/lpd* hanterar den fysiska utskriften till skrivare eller andra enheter och startas normalt via systemfilen /etc/rc då datorsystemet startas och är alltid aktivt tills hela systemet stängs av eller går ner i enanvändarnivå. När */usr/lib/lpd* skall startas vägrar den om filen **/usr/spool/lpd/lock** finns, vilket hindrar att flera köhanterare startas. Annars skapas denna fil då */usr/lib/lpd* startas.

När köhanteraren startas bör den alltid startas som en bakgrundsprocess genom att tecknet **&** ges efter kommandot. Exempel ur filen **/etc/rc**:

```
/usr/lib/lpd &
```

10

11

11. Säkerhetskopiering

11. Säkerhetskopiering

11.1. Motivering till och metoder för backup

Säkerhetskopiering är mycket viktig i en datorinstallation.

Det är betydligt vanligare att förlora data än man tror. Ofta beror det på olyckshändelser och ovarsamhet från användare. Men även systemadministratören (**root** eller **su**) kan lätt förlora data med katastrofala följder. Man kan även förlora data när systemet kraschar.

Ju fler användare man har i systemet desto större potentiell risk har man för krascher. Och kom ihåg det finns inget krasch-säkert system.

Det finns många sätt att utföra säkerhetskopiering på ett datorsystem. Men oberoende av metoden, så skall skaärhetskopiering tas för att försäkra sig om att man kan återfå all viktig information ifall systemet kraschar eller filer förloras/förstörs.

Av största vikt är att se till man har släpande säkerhetskopieringar, d v s sparar band så att man har möjlighet att återskapa filer som är ett par månader gamla. I många fall upptäcks det att filer är förstörda eller raderade av misstag först efter relativt lång tid.

Om säkerhetskopiering tas dagligen skall t ex fredagens säkerhetskopiering sparas 4 veckor. Varje månad läggs en säkerhetskopiering undan som sparas t ex 6 månader. Detta kräver då:

måndag - torsdag	4 band	
fredagar	4 band	
6 månader	6 band	
Totalt	**14 band**	**Kostnad ca 6.500 kr**

Denna kostnad skall vägas mot eventuell förlust av data som oftast orsakar dryga utgifter i form av stillestånd och konsultarvoden.

Följande kan inträffa:

En kund förlorar en Mimerdatabas på 20 Mbyte., motsvarande cirka 3 manmånaders arbete.

Säkerhetskopiering utförs som total säkerhetskopieringer med ett band för måndag, ett för tisdag o s v fram till fredag. En dag uppstår ett diskfel som medför att en sektor i Mimerdatabanken blir oläsbar. Databanken blir i och med detta fel oanvändbar.

Säkerhetskopiering som görs med *tar* avbryts när kommandot kommer till den den dåliga sektorn på disken. Av en slump har den systemansvarige semester denna vecka, och någon kontroll av att säkerhetskopieringen gått utan problem, görs inte. Efter fem dagar har samtliga användbara säkerhetskopieringar av databanken skrivits över med den förstörda!

Kom ihåg: Den dag ni råkar ut för förlorat data kommer ni **att uppskatta ordentliga säkerhetskopiering.**

Säkerhetskopiering

En lämplig teknik för säkerhetskopiering av systemet beror på flera faktorer. En snabb och enkel metod kan resultera i att man måste ägna timmar åt att återskapa systemet. Den tid man ägnar åt säkerhetskopiering vägs emot den tid det tar att reparera systemet.

De två huvudalternativen för säkerhetskopiering är:

Total säkerhetskopiering Säkerhetskopiering av hela filsystemet även operativsystemet.

Del säkerhetskopiering: Alla metoder där man sparar mindre än hela filsystemet.

Variationer: Hela systemet förutom systemfilerna, men inkluderande filer som ändras ofta som:

/etc/passwd
/etc/group
/etc/rc

Inkrementell säkerhetskopiering. Man sparar filer baserat på det datum då filen ändrades senast.

rutinerna för säkerhetskopiering måste anpassas till tillgängligt medium för säkerhetskopiering. Om systemet har magnetbandsstation eller streamer-tape så kan en helt annan teknik användas än om man bara har disketter.

På ett system med streamer eller magnetband kan följande förslag till rutiner för säkerhetskopiering användas.

1. Efter varje större förändring i systemet görs total säkerhetskopiering. T ex om en ny version av någon programvara installerats.
2. Delkopiering gjorda varje natt och återanvända vecka för vecka.
3. Total säkerhetskopiering varje vecka, t ex fredag. Sparas 4 veckor.
4. Total säkerhetskopiering varje månad. Sparas minst 6 månader.

11.2. Säkerhetskopiering - tar

Kommandot *tar* sparar filer till och från en yttre enhet i ett speciellt standardiserat format som är oberoende av det aktuella systemets fysiska filstruktur. Lämpliga medier är streamerkassetter och disketter.

Filerna kopieras direkt till det fysiska mediet med ett block som innehåller information om filernas ursprungliga plats i biblioteksstrukturen, filåtkomsttillstånd o s v i det filsystem från vilken kopieringen sker. Tillräcklig information för att ett senare *tar* kommando skall kunna rekonstruera de kopierade filerna eller biblioteken. Enstaka eller alla filer och bibliotek kan läsas tillbaka till systemet vid behov.

Observera! Det rekommenderas att filer normalt kopieras utan inledande / i filnamnet. Man ställer sig istället med kommandot cd i det bibliotek där filerna eller biblioteken finns innan *tar* kommandot ges. På detta sätt finns en möjlighet att läsa tillbaka biblioteken och filerna till ett temporärt bibliotek utan att förstöra de originalfiler som finns på skivminnet. Alternativt kan inledande / vid inläsning tas bort automatiskt, om tillvalet **-A** anges till *tar* kommandot.

Vid inläsning av sparade filer till ett annat datorsystem bör användar- och gruppnamn ändras med tillvalet **-o** till *tar* kommandot, eftersom *tar* endast spar denna information i numerisk form.

Om säkerhetskopiering sker med *tar* kommandot i bakgrunden bör utgångsstatus från *tar* testas för att säkerställa att kopieringen gick bra. Detta kan t ex göras med kommandot *test*. Exemplen nedan visar kopiering till streamer. Dessa kommandon lägges lämpligen i en fil som utförs som en shellprocedur i bakgrunden (med **&**), varvid texten **Fel vid säkerhetskopiering** ges till huvudkonsolen om kopieringen inte gick bra. Eventuella felmeddelanden läggs i den temporära filen **/usr/adm/backuplog** för att eventuellt senare kontrolleras.

Följande shell-procedur (**/usr/adm/fbackup**) kan användas för total säkerhetskopiering

```
# /usr/adm/fbackup
# shell script to do full backup
# logg info is stored in /usr/adm/backuplog
# Diab Data AB
# BLOCK is block factor used
# TSIZE is tape size in kbytes (60000 means 60 Mbyte)
BLOCK=300
TSIZE=60000
date >/usr/adm/backuplog
cd /
tar cvfbkS /dev/st0 $BLOCK $TSIZE . >>/usr/adm/backuplog 2>&1
if test $? -ne 0; then
     echo "Fel vid säkerhetskopieringen" >/dev/console
     echo "Fel vid säkerhetskopieringen" >>/usr/adm/backuplog
fi
date >>/usr/adm/backuplog
```

Denna skript kan startas i bakgrunden eller, om *cron* kommandot finns i systemet, med kommandot *crontab*.

Om total säkerhetskopieringen skall startas fredag kväll vid 23:00, lägg till följande rad till systemadministratörens *crontab*-fil.

```
0 23 * * 5 /usr/adm/fbackup
```

Följande procedurer kan användas för inkrementell säkerhetskopiering:

```
# /usr/adm/ibackup
# incremental backup script
# Diab Data AB
BLOCK=300
TSIZE=60000
case $# in
     1)   days=$1;;
     *)   echo "Usage: ibackup n - where n is number of days back"
esac
case $days in
     Å1-9Å)     ;;
     Å1-9ÅÅ0-9Å)     ;;
     *)   echo "Bad argument, should be numeric"; exit;;
esac
echo "backing up last $days day(s)" >/usr/adm/backuplog
cd /
find . -type f -mtime -$days -print >/tmp/backupfiles
# if some parts of the file system should not be backuped
# change the file /tmp/backupfiles here. Example:
# cat /tmp/backupfiles ö /usr/adm/filter >/tmp/backupfilx
# mv /tmp/backupfilx /tmp/backupfiles
date >>/usr/adm/backuplog
tar cvfbkF /dev/st0 $BLOCK $TSIZE /tmp/backupfiles \
     >>/usr/adm/backuplog 2>&1
if test $? -ne 0; then
     echo "Fel vid säkerhetskopieringen" >/dev/console
     echo "Fel vid säkerhetskopieringen" >>/usr/adm/backuplog
fi
date >>/usr/adm/backuplog
rm -f /tmp/backupfiles
```

Shell procedur som undantar vissa filer från säkerhetskopiering, kan se ut så här:

```
# Shell script to remove unwanted files from backupfile
# list
# Diab Data AB
# /usr/adm/filter

while true
do
     read f
     if test $f
     then
          x=`dirname $f`
          case $x in
          # here you specify directories NOT to be backuped
               ./..systemfiles) ;;
               ./usr/spool) ;;
               ./tmp) ;;
               ./usr/tmp) ;;
               * | .*)
               case $f in
               # here you specify file names NOT to be
               # backuped
                    ./etc/mtab) ;;
                    ./etc/mnttab) ;;
                    ./etc/utmp) ;;
                    * | .*) echo $f;;
               esac
          esac
     else
          exit;
     fi
done
```

11.3. Säkerhetskopiering - bup

Kommandot *bup,* som är ett hjälpprogram, utför periodiskt återkommande säkerhetskopiering av specificerade filer. *bup* använder *tar*-kommandot för att göra själva säkerhetskopieringen. Kommandot *cron* används för automatisk regelbunden säkerhetskopiering. *bup* används också till att återladda filer från kopieringsmedierna.

De medier som används för säkerhetskopieringen (band och disketter) ges volymnamn. En loggfil underhålls vilken innehåller filnamn, säkerhetskopieringens volymnamn och tidpunkten för säkerhetskopieringen för alla filer på kopieringsmediet. Loggfilen heter **/usr/lib/bup/log**. Då *bup* används för att återladda valda filer, kommer kommandot att informera användaren om från vilket band eller diskett som filen skall tas. Volymnamn som är standard kan specificeras i filen **/usr/lib/bup/volumenames** och de kommer att växla automatiskt. Loggfilen eller valda delar av den kan tas fram och visas med kommandot *bup -l.*

bup-kommandot skall utföras i tre steg.

- Först lagras standard volymnamn i filen **/usr/lib/bup/volumenames** med hjälp av en editor, t ex *dmacs*.
- Sedan körs *bup* för att lagra namnen på alla filer (eller bibliotek), som ska säkerhetskopieras, i kommandofilen **/usr/lib/bup/cmd**.
- Som tredje steg görs själva kopieringen med *bup*-kommandot och ett av följande tillval; **-bd**, **-bw**, **-bm** eller **-ba**, valfritt regelbundet aktiverat genom att använda kommandot *crontab*.

För att specificera fil- respektive biblioteksnamnen körs *bup* med ett av följande optioner; **-m**, **-w**, **-d** eller **-a**. Optionen **-m** används för filer som kopieras månatligen, **-w** för de som kopieras veckovis, och **-d** för sådana som kopieras dagligen. **-a** används bara då filer skall slutlagras på ett externt medium och sedan tas bort ur systemet. *bup* kommer att lägga in fil- eller biblioteksnamnet i *bup*:s kommandofil **/usr/lib/bup/cmd** för senare användning då själva säkerhetskopieringen äger rum.

Fil- eller biblioteksnamnet kan specificeras antingen som normala sökvägsnamn, wildcards eller som argument till ett *find*-kommando, *bup -r* optionen, för att välja filer. *bup* kommer alltid att lägga till nuvarande bibliotek till alla filnamn som inte är fullständiga sökvägsnamn. Om namnet på ett bibliotek anges tages hela biblioteket. Om wildcards eller *find*-argument ges kommer de inte att bli utvärderade förrän det är dags för själva säkerhetskopieringen förutsatt att de är omgivna av apostroftecken, (t ex '*.doc'), vilka skall hindra shell att avkoda dessa specialtecken.

Filer kan tas bort från kommandofilen med hjälp av kommandot *bup -k.* Kommandofilen kan listas med det kommandot *bup -L.*

Filen **/usr/lib/bup/volumenames** skall innehålla tre rader med namn på den dagliga, veckovisa och månatliga säkerhetskopieringen. Raderna

är märkta d:, w: eller m: och volymnamnen är skilda åt med ett kommatecken och valfritt antal mellanslag eller TAB-tecken. Volymnamnen kommer automatiskt att växlas under användningen, dvs första namnet på raden kommer att användas vid nästa kopiering och detta namn kommer att flyttas till slutet av raden. Inget mellanslag får finnas i namnet.

Exempel på en fil **/usr/lib/bup/volumenames**:

```
Före backup:    d:   daytape1,   daytape2,   daytape3
                w:   weektape1,  weektape2,  weektape3
                m:   monthtape1, monthtape2, monthtape3

Efter första    d:   daytape1,   daytape2,   daytape3
vecko-kopier-   w:   weektape2,  weektape3,  weektape1
ingen:          m:   monthtape1, monthtape2, monthtape3
```

bup spar filerna på kopieringsmediet i överensstämmelse med följande tillvalskommando till *tar* och listan med filer lägges till loggfilen tillsammans med volymnamnet.

```
tar -cvfSF  enhet fil
```

eller

```
tar -cvfSFbk  enhet blockstorlek volymstorlek fil
```

I loggfilen **/usr/lib/bup/log** är varje fil, som lagrats på en kopieringsvolym, listad i ett format liknande kommandot *ls -l* med det fullständiga sökvägsnamnet tillsammans med volymnamn för säkerhetskopiering och tidpunkten för kopieringen. Endast den senaste tidsangivelsen för kopiering av varje fil sparas i loggfilen. Om samma fil finns på olika medier, finns den listad en gång per media. Loggfilen kan listas med kommandot
bup -l fil.

Tillvalet **-u** specificerar volymnamn, som vanligtvis är ett bandnummer eller liknande.

När säkerhetskopiering görs lagras detta namn i loggfilen, och vi rekommenderar därför att bandet/disketten märks på samma sätt. Om tillval **-u** inte är specificerad och något av tillvalen **-bm**, **-bw** eller **-bd** används kommer *bup* att försöka hämta det volymnamn som anges i filen **/usr/lib/bup/volumenames**.

När filer återläses från kopian med tillval **-x**, kan ett volymnamn anges så att enbart filer på dessa volymer läses in. Wildcard kan användas när volymnamnet anges om de omges av apostrofer för att hindra shell av avkoda dem. Endast volymnamn som finns i logg-filen kan anges.

Tillvalet **-M** gör att *bup* sänder meddelande med *mail* till användaren om något fel uppstår under säkerhetskopieringen.

Det finns även ett antal tillval för att lägga till, ändra eller lista filer i kommandofilen för *bup*.

Tillvalet **-m** anger filer, som skall säkerhetskopieras, då den månatliga kopieringen skall göras (med *bup -bm*).

Tillvalet **-w** anger filer, som skall säkerhetskopieras, då veckans kopiering skall göras (med *bup -bw*).

Tillvalet **-d** anger filer, som skall säkerhetskopieras, då den dagliga kopieringen skall göras (med *bup -bd*).

Tillvalet **-a** anger filer, som skall säkerhetskopieras och sedan raderas, då kopiering för arkivering görs (med *bup -ba*). I detta fall raderas även motsvarande rad i *bup*:s kommandofil då arkivkopieringen görs.

Tillvalet **-k** tar bort en tidigare angiven fil från *bup*:s kommandofil. Namnet på filen/biblioteket skall återges på exakt samma sätt som den ges i kommandofilen, **/usr/lib/bup/cmd**, inklusive eventuella wildcards. Om filer angivits med optionen **-r** måste de anges på samma sätt om de skall tas bort.

Tillvalet **-r *findargument*** ger rekursiva parametrar till *find*. Detta tillval gör att alla följande parametrar på kommandoraden behandlas som parametrar till kommandot *find*. Parametrarna lagras i *bup*:s kommandofil och används av *find* för att hitta filerna då själva säkerhetskopieringen utförs. Ett mellanslag måste skilja **-r** och parametrarna åt. Observera att inga **-exec** eller **-print**-parametrar kan specificeras. Då **-r** används kan inga normala filnamn specificeras.

Tillvalet **-L** listar köade filer i kommandofilen för *bup*. Dessa filer kommer att kopieras när något av tillvalen **-b[mwda]** ges.

Tillval för att lista och ändra logg-filen för bup:

Tillvalet **-l** listar filinformationen från *bup*:s loggfil. **-l** kan kombineras med **-u** då man vill lista filer enbart med ett specificerat volymnamn. Om filnamn är specificerade som argument kommer endast information om dessa att listas. Filinformationen listas som med kommandot ***ls -l***. Om även optionen **-v** ges, listas också tiden ut då de kopierades. Wildcards kan användas när filnamn och volymnamn (med **-u**) anges för att söka filer i logg-filen. Filnamn som ges utan full sökväg anses vara i nuvarande bibliotek.

Tillvalet **-e** radera ut alla logg-rader för en viss volym (given med optionen **-u**) från *bup*'s logg-fil.

Tillval för att lista standard volymnamn:

Tillvalet **-nm** visar de standardvolymnamn som skall användas vid nästa månatliga säkerhetskopiering. Namnet hämtas från filen **/usr/lib/bup/volumenames**.

Tillvalet **-nw** visar de standard volymnamn som skall användas vid nästa veckas säkerhetskopiering. Namnet hämtas från filen **/usr/lib/bup/volumenames**.

Tillvalet **-nd** visar de standard volymnamn som skall användas vid nästa dags säkerhetskopiering. Namnet hämts från filen **/usr/lib/bup/volumenames**.

Tillval för att utföra säkerhetskopieringen:

Tillvalet **-bm** utför den månatliga säkerhetskopieringen. Alla filer som har angivits med **-m**, **-w** eller **-d** kommer att kopieras till den enhet som anges.

Tillvalet **-bw** utför veckokopieringen. Alla filer som har angivits med **-w** eller **-d** kommer att kopieras till den enhet som anges.

Tillvalet **-bd** utför den dagliga kopieringen. Alla filer som har angivits med **-d** kommer att kopieras till den enhet som anges.

Tillvalet **-ba** utför arkivkopiering till den angivna enheten och ***raderar*** filerna. Detta påverkar enbart filer som angivits med **-a**. Motsvarande rader med **-a** i *bup*:s kommandofil raderas automatiskt efter det att säkerhetskopieringen är genomförd. Om inget volymnamn angivits (med **-u**) kommer *bup* att skapa ett volymnamn från dagens datum och klockslaget. Detta volymnamn bör anges på volymen för identifiering.

Tillvalet **-c** testar volymen efter säkerhetskopiering. *bup* använder *tar* och läser volymen efter säkerhetskopiering för att kontrollera att kopiering gått bra.

Tillvalet **-tk *volymstorlek*** använder nästa argument (volymstorlek) som storleken på volymen i kbyte. Detta sänds vidare till *tar*-kommandot.

Tillvalet **-tb *blockfaktor*** använder nästa argument (blockfaktor) som blockfaktor på volymen. Detta sänds vidare till *tar*-kommandot.

Tillval för att läsa tillbaka filer från säkerhetskopieringvolymen:

Tillvalet **-x** hämtar filer från ett kopieringsmedium på den givna enheten. *bup* skriver ut volymnamnet på det medium (band eller diskett) som användaren skall sätta in och hämtar sedan de filnamn som är givna som argument. Om fil- eller biblioteksnamnet inte är givet med det fullständiga sökvägsnamnet så lägger *bup* till det nuvarande bibliotekets fullständiga sökvägsnamn. Wildcards kan användas när filnamnen specificeras, om de omges av apostrofer för att hindra shell att avkoda dem. bup sparar alltid filerna med fulls sökvägsnamn. Om användaren vill läsa in en fil till ett annat bibliotek för att där kunna undersöka den, kan tar-kommandot med optionen **-A** användas istället för *bup -x*.

EXEMPEL

```
bup -w '*'
```

Detta kommando anger att alla filer i det nuvarande bibliotek skall kopieras vid nästa veckas säkerhetskopiering.

```
bup -m -r -name \*.c
```

Detta kommando anger att alla *.c-filer i det gällande biblioteket skall kopieras vid nästa månatliga säkerhetskopiering. Wildcard-tecknet '*' citeras här från shell med \.

```
bup -bw -M -utape_123 /dev/st0
```

Detta kommando utför veckans säkerhetskopiering på ett band med volymnamnet 'tape_123'. Om tillvalet **-u** inte hade givits skulle volymnamnet vara det första namnet på raden märkt w: i filen **/usr/lib/bup/volumenames**. Eventuellt fel som uppstår kommer att rapporteras med *mail* till användaren.

```
bup -x /dev/st0 file1 file2 file3
```

Detta kommando återladdar filerna file1 file2 och file3 till det aktuella biblioteket. *bup* visar det riktiga volymnamnet på det media (band eller diskett), som skall sättas in innan filerna laddas. För filer, som inte skall tillhöra nuvarande bibliotek, måste fullständiga sökvägsnamn anges.

I biblioteket **/usr/lib/bup** finns tre kommandofiler med standardformat, som kan användas för säkerhetskopiering med *bup*. Deras innehåll listas nedan. Super-user privilegier krävs för att använda dessa kommandon. Vilka filer som ska sparas har dessförinnan definierats i filen **/usr/lib/bup/cmd** med hjälp av *bup*-kommandon.

/usr/lib/bup/daily (gör daglig säkerhetskopiering)

/usr/lib/bup/weekly (gör veckovis säkerhetskopiering)

/usr/lib/bup/monthly (gör månadsvis säkerhetskopiering)

```
# /usr/lib/bup/daily
# bup script for daily backup.
# This script should be called when the daily backup shall be
# run. The daily backup can be a total backup of the complete
# disc or it can be an incremental backup (only those files that
# have been changed the last day).
#
bup -bd -M -tk 60000 -tb 300 -c /dev/st0
```

```
# /usr/lib/bup/weekly
# bup script for weekly backup.
# This script should be called when the weekly backup shall be
# run. The weekly backup can be a total backup of the complete
# disc or it can be an incremental backup (only those files that
# have been changed the last week).
#
bup -bw -M -tk 60000 -tb 300 -c /dev/st0
```

```
# /usr/lib/bup/monthly
# bup script for monthly backup.
# This script should be called when the monthly backup shall
# be run. The monthly backup can be a total backup of the
# complete disc or it can be an incremental backup (only those
# files that have been changed the last month).
#
bup -bm -M -tk 60000 -tb 300 -c /dev/st0
```

12

12. Cron

12. Cron

cron är en administrativ demon-process som startar program vid specificerade tidpunkter. I system som baserades på tidigare UNIX-versioner specificerades alla program som skulle exekveras i en enda fil (*crontab*), vilken normalt hade speciella åtkomsttillstånd för att kunna hantera systemadministrativa processer. För att använda dessa funktioner krävdes särskilda tillstånd.

I D-NIX hanterar *cron* både administrativa filer och användarnas egna crontab-filer, och kommandot *crontab* används av vanliga användare för att hantera dessa filer.

crontab hanterar användarnas crontab-filer. Programmet testar tillstånd och kopierar användarens filer och kommandon till rätt bibliotek för att *cron*-demonprocessen ska kunna utföra dem.

cron hanterar även begäran från kommandona *at*(1) och *batch*(1) i utvecklingssystemet. Senare i kapitlet ges en kort beskrivning av detta, i övrigt hänvisas till dokumentationen för utvecklingspaketet.

12.1. Administrativa filer

Detta kapitel beskriver filer som används av *cron* och som måste skapas och ändras av den systemansvarige. Alla dessa filer finns i biblioteket **/usr/lib/cron**.

Filer med kommandon som skall exekveras av *cron* samt crontab-tabeller finns i biblioteket **/usr/spool/cron**.

Kommunikationen mellan användar-kommandon och **cron** sker genom namngivna pipes (FIFO-filer) i filsystemet.

12.1.1 Filerna allow och deny

Olika användare kan tillåtas eller förbjudas att använda sig av kommandot *crontab* genom listor i filerna **cron.allow** eller **cron.deny**.

Filen **cron.allow** kan innehålla en lista över de användare (inloggningsnamnen) som får lägga kommandofiler i kö med *crontab*. Om filen cron.allow inte finns, kan istället en lista över de användare som inte får använda *crontab* finnas i filen **cron.deny**.

Om filen **cron.allow** existerar, ignoreras filen **cron.deny** helt. Om varken **cron.allow** eller **cron.deny** existerar är det endast root (med superuserprivilegier) som kan använda kommandot *crontab*.

En sammanfattande tabell visar tillstånden att använda kommandona:

allow	**deny**	**Användare med tillstånd**
saknas	saknas	Endast root
saknas	tom	Alla användare
saknas	finns	Hindrar listade användare
tom	ignoreras	Endast root
finns	ignoreras	Tillåter listade användare

12.1.2 Köer för cron

cron kan dynamiskt begränsa antalet processer som exekveras samtidigt från köerna. Den klarar av upp till 26 olika köer och begränsar antalet processer ur varje kö separat. I filen **queuedefs** ska finnas definitioner av de köer som används.

Observera! I grundpaketet används endast den kö som kallas **c** med kommandot *crontab*, och om denna kö inte definierats i **queuedefs** används standardvärden istället.

Varje rad i filen **queuedefs** skall ha följande format:

```
q.NNjNNnNNw
```

där:

q	är en bokstav a-z som är könamn.
NNj	är max antal processer ur denna kö som får exekveras samtidigt. **NN** är ett heltal och standardvärdet är 100 om det ej anges.
NNn	är argumentet till *nice*-kommandot som används när en process startas från kön. Standardvärdet är 2 om det ej anges.
NNw	är tiden (i sekunder) innan cron försöker starta en process ur denna kö igen efter att villkoren visat sig inte uppfyllda. Standardvärdet är 60 (1 minut) om det inte anges.

Ett tomt fält sätts automatiskt till standardvärdet. Följande är ett exempel på en **queuedefs**-fil:

```
a.4j1n
b.2j2n90w
c.0n10j
n.120w4n1j
```

Tredje raden beskriver kön **c**, som används för kommandot *crontab*. Raden anger att max tio processer ur kön får köras samtidigt och att nice-värdet är 0. Om kön innehåller fler än tio begäran kommer alla utom de tio första att få vänta. Eftersom ingen väntetid definierats här kommer *cron* att testa varje minut (standardtiden 60 sek) och starta den elfte processen ur denna kö då någon av de första tio blivit klar. Med *nice*-värdet 0 kommer processerna att exekveras med samma prioritet som normala kommandon. Med ett högre *nice*-värde kan de exekveras med en lägre prioritet.

De andra raderna i exemplet beskriver andra köer, som endast används med *at*(1) och *batch*(1) som inte ingår i grundsystemet. Till exempel anges att bara en process får köras i kö **n**, (aktiverad av *at* **-qn**), varvid systemet väntar minst 120 sekunder mellan testerna om nästa kommando i kön ska få exekvera. Processen körs med *nice*-värdet 4.

Om kö-definitionsfilen ändras läser *cron* den innan nästa process ur kön startas. Om filen inte existerar används standardvärdena.

Standardköer som används är följande:

a	Standard med *at*(1)-kommandot
b	Standard med *batch*(1)-kommandot
c	Standard med *crontab*-kommandot
e,f,....	Övriga köer kan användas genom att ge optionen **-qn** till *at*(1)-kommandot, där **n** är den kö som önskas.

12.1.3 Log-fil

cron loggar alla kommandon som startas, avslutas och deras status i filen **log**. Poster som inleds med tecknet > mostvarar start av ett kommando. Två poster skrivs för varje kommando. Först lagras kommandot och sedan kommer en post med login-namn, process-id, könamn och tiden när kommandot startades. Poster som inleds med tecknet < motsvarar terminering av ett kommando och liknar start-poster, med tillägget att kommandots utgångsstatus också lagras. Poster som inleds med tecknet ! innehåller statusinformation.

12.1.4 crontab-filernas format

Med den nya *cron*-processen har varje användare en egen individuell fil för *crontab*. Dessa filer ligger i biblioteket **/usr/spool/cron/crontabs**. Namnet på crontab-filen används som användarnamn vid exekveringen för att erhålla rätt filåtkomsttillstånd för användare och grupp.

Nedan är ett exempel på en rad i en crontab-fil med namnet **sys**. Den exekveras då med alla tillstånd som användare **sys** har.

```
0 19-7 * * * /usr/lib/sa/sal >/dev/null &
```

12.2. Allmänt om funktionen hos cron

Kommandot *crontab* skapar kommandon till *cron* och sänder dessa genom en spciell fil **/usr/lib/cron/FIFO**. När *crontab* gör en ny crontab-fil eller ändrar eller tar bort en redan befintlig, sänds ett meddelande via FIFO som gör att *cron* läser av filen och lägger in exekveringsbegäran i sin tidskö.

Eftersom *cron* läser alla filer i **/usr/spool/cron/crontabs** varje gång *cron* startas kommer exekveringsbegäran i köerna att aktiveras även om systemet stängs av och startas igen. Dessutom är det även möjligt att manuellt lägga in crontab-filer, förutsatt att användaren har lämpliga skrivtillstånd, varvid även dessa läses in nästa gång *cron* startas. Därvid används inte **cron.allow** eller **cron.deny**, vilka enbart styr *crontab*-kommandot. *cron* startas normalt på nytt då datorsystemet startas, eller vid övergång till fleranvändarmod i operativsystemet. Därför bör enbart super-user ha skrivtillstånd i dessa bibliotek.

cron-processen försöker rapportera fel som orsakar att den stoppas genom felmeddelande till systemkonsolen. Varningsmeddelanden ges även vid användning av *crontab* om *cron* inte är aktiverad i systemet.

12.3. Användning av cron med utvecklingspaketet

I utvecklingspaketet som beställs separat finns ytterligare två kommandon som sänder exekveringsbegäran till *cron*. *cron* hanterar dessa på liknande sätt som från *crontab*. För detaljer, se handböckerna för utvecklingssystemet.

Med *batch*(1) används kön med namnet **b** i filen **queuedefs**, medan kommandot *at*(1) normalt använder kön **a**. *at*(1) kan använda vilken kö som helst genom optionen **-q*n***, där ***n*** är könamnet. De ytterligare filer som då finns är:

/usr/spool/cron/atjobs/*	Bibliotek med kommandofiler som väntar på exekvering.
/usr/lib/cron/at.allow	Motsvarande **cron.allow**, men för *at* och *batch*.
/usr/lib/cron/at.deny	Motsvarande **cron.deny**.
/usr/lib/cron/.proto	Prototypfil för att skapa *at*-filer.
/usr/lib/cron/.proto.q	Separata prototypfiler kan användas för olika köer.

Prototypfiler används för att standardisera informationen som skrivs till de kommandofiler som skapas med *at*. Om en fil med namnet **.proto.q** finns, där **q** anger den kö den skall användas till, kopieras denna fil till den skapade kommandofilen. Annars används den allmänna prototypfilen **.proto**. Kommandot *at* avbryts med ett felmeddelande om inte prototypfil existerar.

Följande shellvariabler kan användas i prototypfilerna och substitueras då en kommandofil skapas av *at*.

$m	användarens nuvarande tillståndsmask vid skapande av filer (se *umask*).
$l	användarens nuvarande max filstorlek vid skapande av filer (se *ulimit*(2)). Det är bokstaven lilla **l** som används som variabelnamn.
$d	namnet på nuvarande aktuellt bibliotek.
$t	tiden då processen skall exekveras. Anges som ett numeriskt värde i sekunder.
$<	standard input läses härifrån och tills filslut och lägges in i kommandofilen.

Nedan är ett exempel på en prototypfil. Dessa kommandon lägges in i den kommandofil som skapas, före användarkommandona, men efter det att normala standardvariabler definierats.

```
cd $d
ulimit $l
umask $m
$<
```

13

13. UUCP Nätverkssystem

13. UUCP Nätverkssystem

13. UUCP Nätverkssystem

13.1. Introduktion

Dessa hjälpprogram, som ingår i D-NIX utbyggnadspaket, gör det möjligt för datorer som använder operativsystemet UNIX att kommunicera med varandra och med fjärrterminaler. Dessa hjälpprogram kan användas till en mängd olika saker, från att kopiera filer mellan datorer (*uucp* och *uuto*), beordra fjärrexekvering av kommandon (*uux*) samt utföra fjärrlogin (*cu*).

Som administratör måste du vara bekant med de administrativa verktygen, logg- och databasfilerna som används. Detta avsnitt kommer i de talj att beskriva dessa filer, bibliotek, demoner och kommandon.

Kommentarer finns även inlagda i de databasfiler som levereras med systemet för att underlätta vid ändringar och tillägg.

Innan konfigurationsfilerna skapas måste vissa grundläggande beslut tas angående tillträde till datorerna i nätverket. Om en administratör är ansvarig för endast ett datorsystem och detta skall anslutas till ett nätverk måste administratören bestämma vilken nivå av åtkomst de övriga systemen skall få. De övriga medlemmarna i nätverket måste göra samma sak för det nya systemet. Lösenordssystemet i varje dator används för att ge åtkomstprivilegier för andra system. Filen **/usr/lib/uucp/Permissions** begränsar åtkomsten från andra system till delar av filsystemet. Dessutom kan i denna fil läggas in begränsningar vid anrop från den lokala datorn till andra datorer. Eftersom filöverföringsprogrammen arbetar under identiteten *uucp*, krävs dessutom att användare *uucp* har vanliga åtkomsttillstånd till de bibliotek och filer som ska kunna nås. Filen **/usr/lib/uucp/Systems** på den lokala datorn bestämmer vilka andra system i nätverket som kan nås, vilka tider som är tillåtna och på vilket sätt kontakt skall tas mellan systems.

När fler än ett system upprättas har administratören kontroll över en större del av nätverket och kan ta fler beslut angående installationen. Till exempel kan nätverket installeras som ett privat nätverk där endast de system vilka ligger direkt under administratörens kontroll har tillgång till varandra. Att inte ge system utanför nätverket tillträde kan ordnas, men detta är vanligtvis opraktiskt. Tillträdet från utomstående system till enskilda datorer i det privata nätverket kan begränsas. Alternativt är att tillträde till/från utomstående system begränsas till endast ett system. Detta är ganska vanligt för att minimera arbetet med uppdatering av åtkomstinformation (lösenord, telefonnummer, loginsekvenser m.m.) och för att minimera antalet möjliga säkerhetsläckor i det privata nätverket.

Exempel

Några enkla exempel ges här för att illustrera hur UUCP nätverkssystem kan användas. Beteckningen "systemnamn" anger vanligen nodnamnet för ett annat datorsystem. Nodnamnet kan visas med kommandot *uname -n* lokalt på varje system. Den verkliga dataöverföringen mellan systemen sker inte alltid direkt utan kan ske vid en senare tidpunkt beroende på informationen i systemfilerna.

Exempel 1. Kopiera en fil till en annan dator, förutsatt att det finns åtkomsttillstånd för detta. Något allmänt bibliotek brukar finnas tillgänligt på varje dator, vanligen **/usr/spool/uucppublic** för att användas för filöverföringar. Notera att även den lokala filen (minfil) måste kunna läsas av *uucp*. Beteckningen **~användare** anger hembiblioteket för en användare. Beteckningen **~/användare** anger ett bibliotek, med användarens namn, som ligger i det allmänna biblioteket, t ex **/usr/spool/uucppublic/användare**.

```
uucp  minfil  systemnamn!tillbibliotek
uucp  minfil  systemnamn!tillbibliotek/dittfilnamn
uucp  minfil  systemnamn!~användare
uucp  minfil  systemnamn!~/användare
```

Kopiering kan även ske mellan två fjärrdatorer eller från en fjärrdator till den lokala datorn genom att även källfilen kan anges med systemnamn.

Exempel 2: Kopiera en fil till en annan dator, men denna gång till en användare, som själv får hämta filen med kommandot *uupick*.

```
uuto  minfil  systemnamn!användare
```

Filer som sänts med *uuto* kan hämtas med *uupick*, som söker efter användarens filer i PUBDIR. För varje fil som hittas frågar systemet vad användaren vill göra med filen. För att t.ex. flytta en fil till ett lämpligt bibliotek kan användaren skriva "*m mittbibliotek*" (se *UUTO(1C)*

Exempel 3: Sänd ett brev med kommandot *mail* till en användare på en annan dator. Med lämpliga hjälpfiler kan detta kommando automatiskt sända brevet via UUCP-systemet till den andra datorn.

```
mail systemnamn!användare < brevfil
```

Exempel 4: Koppla upp dig för inloggning som en vanlig användare till en annan dator, via kommandot *cu*. När ett systemnamn anges i cu-kommandot sköts uppkopplingen automatiskt av systemet, men inloggning sker manuellt.

```
cu  systemnamn
```

Referenser

Utöver denna beskrivning finns mer information om UUCP nätverkssystem i följande handböcker:

AT&T User's Reference:	uucico(1M) uucp(1C) uulog(with uucp in 1C) uuname(with uucp in 1C) uustat(1C) uupick(with uuto in 1C) uuto(1C) uux(with uucp in 1C)
AT&T Administrator Manual:	uucleanup(1M) uusched(1M) uucheck(1M) Uutry(1M) uuxqt(1M)
D-NIX Referenshandbok:	cron(1) crontab(1) cu(1) getty(1) kermit(1) mail(1) rmail(1)

Det finns även böcker, som behandlar administrationen av UUCP, att köpa i allmänna bokhandeln.

13.2. Nätverkshårdvara

Innan datorn kan kommunicera med andra datorer måste du installera hårdvaran för att få en komplett kommunikationslänk. De kablar och annan hårdvara, som kommer att behövas beror på hur du vill koppla samman datorerna: med seriell direktförbindelse, via telefonledning eller ett lokalt nätverk.

Direktförbindelse

Skapa en seriell direktförbindelse mellan två datorer genom att dra en kabel mellan serieportarna. En direktförbindelse är användbar då två datorer kommunicerar regelbundet och är placerade nära varandra - inom 15 meter. Du kan använda ett korthållsmodem för att öka avståndet något. En överföringshastighet på upp till 38400 bitar per sekund är möjlig då datorerna har direktförbindelse.

Telefonledning

Genom att använda ett modem med automatisk uppringning eller en separat uppringningsenhet (ACU, eng. Automatic Call Unit) kan datorn kommunicera med andra datorer över en vanlig telefonlinje. Den automatiska uppringningsenheten eller modemet slår numret som nätverksprogrammet begär. Den dator som kontaktas måste ha ett telefonmodem med möjlighet att svara på inkommande samtal.

Lokalt nätverk

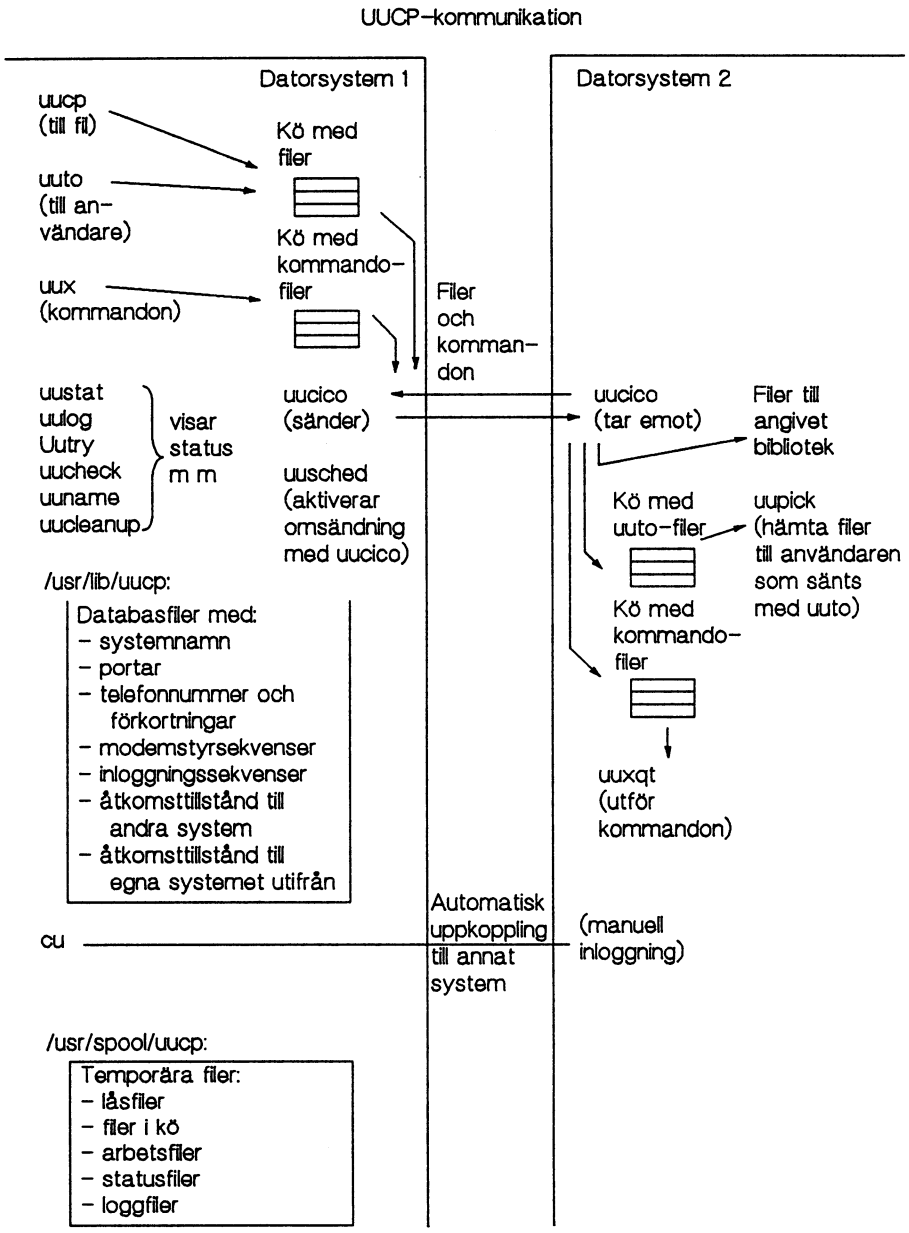
Om datorn är inkopplad som en nod i ett lokalt nätverk (LAN) kan den kommunicera med övriga datorer anslutna till nätverket, till exempel via en Ethernetförbindelse.

Globalt nätverk

Om datorn har en anslutning till ett globalt nätverk (WAN) kan den kommunicera med övriga tillgängliga datorer, till exempel via en X.25 anslutning.

13.3. Nätverkskommandon

Program för grundläggande nätverksanvändning kan delas in i två kategorier: användarprogram och administrativa program. Här följer en beskrivning av programmen i respektive kategori. Nedanstående bild ger först en liten förenklad översikt.



13.3.1 Användarprogram

Användarprogrammen för grundläggande nätverksanvändning finns i biblioteket **/usr/bin**. Du behöver inga speciella privilegier för att använda dessa program. Kommandona beskrivs i *AT&T User's Reference Manual*, utom *cu*(1) som återfinns i *D-NIX Referenshandbok*.

cu(1)	Kopplar datorn till en fjärrdator så att du kan vara inloggad på båda samtidigt. Detta gör det möjligt att flytta filer eller utföra kommandon på någon av datorerna utan att den ursprungliga uppkopplingen försvinner. Dock kan enbart textfiler överföras och ingen kontroll sker av överföringen.
uucp(1C)	Låter en användare kopiera en fil från en dator till en annan. *uucp* skapar arbetsfiler och datafiler, köar jobbet för överföring och anropar *uucico*-demonen vilken i sin tur försöker kontakta fjärrdatorn.
uuto(1C)	Kopierar filer från en dator till ett allmänt köbibliotek på en annan dator (**/usr/spool-/uucppublic/receive**). Till skillnad från *uucp*, som låter dig kopiera en fil till ett valfritt tillgängligt bibliotek på fjärrdatorn, placerar *uuto* filen i ett lämpligt köbibliotek och talar om för fjärranvändaren att han skall hämta den med *uupick*.
uupick(uuto 1C)	Hämtar filer under biblioteket **/usr/spool-/uucppublic/receive** när filer överförts till en dator med hjälp av *uuto*.
uux(1C)	Skapar de arbets-, data- och exekverbara filer som behövs för att utföra ett kommando på en fjärrdator. Arbetsfilen innehåller samma information som de arbetsfiler som skapas av *uucp* och *uuto*. Exekveringsfilerna innehåller den kommandosträng som skall exekveras på fjärrdatorn samt en lista över datafilerna. Datafilerna är de filer som behövs för att utföra kommandot.
uustat(1C)	Visar status för begärda överföringar (*uucp*, *uuto* och *uux*). Den ger dig även möjlighet att kontrollera köade överföringar eller låter användaren ta bort dem ur kön.
uuname(uucp 1C)	Visar vilka systemnamn som finns inlagda i systemfilerna.

13.3.2 Administrativa program

De flesta administrativa program finns i biblioteket **/usr/lib/uucp** tillsammans med databasfiler för nätverksanvändning och olika shell-procedurer för administrationen. Ett undantag är *uulog* vilken finns i biblioteket **/usr/bin**. Dessa kommandon beskrivs i AT&*T Aministrator's Reference Manual*.

Du skall använda loginidentiteten uucp, eftersom den äger hjälpfilerna och de köade datafilerna. Hembiblioteket för loginidentiteten *uucp* är **/usr/lib/uucp**. Andra loginidentiteter används av fjärrdatorer då de vill få tillgång till din dator, t ex nuucp med flera. Anrop från fjärrdatorer svaras av *uucico*, vilken dock alltid arbetar med identiteten *uucp* för full tillgång till alla filer.

uulog(uucp 1C) Visar innehållet i en specificerad dators loggfiler. Loggfiler skapas för varje fjärrdator som din dator kommunicerar med, separat för de olika typerna *uucp*, *uuto* och *uux*.

uucleanup(1M) Städar i spool-biblioteket. Den utförs vanligen från shell scripten **uudemon.cleanu**, vilken startas av *cron*.

Uutry(1M) Testar möjligheterna till hantering av anrop och klarar även viss felsökning. Den anropar *uucico*-demonen för att upprätta en kommunikationslänk mellan din dator och den fjärrdator du specificerar.

uucheck(1M) Kontrollerar närvaron av program, bibliotek och stödfiler för nätverksadministration. Den kan också kontrollera vissa delar av filen med åtomsträttigheter, **Permissions**, så att det inte finns uppenbara syntaktiska felaktigheter.

uudemon.hour Shellprocedur som normalt startas av *cron* vid lämpliga tidpunkter för att åter försöka med överföringar som ligger i kö eller köade kommandon, som inte kunnat genomföras.

uudemon.poll Shellprocedur som normalt startas av *cron* vid lämpliga tidpunkter (alltid strax före uudemon.hour) för att anropa fjärrsystem som är passiva. Detta kallas pollning. Filen **Poll** innehåller information om vilka fjärrsystem som skall pollas.

uudemon.cleanu Shellprocedur som normalt startas av *cron* vid lämpliga tidpunkter för att rensa bland gamla temporärfiler (via *uucleanup*). Dessutom kombineras gamla loggfiler och flyttas till **/usr/spool/uucp/.Old**, där de sparas minst en vecka.

uudemon.admin Shellprocedur som normalt startas av *cron*, och sänder mail till UUCP-administratören om överföringar med uucico. Glöm ej att regelbundet logga in som *uucp* och ta hand om denna information.

13.4. Demoner

Det finns tre demoner bland hjälpprogrammen för nätverkshantering. En demon är ett program som utförs som en bakgrundsprocess och utför allmänna funktioner över hela systemet. Dessa demoner hanterar filöverföringar och kommandoexekvering. De kan även startas manuellt. Dessa programfiler skall ha 'Set-user-id'-biten satt, och ägas av *uucp*, för att arbeta med filåtkomsttillstånd enligt *uucp*.

uucico(1M) Väljer den enhet som skall användas för länken, upprättar länken till fjärrdatorn, utför nödvändig loginsekvens och åtkomstkontroll, överför data och exekveringsfiler, loggar resultat samt skickar *mail* till användaren då överföringen är klar. När den lokala *uucico*-demonen anropar en fjärrdator "talar" den med *uucico*-demonen på fjärrdatorn under sessionen.

uucico-demonen startas av *uucp*, *uuto* och *uux* program, då nödvändiga filer har skapats, för att upprätta en förbindelse till fjärrdatorn. Den startas även av programmen *uusched och Uutry*.

uuxqt(1M) Utför begäran om fjärrexekvering. Den söker i spool-biblioteket efter exekveringsfiler (alltid namnet **X.***fil*) vilka kommer från en fjärrdator. När en **X.***fil* hittas öppnas den av *uuxqt* för att komma åt namnen på de datafiler som behövs. Sedan kontrolleras att de nödvändiga datafilerna finns tillgängliga. Om filerna finns tillgängliga kontrolleras filen med åtkomsträttigheter, **Permissions**, för att verifiera att den har rätt att utföra önskat kommando. *uuxqt*-demonen startas av shellscriptet **uudemon.hour** som startas av *cron*.

uusched(1M) Planerar det köade arbetet i spool-biblioteket. Innan *uucico*-demonen startas kommer *uusched* att slumpmässigt bestämma i vilken ordning fjärrdatorer kommer att anropas. *uusched* startas av shellscriptet **uudemon.hour** som startas av *cron*.

13.5. Hjälpfiler

Hjälpfilerna för nätverksprogrammen finns i biblioteket **/usr/lib/uucp**. Alla användare skall ha läs- och exekveringstillstånd i **/usr/lib/uucp**. Beskrivningen nedan innehåller detaljer om strukturen på dessa filer så att du kan editera dem manuellt.

Systems
Innehåller information till *uucico*-demonen och programmet *cu* vid upprättandet av en förbindelse till en fjärrdator. Den innehåller information som t.ex namnet på fjärrdatorn, namnet på uppkopplingsenheten associerad med fjärrdatorn, när fjärrdatorn kan nås, telefonnummer, login ID samt lösenord. Dessutom tillåts normalt inloggning enbart från fjärrdatorer listade i denna fil. Filen **Systems** skall av säkerhetsskäl ha åtkomsttillstånden 600 oktalt för att kunna läsas enbart av användaren uucp.

Devices
Innehåller information om placering och överföringshastighet för uppringande modem, direktförbindelser och nätverksenheter.

Dialers
Innehåller de teckensträngar som används för att förhandla med modem och nätverksenheter (automatisk uppringning m m) vid etablerandet av förbindelser med fjärrdatorer (inte typ 801-uppringare).

Dialcodes
Innehåller nummerkodsförkortningar vilka kan användas i fältet för telefonnummer i filen **Systems**.

Permissions
Denna fil definierar nivån av åtkomsträttigheter för olika datorer då de försöker överföra filer eller fjärrexekvera kommandon på din dator. Här anges även begränsningar vid anrop från den lokala datorn till olika fjärrdatorer.

Poll
Denna fil definierar de datorer som skall pollas av ditt system och när de skall pollas.

Devconfig
Denna fil används enbart för att konfigurera hjälpfilerna för ett nätverk av typen STARLAN eller enligt AT&Ts Transport Interface. Annars lämnas denna fil tom.

Sysfiles
Denna fil används för att ange om olika filer skall användas av *uucico* och *cu* som **Systems, Devices** och **Dialers** filer.

Det finns flera andra filer som skulle kunna ingå bland dessa hjälpfiler, men de är inte direkt relaterade till processen att koppla upp en förbindelse och överföra filer. Dessa är **Maxuuxqts**, **Maxuuscheds** och **remote.unknown**. De allmänna systemfilerna **/etc/passwd**, **/etc/group** och **/etc/inittab** skall också vara anpassade till nätverkssystemet och datorernas **systemnamn** vara definierade. Detta beskrivs i slutet på detta delkapitel.

För att skapa en länk till en ny fjärrdator skall följande utföras:

1. Upprätta den fysiska förbindelsen, t ex via en serieport och ett modem på båda systemen. Filen **/etc/inittab** kan behöva ändras för de serieportar som används, så att inloggning respektive utgående anrop tillåts och så att båda datorerna använder samma parametrar, t ex överföringshastiget.

2. Se till att båda systemen har unika **systemnamn**.

3. Skapa nya rader i filen **Systems** i båda datorerna, så de får veta varandras systemnamn, hur kontakten skall ske och eventuella tidsbegränsningar.

4. Eventuellt kan en ny enhet behöva definieras i filen **Devices** med det modem som används.

5. Om en ny typ av modem införs kan det bli nödvändigt att ändra eller införa nya uppringningskommandon i filen **Dialers**.

6. Inför säkerhetsinformation i filen **Permissions**, eventuellt olika för olika fjärrdatorer.

7. Eventuellt ökas säkerheten genom olika begränsningar, t ex separata login-namn och lösenord för olika fjärrdatorer, genom att anrop enbart får göras från den ena datorn (filen **Poll**) eller genom att motringning krävs.

Efter varje ändring bör hjälpfilerna kontrolleras med följande kommando, vilket dessutom listar hur filen **Permissions** tolkas av programvaran:

```
uucheck -v
```

13.5.1 Referenser mellan hjälpfilerna

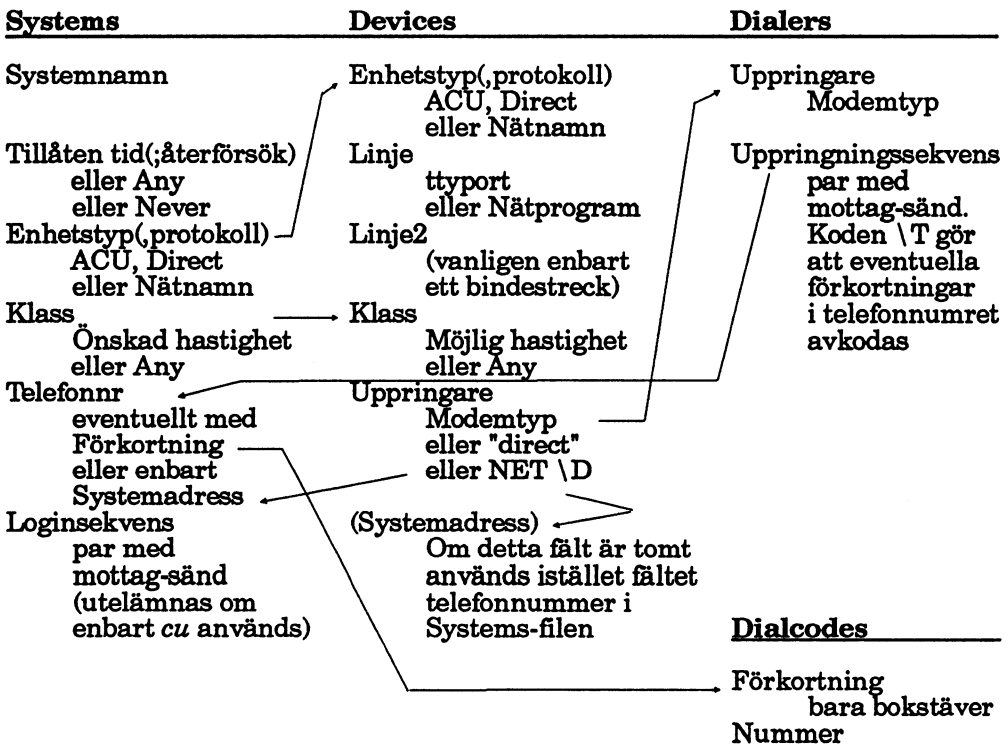
I tabellen nedan ges en starkt förenklad sammanfattande bild av vilka fält i de viktigaste databasfilerna som hänger ihop och vilka parametrar de innehåller. Varje fil beskrivs sedan i följande delkapitel.

Anrop från användarprogrammen sker vanligen till "Systemnamn" enligt filen **Systems**, varefter lämplig ledig kommunikationsväg kopplas upp automatiskt. I **Systems** anges vilka hastigheter och vilken enhetstyp som krävs (t ex modem med automatisk uppringning ACU). I filen **Devices** sökes sedan en lämplig linje och vilken uppringare (t ex modem) som är kopplad till den. I filen **Dialers** finns information hur uppringaren styrs. Telefonnummer hämtas från **Systems**. Telefonnumret kan innehålla förkortningar, som återfinns i filen **Dialcodes**. När kontakten med fjärrdatorn är upprättad sker inloggningen enligt loginsekvensen i Systems, varefter *uucico* i båda systemen sköter överföringen av datafiler och kommandofiler. Exempel:

```
Systems: eagle Any ACU 1200 sthlm933251 ogin: nuucp ssword: Oak
Devices: ACU tty25 - 1200 hayes
Dialers: hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
Dialcodes: sthlm 08
```

Om uppkoppling ska ske via ett lokalt nätverk, t ex över ethernet med telnet, anges ett nätnamn som enhetstyp och nätverksprogrammet (ex telnet) anges som Linje i filen **Devices**. Då anges systemadressen i filen **Systems** istället för telefonnummer. Vanligen är systemnamn och systemadress samma sträng och består av datorns nodnamn.

```
Systems: eagle Any ETHNET - eagle ogin: nuucp sword: Oak
Devices: ETHNET telnet - - NET \D
```



Förklaringar:

Nätnamn som enhetstyp är enbart en valfri sträng för att referera från filen **Systems** till en viss rad i filen **Devices**. Enhetstypen **Direct** i filen **Devices** krävs av *cu* då enbart ttyporten anges, ex *cu -l /dev /tty25*.

NET \D indikerar internt lokalt nätverk, varvid **Nätprogram** sköter uppkoppling till adressen **Systemadress**.

mottag-sänd anger par av strängar som sänds och förväntas vid styrning av modem respektive under inloggningen.

13.5.2 Filen Systems

Filen **Systems** (**/usr/lib/uucp/Systems**) innehåller information som behövs av uucico-demonen för att etablera en kommunikationslänk till en fjärrdator. Varje rad i filen representerar en dator som kan anropas av din dator. Vidare kan programvaran förhindra att datorer som inte finns i denna fil loggar in på din dator (se avsnittet "Andra Nätverksfiler" i detta kapitel för en beskrivning av filen **remote.unknown**). Fler än en rad kan finnas för en specifik dator. De extra raderna representerar alternativa kommunikationsvägar vilka kommer att prövas i sekventiell ordning.

Observera! I denna fil finns både login-namn och lösenord i klartext för inloggning till alla tillgängliga fjärrdatorer, varför denna fil alltid bör skyddas genom att enbart tillåta användare *uucp* att läsa/skriva i den. Kommandot *chmod 600 Systems* sätter detta skydd.

Fälten i denna fil är nära relaterade till fält i filerna **Devices, Dialers** och **Dialcodes**. Innan du gör några ändringar i någon av dessa filer bör du vara bekant med dem alla. En ändring av en rad i en fil kan kräva en ändring av en rad i en annan fil. Rader som börjar med # är kommentarrader.

Genom att använda filen **Sysfiles** kan du definiera flera filer vilka kan användas som Systems-filer. Se beskrivningen av filen **Sysfiles** för detaljer.

Varje rad i filen **Systems** har följande format:

```
Systemnamn Tid Enhetstyp Klass Telefonnummer Loginsekvens
```

Vart och ett av dessa fält definieras i följande avsnitt.

Systemnamn

Detta fält innehåller nodnamnet på fjärrdatorn. Utgående från detta namn söker programvaran genom hjälpfilerna för att koppla upp en lämplig kommunikationskanal.

Tid

Detta fält innehåller en sträng som beskriver vilka dagar i veckan och vilka tider på dagen då fjärrdatorn kan anropas. Tidfältet skall vara en sammanhängande sträng utan mellanslag och har följande format:

```
DagTid[;omförsök]
```

Fältet *Dagtid* kan bestå av flera dagtid-kombinationer separerade med kommatecken (,). Inga mellanslag är tillåtna inom strängen. *Dag*-delen kan vara en lista innehållande något av följande:

Su Mo Tu We Th Fr Sa	Enskilda dagar
Wk	Vardagar (samma som MoTuWeThFr)
Any	Valfri dag

Never Din dator är i passiv mod i förhållande till fjärrdatorn. Om *Tid*-fältet innehåller **Never** kan din dator aldrig initiera ett anrop till fjärrdatorn. Anropet måste initieras av fjärrdatorn. Se även beskrivningen av filen **Permissions**.

Här följer ett exempel:

```
Wk1700-0800,Sa,Su
```

Detta exempel tillåter anrop från 17:00 till 08:00 Måndag till Torsdag, samt när som helst på Lördag och Söndag. Detta skulle kunna vara ett effektivt sätt att tillåta anrop då telefontaxan är låg om omedelbar överföring inte är nödvändig.

Tid-delen skall vara ett tidsintervall som t ex 0800-1230. Om ingen tidsperiod anges antas anrop vara tillåtna oavsett tidpunkt. En tidsperiod som passerar 0000 är tillåten. T ex 0800-0600 betyder att alla tider utom de mellan 0600 och 0800 är tillåtna. Ett valfritt extrafält *;omförsök* finns för att specificera minsta tiden (i minuter) mellan omförsök efter ett misslyckat anrop. Om inget anges är väntetiden 60 minuter. Avskiljningstecknet är semikolon (;) vilket innebär att Any;9 betyder att anrop är tillåtna när som helst men att det dröjer minst 9 minuter efter ett misslyckat anrop till nästa omförsök.

Enhetstyp(,protokoll)

I detta fält anges enhetstypen som skall användas för att upprätta kommunikationslänken till fjärrdatorn. Nyckelordet som används i detta fält motsvarar det första fältet i filen **Devices**, t ex ACU i exemplet nedan.

```
Systems: eagle Any ACU,g D1200 NY3251 ogin: nuucp sword: Oak
Devices: ACU tty11 - D1200 penril
```

Vid en direktförbindelse till en annan dator, kan t ex systemnamnet användas som enhetstyp för att referera till en specifik rad i filen **Devices**. Se fältet *Enhetstyp* i **Devices** för närmare detaljer.

Du kan även definiera det protokoll som skall användas för att kontakta systemet genom att lägga till det till *Enhetstyp*-fältet efter ett komma. I exemplet ovan visas hur du kopplar protokollet **g** till enhetstypen ACU. Se även avsnittet "Protokoll" i beskrivningen av filen **Devices** för detaljer.

Klass

I detta fält anges önskad överföringshastighet för den enhet som ska används för att upprätta kommunikationslänken. Det kan innehålla en bokstav tillsamans med hastigheten (t.ex C1200, D1200) för att skilja på olika grupper av uppringare (se beskrivningen av *Klass*-fältet för filen **Devices**). Nyckelordet **Any** kan anges om valfri överföringshastighet skall vara tillåten. Endast enheter med motsvarande *Klass*-fält i filen *Devices* används, t ex D1200 nedan:

```
Systems: eagle Any ACU D1200 NY3251 ogin: nuucp sword: Oak
Devices: ACU tty11 - D1200 penril
```

Om det inte behövs någon information i detta fält skall du ändå markera fältet med ett - tecken. Detta gäller bl a när interna nätverk används, t ex ethernet.

Telefonnummer

I detta fält anges *telefonnumret* till fjärrdatorn för automatiska uppringare (modem) eller en *token*-sträng (systemadress) om en nätverksomkopplare eller internt nätverksprogram (ex telnet) används.

Telefonnumret består av en valfri alfabetisk förkortning och en numerisk del. Om en alfabetisk förkortning används som en del av eller hela numret, måste den finnas i filen **Dialcodes**. Exempelvis:

```
Systems: eagle Any ACU D1200 NY3251 ogin: nuucp sword: Oak
Dialcodes: NY 9=1212555
```

I denna sträng används ett = tecken för att tala om för den automatiska uppringaren att den skall vänta på ny kopplingston innan den slår återstående siffror. Ett bindestreck (-) i strängen betyder att den automatiska uppringaren skall ta en paus på 4 sekunder innan den slår nästa siffra. Dessa tecken översätts eventuellt till andra i filen **Dialers**, beroende på uppringare.

Om datorn är kopplad till en nätverksomkopplare kan andra datorer som är anslutna till denna omkopplare anropas. Raderna i filen **Systems** för dessa datorer innehåller inte ett telefonnummer i fältet *Telefonnummer*, de innehåller istället det *token* (adressträng) som måste skickas till omkopplaren så att den kan veta vilken dator din dator vill kommunicera med. Detta är vanligtvis samma som systemnamnet. Motsvarande rad i filen **Devices** skall innehålla ett **\D** i slutet på raden så att adressträngen inte översätts med hjälp av filen **Dialcodes**.

Vid förbindelser via ett lokalt nätverk, t ex Ethernet, anges NET \D i fältet *uppringare* i **Devices** och istället för telefonnummer anges den systemadress (vanligen samma som systemnamnet) som ska vara parameter till nätverksprogrammet (angivet i fältet *Linje* i **Devices**). Med följande exempel sker uppkoppling till systemet **sys15**, genom programmet *telnet syst15* då följande rader anges i filerna:

```
Devices: ETHNET telnet - - NET \D
Systems: sys15 Any ETHNET - syst15 ogin: nuucp sword: Oak
```

Vid en direktförbindelse ut via en serieport, markerad med enhetstypen *Direct* i filen **Devices**, anges portnamnet i telefonnummerfältet.

Login

Detta fält innehåller inloggnings-sekvensen skriven som en serie av fält och underfält med formatet:

```
förväntad sänd förväntad sänd ... ...
```

där *förväntad* är den sträng som tas emot och *sänd* är den sträng som sänds då den förväntade strängen mottagits.

Fältet *förväntad* kan vara uppdelat i underfält med bindestreck som separator, enligt:

```
förväntad1[-sänd2-förväntad2] ...
```

där *sänd2* sänds om den tidigare *förväntad1* inte blivit läst. Det *förväntad2* som kommer efter *sänd2* är nästa förväntade sträng. Med strängen **login--login** förväntar sig UUCP **login**. Om UUCP får in **login** fortsätter den med nästa fält. Om den inte får in login kommer den att sända ingenting följt av ett nyrad-tecken och sedan vänta på ett nytt **login**. Om det inledningsvis inte förväntas några tecken från fjärrdatorn skall tecknen "" (en tom sträng) användas i det första *förväntad*-fältet. Notera att alla *sänd*-fält kommer att sändas iväg med ett avslutande nyrad-tecken såvida inte *sänd*-strängen avslutas med ett **\c**.

Här följer ett exempel på en rad i filen **Systems** som använder en förväntad-sänd sträng.

```
owl Any ACU 1200 Chicago6013 "" \r ogin:-BREAK-ogin: \
                                uucpx word: xyzzy
```

Detta exempel betyder, sänd ett RETURN-tecken och vänta på strängen **ogin:** (för Login:). Om den strängen inte kommer, sänds ett BREAK-tecken. När **ogin:** kommer, sänds loginnamnet **uucpx**, och då sedan strängen **word:** kommer (för password:) skickas lösenordet **xyzzy**.

Vid uppkoppling mot ett modem på en fjärrdator kan BREAK behöva sändas bl a för att ändra överföringshastigheten, om *getty* på fjärrdatorn provar olika hastigheter tills sändaren och mottagaren är överens.

Det finns ett flertal escape-tecken vilka har en specifik betydelse då de ingår som en del i en sträng som sänds under loginsekvensen. Följande escape-tecken är användbara vid UUCP kommunikation:

\N	Sänd eller förvänta ett null-tecken (ASCII NUL).
\b	Sänd eller förvänta ett backspace-tecken.
\c	Om det är slutet på en sträng, ta bort det nyradtecken som vanligtvis sänds. Ignorera annars.
\d	Fördröj i två sekunder innan fortsatt sändning eller mottagning av fler tecken.
\p	Infoga en paus på ungefär 1/4 till 1/2 sekund.
\E	Aktivera eko-kontroll. (Från och med nu gäller att så fort ett tecken sänds iväg väntar vi tills tecknet kommer tillbaka innan vi gör något annat).

\e	Koppla bort eko-kontroll.
\n	Sänd ett nyrad-tecken.
\r	Sänd eller förvänta ett RETURN-tecken.
\s	Sänd eller förvänta ett mellanslagstecken.
\t	Sänd eller förvänta ett tab-tecken.
****	Sänd eller fövänta ett \-tecken.
EOT	Sänd eller förvänta EOT nyrad, två gånger.
BREAK	Sänd eller förvänta ett break-tecken.
\K	Samma som BREAK.
\nnn	Gör om det oktala talet (nnn) till ett tecken.

13.5.3 Filen Devices

Filen **Devices** (**/usr/lib/uucp/Devices**) innehåller information för alla de enheter som kan användas för att etablera en länk till en fjärrdator. Dessa är t.ex automatiska uppringningsenheter (modem), direktförbindelser och nätverkskopplingar. Se även filen **Sysfiles** om flera **Devices**-filer skall användas.

Anmärkning: Fälten i denna fil är nära relaterade till fält i filerna **Dialers, Systems** och **Dialcodes.**

Varje rad i filen Devices har följande utseende:

```
Enhetstyp Linje Linje2 Klass Uppringare-Token-par
```

Vart och ett av dessa fält definieras i följande avsnitt.

Enhetstyp(,protokoll)

Detta fält kan innehålla ett av två nyckelord (**Direct** eller **ACU**), namnet på ett nätverk, ett systemnamn eller en omkopplare i ett lokalt nätverk. Fältet är en hänvisning från tredje fältet i filen **Systems** till en eller flera rader i **Devices**. Exempel med typ = ACU:

```
Systems: eagle Any ACU 1200 3251 ogin: nuucp ssword: Oak

Devices: ACU tty11 - 1200 penril
Devices: ACU tty25 - 1200 hayes
```

Det är möjligt att ange ett protokoll som skall användas för denna enhet, genom att lägga till en protokollparameter efter ett komma. Se avsnittet Protokoll i slutet av detta delkapitel.

Direct — Detta nyckelord indikerar en direktförbindelse utan modem (endast för uppkoppling med *cu*).

ACU — Detta nyckelord indikerar att länken till en fjärrdator kopplas via ett automatiskt uppringande modem. Detta modem kan vara kopplat direkt till din dator eller indirekt via en omkopplare i det lokala nätverket.

Nätnamn — Enhetstypen kan vara namnet på ett lokalt nätverk (LAN), varvid kommandot som används för uppkoppling av förbindelsen anges i fältet *Linje*.

Exempel:

```
Systems: diab Any ETHNET - diab \
     in:--in: nuucp

Devices: ETHNET telnet - - NET \D
```

Systemnamn — Vid t ex en direktförbindelse till ett specifikt system kan systemnamnet för fjärrdatorn anges som enhetstyp. Att ange namnet på detta sätt används för att framhålla det faktum att den ledning som associeras med denna rad i filen **Devices** är för en specifik dator i filen **Systems.**

Linje

Detta fält innehåller enhetsnamnet på den ledning (tty-port) som denna rad i **Devices** beskriver. Om t.ex ett automatiskt uppringande modemet är kopplat till ledningen **/dev/tty11** skulle namnet i detta fält vara **tty11**. Om anslutningen skall göras via ett lokalt nätverk, ska denna parameter specificera namnet på det program som används för att koppla upp nätverket, t ex *telnet* eller *ncu*.

Linje2

Detta fält används enbart då en automatisk uppringare (ACU) använder en separat linje, exempelvis för typ 801-uppringare. Annars anges enbart ett bindestrek (-) i fältet.

Om nyckelordet ACU användes i fältet *Enhetstyp* och ACUn är en typ 801-uppringare skall *Linje2* innehålla enhetsnamnet på 801-uppringaren. (typ 801-uppringare har inget modem varför ett separat modem måste användas. Detta kopplas till ledningen som definieras i fältet *Linje*). Det betyder att en ledning allokeras till modemet och en annan till uppringaren. Eftersom denna konfiguration enbart används av 801-uppringare, ignoreras fältet *Linje2* normal. Fältet måste dock finnas varför ett bindestreck normalt anges.

Klass

Om nyckelordet **ACU** eller **Direct** används i fältet *Enhetstyp* kan *Klass* innehålla endast enhetens överföringshastighet. Det kan dock innehålla en bokstav tillsammans med överföringshastigheten (t.ex C1200 eller D1200) för att kunna skilja på olika grupper av uppringare. Detta är nödvändigt då större företag kan använda flera olika telefonnät, ett som tar hand om de interna kommunikationerna medan ett annat tar hand om de externa kommunikationerna. I detta fall är det nödvändigt att kunna bestämma vilken linje som skall användas för intern kommunikation och vilken som skall användas för extern kommunikation. Nyckelordet som används i *Klass*-fältet i filen **Devices** motsvarar det fjärde fältet (Klass) i filen **Systems**, (D1200 i exemplet nedan):

```
Systems: eagle Any ACU D1200 633251 ogin: nuucp ssword: Oak
Devices: ACU tty11 - D1200 penril
```

För de enheter som kan användas för valfri överföringshastighet används nyckelordet **Any**. Om **Any** används kommer linjen att användas med hastighet som anges i fältet *Klass* i filen **Systems**. Om även detta fält är satt till **Any** kommer överföringshastigheten att sättas till 1200 bps.

Fältet *Klass* används ej (ett bindestrek måste ges) när uppkoppling sker via ett lokalt nätverk.

Uppringare-token-par

Detta fält innehåller par av uppringare och token, kallat UTP-fält nedan. *Uppringare* kan vara namnet på ett automatiskt uppringande modem eller en omkopplare i en nätverk. Nyckelordet **direct** anges för en enhet som används för direktförbindelse. *Token* är en

adressträng som skall sändas till uppringaren och kan skrivas in efter delfältet *uppringare*, men den kan också utelämnas. I så fall hämtas den från fältet *Telefonnummer* i filen **Systems**.

Du kan ha valfritt antal uppringare-token-par. Detta kan t ex utnyttjas om ett modem någonstans i ett nätverk skall användas för att ringa ut, varvid uppkopplingen sker i två steg, först över nätverket till modemet och sedan via modemet till det önskade datorsystemet.

Detta fält har formatet (alla fält utom det första kan utelämnas):

```
uppringare token uppringare token ... ...
```

I de flesta fall består det sista paret endast av en uppringare varvid token hämtas från fältet *Telefonnummer* i filen **Systems**. Exempel, där uppringaren är ett hayes-kompatibelt modem och telefonnumret är 1234567:

```
Systems: eagle Any ACU 1200 1234567 ogin: nuucp ssword: Oak
Devices: ACU tty11 - 1200 hayes
Dialers: hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
```

Fältet *uppringare* hänvisar normalt till en rad (modemtyp) i filen **Dialers** eller vara kan en av flera speciella uppringartyper. Dessa speciella uppringartyper kompileras in i programvaran och är därmed tillgängliga utan att finnas med i filen **Dialers**.

801 - Bell 801 automatisk uppringare
NET - Nätverksgränssnitt (t ex en Ethernet-anslutning)

Om intern nätverksprogramvara används för uppkoppling, angivet med strängen NET i fältet uppringare, används ej filen **Dialers** och token (=systemadress) tas från femte fältet i filen Systems. Andra fältet i **Devices** (telnet i exemplet) anger då det program som används för uppkoppling. Exempelvis sker uppkoppling med följande rader enligt *telnet syst15*.

```
Systems: syst15 Any ETHNET - syst15 ogin:--ogin: nuucp
Devices: ETHNET telnet - - NET \D
```

Det finns två escape-tecken som kan förekomma i ett *UTP*-fält:

\T Indikerar att förkortningar i fältet *token* (vanligen ett telefonnummer) skall översättas med hjälp av filen **Dialcodes**. Detta escape-tecken är normalt istället placerat i filen **Dialcodes** för varje uppringar-procedur som är associerad med ett automatiskt uppringande modem (hayes, penril, ventel etc). Då sker inte översättningen förrän uppringarproceduren utförs.

\D Indikerar att fältet *token* **inte** skall översättas med hjälp av filen **Dialcodes**. Om inget escape-tecken är definierat i slutet på en rad i filen **Devices** antas **\D** som standard. Ett **\D** används också i filen **Dialers** för rader associerade med nätverksomkopplare (develcon och micom).

Fältet *uppringare-token-par* (*UTP*) kan struktureras på fyra olika sätt beroende på den enhet som beskrivs på raden.

1. Modem kopplat till en port på datorn

Om ett automatiskt uppringande modem är kopplat direkt till en port på din dator har *UTP*-fältet på motsvarande rad i filen **Devices** endast en sträng. Denna sträng är normalt namnet på modemet och används för att koppla ihop raden i filen **Devices** med en rad i filen **Dialers**. Uppringare är första fältet på en rad i filen **Dialers**.

Exempel, där 'ventel' är en modemtyp som används som uppringare och måste finnas i filen **Dialers**:

```
Systems: eagle Any ACU 1200 633251 ogin: nuucp ssword: Oak
Devices: ACU tty11 - 1200 ventel
Dialers: ventel =&-% "" \r\p\r\c $ \t%%\r\c ONLINE!
```

Notera att endast uppringardelen (**ventel**) finns med i *UTP*-fältet på raden i filen **Devices**. Detta betyder att *token* som skall skickas till uppringaren (i detta fall ett telefonnummer) hämtas från fältet *Telefonnummer* på en rad i filen **Systems**. (**\T** är underförstådd i UTP-fältet, se ovan).

2. Direktförbindelse till en dator

Om en direktförbindelse är upprättad till en speciell dator så skall *UTP*-fältet i motsvarande rad innehålla nyckelordet **direct**. Detta gäller för båda typerna av direktförbindelse, *Direct* och *Systemnamn* (se beskrivningen för fältet *Enhetstyp*).

```
Systems: eagle Any eagle 9600 tty11 ogin: nuucp ssword: Oak
Devices: eagle tty11 - 9600 direct
```

3. Förbindelse över en nätverksomkopplare

Om den dator du vill kommunicera med finns på samma nätverksomkopplare som din dator, måste din dator koppla upp sig mot nätverket, vilket i sin tur kopplar upp sig mot den andra datorn. För denna typ av enhet finns det endast ett par i UTP-fältet.

Om uppkoppling sker via en nätverksomkopplare skall delen *uppringare* motsvara en rad i filen **Dialers**, tex enligt exemplet nedan. Ofta används då omkopplarens namn både som enhetstyp och som uppringare.

```
Systems: syst15 Any develcon - syst15 ogin:--ogin: nuucp
Devices: develcon tty13 - 1200 develcon \D
Dialers: develcon "" "" \pr\ps\c est:\007 \E\D\e \007
```

Token-delen är lämnad blank vilket indikerar att den hämtas från filen **Systems**. *Token* är den adress som omkopplaren har till datorn och finns i *Telefonnummer*-fältet på raden för den här specifika datorn i filen **Systems**. Den är ofta samma sträng som systemnamnet. Denna typ av *UTP*-fält innehåller ett escape-tecken (**\D**), vilket ser till att innehållet i *Telefonnummer*-fältet inte kommer att tolkas som en förkortning i filen **Dialcodes**.

4. Modem anslutet till en nätverksomkopplare

Om ett automatiskt uppringande modem är kopplat till en nätverksomkopplare måste din dator anropa omkopplaren som i sin tur gör uppkopplingen mot det automatiska uppringande modemet. Denna typ av rad kräver två *uppringare-token-par*. Första paret anger uppkopplingen via nätverksomkopplaren till modemet och andra paret anger hur modemet sedan kopplar upp till det önskade systemet.

Uppringardelen för varje par (fälten fem och sju) hänvisar till rader i filen **Dialers** enligt nedan:

```
Systems: eagle Any ACU 1200 633251 ogin: nuucp ssword: Oak
Devices: ACU tty14 - 1200 develcon vent ventel
Dialers: develcon "" "" \pr\ps\c est:\007 \E\D\e \007
Dialers: ventel =&-% "" \r\p\r\c $ \T%%\r\c ONLINE!
```

I det första paret är **develcon** uppringaren och **vent** är det *token* som skickas till Develcon-omkopplaren för att tala om vilken enhet (ventel modem) som skall kopplas till ditt system. Denna *token* skall vara unik för varje LAN-omkopplare eftersom varje omkopplare kan sättas upp olika. När ventel-modemet väl är uppkopplat används det andra paret, där ventel är uppringaren och *token* hämtas från filen **Systems** (telefonnummer).

Protokoll för överföringen

Du kan definiera det protokoll som skall användas med varje enhet som ett tillägg till fältet *Enhetstyp*, antingen i **Systems** eller i **Devices**. Ett protokoll specificeras med formatet *Enhetstyp*, *Protokoll* (t.ex **STARLAN, e**). I de flesta fall är det inte nödvändigt eftersom du kan använda standardprotokoll. De tillgängliga protokollen är:

g Detta protokoll är långsammare och mer tillförlitligt än **e**. Det är bra för överföringar på telefonledningar med mycket störningar.

e Detta protokoll är snabbare än **g**, men det förutsätter en felfri överföring.

För tillförlitliga lokala nätverk kan du använda protokollet **e**. Här är ett exempel på hur du lägger till ett protokoll till en rad för en enhet:

```
STARLAN,e starlan - - TLIS \D
```

Detta betyder att för enheten **STARLAN** skall protokoll **e** användas.

13.5.4 Filen Dialers

Filen **Dialers** (**/usr/uucp/Dialers**) specificerar den inledande konversation som måste ske på linjen innan överföring av data kan göras. Denna konversation är vanligtvis en sekvens av ASCII strängar som sänds och mottages (förväntas). Den används ofta för att ringa ett telefonnummer med ett automatiskt uppringande modem. Se även filen **Sysfiles** om flera **Dialers**-filer ska användas.

Det femte fältet på raderna i filen **Devices** (uppringare) är antingen ett index (en modemtyp) in i filen **Dialers** eller en specifik uppringartyp (801 eller NET). Den rad i filen **Dialers** som börjar med detta index används för att styra uppringaren för motsvarande enhet i **Devices**. Exempel:

```
Devices: ACU tty11 - 1200 ventel
Dialers: ventel =&-% "" \r\p\r\c $ \T%%\r\c ONLINE!
```

Vidare används varje udda fält i filen **Devices**, med början på den sjunde positionen, som index in i filen **Dialers** om uppkopplingen skall ske i flera steg.

Varje rad i filen **Dialers** har följande format:

```
uppringare substitutioner förväntad sänd förväntad sänd ...
```

Fältet *Uppringare* motsvarar det femte och ytterligare udda numrerade fält i filen **Devices**. Fältet *substitutioner* innehåller en översättningssträng: det första tecknet av varje par av tecken ersätts av det andra tecknet i paret. Detta används vanligtvis för att översätta = och - till det som uppringaren använder för "vänta på ton" och "paus".

De återstående fälten *förväntad sänd* består av par av teckensträngar. Första strängen i varje par *(förväntad)* anger en sträng som skall tas emot och andra strängen *(sänd)* den sträng som sedan sänds till uppringaren.

Fältet *förväntad* kan bestå av flera strängar, som anger vad som skall göras om de förväntade tecknen inte kommer. Se beskrivningen av loginsekvensen i filen **Systems** för mer detaljer om detta.

Nedan finns ett antal rader med teckensträngar som bifogas hjälpprogrammen i filen Dialers.

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \
     \E\TP > 9\c OK
ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
rixon =&-% "" \r\r\d $ s9\c )-W\r\ds9\c-) s\c : \T\r\c $ \9\c \
                LINE
vadic =K-K "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon "" "" \pr\ps\c est:\007 \E\D\e \007
micom "" "" \s\c NAME? \D\r\c GO
direct
att2212c =+-, "" \r\c :--: ato12=y,T\T\r\c red
att4000 =,-, "" \033\r\r\c DEM: \033s0401\c \006 \
                \033s0901\c \006 \033s1001\c \006 \033s1102\c \
                \006 \033dT\T\r\c \006
att2224 =+-, "" \r\c :--: T\T\r\c red
nls "" "" NLPS:000:001:1\N\c
```

Tomma rader och rader som börjar med #, TAB eller mellanslag är kommentarrader.

Betydelsen av vissa escape-tecken (de som börjar med "\") vilka används i filen **Dialers** beskrivs nedan.

\p	Paus (ungefär 1/4 till 1/2 sekund)
\d	Fördröjning (ungefär 2 sekunder)
\D	Telefonnummer eller token utan **Dialcodes** översättning
\T	Telefonnummer eller token med **Dialcodes** översätning
\N	Lägg in ett NULL-tecken
\K	Lägg in ett BREAK-tecken
\E	Aktivera eko-kontroll (för långsamma enheter)
\e	Koppla bort eko-kontroll
\r	RETURN-tecken
\c	Inget ny-rad eller retur
\n	Sänd ny-rad
\nnn	Sänd tecken givet som oktal kod nnn

Ytterligare escape-tecken som kan användas finns beskrivna i avsnittet om filen **Systems**.

Exempel:

Penril-raden i filen **Dialers** exekveras på följande sätt.

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \
     \E\TP > 9\c OK
```

Först översätts argumentet för telefonnummer genom att alla = ersätts med ett **W** (vänta på ton) och alla - ersätts med **P** (paus). Handskakningen enligt återstående del av raden utförs på följande sätt:

""	Vänta på ingenting (d.v.s fortsätt till nästa fält).
\d	Fördröjning på 2 sekunder.
>	Vänta på ett >.
Q\c	Sänd ett Q utan efterföljande nyrad-tecken.
:	Vänta på ett :.
\d-	Dröj 2 sekunder, sänd ett minustecken.
>	Vänta på ett >.
s\p9\c	Sänd ett **s**, paus i 1/2 sekund, sänd en **9**, sänd inget avslutande nyrad-tecken.
)-W\p\r\ds\p9\c-)	Vänta på ett). Om det inte tas emot, behandla strängen mellan - tecknen enligt följande. Sänd ett **W**, paus, sänd ett vagnretur,

	fördröjning, sänd ett **s**, paus, sänd en **9**, utan nyrad-tecken och vänta på ett **)**.
y\c	Sänd ett **y** utan nyrad-tecken.
:	Vänta på ett **:**.
\E\TP	Aktivera eko-kontroll. (Från och med nu gäller att så fort ett tecken sänds iväg väntar vi tills tecknet kommer tillbaka innan vi gör något annat). Sedan skickas telefonnumret. **\T** betyder, ta bifogat telefonnummer lägg till översättningen från **Dialcodes** och konvertera tecken enligt fält två på denna rad. Sänd sedan ett **P**.
>	Vänta på ett **>**.
9\c	Sänd en **9** utan nyrad-tecken.
OK	Vänta på strängen **OK**.

13.5.5 Filen Dialcodes

Filen **Dialcodes** (**/usr/lib/uucp/Dialcodes**) innehåller de förkortningar som kan användas i fältet *telefonnummer* i filen **Systems**. Varje rad har följande format:

```
förkortning nummersekvens
```

Förkortning är den alfabetiska förkortning som används i fältet *telefonnummer* i filen **Systems** och *nummersekvens* är den sekvens av tecken som skickas till uppringaren då förkortningssträngen påträffas i filen **Systems**.

Exempel: Följande rad i **Dialcodes**

```
jt 9=847-
```

används tillsammans med ett nummer i *telefonnummer*-fältet i filen **Systems** t ex med utseendet **jt7867**. Då raden innehållande **jt7867** träffas på, sänds sekvensen 9=847-7867 till uppringaren, förutsatt att tecknet **\T** anges antingen i filen **Dialers** eller i uppringare-token-paret i filen **Devices**.

13.5.6 Filen Permissions

Filen **Permissions** (**/usr/lib/uucp/Permissions**) anger rättigheterna för fjärrdatorer vad gäller login, åtkomsträttigheter till filer och kommandoexekvering. Det finns parametrar som begränsar fjärrdatorns möjligheter att begära filer och dess möjlighet att ta emot filer köade av den lokala datorn. En annan parameter kan specificera de kommandon som en fjärrdator får utföra på en lokal dator.

Efter ändringar i filen **Permissions** används följande kommando som listar hur filen tolkas av systemet:

```
uucheck -v
```

Hur raderna är strukturerade

Varje logisk rad kan bestå av flera fysiska rader vilka avslutas med ett \ tecken om logiska raden fortsätter på nästa fysiska rad. Logiska rader består av en eller flera parametrar vilka separeras av blanktecken. Varje parameter består av ett namn/värde-par med följande format:

```
namn=värde
```

Observera att inga blanktecken är tillåtna i en parameter eller ett värde. Kommentarrader inleds med ett #-tecken och gäller till nästa nyradtecken. Tomma rader ignoreras (även i logiska rader som består av flera fysiska rader).

Det finns två typer av rader i filen Permissions

LOGNAME	Dessa rader innehåller en LOGNAME-parameter och specificerar de rättigheter som gäller då en fjärrdator loggar in på (anropar) din dator. Inloggningsnamnen styr då rättigheterna.
MACHINE	Dessa rader innehåller en MACHINE-parameter och specificerar de rättigheter som gäller fjärrdatorer då din dator loggar in (anropar) dem. Fjärrdatorns systemnamn styr då rättigheterna.

En enkel **Permissions**-fil kan innehålla enbart följande rad som ger standardtillstånd för alla datorer som loggar in som nuucp, samt standardtillstånd för din dator när den anropar andra datorer.

```
LOGNAME=nuucp
```

Detta är samma sak som om filen bestod av följande rader:

```
LOGNAME=nuucp \
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic \
SENDFILES=call REQUEST=no COMMANDS=rmail
```

Standardtillstånd för fjärrdatorer som loggar in på din dator som nuucp är då att:

- Mail kan sändas till din dator.
- Filer kan sändas till din dator, men bara till **/usr/spool/uucppublic**.

- Inga filer kan begäras från din dator när fjärrdatorn anropat och inga andra fjärrkommandon kan utföras i din dator.
- Eventuellt arbete som köats från din dator, för att sändas till fjärrdatorn, kan inte överföras när fjärrdatorn loggat in med enbart standardtillstånd.

Standardtillstånd för fjärrdatorer när din dator anropar dem (utan att någon MACHINE-rad finns för dem) är att:

- Efter att din dator avslutat sina överföringar kan fjärrdatorn enbart sända mail till din dator och sända filer till biblioteket **/usr/spool/uucppublic**.
- Din dator kan både sända och begära filer, dock bara i ovanstående allmänna bibliotek. Din dator kan sända *mail*, såväl som andra kommandon för exekvering i fjärrdatorn. Allt detta dock eventuellt begränsat av fjärrdatorns systemfiler.

Två vanliga tillägg till standardtillstånden är att alla fjärrdatorer som loggar in med uucp-systemet även skall kunna läsa filer från det allmänna biblioteket i din dator och att överföringar som användare på din dator har köat upp skall kunna utföras även när fjärrdatorn har kopplat upp förbindelsen. Detta kan erhållas med följande Permissions-fil. Tänk emellertid noga efter om detta ur säkerhetssynpunkt bör vara tillåtet, innan det införs utan ytterligare begränsningar. Du vet ju inte om den uppringande datorn verkligen är den han utger sig för att vara.

```
LOGNAME=nuucp REQUEST=yes SENDFILES=yes
```

Följande saker bör du ta hänsyn till då du via filen **Permissions** begränsar rättigheterna för fjärrdatorer:

- Alla login-namn som används för inloggning från fjärrdatorer, då de loggar in för UUCP-kommunikation, skall finnas på en och endast en LOGNAME-rad.
- Om en dator anropas vars namn inte finns på en MACHINE-rad kommer standardtillstånd (rättigheter/begränsningar) att gälla för dem enligt ovan. Värdet MACHINE=OTHER kan anges på en MACHINE-rad för att ge andra rättigheter/begränsningar än standard för alla fjärrdatorer som inte finns på andra MACHINE-rader.
- LOGNAME- och MACHINE-rader kan kombineras för att ange samma rättigheter för en fjärrdator, oberoende av vem som anropar. För bättre kontroll ska då varje fjärrdator ha ett eget inloggningsnamn (se sid. 13-34).

Parametrar

Varje rad i **Permissions** har en LOGNAME-parameter och/eller en MACHINE-parameter samt tillhörande tillståndsparametrar. Detta avsnitt beskriver varje parameter, hur de används samt deras standardvärden.

LOGNAME

Värdet för denna parameter är ett login-namn. Övriga parametrar enligt denna logiska rad gäller vid inloggning från alla fjärrdatorer, som loggar in med detta login-namn. Exempel:

```
LOGNAME=nuucp
```

nuucp är det namn som vanligen används.

MACHINE

Värdet för denna parameter består av en lista av systemnamn för de fjärrdatorer för vilka denna logiska rad gäller när din dator anropar dem. Systemnamnen i listan separeras med kolon (:). Exempel:

```
MACHINE=eagle:owl:hawk
```

Alternativt kan ordet OTHER anges istället för listan med systemnamn. Raden specificerar då parametervärden för datorer som din dator anropar, vilka inte finns på specifika MACHINE-rader. Detta kan vara intressant då det kommer anrop från många datorer där kommandouppsättningen ändras från gång till gång. Namnet "OTHER" som MACHINE-parameter kan t ex användas på följande sätt:

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/lbin/Photo:/usr/lbin/xp
```

REQUEST

Parametern REQUEST specificerar om en fjärrdator kan begära filöverföring från din dator.

```
REQUEST=yes
```

betyder att fjärrdatorn kan begära filöverföring från din dator.

```
REQUEST=no
```

betyder att fjärrdatorn inte kan begära filöverföring från din dator. Detta är standardvärdet. Det kommer att användas om inte parametern REQUEST specificeras. REQUEST kan finnas antingen på en LOGNAME-rad (fjärrdator anropar dig) eller en MACHINE-rad (du anropar fjärrdatorn). Ett tips angående säkerhet: När en fjärrdator anropar dig, kan du inte veta om fjärrdatorn är den han påstår sig vara, såvida du inte har ett unikt login och lösenord för den datorn.

SENDFILES

När en fjärrdator anropar din dator och avslutar sitt arbete kan den försöka att ta emot arbete som din datorn har köat till den. Tillvalet SENDFILES specificerar huruvida din dator kan skicka arbete köat till fjärrdatorn.

```
SENDFILES=yes
```

betyder att din dator kan skicka arbete som är köat till fjärrdatorn så länge fjärrdatorn är inloggad som ett av namnen i parametern

LOGNAME. Denna sträng är obligatorisk om din dator är i "passiv mod" i hänseende till fjärrdatorn.

```
SENDFILES=call
```

betyder att köade filer i din dator kommer att sändas endast då din dator anropar fjärrdatorn. Detta är standardvärdet för parametern SENDFILES. Denna parameter har endast betydelse för LOGNAME-rader. MACHINE-rader gäller ju då du anropar fjärrdatorn och om parametern används på en MACHINE-rad ignoreras den.

READ och WRITE

Dessa parametrar begränsar de olika delar i filsystemet som *uucico* får läsa från och skriva till. READ och WRITE kan användas på både MACHINE- och LOGNAME-rader. Notera att en ytterligare begränsning är att uucico (användare *uucp*, grupp *other*) dessutom måste ha vanliga filåtkomsttillstånd i de bibliotek/filer som anges.

Standard för både READ och WRITE är biblioteket **uucppublic** enligt följande strängar:

```
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic
```

Strängarna

```
READ=/ WRITE=/
```

ger åtkomsträttigheter till samtliga filer i din dator som användaren *uucp* har tillgång till. Detta är normalt **inte** lämpligt av säkerhetsskäl.

Värdet på dessa rader är en lista med sökvägar separerade med kolon. Parametern READ gäller för begäran av filer och WRITE för lagring av filer på din dator. Sökvägen för en fil som kommer in eller sänds iväg med fullständig sökväg angiven, måste överensstämma med något av de givna värdena. För att tilldela rättigheter att lagra filer i **/usr/news** lika väl som i det allmänna biblioteket kan exempelvis följande värden användas med parametern WRITE:

```
WRITE=/usr/spool/uucppublic:/usr/news
```

Notera! Om parametern READ eller WRITE används måste alla sökvägar anges, eftersom standardvärdena (uucppublic) inte automaitskt läggs till listan. Om t ex filnamnet **/usr/news** var det enda som specificerades i parametern WRITE skulle inte lagring av filer i det allmänna biblioteket tillåtas.

Du bör vara försiktig med de bibliotek som du gör tillgängliga för läsning och skrivning från fjärrsystem. Du vill t ex normalt inte att fjärrdatorer skall kunna skriva över filer i biblioteket /etc varför du inte bör tillåta skrivning till detta.

NOREAD och NOWRITE

Parametrarna NOREAD och NOWRITE specificerar undantag mot parametrarna READ och WRITE eller deras standardvärden. Exempel:

```
READ=/ NOREAD=/etc WRITE=/usr/spool/uucppublic
```

tillåter läsning av alla filer utom de i biblioteket **/etc** (och dess underbibliotek) och skrivning enbart till biblioteket **/usr/spool/uucp-**

public. NOWRITE fungerar på samma sätt som NOREAD. Parametrarna NOREAD och NOWRITE kan användas på både LOGNAME- och MACHINE-rader.

CALLBACK

Parametern CALLBACK används på LOGNAME-rader för att ange att inga transaktioner utförs innan det anropande systemet har motringts. Det finns två exempel på när du kan använda CALLBACK. Ur säkerhetssynpunkt kan identiteten hos en uppringande dator säkerställas genom motringning. Vid längre dataöverföringar kan du välja vilken dator som skall betala för uppkopplingen.

```
CALLBACK=yes
```

betyder att din dator måste motringa fjärrdatorn innan någon filöverföring kan äga rum.

Standardvärdet för parametern CALLBACK är

```
CALLBACK=no
```

Parametern CALLBACK används väldigt sällan. Observera att om två datorer har denna parameter satt för varandra kommer aldrig någon uppkoppling att göras.

COMMANDS

Tillvalet COMMANDS kan vara mycket farligt för säkerheten i ditt system. Använd det med stor försiktighet.

Programmet *uux* kommer att generera begäran om fjärrexekvering och köa dessa för överföring till fjärrdatorn. Filer och ett kommando skickas till målmaskinen för fjärrexekvering. Parametern COMMANDS kan användas på MACHINE-rader för att specificera de kommandon som en fjärrdator kan utföra på din dator. Observera att parametern COMMANDS inte behöver användas på LOGNAME-rader, detta då COMMANDS på en MACHINE-rad gäller både då vi anropar fjärrdatorn och då den anropar oss.

```
COMMANDS=rmail
```

indikerar de standardkommandon som en fjärrdator kan exekvera på din dator (*rmail* tar emot mail från fjärrdatorn). Om COMMANDS-parametern anges på en MACHINE-rad undertrycker denna standardkommandot. Exempelvis undertrycker raden

```
MACHINE=owl:raven:hawk:dove COMMANDS=rmail:rnews:lp
```

standardkommandot så att datorerna owl, raven, hawk och dove kan utföra kommandona *rmail*, *rnews* och *lp* på din dator.

Förutom kommandonamn, enligt ovan, kan fullständiga filnamn för kommandon anges. Exempelvis

```
COMMANDS=rmail:/usr/lbin/rnews:/usr/local/lp
```

specificerar att kommandot *rmail* använder standardsökvägen. Standardsökvägen för din dator kan t ex vara **/bin**, **/usr/bin** och **/usr/lbin**. Vare sig fjärrdatorn då specificerar */usr/lbin/rnews* eller bara *rnews* för kommandot som skall utföras, kommer

/usr/lbin/rnews att utföras oberoende av standardsökvägen. Likaså är */usr/local/lp* det *lp* kommando som kommer att utföras.

Om du inkluderar värdet ALL i listan betyder det att alla kommandon från de fjärrdatorer som anges på denna rad kommer att kunna utföras. Om du använder detta värde ger du fjärrdatorn full tillgång till din dator med rättigheter enligt användare *uucp*. **VAR FÖRSIKTIG**. Detta ger betydligt mer åtkomsträttigheter än vad vanliga användare har.

Strängen

```
COMMANDS=/usr/lbin/rnews:ALL:/usr/local/lp
```

illustrerar två saker: Värdet ALL kan finnas var som helst i strängen och att angivna sökvägar för *rnews* och *lp* kommer att användas (istället för standard) om det begärda kommandot inte innehåller det fullständiga filnamnet för *rnews* eller *lp*.

Parametern VALIDATE skall användas tillsammans med COMMANDS så fort "farliga" kommandon som t.ex *cat* och *uucp* specificerats med COMMANDS. Alla kommandon som läser och skriver kan vara "farliga" för den lokala säkerheten när de utförs av UUCP fjärrexekveringsdemon (*uuxqt*).

VALIDATE

Parametern VALIDATE används tillsamans med parametern COMMANDS vid specificering av kommandon som kan vara "farliga" för din dators säkerhet. Det används för att erhålla en viss grad av verifikation av identiteten på den som utför ett anrop. För att parametern VALIDATE ska kunna användas meningsfullt, krävs att priviligierade datorer använder unika loginnamn och lösenord för UUCP-transaktioner. En viktig aspekt för att kunna tillåta detta är att det login/lösenord som associeras med denna rad är skyddat. (Filen **Systems** på fjärrdatorn får alltså inte vara läsbar för andra än *uucp*). Om en utomstående får tag på denna information kan inte detta specifika VALIDATE anses som säkert. (VALIDATE skall mera ses som en utökad säkerhet ovanpå parametern COMMAND. Det är dock betydligt säkrare och skall alltid anges om farliga kommandon tillåts).

Var mycket restriktiv med att ge en annan maskin eget login och lösenord för UUCP användning med möjlighet att utföra alla kommandon. Att ge eget login och lösenord med åtkomsträttigheter och exekveringsmöjligheter är detsamma som att ge en användare eget login och lösenord direkt på din maskin. Om du är det minsta tveksam till någon person som är användare på den andra maskinen skall du inte ge eget login och lösenord till den maskinen.

En LOGNAME-rad med följande fält

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specificerar att om någon annan dator påstår sig vara eagle, owl, eller hawk vid inloggningen, måste den ha använt loginnamnet **uucpfriend** för att dess kommandon ska kunna utföras. Som du kan se är det mycket lätt att få tillgång till maskinen, eftersom loginnamn

och lösenord ligger i en vanlig textfil (**Systems**) i alla fjärrdatorer, dock förhoppningsvis läsbara endast för system- och uucp-administratörerna.

VALIDATE-parametern länkar samman MACHINE-raden (med systemnamn och en COMMANDS-parameter) till en LOGNAME-rad med ett priviligierat login. Denna länk behövs eftersom exekveringsdemonen *uuxqt* inte arbetar medan den andra datorn är inloggad. Det är en asynkron process och *uuxqt* vet annars inte under vilket logginnamn en begäran om exekvering har sänts.

Varje fjärrdator får ett eget "spool" bibliotek på din dator. Som namn på detta bibliotek används den andra datorns systemnamn. I dessa bibliotek har bara nätverksprogrammen skrivrättigheter. Exekveringsfilerna från de andra datorerna läggs i detta bibliotek när de överförts till din dator. När *uuxqt* demonen arbetar, kan den använda biblioteksnamnet för att hitta motsvarande MACHINE-rad i filen **Permissions** och hämta COMMANDS listan. Om inte datornamnet finns i **Permissions** filen, används standardvärdena istället.

Följande exempel visar relationen mellan MACHINE och LOGNAME rader:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/lbin/rnews \
READ=/  WRITE=/

LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/  WRITE=/
```

Värdet i COMMANDS-listan innebär att *rmail* och */usr/lbin/rnews* kan utföras av fjärranvändare.

På den första raden måste du förutsätta att när du vill anropa någon av de listade datorerna, anropar du verkligen antingen **eagle**, **owl**, eller **hawk**. Om någon fil läggs i något av spoolbiblioteken för **eagle**, **owl**, eller **hawk** har de lagts där av någon av dessa maskiner. Om någon annan maskin loggar in och påstår sig vara en av dessa maskiner kommer även dessa filer att läggas i dessa bibliotek. När parametern VALIDATE=uucpz angivits kommer en extra test att göras att inloggning verkligen skett med inloggningsnamnet uucpz (enligt exemplet) innan dessa kommandofiler accepteras.

Kombinerade MACHINE- och LOGNAME-rader

Det är möjligt att kombinera MACHINE- och LOGNAME-rader till en enda rad där de gemensamma parametrarna är desamma. T.ex kan de två raderna:

```
MACHINE=eagle:owl:hawk REQUEST=yes READ=/ WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes READ=/ WRITE=/
```

dela på parametrarna REQUEST, READ och WRITE. Dessa två rader kan slås samman till följande:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes READ=/ WRITE=/
```

13.5.7 Filen Poll

Filen **Poll** (**/usr/lib/uucp/Poll**) innehåller information för pollning av fjärrdatorer. Varje rad i filen **Poll** innehåller namnet på en fjärrdator som skall anropas följt av ett tab-tecken (mellanslag går inte bra) och slutligen de tidpunkter då datorn skall anropas. Raderna i filen **Poll** har följande format:

```
systemnamn tid ...
```

T ex gör raden

```
eagle      0 4 8 12 16 20
```

att datorn **eagle** pollas var fjärde timme.

Proceduren **uudemon.poll** skall startas regelbundet varje timme av *cron*. Denna procedur utför egentligen inte pollningen. Den lägger upp en arbetsfil för pollning (alltid med namnet C.fil) i spool-biblioteket vilken kommer att upptäckas av nätverksprogrammen (*uusched*), som startas av **uudemon.hour**. Därför skall **uudemon.poll** alltid startas lite före **uudemon.hour** med *cron*.

Rader som börjar med # i filen **Poll** är kommentarrader.

13.5.8 Filen Devconfig

Filen **Devconfig** (**/usr/lib/uucp/Devconfig**) används då din dator kommunicerar över ett STARLAN-nätverk eller något annat Stream-baserat kommunikationsmedia enligt AT&T Transport Interface (TI). Denna fil kan annars lämnas tom.

Raderna i filen **Devconfig** definierar de STREAMS-moduler som används för en specifik TI enhet. Raderna i filen **Devconfig** har då följande utseende:

```
service=x device=y push=z[:z ...]
```

där *x* kan vara *cu*, *uucico* eller båda två separerade med ett kolon, där **y** är namnet på ett TI-nätverk och motsvarar en rad i filen **Devices** och där *z* ersätts med namnet på STREAMS-modulerna i den ordning de skall inkluderas. Olika moduler och enheter kan definieras för *cu*- och *uucp*-tjänster.

Följande rader bör vanligtvis användas i filen:

```
service=cu      device=STARLAN push=ntty:tirdwr:ld0
service=uucico  device=STARLAN push=ntty:tirdwr:ld0
```

Detta exempel lägger ut **ntty**, **tirdwr** följt av **ld0**. Filen kan ändras med en vanlig texteditor.

13.5.9 Filen Sysfiles

Filen **/usr/lib/uucp/Sysfiles** låter dig tilldela olika filer att användas av *uucp* och *cu* som **Systems**-, **Devices**- och **Dialers**-filer. Här följer ett par exempel på då detta kan vara användbart.

- Du kanske vill ha olika **Systems**-filer så att begäran av logintjänster (från *cu*) kan göras till andra adresser än *uucp* använder.

- Du kanske vill ha olika **Dialers**-filer för att utnyttja olika handskakning för *cu* och *uucp*.
- Du kanske vill dela upp listorna i flera **Systems**-, **Dialers**- och **Devices**-filer. Speciellt filen **Systems** kan bli mycket stor och det kan då vara bra att dela upp den i flera mindre filer.

Raderna i filen **Sysfiles** har följande format:

```
service=w systems=x:x dialers=y:y devices=z:z
```

där *w* ersätts med *uucico*, *cu* eller båda två separerade av ett kolon. *x* är en eller flera filer att använda som **Systems**-fil, med filerna separerade av kolon. De läses i den ordning de räknas upp. *y* är en eller flera filer att använda som **Dialers**-fil. *z* är en eller flera filer att använda som **Devices**-fil. Alla filer anses vara i biblioteket **/usr/lib/uucp**, om inte ett fullständigt filnamn ges (inklusive sökväg). Du kan använda bakvänt snedstreck-vagnretur (**RETURN**) för att låta en logisk rad fortsätta över flera fysiska rader.

Här följer ett exempel på hur du kan använda en lokal **Systems**-fil tillsammans med den vanliga **Systems**-filen:

```
service=uucico:cu systems=Systems:Local_Systems
```

Med denna rad i **/usr/lib/uucp/Sysfiles** kommer både *uucico* och *cu* att först titta i filen **/usr/lib/uucp/Systems**. Om det system de försöker anropa inte har en rad i denna fil, eller om raderna i denna fil misslyckas, kommer de att titta i filen **/usr/lib/uucp/Local_Systems**.

När olika **Systems**-filer är definierade för *uucico*- och *cu*-tjänster kommer din dator att lagra två olika listor över system. Du kan skriva ut *uucico*-listan med kommandot *uuname* eller *cu*-listan med kommandot *uuname* **-c**.

13.5.10 Andra Nätverksfiler

Det finns ytterligare tre filer i **/usr/lib/uucp** som direkt inverkar på användningen av nätverkstjänsterna. I de flesta fall duger standardvärdena utan ändring. Filerna kan ändras med en vanlig texteditor.

Maxuuxqts — Denna fil definierar hur många *uuxqt* program som kan köras samtidigt, vanligen 1 eller 2.

Maxuuscheds — Denna fil definierar hur många *uusched* program som kan köras samtidigt, vanligen 1 eller 2.

remote.unknown — Denna fil är en shell-procedur som startas då en dator som inte finns i någon **Systems**-fil startar en konversation. Den kommer att logga kommunikationsförsöket och misslyckas med kommunikationen. Om du ändrar rättigheterna för denna fil så att den inte kan exekveras (*chmod 000 remote.unk-*

nown) kommer ditt system att acceptera alla konversationsbegäran.

13.5.11 Allmänna systemfiler

Förutom nätverksfilerna krävs även att övriga systemfiler för inloggning och regelbunden systemadministration är korrekt initierade.

Lokalt systemnamn

Nodnamnet för det lokala datorsystemet skall vara uppsatt och vara unikt eftersom detta namn används vid kommunikation med andra system. Dessutom bör namnet vara maximalt sex tecken långt eftersom nätverksprogrammet bara använder de första sex tecknen i namnet.

Nodnamnet kan visas med kommandot *uname -n* och kan definieras med kommandot */etc/mkcfig*, vilket uppdaterar operativsystemsfilen. *mkcfig*-kommandot beskrivs separat i denna handbok.

/etc/passwd

För att andra system ska kunna logga in till din dator måste namn och lösenord vara definierade i denna fil. Två eller fler namn ska vara definierade. Ett är namnet **uucp**, som är för UUCP-administration lokalt på systemet och som är ägare till alla uucp-program, bibliotek och filer. Dess loginbibliotek är **/usr/lib/uucp**.

För övriga login-namn som används från fjärrdatorer skall alltid */usr/lib/uucp/uucico* anges istället för shell. Loginbiblioteket skall vara **/usr/spool/uucppublic**, vilket skall ha alla åtkomsttillstånd satta (rwx) för alla i systemet. Dessa inloggningsnamn och lösenord återfinns i filen **Systems** i de fjärrsystem som ska kunna logga in. Genom att använda olika inloggningsnamn för olika fjärrdatorer (åtminstone grupper av fjärrdatorer) kan säkerheten ökas. Jämför VALIDATE i filen **Permissions**. Nedan är ett exempel på rader för detta i filen /etc/passwd:

```
uucp:h8HjG6jdsf:5:1:UUCP owner:/usr/lib/uucp:
nuucp:zaaOkjhNU:6:1:UUCPlogin1:/usr/spool/uucppublic:
                                    /usr/lib/uucp/uucico
xuucp:zaahkUjcv:7:1:UUCPlogin2:/usr/spool/uucppublic:
                                    /usr/lib/uucp/uucico
```

Notera att *uucico*-demonen används som shell vid inloggning från fjärrdatorerna.

/etc/group

Alla loginnamn som används för inloggning från fjärrdatorer bör tillhöra en och samma grupp, lämpligen samma grupp som uucp-administratören tillhör. Exempelvis bör med ovanstående exempel raden

```
others:1:uucp,nuucp,xuucp
```

finnas i filen /etc/group.

/etc/inittab

De tty-portar som används för kommunikation med fjärrdatorer kan sättas upp så att både inkommande och utgående anrop kan hanteras. Detta kräver att motsvarande rad i **/etc/inittab** innehåller ett *getty*-kommando med tillvalet -u, t ex enligt följande exempel:

```
t25:2:respawn:nice -16 /etc/getty -l -r -u 10 tty25 D2400
```

Därvid tvingas getty att använda låsfiler som hindrar att utgående och inkommande anrop krockar på samma port. Notera att det modem som används måste byglas så att det inte ger en DCD signal till datorn förrän någon verkligen försöker logga in. Annars kommer getty att alltid låsa porten. Se kommandot *getty* för detaljer.

crontab-filer

För automatisk nätverkshantering bör filer skapas för demonen *cron* så att nätverksprogrammen startas regelbundet. Särskilt gäller detta för att hantera omförsök och för att städa bort gamla temporärfiler. Det finns shell-procedurer i **/usr/lib/uucp** som kan startas av *cron*. En **crontab**-fil tillhörande användare *uucp* skapas t ex som följande. Notera att **uudemon.poll** alltid ska startas lite tidigare än **uudemon.hour**.

```
48  8,12,16  *  *  * /usr/lib/uucp/uudemon.admin
45  23       *  *  * /usr/lib/uucp/uudemon.cleanu
50  *        *  *  * /usr/lib/uucp/uudemon.poll
56  *        *  *  * /usr/lib/uucp/uudemon.hour
```

13.6. Arbetsfiler och loggfiler

Här beskrivs de administrativa arbetsfiler som används. Dessa filer skapas i spool-biblioteket **/usr/spool/uucp** eller underbibliotek till detta. Ett undantag är låsfilerna som ligger i **/usr/spool/locks**. Arbetsfilerna används för att låsa enheter, innehålla temporära data eller spara information om fjärröverföringar eller exekveringar.

TM. (temporära datafiler)

Dessa temporära datafiler skapas i spool-biblioteket (t.ex **/usr/-spool/uucp/NNN**) då en fil tas emot från en fjärrdator. Biblioteket NNN har samma namn som den fjärrdator som sänder filen. Namnen på de temporära datafilerna har följande format:

```
TM.pid.ddd
```

där *pid* är ett process-id och *ddd* är ett tresiffrigt sekvensnummer som startar på 0.

När hela filen tagits emot flyttas filen TM.*pid.ddd* till biblioteket som specificeras i filen C.*sysnxxxx* (beskrivs nedan) som orsakade överföringen. Om behandlingen avbryts onormalt kan filen TM.*pid.ddd* finnas kvar i NN-biblioteket. Dessa filer skall automatiskt tas bort av *uucleanup*.

LCK.. (låsfiler)

Låsfiler skapas i biblioteket **/usr/spool/locks** för varje enhet som används. Låsfiler förhindrar försök till samtidig användning av en linje eller uppringare opch hindrar kollision mellan inkommande och utgående anrop på en linje. Namnet på en låsfil har följande format:

```
LCK..str
```

där *str* antingen är namnet på en enhet eller namnet på en dator. Dessa filer kan finnas kvar i spool-biblioteket om kommunikationslänken bryts oavsiktligt (t ex om datorn stoppar felaktigt). Låsfilerna kommer att ignoreras (och tas bort av *uusched*) eftersom faderprocessen inte längre är aktiv. Låsfilen innehåller process-id för den process som skapade låset.

C. (arbetsfiler)

Arbetsfiler skapas i spool-biblioteket med fjärrdatorns namn då arbeten (filöverföring eller fjärrexekvering av kommandon) har köats till en fjärrdator. Namnet på arbetsfilerna har följande format:

```
C.sysNxxxx
```

där *sys* är namnet på fjärrdatorn, *N* är det ASCII-tecken som används för att ange prioriteten på arbetet och *xxxx* är det fyrsiffriga jobbsekvensnumret som tilldelats av UUCP. *uucico* bearbetar arbetsfilerna enligt prioritetsbokstaven i ordning A-Z, a-z. Tillvalet -g till uucp eller uux kan placera arbetet på valfri plats i kön. Arbetsfiler innehåller följande information i upp till åtta fält per rad:

- Typen. S=sänd, R=Begär från fjärrdatorn, X=sänd kommando.

• Fullständigt filnamn (sökväg) för den fil som skall sändas eller begäras.

• Fullständigt filnamn för destinationen eller ~användare/filnamn.

• Användarens loginnamn.

• Lista med parametrar och tillval för u*ucp*, *uuto*, *uux*.

• Namn på associerad datafil i spool-biblioteket. Om *uucp -c* eller *uuto -p* specificerats kommer ett dummynamn (**D.0**) att användas.

• Källfilens åtkomsttillstånd i filsystemet.

• Fjärranvändarens loginnamn om denna skall underrättas med *mail* då överföringen är klar.

D. (datafiler)

Datafiler skapas i spoolbiblioteket med fjärrdatorns namn när det på kommandoraden till t ex *uucp* specificeras (-C) att källfilen skall kopieras till spool-biblioteket eller när *mail* sänds. Namnet på datafilerna har följande format:

```
D.systxxxxyyy
```

där *syst* antingen är det lokala systemnamnet (för filer som ska sändas) eller fjärrdatorns systemnamn (för filer som mottagits). *syst* avkortas alltid till max 5 tecken. *xxxx* är det fyrsiffriga jobbsekvensnumret som tilldelats av *uucp*. Jobbsekvensnumret kan åtföljas av undersekvensnummer, *yyy*, som används då det är flera **D.** filer som skapats för en arbetsfil (**C.**).

X. (exekveringsfiler)

Exekveringsfiler skapas i spool-biblioteket med fjärrdatorns namn innan fjärrexekvering av ett kommandon. Namnet på exekveringsfilerna har följande format:

```
X.sysXxxxx
```

där *sys* är namnet på fjärrdatorn, *X* anger prioriteten på arbetet (vilket alltid är X), och *xxxx* är det fyrsiffrig sekvensnumret som tilldelats. Exekveringsfiler innehåller två eller flera rader med en kodbokstav i början av varje. Raderna U och C finns alltid. Övriga beror på vilken begäran som gjorts.

U Loginnamn och systemnamn för den som gör en begäran.

F Namn på de filer som erfordras för exekveringen.

I Standard input för kommandosträngen

O Datorsystem och filnamn som skall ta emot standard output från kommandot.

C Kommandosträngen.

N Eventuella rader för returstatusbegäran

R Fullständig mailadress för den som gör en begäran

Utöver dessa arbetsfiler finns standardbibliotek under **/usr/spool/uucp** för loggfiler av olika typer. Dessa biblioteksnamn börjar med en punkt och listas endast med optionen -a till *ls*-kommandot. De är:

.Admin	Administrativt bibliotek med bl a felmeddelanden (ASSERT errors) och statistik över dataöverföringar (xferstats).
.Corrupt	Felaktiga arbetsfiler.
.Log	Loggfiler för *uucico*, *uucp*, *uux* och *uuxqt*.
.Old	Gamla loggfiler, sammanslagna av *uucleanup*.
.Sequence	Sekvensnummerfiler för kommunikation med fjärrsystem.
.Status	Statusfiler för olika fjärrdatorer.
.Workspace	Temporära arbetsfiler.
.Xqtdir	Temporära filer för *uuxqt*.

Innehållet i loggfilerna (i /usr/spool/uucp/.Log) visas lättast med kommandot *uulog* enligt **AT&T User's Reference Manual**.

Felmeddelanden (ASSERT errors) och statusmeddelanden ges i klartext i filerna **.Admin/errors** samt i respektive statusfil med fjärrsystemens namn **.Status/systemnamn**. Förklaringar till vissa av dessa finns i ett senare delkapitel.

13.7. Standardbibliotek för användarfiler

Biblioteket **/usr/spool/uucppublic** är det enda biblioteket som är tillgängligt för fjärrdatorer för filöverföringar, om ingenting anges i filen **Permissions**.

I underbibliotek under **/usr/spool/uucppublic/receive** läggs alltid filer som sänts från en fjärrdator med kommandot *uuto*.

13.8. Exempel

13.8.1 Direktförbindelser

Detta avsnitt beskriver hur en direktförbindelse upprättas mellan två datorer.

Direktförbindelser är fördelaktig då:

- Det inte är möjligt att koppla ihop datorerna via ett lokalt nätverk (LAN).
- Två datorer överför stora mängder data regelbundet.
- Kabeln mellan två datorer inte är längre än några hundra meter.

Avståndet mellan två direktkopplade datorer beror på miljön som kabeln går igenom. Standarden för RS232-förbindelser är 15 meter eller mindre med en överföringshastighet på upp till 38400 bitar per sekund (bps). När kabellängden ökar kommer också störningarna att bli ett problem vilket betyder att överföringshastigheten måste minskas eller att korthållsmodem måste placeras ut i ändarna på kabel.

Använd inte mer än 300 meter kabel för att koppla ihop två datorer. Denna direktförbindelse skall kunna fungera upp till 9600 bps i en störningsfri omgivning.

Idealt skall system med en direktförbindelse mellan sig exekvera samma version av UNIX för att ha samma uppsättning funktioner. Men även om datorerna inte exekverar samma version UNIX skall detta inte påverka dig när du använder UUCP nätverkssystem.

Detta avsnitt beskriver de programfiler som måste modifieras på din dator för att kunna upprätta en direktförbindelse. Titta i dokumentationen för din dator om du upprättar en direktförbindelse till en dator som är av annan typ än din egen.

Följande filer måste uppdateras för att markera att du har en direktförbindelse, t ex med en nollmodemkabel:

- /usr/lib/uucp/Devices
- /usr/lib/uucp/Systems
- /etc/inittab och eventuellt /dev/ttyXX
- Eventuellt i /usr/lib/uucp/Permissions

Vi förutsätter i exemplen att redan definierade login-namn (nuucp) och lösenord användas. I annat fall måste nya skapas i systemet, t ex med *mkuser*, varvid även /etc/passwd och /etc/group ändras. Då måste dessa även införas i **Permissions** på LOGNAME-rader.

Skapa rader i filen Devices

Filen **Devices** innehåller information som rör placeringen (linjen) och överföringshastigheten för en kommunikationslänk. Rader kan läggas till i filen Devices med följande information:

- enhetstypen **Direct** som anger en direktkopplad förbindelse.
- portens namn (för **/dev/tty21** använder du **tty21**)
- överföringshastighet (9600 eller 19200)
- ordet **direct** istället för en uppringare

Exempel:

```
Devices: Direct tty21 - 9600 direct
```

Skapa rader i filen Systems

En rad måste skapas i filen **Systems** med följande information för den dator som associeras med direktförbindelsen:

- nodnamn (systemnamn) för den andra datorn
- typ av enhet som skall anropas (**Direct**)
- överföringshastighet (9600 eller 19200)
- port som används för länken (för **/dev/tty21** använder du **tty21**)
- login id som du använder för att logga in på systemet (*nuucp* eller något annat login namn som du satt upp)
- lösenordet för ovanstående login (det behövs även vid direkt anslutning, eftersom någon ju kan försöka logga in med samma login-id på en annan port)

Exempel:

```
Systems: dp10 Any Direct 9600 tty21 ogin: nuucp ssword: Oak
```

Notera att du måste lägga in en rad i båda datorernas **Systems**-fil även om bara den ena ska sköta uppkopplingen, eftersom båda måste känna till varandras systemnamn. Om bara en av datorerna skall vara aktiv och sköta uppkopplingen anges **Never** istället för **Any** i andra fältet på raden i filen **Systems** i den passiva datorn. I den aktiva datorn läggs en rad in i filen **Poll** för att aktivera pollning. Observera att ett tab-tecken måste användas efter systemnamnet i filen **Poll**:

```
dp00        8 12 16 20
```

För att undvika möjliga problem vid inloggning när du använder hög överföringshastighet kan du lägga till pauser (**\p**) mellan de tecken som sänds ut som login id och lösenord i sekvensen i filen **Systems**. Som exempel skulle du, istället för nuucp, kunna ange **n\pu\pu\pc\pp\p**.

Ändringar i filen /etc/inittab och eventuellt /dev/ttyXX

Det finns två versioner av hjälpprogrammen för nätverksanvändning. Olikheterna mellan dessa ses i filen **/etc/inittab**. Den nyaste versionen tillåter dubbelriktade loginmöjligheter genom att använda *getty -u*. Detta betyder att om två datorer är kopplade till varandra via en direktförbindelse kan båda begära kommunikation med den andre. Detta gäller inte om endast en av datorerna kan använda *getty -u*.

Om direktförbindelsen kopplar din dator till en dator med den nya versionen av programvaran skall filerna **/etc/inittab** i båda datorerna sättas upp att tillåta "dubbelriktad" trafik på de associerade ledningarna. Detta betyder att de ledningar som används måste exekvera *getty* i varje ände av ledningen. Detta möjliggör att båda datorerna kan begära kommunikation (anropa) den andre.

Om direktförbindelsen kopplar din dator med en dator som inte har den nya versionen av programvaran skall filerna **/etc/inittab** sättas upp olika på de två systemen. Filen **/etc/inittab** på de två datorerna skall sättas upp att antingen tillåta "inkommande" eller "utgående" trafik på ledningen. Om en dator tillåter inkommande trafik måste den andra datorn tillåta endast utgående trafik. *getty -u* kan inte användas på någon av datorerna i detta fall.

En dators rad i filen **/etc/inittab** skulle kunna vara att starta *getty* för inkommande trafik eller stoppa *getty* för utgående trafik. För att denna typ av länk skall fungera måste den ena datorn sättas upp för att polla den andra datorn. Om fjärrdatorn endast tillåter inkommande trafik måste du sätta upp din dator så att den pollar fjärrdatorn. Om fjärrdatorn endast tillåter utgående trafik måste den sättas upp att polla din dator.

Rader i filen **/etc/inittab** innehåller följande information:

- namnet på den port du vill modifiera (för **/dev/tty21** använder du **tty21**)
- riktning på trafiken på porten (dubbelriktat, utgående eller inkommande)
- överföringshastighet (9600 eller 19200)

Exempel med dubbelriktad kommunikation:

```
t21:2:respawn:nice -16 /etc/getty -l -r -u 10 tty21 9600
```

Notera att kommandot *'telinit q'* måste ges efteråt för att aktivera en ändring i **/etc/inittab**.

Den enhet som används (ex /dev/tty21) skall ägas av användare uucp och ha läs- och skrivrättigheter för ägaren. Ändring kan ske med kommandona *chown uucp /dev/tty21* och *chmod u+rw /dev/tty21*.

Eventuella tillägg i filen Permissions

Rättigheter och begränsningar i filen **Permissions** kan ofta vara gemensamma för flera fjärrdatorer. Om dessa avviker från standardtillstånden enligt befintliga rader i **Permissions**, kan dock den nya fjärrdatorn **systemnamn** behöva läggas till på en MACHINE-rad.

Antag att vi vill öka rättigheterna så att båda datorerna skall kunna använda varandras skrivare och begära filöverföring åt båda hållen, oberoende av vem som gör uppkopplingen. Då skall följande parametrar läggas in på både en LOGNAME-rad (anrop från den andra datorn) och en MACHINE-rad (anrop till den andra datorn). Vanligen kombineras då dessa enligt det fullständiga exemplet som följer i nästa stycke.

SENDFILES=yes	(tillåter att filer sänd om andra datorn anropar)
REQUEST=yes	(tillåter andra datorn att begära filer)
COMMANDS=rmail:lp	(lp får exekveras, förutom att mail kan tas emot)

Om andra bibliotek än standard skall kunna användas vid filöverföring skall de nämnas som READ- och WRITE-parametrar, förutom att de måste ha åtkomsttillstånd i filsystemen så att användare uucp kan läsa och skriva dem.

Observera! Om rättigheterna ökas enligt ovan bör inte samma loginnamn användas vid anrop från andra fjärrdatorer. Andra fjärrdatorer kan t ex få standardtillstånd vid anrop till din dator med namnet *xuucp*, om följande rad läggs till i filen **Permissions**.

```
LOGNAME=xuucp
```

Exempel med alla filer för en direktförbindelse:

I datorn dp00 kan finnas följande rader i filerna om vi förutsätter att båda systemen klarar av dubbelriktade portar och att loginnamnet nuucp endast används för dessa datorer. LOGNAME=xuucp är en separat rad i filen **Permissions** för att gälla andra datorer som loggar in.

```
Systems: dp10 Any Direct 9600 tty21 ogin: nuucp ssword: Oak
Devices: Direct tty21 - 9600 direct
inittab: t21:2:respawn:nice -16 /etc/getty -l -r -u 10 tty21 9600
tty21:          Ägare skall vara uucp
Permissions:    LOGNAME=nuucp MACHINE=dp10 \
                READ=/usr/spool/uucppublic:/usr/tmp \
                WRITE=/usr/spool/uucppublic:/usr/tmp \
                SENDFILES=yes REQUEST=yes COMMANDS=rmail:lp
                LOGNAME=xuucp
```

I datorn dp10 finns motsvarande rader, men med systemnamnet dp00 för att referera till den andra datorn i filerna **Systems** och **Permissions**. Kanske en annan ttyport används.

Exempel på kommandon som kan utföras på dp00:

```
uucp /usr/tmp/minfil dp10!/usr/spool/uucppublic
mail dp10!john <brevfil
uuto /usr/spool/uucppublic/minfil dp10!john
uupick -sdp10
```

13.8.2 Att använda automatuppringande modem

En förbindelse med automatuppringande modem upprättas på samma sätt som en direktförbindelse, men med något andra parametrar. Den viktigaste skillnaden är dock att säkerheten blir sämre eftersom vem som helst kan ringa in till ett modem, om loginnamnet och lösenordet är känt, lätt att gissa sig till eller läsa i fjärrdatorns **Systems**-fil.

Beskrivningen nedan är kortfattad. För gemensamma detaljer se exemplet med direktförbindelse.

De filer som berörs är:

- /usr/lib/uucp/Devices
- /usr/lib/uucp/Systems
- /usr/lib/uucp/Dialers
- /etc/inittab och eventuellt /dev/ttyXX
- /usr/lib/uucp/Permissions
- /usr/lib/uucp/Poll (om en dator är passiv)

En eller flera rader i Devices

Flera modem kan anslutas via olika portar och var och en skall ha en egen rad i **Devices**, där modemtypen anges som sista fält och enhetstypen är ACU. Antag att tty20 används med 2400 baud för ett modem som följer hayes standard och ett annat modem (ventel) finns på tty19.

Devices:

```
ACU tty20  - 2400 hayes
ACU tty19  - 2400 ventel
```

En rad i filen Systems

Fjärrdatorns namn måste finnas på en rad i filen **Systems**, vare sig den aktivt loggar in eller om den lokala datorn anropar den. Tidsbegränsningen **Never** kan anges i den dator som aldrig anropar den andre, medan all information anges för den aktiva datorn.

Systems i dp00:

```
dp10 Wk0800-1800 ACU 2400 081234567 ogin:--ogin: xuucp sword: Oak
```

Systems i dp10 (vanligen andra parametrar):

```
dp00 Wk0800-1800 ACU 2400 087654321 ogin:--ogin: xuucp sword: kaO
```

Alternativ Systems i dp10 (om den ska vara passiv):

```
dp00 Never ACU 2400 087654321 ogin:--ogin: xuucp sword: kaO
```

I detta fall anges inte vilken port som ska användas i **Systems**, eftersom en valfri port med modem kan väljas vid anropet. Istället anges parametrarna *enhetstyp* (ACU) och *hastighet* (2400) som valkriterier. I exemplet antas att fjärrdatorn dp10 skall anropas med 2400 baud och att den enbart får anropas under arbetstid kl 0800 - 1800 på vardagar-

na. Telefonnumret till dp10 är 08-1234567. Loginnamn för anrop är xuucp och lösenordet är Oak.

Uppringningsprotokoll i filen Dialers

Kontrollera att det finns rader i filen **Dialers** för de modem som används. Dessa kan se ut t ex enligt följande exempel. Den rad som medföljer systemet bör fungera, men eventuellt kan ytterligare fördröjningar (**\d**) behöva läggas till efter att telefonnumret sänts, om telefonnumret är långt.

Dialers:

```
ventel =&-% "" \r\p\r\c $ <K\T%%\r>\c ONLINE!
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\d\c CONNECT
```

Ändringar i /etc/inittab och eventuellt /dev/ttyXX

Här gäller samma som för direkta förbindelser. Den port som används (t ex tty20) måste ägas av *uucp* och korrekta *getty*-rader ska finnas i **/etc/inittab**, i exemplet nedan med dubbelriktade portar..

```
t19:2:respawn:nice -16 /etc/getty -l -r -u 10 tty19 2400
t20:2:respawn:nice -16 /etc/getty -l -r -u 10 tty20 2400
```

Om en port används enbart för utgående anrop skall den vara markerad **off** i **/etc/inittab** enligt:

```
t19:2:off:nice -16 /etc/getty -l -r -u 10 tty19 2400
```

Observera att det modem, som används till en dubbelriktad port, inte får aktivera DCD till datorn förrän en uppkoppling begärs utifrån. I annat fall kan utgående anrop hindras.

Tillägg i filen Permissions

För bästa säkerhet bör datorer som loggar in ha starkt begränsade rättigheter. Lämpligt är att använda standardtillstånden, varvid enbart biblioteket **/usr/spool/uucppublic** kan användas för att utbyta filer och *mail* kan sändas mellan datorerna. En fjärrdator som anropar kan då inte begära filer, men kan sända filer till **uucppublic**. Detta erhålls genom att enbart följande rad finns för det loginnamn som används vid inkommande anrop, i detta fall xuucp:

```
LOGNAME=xuucp
```

Eventuellt kan en MACHINE-parameter läggas till med ytterligare tillstånd, som kan göra att filer kan sändas från andra delar av filsystemet när din dator anropar den dator som anges på MACHINE-raden.

```
MACHINE=dp10 READ=/usr/spool/uucppublic:/usr/tmp:/usr/john
```

Om en dator är passiv och pollas av den andra måste rättigheterna vid inloggning ökas genom en annan LOGNAME-rad i den passiva datorn och en annan MACHINE-rad in den aktiva.

I dp10 som är passiv:

```
LOGNAME=xuucp SENDFILES=yes
```

I dp00 som är aktiv:

```
MACHINE=dp10 REQUEST=yes
```

Exempel på kommandon som kan utföras på dp00 (aktiv):

```
uucp /usr/tmp/minfil dp10!/usr/spool/uucppublic
mail dp10!john <brevfil
uuto /usr/spool/uucppublic/minfil dp10!john
uupick -sdp10
```

Motsvarande kan utföras på dp10, men om dp10 är passiv anges tillvalet -r till *uucp* eftersom ingen överföring ska ske förrän dp00 anropar dp10.

13.8.3 Förbindelse via Ethernet

Detta exempel visar hur en förbindelse via ett lokalt nätverk av typen Ethernet upprättas. Förbindelsen är dubbelriktad och kopplas upp med hjälp av kommandot *telnet systemnamn*, som utförs automatiskt.

De filer som berörs är:

- /usr/lib/uucp/Devices
- /usr/lib/uucp/Systems
- /usr/lib/uucp/Permissions

En rad i Devices

I detta fall anges ett lämpligt nätverksnamn i fältet *enhetstyp*. Samma namn (en valfri sträng t ex ETHNET) skall återfinnas i motsvarande fält i filen Systems. I andra fältet anges det program som skall användas för uppkopplingen (här *telnet*). Någon överföringshastighet anges inte (ett bindestreck används) och i sista fältet (uppringare) anges den speciella strängen NET följt av \D.

Devices:

```
ETHNET telnet  - - NET \D
```

En rad i Systems

Här anges den andra datorns systemnamn samt i femte fältet den nätverksadress som skall användas för uppkoppling. Nätverksadressen är vanligen samma som systemnamnet. Övriga fält är enligt tidigare exempel. Inlogging görs av programmen i UUCP-systemet på vanligt sätt efter att förbindelsen kopplats upp. Antag att uppkoppling ska ske mot systemet **alfa**.

Systems:

```
alfa Any ETHNET - alfa ogin: nuucp sword: Oak
```

Eventuella ändringar i filen Permissions

Om inte standardtillstånden skall användas, införs ändringar på samma sätt som tidigare beskrivits för dubbelriktade förbindelser.

13.8.4 Uppkoppling mot usenet

Usenet är ett allmänt nätverk med många system i hela världen för utbyte av nyheter och meddelanden via ett fritt programsystem (netnews). Ofta används UUCP-uppkopplingar för att vidarebefordra meddelanden mellan systemen. Meddelanden som sänds över nätet är indelade i olika grupper efter innehållet och för att minska mängden data som skall tas emot kan man begränsa abbonnemanget till vissa grupper.

Uppkoppling mot usenet kan ske som passiv nod (varvid en annan usenet-nod regelbundet anropar din dator) eller som aktiv nod (varvid din dator anropar en annan usenet-nod). Innan denna kommunikation upprättas skall man ta kontakt med den som ansvarar för usenet i landet för att beställa ett abonnemang och komma överens om vilket logginnamn och lösenord som skall används, vilken typ av förbindelse (modem, överföringshastighet m m), samt vilken usenet-nod man skall koppla upp sig mot. Systemnamnet på det egna systemet måste vara unikt i hela världen, bör kunna förknippas med ditt företag och får inte vara ett skyddat affärsnamn ("Trademark").

Förutom den uucp-förbindelse som beskrivs nedan, krävs även programvara för usenet. Denna är fri programvara som skall installeras på din dator och anpassas till ditt system, vilket inte behandlas här.

För att ansluta sig till usenet behöver man gå genom följande:

- Se till att ditt skivminne räcker till. Med många nyhetsgrupper kan mängden information bli stor.
- Kontakta den som ansvarar för usenet i landet för information.
- Sätt upp en förbindelse med en befintlig usenet-nod, så nära dig som möjligt. Helst bör det vara en dubbelriktad förbindelse. Detta kapitel beskriver hur denna förbindelse upprättas med uucp-systemet.
- Anpassa och installera programvaran för usenet.
- Sänd en registreringsbegäran över nätet, enligt ett visst format till gruppen *news.newsites* och till gruppen *comp.mail.lists*.

Uppkoppling som passiv nod

Uppkoppling som passiv nod sker via ett modem med auto-svar, varvid den andra usenet-noden loggar in till din dator och lämnar och hämtar meddelanden. Då skall den andra noden tillåtas att använda kommandona *rmail* och *news* på din dator.

De filer som berörs är:

- /usr/lib/uucp/Systems
- /etc/inittab och eventuellt /dev/ttyXX
- /etc/passwd och /etc/group med en ny användare

- /usr/lib/uucp/Permissions

I filen **Systems** skall en rad med den andra usenet-nodens systemnamn finnas, men med strängen **Never** i tidsfältet. Antag att systemnamnet är beta. då behövs enbart följande rad i **Systems**:

```
beta Never
```

Enheten som används (t ex **/dev/tty20**) skall vara kopplat till ett modem med auto-svar. En getty-process skall startas i filen **/etc/inittab** för inloggning och i filen **/etc/gettydefs** skall rätt överföringshastighet anges (vanligen 1200 eller 2400 baud).

En användare med ett eget inloggningsnamn och lösenord skapas i **/etc/passwd** och **/etc/group** med kommandot */etc/mkuser* för att ta emot anrop från den andra usenet-noden. Gruppnamnet bör vara samma som för användare uucp, inloggningsbiblioteket skall vara **/usr/spool/uucppublic** och istället för shell skall */usr/lib/uucp/uucico* anges.

Inittab:

```
t21:2:respawn:nice -16 /etc/getty -l -r -u 10 tty21 2400
```

/etc/passwd: (med exempelvis loginnamn useNeta)

```
useNeta:x8j)hHk:7:1:UUCP usenet:/usr/spool/uucppublic:
                                        /usr/lib/uucp/uucico
```

I filen **Permissions** skall denna nya användare vid inloggning ges tillstånd att exekvera *rmail* och *rnews*. Den skall enbart ha tillträde för att läsa och skriva i det bibliotek som usenet-systemet använder (vanligen **/usr/spool/news**) förutom det allmänna biblioteket **/usr/spool/uucppublic**. Begäran om överföringar gjorda av din dator skall kunna utföras när denna användare loggar in, men fjärrdatorn skall inte själv kunna begära att läsa filer från din dator. Exempel med loginnamn *useNeta*:

```
LOGNAME=useNeta SENDFILES=yes REQUEST=no \
READ=/usr/spool/uucppublic \
WRITE=/usr/spool/uucppublic:/usr/spool/news \
COMMANDS=rmail:rnews
```

Exempel på kommandon som kan ges på din dator är:

```
mail beta!diab!john <brevfil
uucp -r /usr/spool/uucppublic/minfil beta!diab!~/johnsfiles
```

I sista exemplet överförs filen till systemet med namnet diab och där till biblioteket **/usr/spool/uucppublic/johnsfiles**, förutsatt att datorn *beta* har en vidarekoppling till *diab*. uucp -r betyder att din dator inte försöker sända filen utan bara lägger den i kön.

Uppkoppling som aktiv nod

Uppkoppling som aktiv nod sker via ett modem med auto-uppringning, varvid inloggning sker på den andra usenet-noden, varefter nyheter och meddelanden hämtas därifrån. Antag att den andra usenet-noden har samma systemnamn som i exemplet ovan (beta).

De filer som berörs är:

- /usr/lib/uucp/Systems
- /usr/lib/uucp/Devices
- /usr/lib/uucp/Dialers
- /etc/inittab och eventuellt /dev/ttyXX
- /usr/lib/uucp/Permissions

I filen **Systems** skall en fullständig rad finnas för uppkoppling till den andra usenet-noden. Antag att systemnamnet för den är beta, och att uppringning sker med 2400 baud och med telefonnummer 123456 och med något loginnamn och lösenord.

```
beta Any ACU 2400 123456 ogin: uSnEtx ssword: xyz
```

En eller flera lämpliga enheter skall finnas definierade i **Devices**.

```
ACU tty21 - 2400 hayes
```

Filen Dialers skall innehålla motsvarande modemtyp.

```
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\d\c CONNECT
```

I filen /etc/inittab skall motsvarande enhet antingen vara passiv (off) för enbart utgående anrop eller vara dubbelriktad, som i följande exempel.

```
t21:2:respawn:nice -16 /etc/getty -l -r -u 10 tty21 2400
```

Ägaren till /dev/tty21 skall vara uucp.

I filen **Permissions** skall tillstånd finnas på en MACHINE-rad för fjärrdatorn att använda kommandot *rnews* förutom standardkommandot *rmail*. I övrigt skall behörigheten till filsystemet vara samma som vid passiv anslutning ovan. (Filer får alltid sändas från din dator när den kopplar upp en förbindelse, varför SENDFILES-parametern egentligen inte behövs).

```
MACHINE=beta SENDFILES=yes REQUEST=no \
READ=/usr/spool/uucppublic \
WRITE=/usr/spool/uucppublic:/usr/spool/news \
COMMANDS=rmail:rnews
```

13.9. Underhåll och systemsäkerhet

13.9.1 Systemsäkerhet

uucp-nätverket kan vara ett problem för säkerheten i ett system. Ett kommando som kan utföras av en användare på ett visst fjärrsystem eller en fil som kan kopieras av en användare på ett visst fjärrsystem kan betraktas som tillgängliga för samtliga användare i uucp-nätverket. Lokala användare kan bli släpphänta med att skydda känslig information.

Det finns möjligheter att förbättra säkerheten genom att förstärka och ersätta de normala funktionerna för filskydd. En lista över rekommenderade åtgärder följer nedan. Säkerhetsansvariga rekommenderas att noggrant läsa igenom listan och utnyttja de möjligheter som passar bäst.

Föslag på åtgärder för att förbättra säkerheten vilka kan implementeras av den systemansvarige då *uucp* används:

1. Filen **Systems** på alla åtkomliga datorsystem måste har åtkomsttillstånd som skyddar den mot obehörig åtkomst. Där finns nämligen både loginnamn och lösenord i klartext för inloggning till alla de system som din dator kan kontakta. Filen skall ha åtkomsttillstånd enligt `r--  ---  ---` (oktalt 400) och ägas av uucp.
2. Ägaren till *uucp*-programmen skall vara ett unikt administrativt login, vanligen uucp. Gör detta login olikt fjärranvändares login till uucp-systemet.
3. Programmen *uucp, uucico, uuname, uusched, uustat, uux, uuxqt och uulog* ägs av detta administrativa uucp-logini och biten "set-user-id" skall ettställas för dessa filer. Programmens åtkomsträttigheter skall vara endast execute för övriga användare.
4. Login till uucp-systemet från fjärrdatorer skall inte ge åtkomst till ett standardshell. Istället startas programmet *uucico* så att allt arbete måste utföras via *uucico*.
5. En kontroll av path skall göras på namnen för alla filer vilka skall skickas eller tas emot då ett *uucp* programkommando utförs. I filen **Permissions** begränsas åtkomsten med READ och WRITE till olika delar av filsystemet så mycket som möjligt och helst separat för olika fjärrdatorer.
6. Motringning (CALLBACK i **Permissions**) kan krävas för vissa användaridentiteter. (Motringningen kräver att båda systemen har modem vilka kan ringa och ta emot eller att de är förbundna med en fast ledning).
7. Listor över exekverbara kommandon bör kopplas både till loginnamn och systemnamn i filen **Permissions** med parametern COMMANDS kombinerat med VALIDATE.
8. Upprätta ett extra lösenord för uppringande enheter genom att skapa filerna **/etc/dialups** och **/etc/d_passwd**.

13.9.2 Underhåll och administration

Det mesta arbetet när det gäller underhåll och administration av uucp handlar om upprensning. *cron* används för att regelbundet starta shell-procedurer som förutom att sköta omsändningar m m, även rensar efter bästa förmåga bland arbetsfiler och temporärfiler och komprimerar logg-filerna (se kapitel 13.3.2). Direkt inblandning av administratören krävs för andra uppgifter.

Filer vilka behöver tillsyn:

/usr/spool/uucp/.Admin/errors:

Filen med uucp systemfelmeddelanden. Där rapporteras felaktiga moder på erfordrade filer eller bibliotek, saknade filer och misslyckade läs/skriv systemanrop på överföringskanalen. Felmeddelanden (ASSERT errors) listas senare i detta kapitel.

/usr/spool/uucp/.Admin/xferstats:

Filen med statistik för överföringar till/från fjärrdatorerna. Där visas bl a antal överförda tecken och den tid det tog.

/usr/spool/locks/LCK.xxxx

Låsfiler (notera att de är i locks-biblioteket). Det kan hända att låsfiler lämnas i locks-biblioteket om exekveringar avbryts. Låsfiler ignoreras efter 90 minuter. För att göra ett anrop innan dess kan man ta bort låsfilen med ett *rm*-kommando.

/usr/spool/uucp/.Log/uucico/systemnamn
/usr/spool/uucp/.Log/uucp/systemnamn
/usr/spool/uucp/.Log/uux/systemnamn
/usr/spool/uucp/.Log/uuxqt/systemnamn

Individuella logg-filer för olika kommandon i uucp-systemet och i olika filer för olika fjärrdatorer. Läses t ex med *uulog*.

/usr/spool/uucp/.Old

I detta bibliotek sparas komprimerade logg-filer av kommandot *uucleanup*.

/usr/spool/uucp/.Status/systemnamn

Statusfiler för olika fjärrdatorer med misslyckade (och lyckade) uppkopplingar. När ett anrop till ett system misslyckas tvingar status-filen fram en fördröjning innan det lokala systemet gör ett nytt anrop. *uucico* ger normalt upp efter 26 försök. Läses b a med *uustat -s systemnamn*.

Underhåll med hjälp av cron

cron, klock-demonen, läser poster i filen **/usr/spool/cron/crontabs/uucp** och utför shellprocedurer vid olika tidpunkter. En **crontabs**-fil kan se ut som i följande exempel:

```
48  8,12,16  *  *  * /usr/lib/uucp/uudemon.admin
45  23       *  *  * /usr/lib/uucp/uudemon.cleanu
50  *        *  *  * /usr/lib/uucp/uudemon.poll
56  *        *  *  * /usr/lib/uucp/uudemon.hour
```

uucp, *uuto* och *uux* köar arbete och försöker starta upp programmet *uucico* varje gång de utförs. *uucico* kan ofta inte omedelbart genomföra arbetet utan systemet måste regelbundet starta *uucico* för upprepade försök. Samma sak gäller för *uuxqt* som ska utföra de fjärrexekveringar som kommit in från fjärrdatorer. Detta görs av *uudemon.hour*. Om andra passiva fjärrdatorer skall anropas regelbundet (pollas) utförs detta av *uudemon.poll*.

Individuella loggfiler skapas i spool-biblioteket då programmet *uucp* exekveras. Filerna innehåller information om köade begäran, anrop till fjärrsystem, kommandoexekvering och resultat av filkopieringar. Dessa individuella filer skall regelbundet komprimeras och överföras till /usr/spool/uucp/.Old. *cron* är mycket användbar för hantering av rutinmässiga upprensningar av temporära filer, statusfiler och låsfiler vilka ackumuleras i spool-biblioteket om fel uppträder. Arbetsfiler och datafiler som inte kan exekveras på grund av felaktiga telefonnummer eller gammal login-information måste också rensas bort efter en tid. Shellproceduren *uucleanup*, som utförs varje timme sköter dessa arbetsuppgifter.

En administratör kanske vill ha statistik som visar trafiken mellan specifika system. *uudemon.admin* samlar ihop statistik och sänder *mail* till uucp-administratören.

13.10. Felsökning och statusmeddelanden

13.10.1 Vanliga problem

Följande avsnitt beskriver de vanligast förekommande problemen vilka uppträder på de system som flitigt använder *uucp*.

Slut på utrymme

Filsystemet som används för att köa inkommande eller utgående jobb kan bli utan utrymme och förhindra att jobb exekveras eller mottas från fjärrsystem. Oförmågan att ta emot jobb är det värsta av dessa två tillstånd. När filutrymme sedan blir tillgängligt kommer systemet att få mycket att göra med eftersläpningen i trafiken.

Dåliga modem

Modem på ingångarna orsakar ibland problem som gör det svårt att kontakta andra system eller ta emot filer. Dessa problem är oftast lätt identifierbara eftersom rader i loggfilerna vanligtvis pekar på den dåliga förbindelsen. Om en dålig förbindelse misstänks kan det vara bra att använda kommandot *cu* för att försöka kontakta ett annat system via den ledning som misstänks vara dålig.

Administrativa problem

Vissa uucp-nätverk har så många medlemmar att det blir problem att hålla reda på ändrade lösenord, ändrade telefonnummer eller ändrade inloggningar på fjärrsystem. Detta kan bli ett mycket kostsamt problem eftersom modem kommer att bli upptagna med att försöka kontakta ett system som inte går att kontakta.

Installationsproblem

Filerna **Devices, Systems, Dialers, Permissions** m fl kanske inte har installerats ordentligt. Käll- eller destinationsfilerna i **/usr/spool/uucp** och dess underbibliotek kanske inte har åtkomstprivilegierna 'alla får läsa' eller 'alla får skriva'. Filerna och kommandona i **/usr/lib/uucp, /usr/spool/uucp** och **/usr/spool/uucppublic** kanske inte ägs av **uucp**.

13.10.2 Test av förbindelser

Innan en förbindelse testa bör man förvissa sig om att det egna systemet känner till fjärrdatorn. Alla fjärrdatorer som är kända listas med kommandot *uuname*. Om filen Systems är delad i flera kan *uuname -c* behöva ges för att visa system som kan nås med *cu*-kommandot.

För att kunna verifiera att ett system i nätverket går att kontaktas kan *uucico*-demonen startas upp direkt från en användares terminal. För att enbart testa förbindelsen och att uucico på båda systemen kan kommunicera kan kommandot Uutry användas. Notera att första tecknet i kommandonamnet är en stor bokstav för ovanlighetens skull. För att testa förbindelsen till systemet dp00 ges följande:

```
/usr/lib/uucp/Uutry dp00
```

Därvid visas felsökningstexter under uppkoppling och handskakning. Kommandot avbryts sedan med BREAK eller DEL-tangenten.

För att t.ex verifiera att en begäran kan sändas till dp00 och övervaka hur förbindelsen sker, köas ett jobb för det systemet t ex enligt följande:

```
uucp -r file dp00!/usr/spool/uucppublic/dp20.tmp
```

Tillvalet **-r** köar jobbet men startar inte upp demonen för att utföra jobbet. Kommandot *uucico* kan sedan startas upp direkt:

```
/usr/lib/uucp/uucico -r 1 -x 4 -s dp20
```

Tillvalet **-r1** är nödvändigt för att indikera att demonen skall startas upp i mastermod (d.v.s. det anropande systemet). Tillvalet **-x4** specificerar nivån på felsökningstext som skall skrivas ut. Högre nivåer av felsökning kan skrivas ut (större än 4) men det kräver intern kännedom om *uucico*. Om flera jobb köas till fjärrsystemet är det inte möjligt att få *uucico* att skicka ett bestämt jobb först.

Innehållet i loggfilen (i biblioteket **/usr/spool/uucp/.Log**) skall även undersökas efter felindikationer som lämnas. Använd *uulog -s dp00*. Ofta kan problemen isoleras genom att undersöka posterna i statusfilen knutna till ett bestämt system (i biblioteket **/usr/spool/uucp/.Status**). Använd kommandot *uustat -s dp00*. *Filen* **/usr/spool/uucp/.Admin/errors** innehåller eventuella systemfelmeddelanden.

13.10.3 Status- och loggmeddelanden

Statusmeddelanden lagras i separata filer för varje fjärrdator i biblioteket **/usr/spool/uucp/.Status**. Nedan beskrivs de vanligaste meddelandena.

ASSERT ERROR

Ett allvarligt fel inträffade och ett felmeddelande har lagrats i filen **/usr/lib/spool/.Admin/errors**. Se nästa delkapitel för dessa meddelanden.

BAD LOGIN/MACHINE COMBINATION

Handskakningssekvensen vid inloggning misslyckades då en fjärrdator försökte logga in till din dator. Systemnamn och/eller logginnamn som användes av den anropande datorn är ej tillåtet enligt filen **Permissions**. Detta kan vara ett försök av den anropande datorn att anta en annan dators systemnamn. Uppringningen lyckades dock (om det förekom).

CALLBACK REQUIRED

Den anropade fjärrdatorn kräver att den ska motringa din dator.

CALLER SCRIPT FAILED

Detta är vanligen samma som DIAL FAILED. Dock kan det, om det uppstår ofta, bero på något fel i **Dialers**-filen. Använd *Uutry* för att testa om uppkopplingen fungerar.

CAN'T ACCESS DEVICE

Enheten som systemet försökte använda existerar inte eller har fel åtkomsttillstånd. Kontrollera motsvarande rader i filerna **Systems** och **Devices**.

CONVERSATION FAILED

Kommunikationen avbröts efter att ha startats upp korrekt. Detta beror oftast på att endera datorn stoppades, att programmet avbröts eller att linjen bröts.

DEVICE FAILED

Enheten kunde inte öppnas. Kontrollera åtkomsttillstånd och ägare till enheten.

DEVICE LOCKED

Enheten som användes för utgående anrop är för närvarande låst (**/usr/spool/locks/LCK..enhet**) och används av något annat program.

DIAL FAILED

Fjärrdatorn svarade aldrig vid uppkopplingen. Det kan t ex vara fel modemtyp angiven i **Devices**, fel på sekvensen i **Dialers**, fel telefonnummer eller att fjärrdatorns telefon är upptagen. Det kan även bero på att systemen inte kommer överens om vilken överföringshastighet som skall användas när modemet klarar olika hastigheter eller på att fjärrdatorns modem är bortkopplat.

LOGIN FAILED

Inloggningen till en fjärrdator misslyckades. Det kan vara felaktigt loginnamn/lösenord, fel telefonnummer, en mycket långsam fjärrdator eller fel i uppringningssekvensen eller inloggningssekvensen.

NO CALL (RETRY TIME NOT REACHED)

En uppringning misslyckades eller avbröts och det har inte gått tillräckligt lång tid för att ett nytt försök kan göras. Om du vill göra ett nytt försök omedelbart kan du ta bort statusfilen **/usr/spool/uucp/.Status/systemnamn** och sedan starta en *uucico*-process.

NO DEVICES AVAILABLE

Inga enheter som passar är lediga för utgående anrop. Kontrollera om det finns passande enheter i filen **Devices**. Se i filen **Systems** vilka krav som finns på enheten för uppkoppling till det angivna systemet.

OK

Detta meddelande lagras när allt går bra.

REMOTE DOES NOT KNOW ME

Den anropade fjärrdatorn har ingen rad med din dator i sin **Systems**-fil.

REMOTE HAS A LCK FILE FOR ME

Den anropade fjärrdatorn har en låsfil för ditt system i sitt spoolbibliotek (**/usr/spool/locks/LCK..dittsystemnamn**). Kanske fjärrdatorn just håller på att anropa din dator. Om fjärrdatorn har en äldre version av uucp-systemet kan låsfilen ha lämnats kvar sedan en tidigare uppkoppling avbrutits. Med den nya versionen har kommunikationsprocessen troligen låst sig om den inte verkligen redan kommunicerar med din dator.

REMOTE REJECT AFTER LOGIN

Det loginnamn/lösenord och egna systemnamn som din dator anger vid uppkoppling till fjärrdatorn stämmer inte med tillstånden i fjärrdatorns uucp-system (bl a **Permissions**-filen).

REMOTE REJECT, UNKNOWN MESSAGE

Fjärrdatorn vägrade att kommunicera med din dator av någon okänd orsak. Den kanske inte använder en standardversion av uucp-systemet.

STARTUP FAILED

Inloggningen lyckades men handskakningen mellan uucico-demonerna misslyckades. Förbindelsen kan också ha brutits efter inloggningen.

SUCCEEDED (call to system)

Uppkoppling och eventuella överföringar lyckades.

SYSTEM LOCKED

Ett samtal kan vara på gång då någonting (vanligtvis kommandot *uucp*) påbörjar ett nytt försök. Detta händer ofta då ett flertal uucp-kommandon skickas till samma system; det första startar en förbindelse och efterföljande får detta meddelande och avbryter. Om köfilerna kom på plats fort nog kommer den första uppringningen även att överföra informationen från de efterföljande kommandona.

SYSTEM NOT IN Systems

Någon användare på din dator angav ett systemnamn, som saknas i filen **Systems**.

TALKING

Kommunikation pågår mellan systemen.

WRONG MACHINE NAME

Den anropade fjärrdatorn har ett annat systemnamn än väntat. Din dator förväntar sig att det namn som anges först på raden i filen **Systems** är fjärrdatorns systemnamn, även om uppkopplingen sker med andra adress-strängar.

WRONG TIME TO CALL

Ett anrop gjordes till ett system vid en annan tidpunkt än de som specificerats i filen **Systems**.

13.10.4 Felmeddelanden i .Admin/errors

Felmeddelanden från uucp-systemet lagras i filen **/usr/spool/uucp/.Admin/errors**. Nedan beskrivs de vanligaste felmeddelandena.

BAD UID eller BAD LOGIN_UID

Användar-id saknas i **/etc/passwd**. Denna fil är felaktig eller ett allvarligt fel i filsystemet finns.

BAD LINE

Det finns en felaktig rad i filen **Devices**.

BAD SPEED

En felaktig överföringshastighet har angivits i fältet *Klass* i filerna **Devices** eller **Systems**.

CAN'T OPEN file

uucico kan inte öppna filen, open(), fopen().

CAN'T WRITE file

Det gick inte att skriva till filen, write(), fwrite() fprint() etc.

CAN'T READ file

Det gick inte att läsa från filen, read(), fgets() eller liknande.

CAN'T CREATE file

Filen kan inte skapas, create().

CAN'T ALLOCATE file

Filen kan inte dynamiskt allokeras.

CAN'T LOCK

Ett försök att skapa en låsfile (LCK) misslyckades.

CAN'T STAT

uucico kan inte läsa status för en fil eller ändra stty-parametrarna för en enhet, stat().

CAN'T CHMOD

Åtkomsttillstånden för en file kan inte ändras, chmod().

CAN'T LINK

uucico kan inte skapa en länk till en fil, link().

CAN'T CHDIR

Ett byte av bibliotek misslyckades, chdir().

CAN'T UNLINK

uucico kan inte ta bort en länk till en file eller ta bort filen helt.

CAN'T MOVE TO CORRUPTDIR

Ett försök att flytta felaktiga filer C.xxx, X.xxx till biblioteket **/usr/spool/uucp/.Corrupt** misslyckades. Biblioteket finns antingen inte eller har fel åtkomsttillstånd eller ägare.

CAN'T CLOSE

En fil eller enhet kan inte stängas, close().

CAN'T FORK

Ett försök att skapa en ny process och starta ett program misslyckades, fork() och exec(). Nuvarande process bör kunna avslutas korrekt men ett nytt försök kommer att göras senare (*uuxqt*). Ingen åtgärd krävs.

FILE EXISTS

En C.xxx eller D.xxx file skulle skapas, men filen finns redan. Detta visar på ett problem med sekvensnumreringen, eventuellt sekvensfilen. Det kan även vara ett programfel.

FSTAT FAILED IN EWRDATA

Det är något fel på nätverket när Ethernet används.

LINK ERROR

uuxqt kan inte länka en exekverbar fil till **/usr/spool/uucp/.Xqtdir**

NO UUCP SERVER

En uppkoppling via ett lokalt nätverk försöktes med TCP/IP, men det fanns inget server-program för UUCP.

PERMISSIONS FILE: BAD OPTION

Det finns en felaktig rad eller ett felaktigt värde i filen **Permissions**. Det bör av säkerhetsskäl rättas till omgående.

PKCGET READ

Fjärrdatorn kopplade troligen ner linjen. Ingen åtgärd krävs.

PKXSTART

Fjärrdatorn avbröt kommunikationen plötsligt. Ingen åtgärd krävs.

SYSLST OVERFLOW eller TOO MANY SAVED C FILES

En intern systemtabell räcker inte till. En alltför stor eller felaktig begäran har givits.

SYSTAT OPEN FAIL

Statusbiblioteket **/usr/spool/uucp/.Status** har fel åtkomsttillstånd eller det finns en fil med felaktiga åtkomsttillstånd i detta bibliotek.

TOO MANY LOCKS

Detta är ett internt fel i programvaran.

ULIMIT TOO SMALL

Parametern ulimit är för liten för processen. Filöverföringar kan gå fel, varför inga överföringar kommer att ske.

WRONG ROLE

Detta är ett internt programfel.

XMV ERROR

Något fel finns med några filer eller bibliotek. Troligen i spoolbiblioteket, eftersom åtkomsttillstånden till filer som ska skrivas antas testade innan denna process startade.

14

15

15. Yttre enheter

15. Yttre enheter

Alla yttre enheter som ska anslutas till systemet måste uppfylla två kriterier. Det första är att de är mekaniskt/elektriskt anslutna till systemet. Det andra är att de måste kopplas samman logiskt med operativsystemet via drivrutiner och enhetsnummer. Detta kapitel beskriver de logiska anslutningarna. De fysiska anslutningarna finns beskrivna i **Installation** för respektive enhet. För yttre enheter av SCSI-typ beror de logiska enhetsnumren på hur de fysiskt ansluts, vilket visas i tabellerna i kapitel 15.9.

I grundutförande levereras de flesta system med fyra seriella portar, ett skivminne av winchestertyp, en diskettenhet, eventuellt kassettbandstreamer samt möjlighet att ansluta ett flertal expansionsenheter.

Vilka enheter som kan anslutas beror på datortypen. De vanligaste enheterna är:

- Två interna terminalexpansionkort med vardera fyra extra terminalanslutningar. Dessa monteras i direkt anslutning till datorkortet.
- VME-kort kan sättas in i de interna VME-positionerna. Här kan t ex ett större antal extra terminalanslutningar erhållas via terminalkoncentratorkort och nätverk kan anslutas.
- I/O-kort ur DataBoardserien kan sättas in i den interna racken, för t ex parallella skrivare, nätverk eller andra speciellaändamål.
- Extra massminnesenheter ansluts till kontakter på baksidan. I vissa system kan ett antal extra enheter monteras internt.

Observera! Då yttre massminnesenheter ansluts till SCSI-kontakten på baksidan krävs att termineringarna görs korrekt. Koppla därför aldrig in eller ur massminnesenheter utan att följa anvisningarna i **Installation** för enheten.

15.1. Major och minor device

Alla yttre enheter som finns anslutna finns i biblioteket **/dev**. Där finns även logiska enheter för vissa interna drivrutiner, som t ex primärminnet. Drivrutinerna för alla logiska enheter finns internt i operativsystemet och aktiveras med kommandot */etc/mknod*.

Observera att dessa enhetsnummer och enhetsnamn inte gäller i laddningssystemet vid manuell uppstart av datorn.

Ett exempel på hur några yttre enheter kan vara representerade visas nedan, vilket är några rader ur det som visas med kommandot *ls*.

```
ls -l /dev.

crw--w--w- 1 bc   other   1,  1 Dec 17  1985 console
crw-rw-rw- 1 root root    9,  8 Sep 18 13:50 mf0
brw-rw-rw- 1 root root    9,  8 Sep 14 12:17 bmf0
crw-rw-rw- 1 root root    1,  5 Sep  1  1985 tty05
```

Det första tecknet visar av vilken typ den anslutna enheten är, de följande nio tecknen visar vilka som får använda enheten och hur. De två talen till vänster om datumet anger de primära och sekundära enhetsnumren. Sist på raden visas enhetsnamnet i biblioteket **/dev**.

Enhetstypen är normalt **c**, men kan även vara **b**. Enhetstypen **b** används enbart i speciella fall när man vill läsa eller skriva direkt på enheten med valfri blockstorlek istället för med den fysiska blockstorleken som är definierad för enheten. Anges **b** sker buffring i systemet även vid direkt läsning och skrivning till den fysiska enheten.

Major device — Major device eller Primärt enhetsnummer anger vilken drivrutin i systemet som skall användas.

Minor device — Minor device eller Sekundärt enhetsnummer är en parameter som dels kan peka ut en viss fysisk enhet och dels kan ange någon egenskap för enheten, t ex diskettformat.

Namn — Namn är det enhetsnamn i biblioteket **/dev** som används vid åtkomst till den fysiska enheten.

En tabell över de vanligaste primära och sekundära enhetsnumren finns i kapitel 15.9.

Notera att dessa enhetsnummer och beteckningar kan skilja sig mellan olika maskinvarukonfigurationer. Även mellan olika systemmodeller skiljer sig parametrarna för winchesterskivminnena, kassettstreamer och diskettenheterna.

15.2. Aktivering av drivrutiner till yttre enheter

Alla yttre enheter som har blivit mekaniskt anslutna används via biblioteket **/dev**. Men för att det ska vara möjligt måste drivrutinerna aktiveras i systemet. De drivrutiner som kan aktiveras är de som från början finns inlagda i operativsystemet. För att aktivera dessa använder man sig av kommandot *mknod*.

mknod används t ex på följande vis:

```
/etc/mknod /dev/tty08 c 1 8
```

I det här exemplet definierar vi en terminal med namnet tty08. Enheten arbetar med den fysiska blockstorleken, **c**, i detta fall teckenvis överföring. Drivrutinen för seriell överföring ska användas, **1**, och den ansluts till I/O-kanal **8**.

mknod skapar en specialfil i biblioteket **/dev**. Första argumentet är filens namn. Det andra argumentet är enhetstypen, **c** eller **b**. De sista argumenten är tal som anger primärt enhetsnummer (major device) och sekundärt enhetsnummer (minor device).

Tilldelningen av enhetsnummer är specifik för varje system enligt tabellen i kapitel 15.9.

Observera! Om extra enheter utöver grundsystemet är aktiverade i systemet bör systemadministratören inkludera dessa vid en totalbackup av systemet, för den händelse att en total omgenerering av det skivminne, som systemet har som root device, måste göras. Vid en installation av systemet från början aktiveras de i grundsystemet befintliga enheterna av installationsprogrammet *harddoit* på den medföljande mount-disketten. Om sekundära eller primära enhetsnummer har ändrats för sådana enheter, måste dessa därför ändras med *mknod* efter omgenerering av systemet.

15.3. Diskettenheter - /dev/mfN

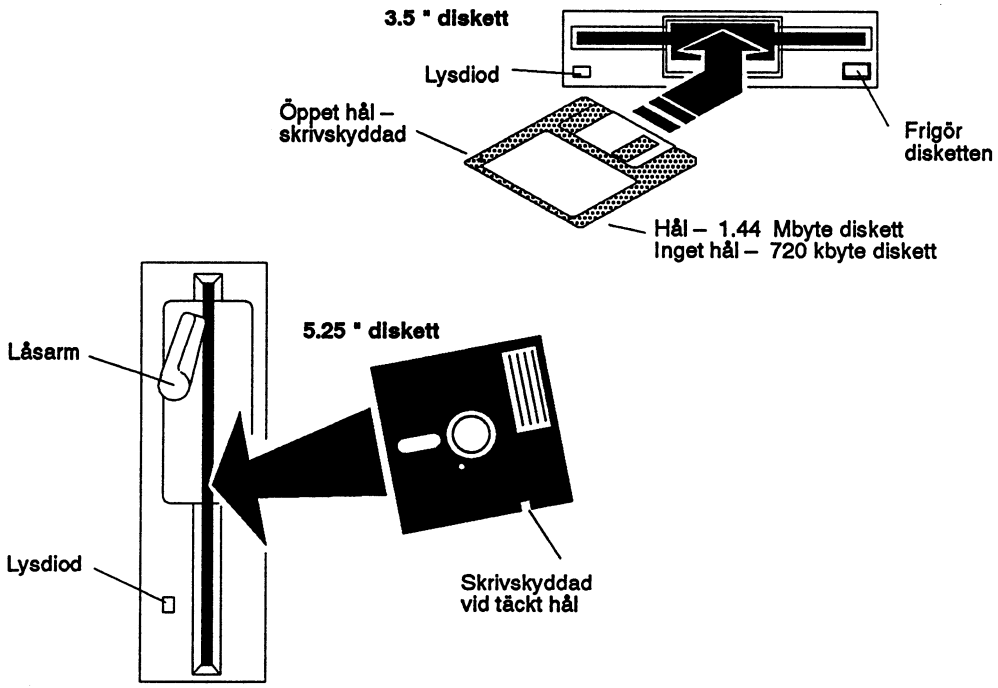
Den interna diskettenheten är en 3.5 tum eller 5 1/4 tum diskettenhet och kan vanligen hantera dubbelsidiga disketter med 80 spår. Normal lagringskapacitet är 720 kbyte, 1.2 Mbyte eller 1.44 Mbyte, men drivrutinen kan hantera även andra varianter om drivenheten klarar detta.

Diskettenheten nås genom enheten **/dev/mf0**. Eventuell ytterligare diskettenhet kallas normalt **/dev/mf1**. Vid behov skapas extra enhetsnamn med andra minornummer med kommandot **/etc/mknod**.

Den är ansluten logiskt till systemet via det primära enhetsnumret **9** (major device) och ett sekundärt enhetsnummer (minor device), som beror på datortypen enligt tabellerna i kapitel 15.9.

Vid systemstart från disketten via systemets bootstrap PROM anges ett enhetsnamn på formen **mf(X,0)**, där X är ett kanalnummer som beror på datortyp, diskett-typ samt vilken fysisk enhet som används.

För bästa resultat och största säkerhet i datalagringen rekommenderas dubbelsidiga disketter avsedda för lagring med dubbel packningstäthet på 80 spår. För 1.2 Mbyte resp. 1.44 Mbyte lagringskapacitet krävs att "high-density"-disketter används.



Lysdioden lyser när datorn skriver/läser på disketten. Under tiden får inte disketten tas ut.

Automatisk avkänning av typen av diskett (sense)

Normalt används ett standardvärde (**sense**) på sekundära enhetsnumret som gör att systemet känner av vilken diskett-typ som används. Automatisk avkänning kan dock inte användas vid systemstart via BOOT-PROM eller i laddningsprogrammet.

Drivrutinen kan fås att visa de funna diskettparametrarna, genom att ett annat standardvärde (**display**) används. Parametrarna visas då på konsolterminalen vid läsning från disketten.

Exempel: En enhet /dev/mf00 skapas med enhetsnummer för "display" XX, beroende på datortyp. Läsning kan t ex ske med kommandot *dd* för att vara oberoende av eventuellt filsystem.

```
/etc/mknod /dev/mf00 c 9 XX          (XX beror av datortyp)
dd if=/dev/mf00 of=/dev/null bs=1k
```

Med avkänning (sense) kan även en 5 1/4 tums 40-spårs diskett läsas (dock inte skrivas) i en diskettenhet som har 80 spår. För att skriva till en 40-spårs diskett, måste en enkelspårs drivenhet användas.

Diskettformat kodat i sekundära enhetsnumret (minor)

Sekundära enhetsnumret kan kodas för att ange ett visst diskettformat istället för att låta systemet känna av diskettens format. De minornummer som används är något olika för olika datorsystem. Nya enhetsnamn skapas då med önskade minornummer.

Speciella diskettformat

Disketter med D-NIX filsystem initieras med */etc/mkfs* och hanteras direkt av operativsystemet.

Disketter med MS-DOS-format eller format enligt SBC-DOS/ABC-DOS kan hanteras med egna filhanterare, som kan köpas till systemet. Då kan de flesta vanliga kommandon användas för att läsa/skriva data, lista filer, skapa och ta bort filer.

På disketter utan filsystem kan filer lagras med kommandot *cpio* eller med *tar*, vilka är standard i alla Unix-system. I vissa fall måste dock vissa anpassningar göras, som t ex byte-swapping.

I alla dessa fall måste disketten vara formatterad. Om utbyte av disketter mellan olika datorer skall ske måste formatteringen överensstämma.

15.4. Skivminnesenheter (SCSI) - /dev/siN

Det första fasta skivminnet har enhetsnamnet **/dev/si0**.

Denna enhet är aktiverad vid leverans och används normalt för systemets root-filsystem.

Enheter av SCSI-typ har det primära enhetsnumret **5** (major device). Det sekundära enhetsnumret (minor device) varierar mellan datormodellerna eftersom olika SCSI-portar används.

Vid systemstart med uttryckligt angivande av enhet via systemets bootstrap PROM anges kanalnummer genom enhetsnamnet **si(X,0)**, där **X** är det sekundära enhetsnumret. Se kapitlet 15.9.

Extra skivminnesenheter

Vissa system har plats internt för extra skivminnen. De flesta system är försedda med anslutningskontakt för att ansluta extra yttre skivminnen av SCSI-typ. I vissa datorer krävs att en SCSI-option monteras i datorn för att yttre skivminnen skall kunna anslutas.

Det första yttre skivminnet bör ha det logiska enhetsnamnet **/dev/si2** och de följande **/dev/si4**, **/dev/si6** o s v med jämna nummer.

Även extra skivminnen skall ha det primära enhetsnumret **5** (major device). Det sekundära enhetsnumret (minor device) beror av vilken datormodell som används, till vilken SCSI-port skivminnet ansluts samt eventuella andra enheter anslutna till samma port.

Vid systemstart från yttre winchesterenheten via systemets bootstrap PROM anges kanalnummer genom enhetsnamnet **si(X,0)**, där X är det sekundära enhetsnumret.

I kapitlet 15.9 finns en tabell och en beskrivning av hur enhetsnumret beräknas samt vilka SCSI-portar enheterna ansluts till i de olika datorerna.

Yttre skivminnen ansluts mekaniskt till de olika datormodellerna enligt installationsanvisningarna för skivminnen. Koppla inte in eller ur en enhet utan att stänga av systemet och följa installationsanvisningarna.

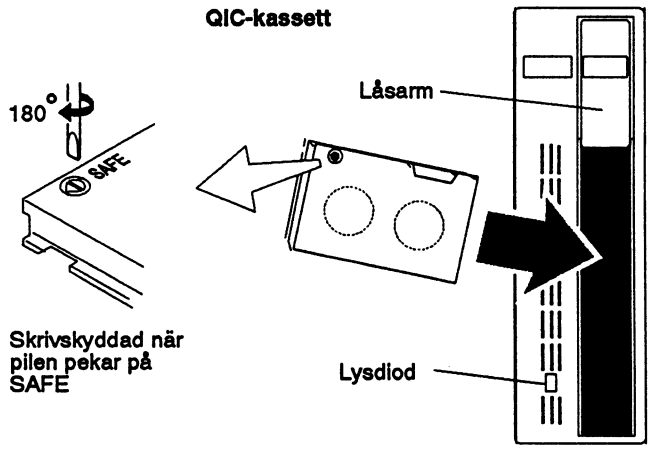
15.5. 1/4 tums kassettenhet (QIC) - /dev/stN

En 1/4 tums kassettenhet finns vanligen monterad i frontpanelen och har enhetsnamnet **/dev/st0**. För vissa modeller är kassettenheten en option. I de fall en extra kassettenhet ansluts till den yttre kontakten används namnet **/dev/st1**.

Drivenheten är enligt QIC-24 i äldre datorsystem och enligt QIC-150 i nyare modeller. För kompatibilitet kan 60 MB kassetter läsas även i de nya enheterna. Drivenheten känner själv av vilken typ av kassett som används. Följande bandkassetter skall användas.

Typ av enhet, format	Kassett-typ	Kapacitet	
150 MB, QIC-150:	DC6150	150 Mbyte	
	DC6320	150 Mbyte	
	DC600A	120 Mbyte	(Enligt QIC-120)
	DC600A	60 Mbyte	(Endast läsning)
60 MB, QIC-24:	DC600A	60 Mbyte	
	DC6150	60 Mbyte	

Använd **inte** band med utökad längd (extended length tapes) eftersom de sliter på huvudet i drivenheten.



Ta inte ut kassetten så länge lysdioden lyser.

Låsarmen dras in över kassetten när kassetten satts in. När låsarmen förs undan skjuts kassetten automatiskt ut en bit.

Skrivhuvudet rengörs regelbundet, beroende på hur mycket enheten används, med en rengöringskassett eller manuellt med Isopropanol-A på en bommullspinne.

Drivrutinen för kassettenheten har det primära enhetsnumret **5** (major device), eftersom den styrs av samma drivrutin som winchesterenheten. Sekundära enhetsnumret (minor device) varierar för datormodell- erna, men har alltid bit 7 satt för att markera att enheten är en sekventiell enhet. Se kapitel 15.9.

En speciell rutin finns för kassettenheten om man vill att bandet inte ska återspolas efter läsning. Enhetsnamnet är då **/dev/st0n** och det sekundära enhetsnumret (minor device) har bit 0 satt (udda). **/dev/st0n** skapas inte av *harddoit* från BOOT-disketten, utan måste aktiveras av användaren om den behövs.

Observera! Skrivning till kassetten kan bara ske från bandets början. Därför kan inte tillvalen **r** och **u** användas i *tar* och bandet måste alltid återspolas före skrivning. Läsning kan däremot göras från valfri position.

Kassettstreamer specialformat

Med den interna kassettsenheten kan exempelvis kassetter utbytas med NCR-datorer. Då måste emellertid noteras att data lagras på kassetterna i NCR med byte-swap. Vid hantering av kassetter i format enligt kommandot *cpio* används då optionen **s** i D-NIX för att erhålla byte-swap. Vid läsning och skrivning med kommandot *tar* måste data filtreras genom kommandot *dd* för att erhålla byte-swap enligt exemplen nedan. Tecknet | är pipe-symbolen i shell.

Läsning av NCR-kassett med *tar*:

```
dd if=/dev/st0 conv=swab | tar xvf -
```

Skrivning av biblioteket mydir till NCR-kassett med *tar*:

```
tar cvfbk - 200 60300 mydir | dd conv=swab of=/dev/st0
```

15.6. Serieanslutningar - /dev/ttyNN

15.6.1 Terminalkoncentrator VME/TC

Drivrutiner finns för att expandera systemet med extra terminalanslutningar via VME-kort med vardera 10 anslutningar för de datormodeller där VME-kort kan installeras.

Drivrutinen har det primära enhetsnumret **11** (major device). De sekundära enhetnumren (minor device) är enligt tabellen i slutet av kapitlet.

Dessa anslutningar kan användas för terminaler och skrivare eller för kommunikationslinjer. Asynkron kommunikation används. Terminaler ansluts via extra kontakter av typen DA15P. Två kablar kopplas mellan varje VME-kort och kontaktplåten. Terminalkontakterna är markerade med sina enhetsnamn. Stiftplaceringen kan ses i **Förberedelser för Installation** i systempärmen. Närmare detaljer om inkopplingen beskrivs i **Installation** av terminalkoncentratorkortet.

Enhetsnamnen är normalt **/dev/tty*nn***, där *nn*=12 och uppåt. För varje kort finns en speciell enhet **/dev/tc*n***, n=0, 1, 2, ..., genom vilken styrinformation och status för kortet erhålles. Dessa enheter skapas automatiskt när programvaran */etc/tc* för terminalkoncentratorn aktiveras vid systemstart med följande kommando i filen **/etc/inittab**:

```
tc:s123456:respawn:/etc/tc # tc demon
```

För terminalportar där inloggning skall kunna ske, skall motsvarande rader med */etc/getty*-kommandon finnas i filen **/etc/inittab** enligt samma princip som beskrivs i kapitel 8.

15.6.2 Extra interna V.24 serieanslutningar

De flesta system kan förses med upp till tolv seriella V.24 (RS232S) kanaler om två interna terminalexpansionkort monteras i datorn tillsammans med CPU-kortet. Varje kortenhet som ansluts till datorn är försedd med fyra V.24 anslutningar.

Ytterligare expansion sker genom VME-kort.

Dessa anslutningar kan användas för terminaler och skrivare eller för anslutning av kommunikationslinjer med asynkrona protokoll.

CPU-kortet har fyra portar:

Enhetsnamn	Primärt enhetsnummer	Sekundärt enhetsnummer
/dev/console	1	1
/dev/lp	1	0
/dev/tty02	1	2
/dev/tty03	1	3

Det första extra kortet har:

Enhetsnamn	Primärt enhetsnummer	Sekundärt enhetsnummer
/dev/tty04	1	4
/dev/tty05	1	5
/dev/tty06	1	6
/dev/tty07	1	7

Det andra extra kortet har:

Enhetsnamn	Primärt enhetsnummer	Sekundärt enhetsnummer
/dev/tty08	1	8
/dev/tty09	1	9
/dev/tty10	1	10
/dev/tty11	1	11

15.7. Expansion med VME- och DataBoard-kort

15.7.1 Anslutning av VME-kort

Beroende på vilken modell det gäller kan upp till sex VME-kort anslutas i datorenheten och i vissa fall fler i yttre expansionsrack. VME-korten ska ha dubbel Europakortsstorlek och en kontakt för 16 data- och 24 adressledningar. I vissa datorer kan 32-bitars VME-kort med två kontakter användas.

Observera! Detta gäller inte de datortyper som saknar intern VME-rack.

Anslutningen av VME-kort sker med spänningen frånslagen och korten sättes in med komponentsidan åt höger eller uppåt.

Om inte alla positioner används sätts VME-korten i de vänstra eller nedersta positionerna, första kortet i VME-1, andra i VME-2 etc.. Detta förutsätter att alla byglingar på bakplanet är öppna, vilket är normalt.

- Om någon VME-position är tom medan kort måste sättas i en annan position med högre nummer, sluts byglingarna på bakplanet mellan den tomma positionen och nästa VME-position.

Se kapitel 15.9 för närmare detaljer om vad som gäller för respektive datormodell.

15.7.2 Generella drivrutiner för VME-bussen

/dev/VMEdrv är en drivrutin som kan användas för buffrad avbrottstyrd åtkomst av VME-bussen. Rutinen kan användas för att sända kommandon och data till slavenheter på VME-kort och hantera avbrottssignaler på VME-bussen.

Drivrutinen för **/dev/VMEdrv** har det primära enhetsnumret **12** (major device) och det sekundära enhetsnumret **0** (minor device).

Med utvecklingspaketet följer ett bibliotek med **C-rutiner** som kan användas vid egen programmering. Med dessa kan hanterare utvecklas för snabb åtkomst av VME-bussen.

För detaljer hänvisas till **D-NIX Programmers Manual**.

15.7.3 Anslutningar av DataBoardkort

DataBoardkort kan anslutas till expansionskontakterna internt i datorenheten och i vissa fall i yttre expansionsrack. Anslutningen sker med spänningen frånslagen och korten sättes in med komponentsidan åt höger eller uppåt.

Observera! Detta gäller inte de datortyper som saknar intern DataBoard-rack.

Alla kontakter kan användas för DataBoard I/O-kort men endast en av kontakterna har extra signaler för anslutning av expansionskort till yttre expansionsrack. I positionen för expansionskort finns extra signaler för hantering av externa interrupt och kort i expansionsracken.

DataBoard-kort som inte använder avbrottssignaler kan placeras i valfri position, men vissa begränsningar gäller övriga kort för några datormodeller enligt kapitel 15.9.

15.7.4 DataBoard generell I/O

/dev/DBinoutb är en drivrutin för direkt åtkomst till DataBoard-kort. Vid skrivning och läsning via /dev/DBinoutb, ges kortvalsadressen och kommandostroben som en kombinerad enhetsadress.

Drivrutinen för /dev/DBinoutb har det primära enhetsnumret **3** (major device), och det sekundära enhetsnumret **8** (minor device).

/dev/DBIdrv är en drivrutin som kan användas för buffrad avbrottsstyrd åtkomst till DataBoard-kort. Rutinen kan användas för att skriva kommandon och utgående data samt hantera avbrottssignaler från DataBoard-korten.

/dev/DBIdrv har primärt enhetsnummer **13** (major device) och sekundärt enhetsnummer **0** (minor device).

Med utvecklingspaketet följer ett bibliotek med **C-rutiner** som kan användas vid egen programmering. Med dessa kan hanterare utvecklas med snabb åtkomst till DataBoard-korten.

/dev/vdbm är en drivrutin som finns i vissa datorer. Den fungerar som en virtuell maskin för hantering av DataBoard. Användaren sänder kompilerade program till drivrutinen, vilka utförs av systemet och kommunicerar med användarprocesser via FIFOn. Det primära enhetsnumret (major) varierar mellan datorerna men det sekundära enhetsnumret (minor) är normalt lika med kortvalskoden på det DataBoard-kort som används. Flera olika enheter definieras därför med varierande namn.

För detaljer hänvisas till **D-NIX Programmers Manual**.

15.7.5 Centronics/SP1 gränssnitt - sp1

I systemet finns en drivrutin som hanterar DataBoard gränssnittskort med parallell dataöverföring för anslutning av skrivare enligt Centronics samt för sänding och mottagning av data enligt SP1. De kort som används är 4001 Centronicskort, 4015 SP1 sändarkort och 4016 SP1 mottagarkort.

Om två datorer kopplas samman över ett SP1 gränssnitt sker överföringen kontinuerligt asynkront i 8-bitars rå mod utan behov av någon styrinformation blandad med data.

När tecken sänds till en skrivare bör inte rå mod användas. Drivrutinen kan då arbeta i textmod med radvis överföring där varje block avslutas med ASCII-tecknet för ny rad (LF). Då sker även vissa konverteringar, som t ex expansion av tabuleringstecken och konvertering av LF till CR/LF eftersom CR (carriage return) inte normalt ingår i radslut i D-NIX och Unix system. Vid mottagning i textmod sker motsvarande konvertering tillbaka.

Drivrutinen har primära enhetsnumret **7** (major device) och det sekundära enhetsnumret (minor device) beror av den kortadress (0-63) som byglats på kortet samt korttyp och den mod som väljs nedan. Enhetsnamnet är normalt **/dev/sp1**.

Minor = Kortadress + Typ + Mod

Typ=0 vid sändning av data och Typ=64 vid mottagning. Mod=0 vid textmod och Mod=128 vid rå mod.

Exempelvis aktiveras enheten för ett Centronics-gränssnitt till en skrivare i textmod med kortadressen 10 med följande kommando (minor=10):

```
/etc/mknod c /dev/sp1 7 10
```

Enheten aktiveras i rå mod som ett Centronicsgränssnitt eller en SP1 sändare med kortadressen 10 genom att ange minor 138=10+128 eller som motsvarande SP1-mottagare med samma kortadress med minor 202=10+64+128.

15.7.6 Terminaler och modem via DataBoard - ua

Drivrutinen för dessa kort har primära enhetsnumret **8** (major device) och de sekundära enhetsnumren (minor device) innehåller bl a den kortadress (0 - 61) som byglas på korten. Maximalt åtta portar av denna typ kan hanteras av drivrutinen. Alla parametrar som t ex överföringshastighet byglas på korten och ignoreras av drivrutinen. Byglingarna beskrivs i **Installation** för respektive kort där även exempel visas hur korten och programvaran konfigureras på olika sätt.

Följande rader i **/etc/gettydefs** och **/etc/inittab** kan användas för att möjliggöra inloggning. Enhetsnamnen är här enligt exemplen nedan.

I **/etc/gettydefs**:

```
uamo# B1200 HUPCL # B1200 HUPCL SANE IXANY TAB3 ECHOE
   #Login: #uamo
```

I **/etc/inittab** för 5124-modem respektive 4117 eller 4118-korten.

```
mo2:2:respawn:nice -16 /etc/getty /dev/modem uamo
ua1:2:respawn:nice -16 /etc/getty /dev/ua01 uamo
```

Modemkort 5124

DataBoardkortet 5124 är ett modem för direkt anslutning till telenätet för överföringshastigheten 300 baud dubbelriktat eller 1200/75 eller 75/1200 baud split speed. Kortet och programvaran hanterar inkommande anrop med auto-svar för inloggning utifrån till systemet. För att ringa ut krävs dels att kortet byglas annorlunda och dels att användaren manuellt slår telefonnumret. Se **Installation av 5124** för detaljer.

För modemkortet är minor device = 64 + kortadressen. Endast kortadresserna 1, 2, ... 7 är tillåtna. Följande enhetsnamn rekommenderas. Här visas kommandot för att aktivera enheten för 5124 med kortadressen 2.

```
/etc/mknod  /dev/modem  c  8  66
```

Terminalkort 4117 och 4118

DataBoardkorten 4117 och 4118 ger systemet extra terminalportar. Till skillnad mot övriga terminalportar byglas överföringsparametrarna på dessa kort. Se **Installation av 4117** eller **4118** för detaljer.

För 4117/4118-korten är minor device = kortadressen + portnumret. Portnumret är normalt 0. I vissa datormodeller där båda portarna på 4118-kortet kan användas kan portnumret även vara 1. Enhetsnamnet **/dev/uaXX** rekommenderas, där XX är kortadressen. Här visas ett kommando för att aktivera enheten för 4117/4118 med kortadressen 1.

```
/etc/mknod  /dev/ua01  c  8  1
```

15.8. Andra yttre enheter

15.8.1 Skivminnen av SMD-typ - /dev/smdN

Skivminnen av SMD-typ ansluts via VME-styrkort, där upp till två skivminnen kan hanteras av varje VME-kort. SMD-skivminnen kan ej anslutas till de datorer som saknar VME-rack.

Drivrutinen för dessa skivminnen har primära enhetsnumret **4** (major device), medan sekundära enhetsnumret (minor device) är olika för varje skivminne enligt tabellen i kapitel 15.9. Varje skivminne kan delas upp i två partitioner, som då skall ha olika sekundära enhetsnummer och olika enhetsnamn.

Det logiska enhetsnamnet för ett SMD skivminne är **/dev/smdX**, där X är 0, 1, ...

Systemet kan aldrig startas upp (bootas) från ett SMD-skivminne.

Detaljer om hur enheterna ansluts finns i **Installationsanvisningen för SMD-skivminnen.**

15.8.2 Bandstation med 1/2 tums band - /dev/mt0

För bandstationer skall enhetsnamnet vara **/dev/mt0**.

Primära enhetsnumret (major device) ska vara **6**, medan det sekundära (minor device) normalt är **0**.

Yttre bandstationer kan anslutas på olika sätt, se installationsanvisningen för detaljer. I de flesta fall används ett VME-kort för direktanslutning till bandstationer med Pertec styrenhet. I vissa datorer används istället ett DataBoard-kort.

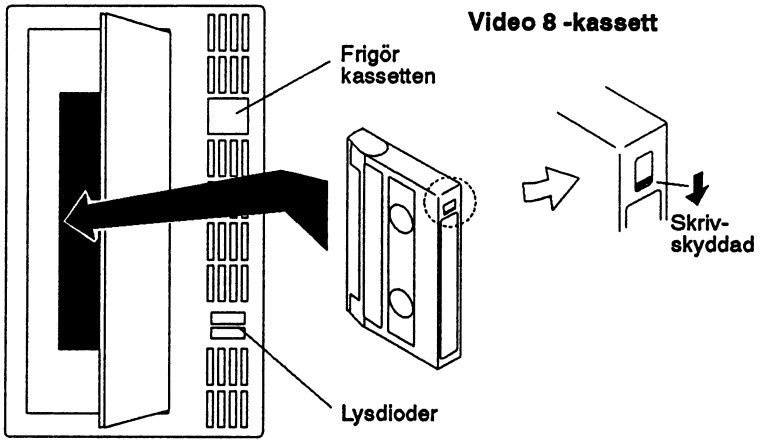
15.8.3 Kassettenhet, 8 mm (Video8) - /dev/vt0

Backup-enheten för en 2 Gbyte kassett ansluts till en SCSI-port på datorn. Vid intern montering ansluts den till samma port somden ordinarie kassettenheten. I övriga datorer ansluts den till den yttre SCSI-kontakten, se installationsanvisningen samt kapitel 15.9.

Enhetsnamnet för Video8 kassettenhet är **/dev/vt0**.

Primära enhetsnumret (major device) är **5**. Det sekundära enhetsnumret (minor device) varierar mellan datormodellerna eftersom olika SCSI-portar används. Dessutom beror enhetsnumret på om andra enheter, t ex extra skivminnen, är kopplade till samma SCSI-port.

Extra långa kassettband av typen **Exabyte 8 mm data-kassett, 2048 Mbyte** skall användas, eller kompatibelt band (t ex Sony QG-112M). Enheten hanteras på samma sätt som den ordinarie 1/4 tums kassettenheten (/dev/st0).



Den gröna lysdioden visar att kassetten är laddad. Den gula lyser när datorn skriver eller läser på kassettenheten.

Ta aldrig ut bandet medan den gula lysdioden lyser.

När knappen trycks in lösgörs bandet och skjuts automatiskt ut. Knappen ska även tryckas in för att öppna luckan när en kassett skall sättas in.

Rengöring skall göras 1 gång/månad eller efter ca 30 Gbyte, med Exabyte 8 mm Cleaning Cassette.

Observera! Skrivning till kassetten kan bara ske från bandets början. Därför kan inte tillvalen **r** och **u** användas i *tar* och bandet måste alltid återspolas före skrivning. Läsning kan däremot göras från valfri position.

Om man inte vill att bandet skall återspolas efter läsning anges ett sekundärt enhetsnummer (minor device) med bit 0 satt (udda) och det enhetsnamn som då bör användas är **/dev/vt0n**.

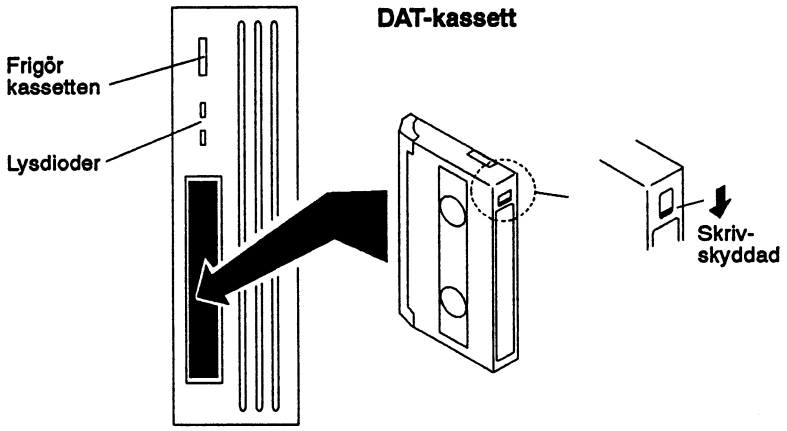
15.8.4 Kassettkassett, 4 mm (DAT) - /dev/dat0

Backup-enheten för en DAT-kassett ansluts till en SCSI-port på datorn. Vid intern montering ansluts den till samma port somden ordinarie kassettenheten. I övriga datorer ansluts den till den yttre SCSI-kontakten, se installationsanvisningen samt kapitel 15.9.

Enhetsnamnet för DAT kassettenhet är **/dev/dat0**.

Primära enhetsnumret (major device) är **5**. Det sekundära enhetsnumret (minor device) varierar mellan datormodellerna eftersom olika SCSI-portar används. Dessutom beror enhetsnumret på om andra enheter, t ex extra skivminnen, är kopplade till samma SCSI-port.

4 mm Digital Audio Tape (DAT) kassettband skall användas, där data lagras enligt DDS format. Exempelvis 3M R120 för 1300 Mbyte. Enheten hanteras på samma sätt som den ordinarie 1/4 tums kassettenheten (/dev/st0).



D

en gröna lysdioden visar att kassetten är laddad. Den gula lyser när datorn skriver eller läser på kassetten. Vid något fel blinkar någon lysdiod snabbt. Om den gröna blinkar långsamt kan rengöring enligt nedan eventuellt hjälpa.

Ta aldrig ut bandet medan den gula lysdioden lyser.

När knappen trycks in lösgörs bandet och skjuts automatiskt ut.

Rengöring skall göras 1 gång/månad eller efter ca 30 Gbyte, t ex med Sony DG-5CL Cleaning Cassette.

Observera! Skrivning till kassetten kan bara ske från bandets början. Därför kan inte tillvalen **r** och **u** användas i *tar* och bandet måste alltid återspolas före skrivning. Läsning kan däremot göras från valfri position.

Om man inte vill att bandet skall återspolas efter läsning anges ett sekundärt enhetsnummer (minor device) med bit 0 satt (udda) och det enhetsnamn som då bör användas är **/dev/dat0n**.

15.8.5 Optiskt skivminne, WORM - /dev/worm0

Enheten med optiskt skivminne av typen WORM (data kan ej raderas) ansluts till en SCSI-port på datorn. Vid intern montering ansluts den till samma port som den ordinarie streamerenhet. I övriga datorer ansluts den till den yttre SCSI-kontakten, se installationsanvisningen samt kapitel 15.9.

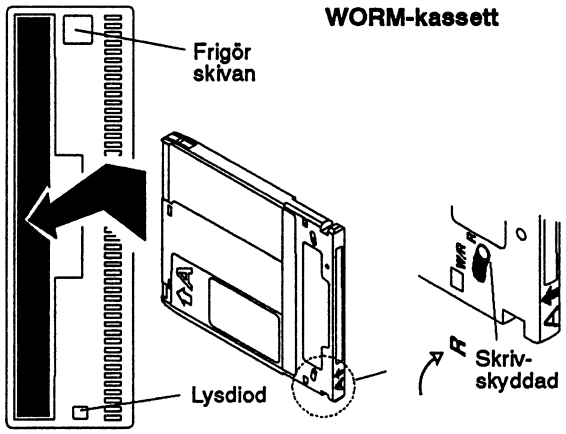
Enhetsnamnet för optiska skivminnet är **/dev/worm0**.

Primära enhetsnumret (major device) är **5**. Det sekundära enhetsnumret (minor device) varierar mellan datormodellerna eftersom olika SCSI-portar används. Dessutom beror enhetsnumret på om andra enheter, t ex extra skivminnen, är kopplade till samma SCSI-port.

De optiska skivorna är dubbelsidiga och inspelning sker på en sida i taget. Ett speciellt kommando */etc/wormmkfs* används för att skapa ett filsystem på det optiska skivminnet. En speciell filhanterare, **/hnd/dworm**, startas för att skriva och läsa filer på skivminnet med normal kommandon i operativsystemet. Programvaran levereras med skivminnet.

De skivor som används skall vara av typen **5 1/4 tums optisk skivminneskassett WORM för 786 Mbyte, med 393 Mbyte per sida**, t ex Maxtor OC800.

Skivan sätts in med den valda sidan (A/B) vänd från lysdioden.



Ta aldrig ut skivan medan lysdioden lyser och koppla alltid bort enheten från filsystemet (kommandot **umount /dev/worm0**) innan den tas bort.

När knappen trycks in lösgörs skivan och skjuts automatiskt ut.

15.9. Datorberoende tabeller och anslutningar

Vissa namn och parametrar varierar mellan de olika datorsystemen. Dessa har i tidigare kapitel beskrivits enbart allmänt. Här listas dessa för de olika datortyperna.

Det finns ett kommando för varje datortyp, vilket returnerar ett sant värde (0) enbart i det systemdet körs. Se **Referenshandboken**, under machid(1).

15.9.1 Start och avstängningsförfarande

I datorerna DS90-11 och DIAB1130 sker start och avstängning på ett något annorlunda sätt än beskrivet i kapitel 2. Nyckelläget OFF finns inte utan istället ett läge MAN som anger manuell start. I DIAB1130 används dessutom en omkopplare istället för en nyckel.

Start sker alltid vid spänningstillslag, i manuell eller auto-mod. Läget MAN kan ställas in när som helst utan att systemet stängs av.

Avstängning sker alltid genom att **/etc/shutdown -k** utförs, lämpligen med nyckeln/omkopplaren i läge MAN. Endast efter att texten "**System halted**" visats på huvudkonsolen får operatören stänga av spänningen.

15.9.2 Parametrar i /sas/bootpar

BOOT device

Parametern anger varifrån systemet skall laddas och i princip samma enheter som i laddningsprogrammet kan anges, men istället för enhetsnamnet anges direkt primära enhetsnumret (major) och kanalnumret.

(5,X) motsvarar skivminnesenheten si(X,0), vanligtvis (5,16) eller (5,0)
(9,X) motsvarar diskettenheten mf(X,0), exempel (9,8), (9,40), (9,0), (9,10).

För möjliga kanalnummer, se listan på nästa sida över enhetsnamn i laddningsprogrammet.

CPU-klockfrekvensen

Denna beror på datortypen och eventuell variant av datorkortet.

DS90-10/11		10000000
DS90-20/21		16670000
DS90-30/31	(25/33 MHz)	25000000 eller 33000000
DS90-30S		16000000
DIAB1320/DIAB1420		16000000
DIAB1130		16000000
DS90-41	(25/33 MHz)	25000000 eller 33000000
DIAB2420		25000000
DIAB2430	(25/33 MHz)	25000000 eller 33000000

15.9.3 Enhetsnamn vid start och i laddningsprogrammet

Enhetsnamnen som anges vid uppstart av systemet på bootnivå 0 är samma som används i laddningsprogrammet (bootnivå 1), d v s med program i biblioteket /sas. Enhetstyp och kanalnummer anges. Valfritt skivminne av SCSI-typ kan anges med si(X,0) där X är sekundära enhetsnumret (minor). För disketter anges mf(X,0) där X anges i listorna nedan. På denna nivå finns ingen avkänningsfunktion (sense) för disketter och endast listade diskett-typer kan användas.

DS90-11

si(0,0)	Första skivminnet
mf(64,0)	720 kbyte diskett (normal 5 1/4 tum BOOT-diskett)
mf(8,0)	1.2 Mbyte diskett (5 1/4 tum). Ej bootning.

DS90-10,20,21,30,31,DS101

si(16,0)	Första skivminnet
mf(8,0)	1.2 Mbyte diskett (normal 5 1/4 tum BOOT-diskett)
mf(64,0)	720 kbyte diskett (5 1/4 tum)

DS90-30S

si(16,0)	Första skivminnet
mf(8,0)	1.2 Mbyte diskett (normal 5 1/4 tum BOOT-diskett)
mf(10,0)	1.44 Mbyte (bara om inre enheten byts mot 3.5 tum)
mf(0,0)	720 kbyte diskett (interna enheten)

DIAB1320, DIAB1420

si(16,0)	Första skivminnet
mf(10,0)	1.44 Mbyte diskett (normal 3.5 tum BOOT-diskett)
mf(24,0)	1.2 Mbyte diskett (extra 5 1/4 tums enhet)
mf(0,0)	720 kbyte diskett (interna enheten)
mf(16,0)	720 kbyte diskett (extra 5 1/4 tums enhet)

DS90-41

si(16,0)	Första skivminnet
mf(0,0)	1.2 Mbyte diskett (normal 5 1/4 tum BOOT-diskett)
mf(40,0)	1.44 Mbyte (bara om inre enheten byts mot 3.5 tum)
mf(32,0)	720 kbyte diskett (interna enheten)

DIAB2420, DIAB2430

si(16,0)	Första skivminnet
mf(40,0)	1.44 Mbyte diskett (normal 3.5 tum BOOT-diskett)
mf(37,0)	1.2 Mbyte diskett (extra 5 1/4 tums enhet)
mf(32,0)	720 kbyte diskett (interna 3.5 tums enheten)
mf(33,0)	720 kbyte diskett (extra 5 1/4 tums enhet)

DIAB1130

si(0,0)	Första skivminnet
mf(0,0)	1.44 Mbyte diskett (normal 3.5 tum BOOT-diskett) eller mf(40,0) vilket också är 1.44 Mbyte
mf(100,0)	1.2 Mbyte diskett (extra 5 1/4 tums enhet via en MSI)
mf(64,0)	1.44 Mbyte diskett (extra 3.5 tums enhet via en MSI)
mf(32,0)	720 kbyte diskett (interna 3.5 tums enheten)
mf(96,0)	720 kbyte diskett (extra 5 1/4 tums enhet via en MSI)

15.9.4 Tabell över enhetsnamn och major/minor-nummer

På opertivsystemnivå anges yttre enheter med enhetsnamn i biblioteket **/dev** och har primära (major) och sekundära (minor) enhetsnummer enligt denna tabell. Även för vissa inre enheter i datorn finns enhetsnamn och nummer.

Primär Major	Sekundär Minor	Namn	Förklaring
1			Enheten styrs av drivrutinen för de seriella portarna.
	0	lp	Skrivarporten
	1	console	Porten för huvudkonsolen
	1	systty	Porten för huvudkonsolen
	1	syscon	Virtuell huvudkonsol (enanvändarmod)
	2	tty02	tty02
	...	...	...
	11	tty11	tty11
2	0	tty	Enhet för den terminal användaren är inloggad på.
3			Enheten styrs av den drivrutin som hanterar den interna maskinvaran.
	0	mem	Primärminnet
	1	kmem	kmem
	2	null	null
	3	inoutb	Rutin för I/O på bytenivå
	4	inout	Rutin för I/O på ordnivå
	5	autosw	Läser nyckelbrytaren på frontpanelen
	6	bclock	Systemklockan med batteribackup
	7	nvram	Startparametrar i non-volatile RAM
	8	DBinoutb	DataBoard I/O drivrutin
	9	error	Log för systemmeddelanden
	12	syslog	Enhet för generering av systemmeddelanden
	13	IU	Enhet för IU-kortet, enbart DIAB1130, se IU(7)
4			Enheten styrs av drivrutinen för skivminnen av SMD-typ. Varje skivminne kan vara uppdelat i två partitioner.
	0	smd0	Kort 0, id 0, huvudenhet
	1	smd1	Kort 0, id 0, 2:a partitionen
	2	smd2	Kort 0, id 1, huvudenhet
	3	smd3	Kort 0, id 1, 2:a partitionen
	8	smd4	Kort 1, id 0, huvudenhet
	9	smd5	Kort 1, id 0, 2:a partitionen
	10	smd6	Kort 1, id 1, huvudenhet
	11	smd7	Kort 1, id 1, 2:a partitionen
	16	smd8	Kort 2, id 0, huvudenhet
	17	smd9	Kort 2, id 0, 2:a partitionen
	18	smd10	Kort 2, id 1, huvudenhet
	19	smd11	Kort 2, id 1, 2:a partitionen

Primär Major	Sekundär Minor	Namn	Förklaring
5			Enheten styrs av drivrutinen för SCSI-enheter. Sekundära enhetsnummer är olika för olika datorer p g a den fysiska uppbyggnaden, använda SCSI-portar och byglat id-nummer i enheterna. I följande delkapitel visas hur enhetsnummer beräknas samt vilka enhetsnummer som är standard i de olika datormodellerna. Exempel på enhetsnamn är:
	16 eller 0	si0	Interna fasta skivminnet (vanligen minor=16)
		si2	Extra fast skivminne nr 1
		si4	Extra fast skivminne nr 2
		si6	Extra fast skivminne nr 3 etc
		worm0	Extra optiskt skivminne, WORM
		st0	1/4 tums kassettenhet, QIC (streamer)
		st0n	Icke återspolande kassettenhet, QIC
		st1	Extra 1/4 tums kassettenhet, QIC
		st1n	Icke återspolande extra kassettenhet, QIC
		vt0	Kassettenhet, Video-8
		vt0n	Icke återspolande kassettenhet, Video-8
		datt0	Kassettenhet, DAT
		datt0n	Icke återspolande kassettenhet, DAT
6			Enheten styrs av drivrutinen för 1/2 tums bandstationer.
	0	mt0	Yttre bandstation
7			Enheten styrs av drivrutinen för Centronics- och SP1-gränssnitt via DataBoard-kort.
	xx	centr	xx=kortadress+typ+mod Typ 0 = Centronics eller SP1 sändare Typ 64 = SP1 mottagare Mod 0 = Textmod i överföringen Mod 128 = Rå mod i överföringen
8			Enheten styrs av drivrutinen för seriella kanaler eller modem på DataBoard-kort typ 4117, 4118 eller 5124.
	xx	uaxx	xx=kortadress för 4117, 4118
	yy	modem	yy=64+kortadress för 5124-modem
9			Enheten styrs av drivrutinen för diskettenheterna.
	0 eller 8	mf0	Intern diskettenhet, 3.5 tum eller 5 1/4 tum.
	1, 9 eller 24	mf1	Extra diskettenhet, 3.5 tum eller 5 1/4 tum. Minornumret beror på datortypen enligt följande delkapitel. Även andra nummer kan väljas för speciella ändamål.
10			Om ett kommuniationskort ansluts (4004) används denna drivrutin.

Primär Major	Sekundär Minor	Namn	Förklaring
11			Enheten styrs av drivrutinen för terminal-koncentrator på ett eller flera VME/TC-kort.
	0-9	tty12-tty21	Serieportar på VME/TC-kort 0
	15	tc0	Styrning/status på kort 0
	16-25	tty22-tty31	Serieportar på VME/TC-kort 1
	31	tc1	Styrning/status på kort 1
			Sekundära enhetsnummer ökas med 16 för
12			Drivrutin för hantering av VME-bussen.
	0	VMEdrv	Rutin som sänder kommandon och väntar på avbrottssignal.
13			Drivrutin för hantering av DataBoardbussen.
	0	DBIdrv	Rutin som sänder kommandon och väntar på avbrottssignal.

Primära enhetsnummer ovanför 13, såväl som vilka enheter som finns, är datorberoende och initeras normalt automatiskt av systemet. Enhetsnamnen enligt följande lista används normalt men primära enhetsnumren är bara exempel.

Primär Major	Sekundär Minor	Namn	Förklaring
16	0	ethdrv	Drivrutin för Ethernet lokalt nätverk i datorer med Ethernet inbyggt på datorkortet.
17	0-255	ptyNN	ptyp0-ptyzf, ptya0-ptyff. Pseudo-ttyer, pty-del.
18	0-255	ttyNN	ttyp0-ttyzf, ttya0-ttyff. Pseudo-ttyer, tty-del.
19	0 1	comcboot comcraw	ComC kommunikation
23	0-255	tndNN	Portar för telnet inloggning. Se /etc/telnettab
24	0	nfsd	NFS hanterare
25	0 1	tcp udp	TCP hanterare UDP hanterare
26	0-255	rldNN	Portar för remote login. Se /etc/rlogintab
27	0-255	padNN	Portar för PAD . Se /etc/padtab.
28	0	lockd	NFS lock device
DIAB1130-specifika enheter:			
14	xx	dram	Temporär systemresurs (enbart DIAB1130) Se dram(7).
16	cs	vdbmN	Virtuell DataBoard maskin (enbart DIAB1130) Se vdbm)(7).
19	0,128	iomr,iomp	MEM disk (enbart DIAB1130). Se iom(7).
23	0,1	bramN	Batteri-RAM för systemdata (enbart DIAB1130). Se bram(7).
34	xx	ttyNN	Terminalportar på USI-kort. Se usi(7).

15.9.5 Diskettenheter, enhetsnummer och format

Disketter i DS90-10 / 11 / 20 / 21 / 30 /31

Den interna diskettenheten är normalt en 5 1/4 tums enhet. Normalt används minornummer 8 som standard med avkänning (sense).

Med "sense"-biten satt är minor-numret:

Bit 7	= 1 för 8 tums diskett (enbart DS90-11) = 0 för 5 1/4 tums diskett.
Bit 6,5	Steghastighetsvärde, normalt 0.Se den tekniska beskrivningen för enheten. Generellt ger högre värden längre stegtid.
Bit 4	Reserverad, bör vara 0.
Bit 3	**Sense** bit = 1.
Bit 2	Reserverad, bör vara 0. Temporärt anger 1 att diskparametrarna visas på skärmen då disketten läses (**display**).
Bit 1,0	Fysisk enhet 0..3.

Utan "sense" anger minornumret formatet enligt:

Bit 7	= 1 för 8 tums diskett (enbart DS90-11) = 0 för 5 1/4 tums diskett.			
Bit 6,5	Sektorstorlek.	0x00 (0)	= 256 byte/sektor.	
		0x20 (32)	= 128 byte/sektor.	
		0x40 (64)	= 512 byte/sektor.	
		0x60 (96)	= 1024 byte/sektor.	
Bit 4	Enkelsidig diskett. Endast sida 0 söks.			
Bit 3	Sense bit = 0.			
Bit 2	Antal sektorer/spår	Bit-värde:	0	4
		128 8 tum	26	25 sekt/spår
		256 8 tum	26	25
		512 8 tum	16	15
		1024 8 tum	8	7
		128 Övriga	16	25
		256 Övriga	16	25
		512 Övriga	9	15
		1024 Övriga	5	7
Bit 1,0	Fysisk enhet 0..3.			

Exempel på vanliga diskettformat kan ses i tabellerna för övriga datortyper på följande sidor.

I DS90-11 hanteras även 8 tums disketter via en yttre ansluten 8 tums enhet. En 8 tums standard diskett har 77 spår. Drivrutinen klarar även av 8 tums disketter med enkel densitet om disketten har 128 bytes/sektor. Används sekundära enhetsnumret (minor) 128+32+16+2 kommer drivrutinen att läsa enkel densitets 8 tums standarddiskett på enhet 0. Detta är det enda sättet att läsa 8 tums disketter med enkel densitet. Sense optionen är inte implementerad för enkel densitet.

Disketter i DS90-30S / DIAB1320 / DIAB1420

Den interna diskettenheten är i DIAB1320 och DIAB1420 en 3.5 tums enhet, men i DS90-30S en 5 1/4 tums enhet. Normalt används minornummer **8** som standard med avkänning (sense). En eventuell andra diskettenhet har normalt minor **24**. Minornumret formas som

Minor = Typ + 16*Enhet

där Enhet =0-2 och anger fysisk diskettenhet, med 0 för den interna enheten. Typ är diskett-typen enligt tabellen nedan.

Typ	Storlek	Namn	Sidor	Cyl	S/C	B/S	Rpm:Rate
0	720 kbyte	PC (DD)	2	80	9	512	300:250
1	Display = Avkänn och visa parametrarna						
2	160 kbyte	M2/ABC	1	40	16	256	300:250
3	180 kbyte	PC	1	40	9	512	300:250
4	320 kbyte	ABC, DT/SS	1	80	16	256	300:250
5	320 kbyte	ABC, ST/DS	2	40	16	256	300:250
6	360 kbyte	PC	2	40	9	512	300:250
7	640 kbyte	M8/SBC/ABC	2	80	16	256	300:250
8	Sense = Avkänn parametrarna automatisk på disketten						
9	1.2 Mbyte	PC (HD)	2	80	15	512	360:500
10	1440 kbyte	PC (HD 3.5 tum)	2	80	18	512	300:500
11	2400 kbyte	PC	2	80	30	512	360:1000
12	2880 kbyte	PC	2	80	36	512	300:1000

13-15 Reserverade

Sidor:	Enkelsidig (SS) eller dubbelsidig (DS) diskett
Cyl	Cylindrar (spår) per sida (40=enkelspår, 80=dubbelspår)
S/C	Sektorer / cylinder
B/S	Bytes / sektor
Rpm	Rotationshastighet
Rate	Dataöverföringshastighet i kbits/sekund
DT/ST	Beteckning för dubbelspårs/enkelspårs disketter
HD/DD	Beteckning för hög densitet / normal dubbel densitet

Disketter i DS90-41 / DIAB1130 / DIAB24xx

Den interna diskettenheten är i DIABxxxx-datorerna en 3.5 tums enhet, men i DS90-41 en 5 1/4 tums enhet. Normalt används minornummer **0** som standard med avkänning (sense). En eventuell andra diskettenhet har normalt minor **1**. Minornumret formas som

Minor = Typ + Enhet

Minor = Typ + Enhet + 64 (med extra MSI-kort i DIAB1130)

där Enhet =0-2 och anger fysisk diskettenhet, med 0 för den interna enheten. Typ är diskett-typen enligt tabellen nedan. Ett extra MSI-kort i DIAB1130 har kortval 25 decimalt.

Typ	Storlek	Namn	Sidor	Cyl	S/C	B/S	Rpm:Rate
0	Sense = Avkänn parametrarna automatisk på disketten						
4	Display = Avkänn och visa parametrarna						
8	160 kbyte	M2/ABC	1	40	16	256	300:250
12	180 kbyte	PC	1	40	9	512	300:250
16	320 kbyte	ABC, DT/SS	1	80	16	256	300:250
20	320 kbyte	ABC, ST/DS	2	40	16	256	300:250
24	360 kbyte	PC	2	40	9	512	300:250
28	640 kbyte	M8/SBC/ABC	2	80	16	256	300:250
32	720 kbyte	PC (DD)	2	80	9	512	300:250
36	1.2 Mbyte	PC (HD)	2	80	15	512	360:500
40	1440 kbyte	PC (HD 3.5 tum)	2	80	18	512	300:500
44	2400 kbyte	PC	2	80	30	512	360:1000
48	2880 kbyte	PC	2	80	36	512	300:1000

52-60 Reserverade

Sidor:	Enkelsidig (SS) eller dubbelsidig (DS) diskett
Cyl	Cylindrar (spår) per sida (40=enkelspår, 80=dubbelspår)
S/C	Sektorer / cylinder
B/S	Bytes / sektor
Rpm	Rotationshastighet
Rate	Dataöverföringshastighet i kbits/sekund
DT/ST	Beteckning för dubbelspårs/enkelspårs disketter
HD/DD	Beteckning för hög densitet / normal dubbel densitet

15.9.6 SCSI ID och minor device i olika system

Datorsystemen har olika antal SCSI portar. Till varje port kan upp till sju enheter anslutas med SCSI id-nummer 0, 1, ... 6. Portens eget id-numer är 7 och får ej användas av någon enhet. Normalt har det interna skivminnet (**/dev/si0**) id = 0. Där så är möjligt är interna kassettenheten (**/dev/st0**) ansluten till en annan SCSI-port och har id = 0 eller 6.

Sekundära enhetsnummer (minor device) för enheter anslutna till SCSI-portarna beräknas utgående från det id-nummer som byglats på enheterna. Bit 7 i sekundära enhetsnumret anger att enheten är en streamerenhet. Om enheten är en streamerenhet så anger bit 0 i enhetsnumret (udda nummer) att streamerenheten skall vara icke återspolande, därav de dubbla numren för streamer i tabellen.

Tabellen visar sekundära enhetsnummer (minor device).

Id	SCSI port 0		SCSI port 1		SCSI port 2		DIAB1130, MSI	
	Disk	Streamer	Disk	Streamer	Disk	Streamer	Disk	Streamer
0	0	128,129	16	144,145	32	160,161	64	192,193
1	2	130,131	18	146,147	34	162,163	66	194,195
2	4	132,133	20	148,149	36	164,165	68	196,197
3	6	134,135	22	150,151	38	166,167	70	198,199
4	8	136,137	24	152,153	40	168,169	72	200,201
5	10	138,139	26	154,155	42	170,171	74	202,203
6	12	140,141	28	156,157	44	172,173	76	204,205

id	Id-nummer byglat på enheten
SCSI-port	Den port där enheten är ansluten
Disk:	Skivminne
Streamer	Kassettstreamer (normalt samt icke återspolande)

Där så är möjligt vid anslutning av yttre massminnesenheter till SCSI-portarna bör följande allmänna regler följas:

- Skivminnen med hög kapacitet bör inte anslutas till samma port som streamerenheter, optiska skivminnen eller andra långsamma enheter.
- Vid användning av spegel-skivminnen skall dessa alltid anslutas till olika SCSI-portar.

Terminering av SCSI kabel-kedjan

Då flera enheter kopplas till samma SCSI-port skall enbart den yttersta enheten vara försedd med termineringsmotstånd enligt installationsanvisningarna. Enheterna får inte kopplas isär och kopplas ihop utan att anvisningarna om detta följs.

Om ingen enhet är ansluten till den yttre SCSI-porten, skall det finnas termineringar på porten. I vissa datorer används en termineringsplugg. I andra finns termineringsbyglingar som skall vara slutna när inget anslutits och öppna vid inkoppling av yttre enheter. I vissa datorer görs en intern omkoppling när yttre enheter kopplas in och ur.

I de olika datorsystemen används SCSI-portarna normalt enligt följande tabeller och figurer.

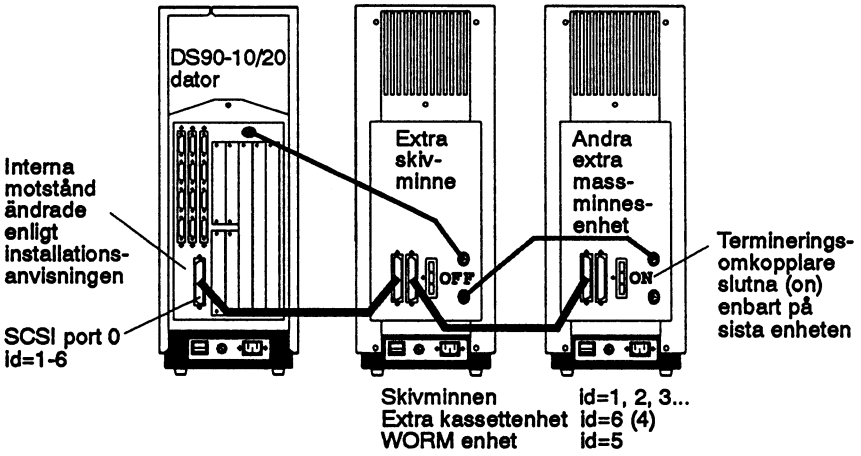
Datorsystemen DS90-10/20

Inne i dessa datorer skall två termineringsmotstånd vara borttagna när yttre massminnesenheter är inkopplade. De skall åter installeras om de yttre enheterna kopplas bort från den yttre kontakten på datorn (se installationsanvisningarna).

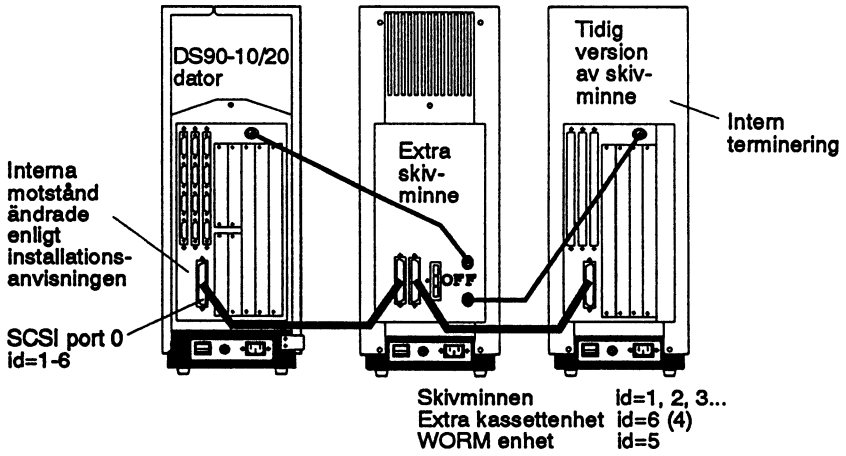
SCSI 0 (yttre) Intern kassettstreamer (id=0) och övriga extra enheter. Enbart denna port finns i yttre kontakten.

SCSI 1 Internt skivminne (id=0).

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
128	st0	SCSI 0, id 0: Intern kassettstreamer, 1/4 tum
129	st0n	Icke återspolande streamer
2	si2	SCSI 0, id 1: Extra skivminne nr 1
4	si4	SCSI 0, id 2: Extra skivminne nr 2
10	worm0	SCSI 0, id 5: WORM skivenhet
140	vt0,dat0	SCSI 0, id 6: Extra streamerenhet,Video8,DAT
141	vt0n,dat0n	Icke återspolande streamer



Nedan visas anslutning med en tidigare version av skivminnesenhet, som innehåller interna termineringar.

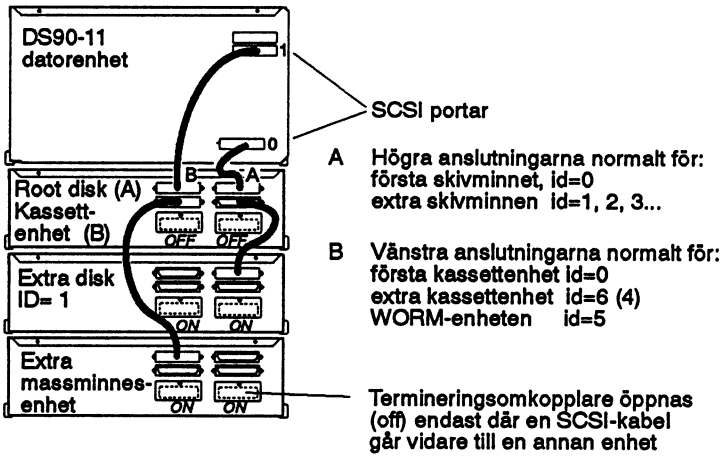


Datorsystem DS90-11

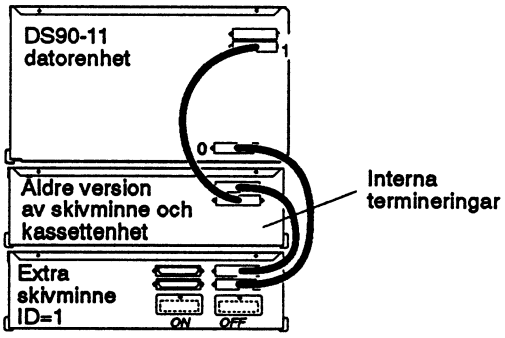
SCSI 0 (undre) Internt skivminne (id=0) och extra skivminnen. Ett spegelskivminne (mirror disk) måste anslutas till port 0 i denna dator.

SCSI 1 (övre) Intern kassettstreamer (id=0) och extra streamer samt optiska skivminnen

Minor	Namn	Port, ID och enhet
0	si0	SCSI 0, id 0: Intern winchester
144	st0	SCSI 1, id 0: Intern kassettstreamer, 1/4 tum
145	st0n	Icke återspolande streamer
2	si2	SCSI 0, id 1: Extra skivminne nr 1
4	si4	SCSI 0, id 2: Extra skivminne nr 2
26	worm0	SCSI 1, id 5: WORM skivenhet
156	vt0,dat0	SCSI 1, id 6: Extra streamerenhet,Video8,DAT
157	vt0n,dat0n	Icke återspolande streamer



Nedan visas anslutning av en tidigare version av enhet, som innehåller interna termineringar.

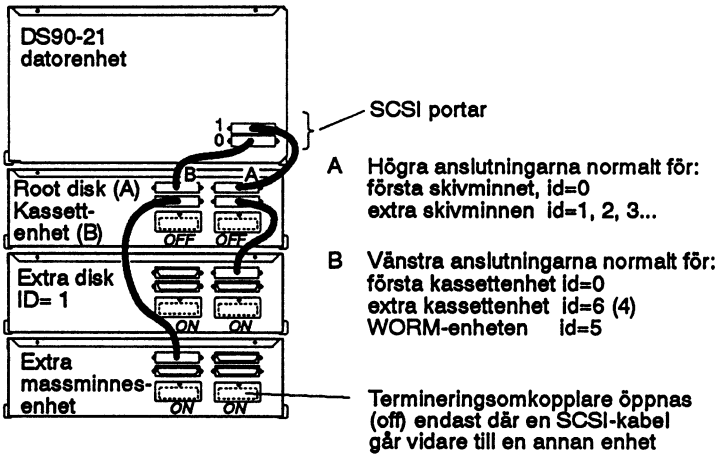


Datorsystem DS90-21

SCSI 0 (undre) Intern kassettstreamer (id=0) och extra streamer samt optiska skivminnen. Ett spegelskivminne (mirror disk) måste anslutas till port 0 i denna dator.

SCSI 1 (övre) Internt skivminne (id=0) och extra skivminnen

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
128	st0	SCSI 0, id 0: Intern kassettstreamer, 1/4 tum
129	st0n	Icke återspolande streamer
18	si2	SCSI 1, id 1: Extra skivminne nr 1
20	si4	SCSI 1, id 2: Extra skivminne nr 2
10	worm0	SCSI 0, id 5: WORM skivenhet
140	vt0,dat0	SCSI 0, id 6: Extra streamerenhet,Video8,DAT
141	vt0n,dat0n	Icke återspolande streamer



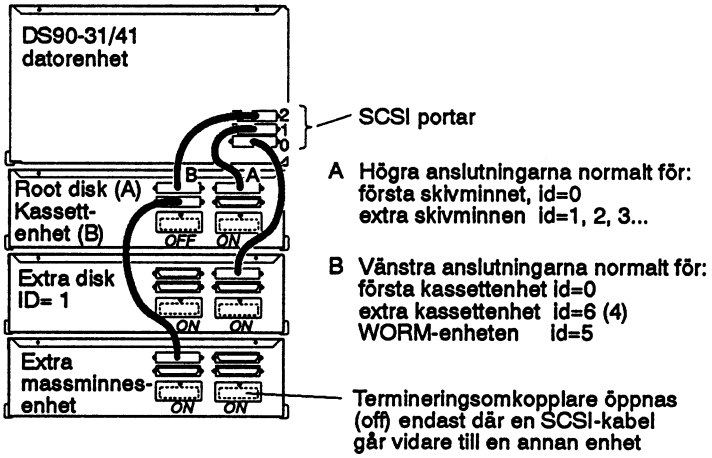
Datorsystem DS90-31 och DS90-41

SCSI 0 (undre) Fri för extra enheter, normalt extra skivminnen. Om inga extra skivminnen är anslutna kan eventuellt optiska skivminnen eller extra streamer anslutas här förutom till SCSI port 2.

SCSI 1 (mellan) Internt skivminne (id=0) och extra skivminnen

SCSI 2 (övre) Intern kassettstreamer (id=0) och extra streamer samt optiska skivminnen

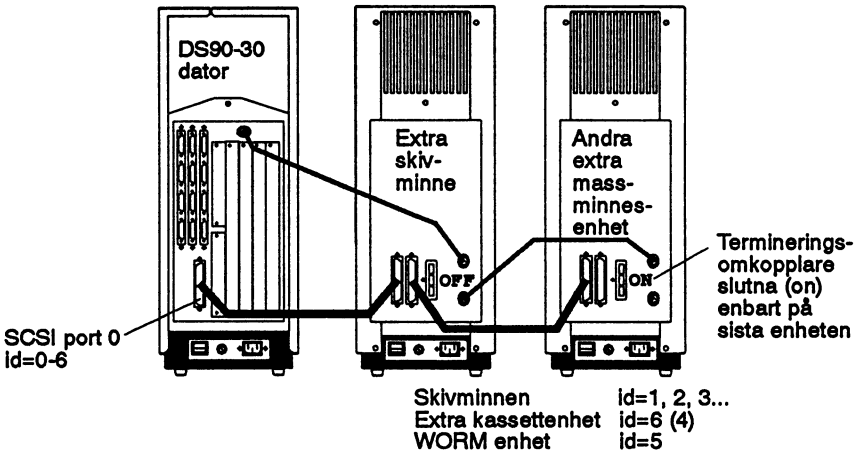
Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
160	st0	SCSI 2, id 0: Intern kassettstreamer, 1/4 tum
161	st0n	Icke återspolande streamer
2	si2	SCSI 0, id 1: Extra skivminne nr 1
4	si4	SCSI 0, id 2: Extra skivminne nr 2
42	worm0	SCSI 2, id 5: WORM skivenhet
172	vt0,dat0	SCSI 2, id 6: Extra streamerenhet,Video8,DAT
173	vt0n,dat0n	Icke återspolande streamer



Datorsystem DS90-30

SCSI 0 (yttre)	Fri för extra enheter. Enbart denna port finns i yttre kontakten.
SCSI 1	Internt skivminne (id=0).
SCSI 2	Intern kassettstreamer (id=0).

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
160	st0	SCSI 2, id 0: Intern kassettstreamer, 1/4 tum
161	st0n	Icke återspolande streamer
2	si2	SCSI 0, id 1: Extra skivminne nr 1
4	si4	SCSI 0, id 2: Extra skivminne nr 2
10	worm0	SCSI 0, id 5: WORM skivenhet
140	vt0,dat0	SCSI 0, id 6: Extra streamerenhet,Video8,DAT
141	vt0n,dat0n	Icke återspolande streamer



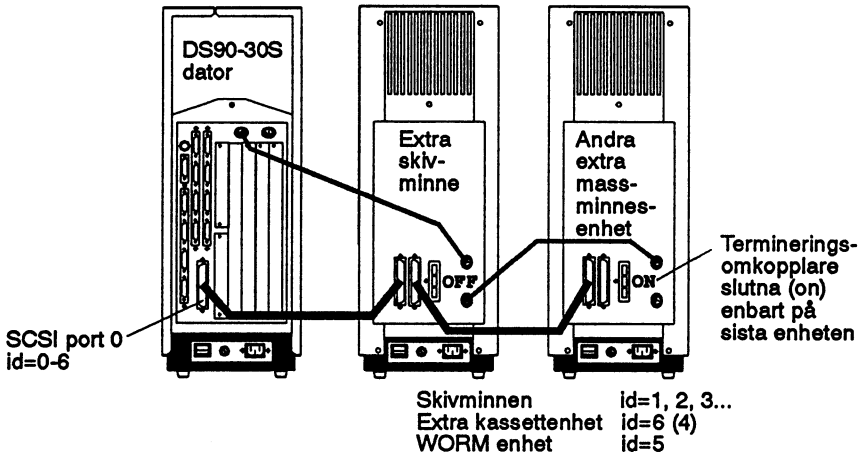
Datorsystem DS90-30S

Anslutning av yttre massminne i denna dator kräver att SCSI-tillvalet är installerat.

SCSI 0 (yttre) Extra enheter. Denna port är ej standard men krävs då extra enheter skall kopplas in i den yttre kontakten.

SCSI 1 Internt skivminne (id=0) och intern kassett-streamer (id=6).

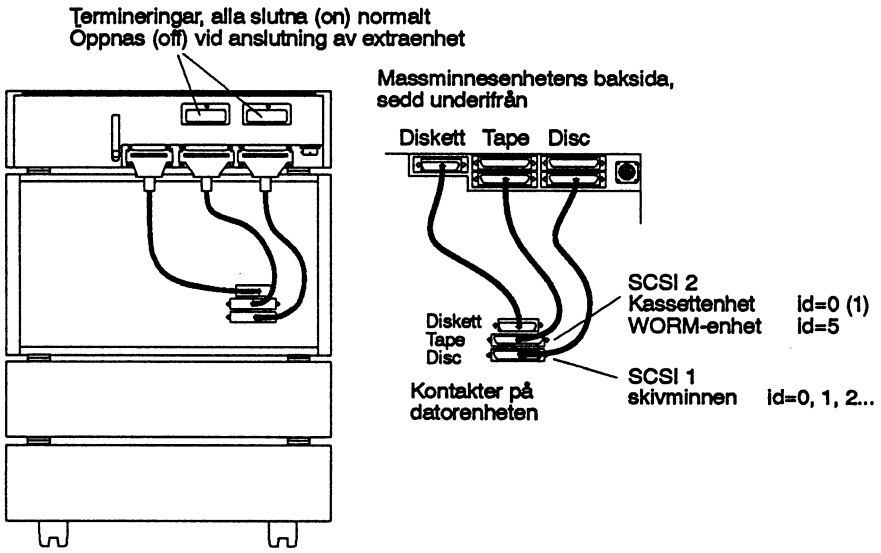
Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
156	st0	SCSI 1, id 6: Intern kassettstreamer, 1/4 tum
157	st0n	Icke återspolande streamer
2	si2	SCSI 0, id 1: Extra skivminne nr 1
4	si4	SCSI 0, id 2: Extra skivminne nr 2
10	worm0	SCSI 0, id 5: WORM skivenhet
140	vt0,dat0	SCSI 0, id 6: Extra streamerenhet,Video8,DAT
141	vt0n,dat0n	Icke återspolande streamer



Datorsystem DS101

SCSI 1	Internt skivminne (id=0) och extra skivminnen
SCSI 2	Intern kassettstreamer (id=0), intern extra streamer (id=1) och extra optiska skivminnen.

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
160	st0	SCSI 2, id 0: Intern kassettstreamer, 1/4 tum
161	st0n	Icke återspolande streamer
162	vt0,dat0	SCSI 2, id 1: Extra streamerenhet,Video8,DAT
163	vt0n,dat0n	Icke återspolande streamer
18	si2	SCSI 1, id 1: Extra skivminne nr 1
20	si4	SCSI 1, id 2: Extra skivminne nr 2
42	worm0	SCSI 2, id 5: WORM skivenhet



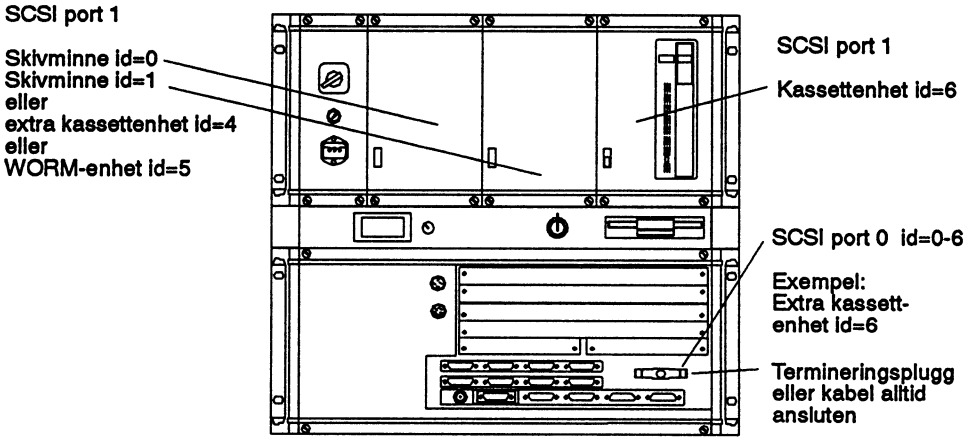
Datorsystem DIAB1320

Anslutning av yttre massminne i denna dator kräver att SCSI-optionen är installerad.

SCSI 0 (yttre) Extra enheter. Denna port är ej standard men krävs då extra enheter skall kopplas in i den yttre kontakten, vilken är av typen "high density". En termineringsplugg krävs om ingen yttre enhet är ansluten.

SCSI 1 Interna skivminnen (id=0, 1) och intern kassettstreamer (id=6). Istället för ett andra skivminne kan istället en andra kassettstreamer (id=4) eller ett skivminne av WORM-typ (id=5) installeras.

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
18	si2	SCSI 1, id 1: Intern winchester nr 2
152	vt0,dat0	SCSI 1, id 4: Intern extra streamer,Video8,DAT
153	vt0n,dat0n	Icke återspolande streamer
26	worm0	SCSI 1, id 5: Intern WORM skivenhet
156	st0	SCSI 1, id 6: Intern kassettstreamer, 1/4 tum
157	st0n	Icke återspolande streamer
2	si4	SCSI 0, id 1: Yttre extra skivminne nr 1
4	si6	SCSI 0, id 2: Yttre extra skivminne nr 2
10	worm1	SCSI 0, id 5: Yttre WORM skivenhet
140	st1	SCSI 0, id 6: Yttre extra streamerenhet
141	st1n	Icke återspolande streamer



Datorsystem DIAB1130

SCSI 0 (MSU)

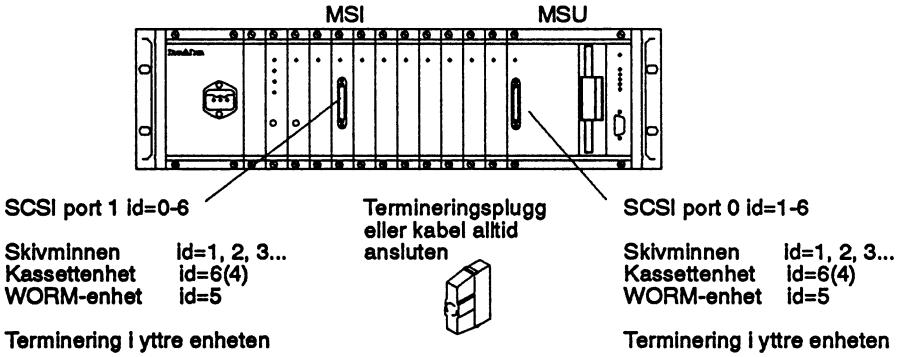
Internt skivminne (id=0). Dessutom finns en kontakt för yttre enheter. Kontakten är av typen "high density". En termineringsplugg krävs om ingen yttre enhet är ansluten.

OBS! Stäng alltid av datorn vid in- och urkoppling.

SCSI 1 (MSI)

Enbart med ett extra MSI-kort. Till denna port kan t ex en yttre streamerenhet kopplas in och ur utan att datorn stängs av.

Minor	Namn	Port, ID och enhet
0	si0	MSU, id 0: Intern winchester
2	si2	MSU, id 1: Yttre skivminne, MSU
140	st0	MSU, id 6: Yttre kassettstreamer, 1/4 tum,MSU
141	st0n	Icke återspolande streamer
66	si4	MSI, id 1: Yttre skivminne, MSI
204	st1	MSI, id 6: Yttre streamer, 1/4 tum, MSI
205	st1n	Icke återspolande streamer



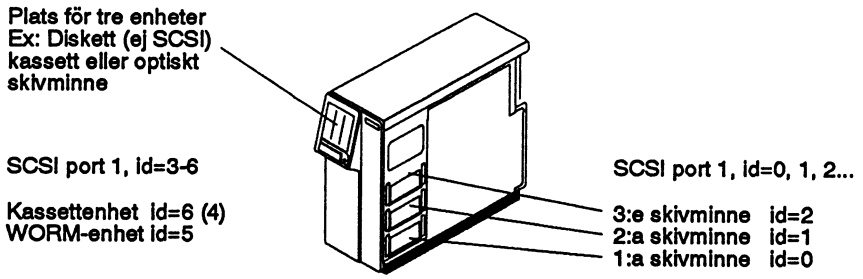
Datorsystem DIAB1420 / 2420

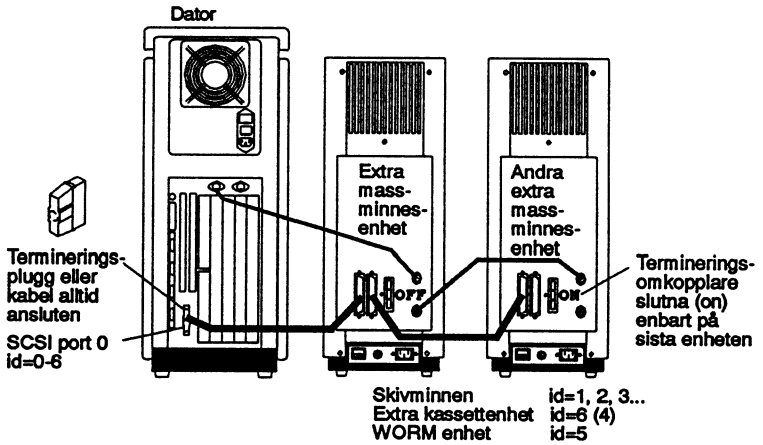
Anslutning av yttre massminne i denna dator kräver att SCSI-optionen är installerad.

SCSI 0 (yttre) Extra enheter. Denna port är ej standard men krävs då extra enheter skall kopplas in i den yttre kontakten, vilken är av typen "high density". En termineringsplugg krävs om ingen yttre enhet är ansluten.

SCSI 1 Upp till tre interna skivminnen (id=0, 1, 2). Dessutom upp till tre interna enheter (id=6, 5 och/eller 4) som monteras i fronten, t ex kassettstreamer eller WORM skivminne.

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
18	si2	SCSI 1, id 1: Intern winchester nr 2
20	si4	SCSI 1, id 2: Intern winchester nr 3
152	vt0,dat0	SCSI 1, id 4: Intern extra streamer,Video8,DAT
153	vt0n,dat0n	Icke återspolande streamer
26	worm0	SCSI 1, id 5: Intern WORM skivenhet
156	st0	SCSI 1, id 6: Intern kassettstreamer, 1/4 tum
157	st0n	Icke återspolande streamer
2	si6	SCSI 0, id 1: Extra skivminne, yttre nr 1
4	si8	SCSI 0, id 2: Extra skivminne, yttre nr 2
10	worm1	SCSI 0, id 5: Yttre WORM skivenhet
140	st1	SCSI 0, id 6: Yttre extra streamerenhet
141	st1n	Icke återspolande streamer





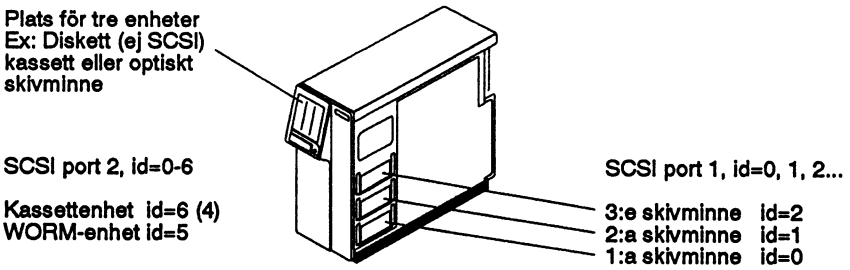
Datorsystem DIAB2430

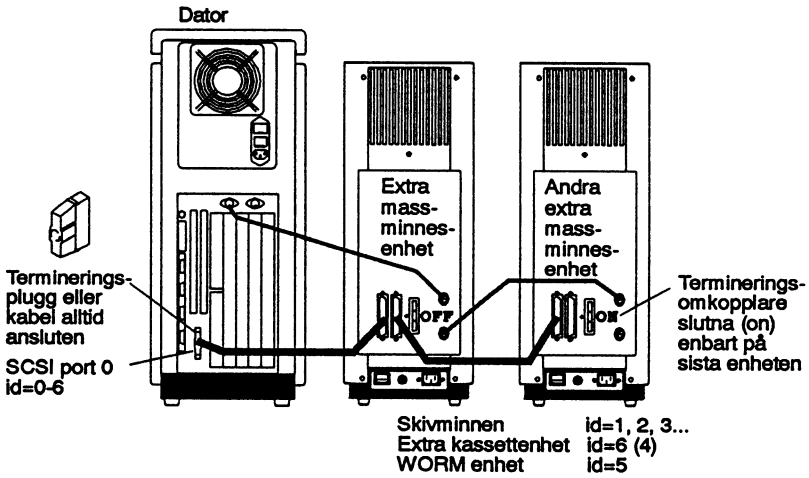
SCSI 0 (yttre) Fri för extra enheter. Enbart denna port finns i yttre kontakten, vilken är av typen "high density". En termineringsplugg krävs om ingen yttre enhet är ansluten.

SCSI 1 Upp till tre interna skivminnen (id=0, 1, 2).

SCSI 2 Upp till tre interna enheter (id=6, 5 och/eller 4) som monteras i fronten, t ex kassettstreamer eller WORM skivminne.

Minor	Namn	Port, ID och enhet
16	si0	SCSI 1, id 0: Intern winchester
18	si2	SCSI 1, id 1: Intern winchester nr 2
20	si4	SCSI 1, id 2: Intern winchester nr 3
168	vt0,dat0	SCSI 2, id 4: Intern extra streamer,Video8,DAT
169	vt0n,dat0n	Icke återspolande streamer
42	worm0	SCSI 2, id 5: Intern WORM skivenhet
172	st0	SCSI 2, id 6: Intern kassettstreamer, 1/4 tum
173	st0n	Icke återspolande streamer
2	si6	SCSI 0, id 1: Extra skivminne, yttre nr 1
4	si8	SCSI 0, id 2: Extra skivminne, yttre nr 2
10	worm1	SCSI 0, id 5: Yttre WORM skivenhet
140	st1	SCSI 0, id 6: Yttre extra streamerenhet
141	st1n	Icke återspolande streamer



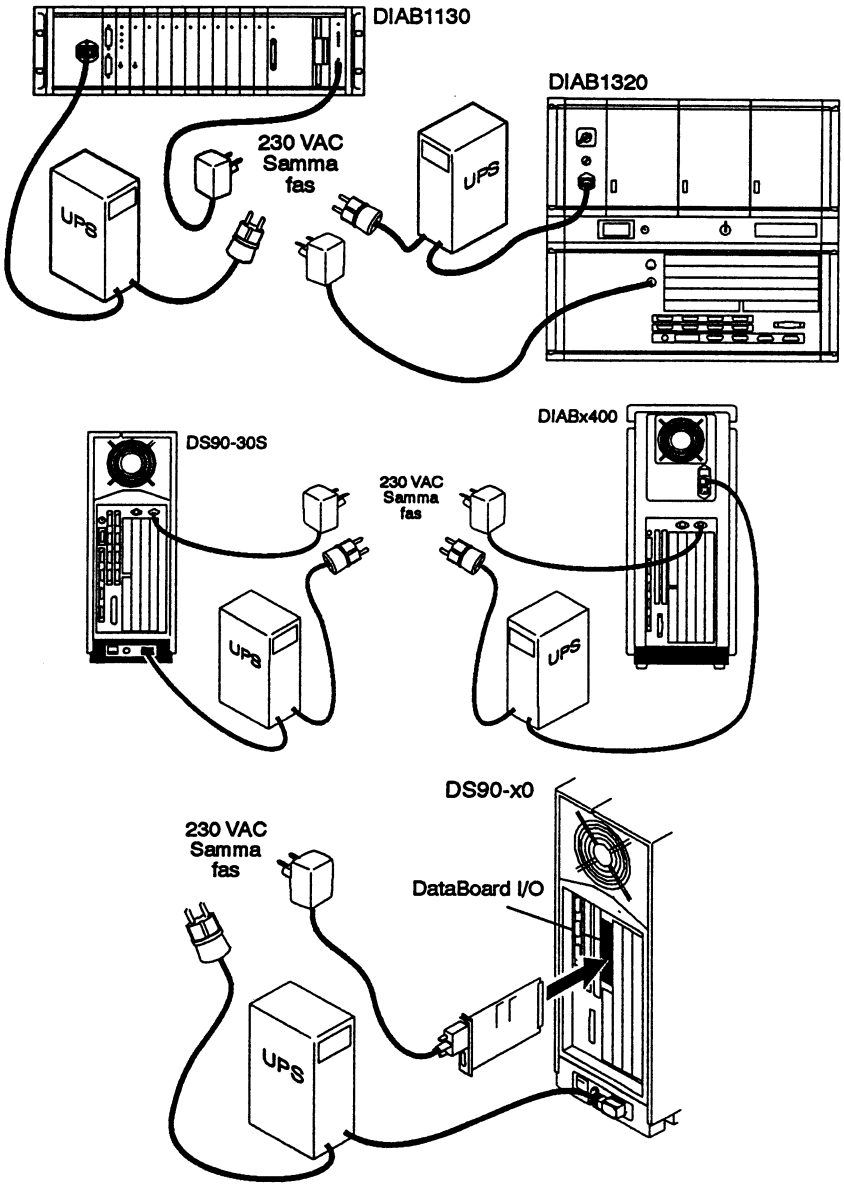


15.9.7 Hantering av kraftavbrott i olika datorer

Med en UPS ansluten till datorn (Uninterruptible Power Supply) skall även en avkänningssignal ges till datorn så att ett kraftavbrott kan detekteras. Systemet kan då kopplas ner efter en given tid (se */etc/mkcfig* i kapitel 5) om inte spänningen återvänder. Dessförinnan fås en oavbruten spänningsförsörjning från den anslutna UPS-enheten.

För datorer av rackmodell kopplas alltid två 230V-kablar in enligt installationshäftet för datorn, varav den ena används som avkänningssignal.

För övriga datorer ansluts en avkänningssignal från en separat adapter till datorn. Hur adaptern skall vara ansluten ses i följande bild.



Efter installation av adaptern för spänningsavkänning måste alltid adaptern vara ansluten för att datorn skall kunna användas. Se detaljer i **Installation Power Fail Indicator**.

15.9.8 Seriekontakter, stiftplacering

För serieanslutning finns DA15P-kontakter på alla datormodeller.

I handboken **Förberedelser för installation** finns detaljer om stiftplaceringen i kontakterna för varje dator, samt beskrivningar av de seriella kablar som används.

15.9.9 VME- och DataBoard-positioner i olika datorer

De olika datorsystemen har olika typer av interna expansionplatser för anslutning av VME-kort och DataBoard-kort.

I vissa enklare system ingår inte någon rack med expansionsplatser i grundsystemet, men dessa kan normalt byggas ut med sådana.

VME- och DataBoardkorten sätts in med spänningen avslagen och med korten vända enligt bilderna för respektive dator.

VME-positioner

VME-positionerna skall alltid fyllas från position VME-1, dvs från vänster (eller nerifrån i rackmonterade datorer). Om någon position måste lämnas tom av mekaniska skäl skall de byglingar som finns till höger om (eller ovanför) den tomma positionen i bakplanet slutas för att leda vissa signaler förbi den tomma positionen.

VME-positionerna har dubbla kontakter i de nyare modellerna (med 32-bitars buss) medan bara en kontakt används i andra modeller. VME-kort med endast en kontakt kan ändå användas i de nyare VME-positionerna.

DataBoard-positioner

Alla DataBoard-positioner kan användas för DataBoard I/O-kort men endast en position är anpassad för expansionskort till en yttre expansionsrack.

Normalt finns enbart en avbrottsnivå per position, vilket innebär att endast en kanal utnyttjas med kort som annars använder flera avbrottsnivåer. Inga DataBoard-kort som kräver DMA kan användas om inte annat sägs för respektive modell. I övrigt gäller vissa begränsningar enligt beskrivningen för olika datormodeller.

DS90-10 / 10E

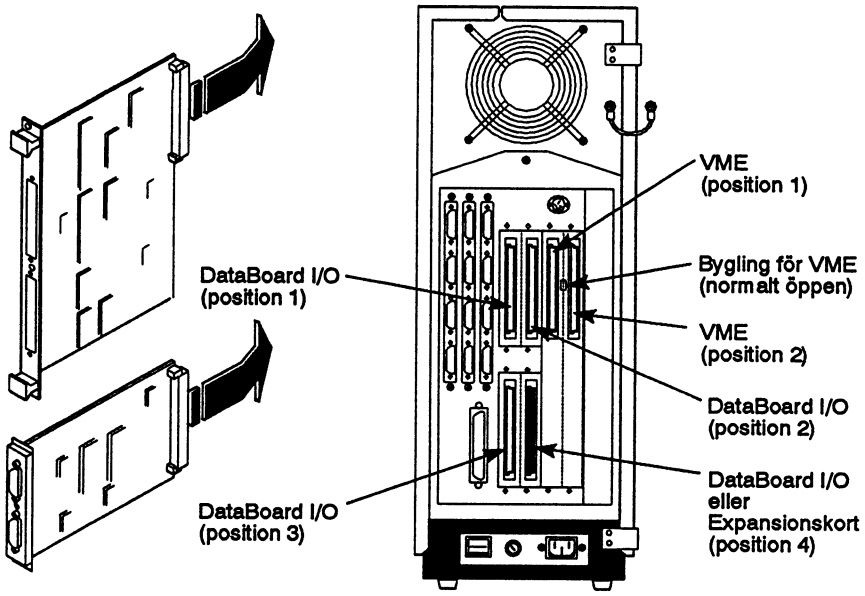
Interna racken med VME/DataBoard är ett tillval i DS90-10E.

VME-positioner: Enbart en VME-kontakt finns per position, varför kort som kräver 32-bitars kommunikation inte kan användas.

DataBoard-positioner: Även DataBoardkort med DMA kan användas i denna dator.

Om expansionskort används i nedre högra positionen får *inte övre vänstra* positionen användas för kort som använder avbrottssignalen.

Två kanaler kan i princip användas på ett DataBoardkort i nedre högra positionen, men då får *inte nedre vänstra* positionen användas för kort som använder avbrottssignalen.

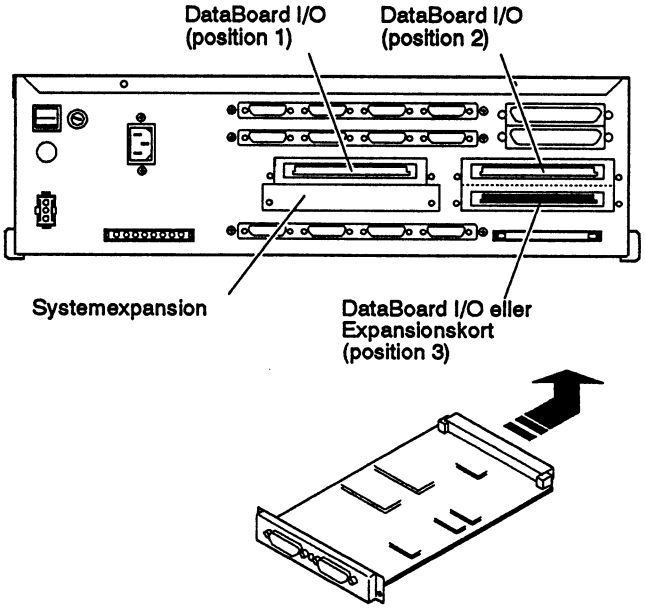


DS90-11

DataBoard-positioner: Även DataBoardkort med DMA kan användas i denna dator.

Om expansionskort eller två kanaler på ett DataBoardkort används i nedre högra positionen får *inte övre vänstra* positionen användas för kort som använder avbrottssignalen.

Observera! Nedre vänstra positionen i racken är en speciell systemexpansionsport och *inte* en DataBoardposition.

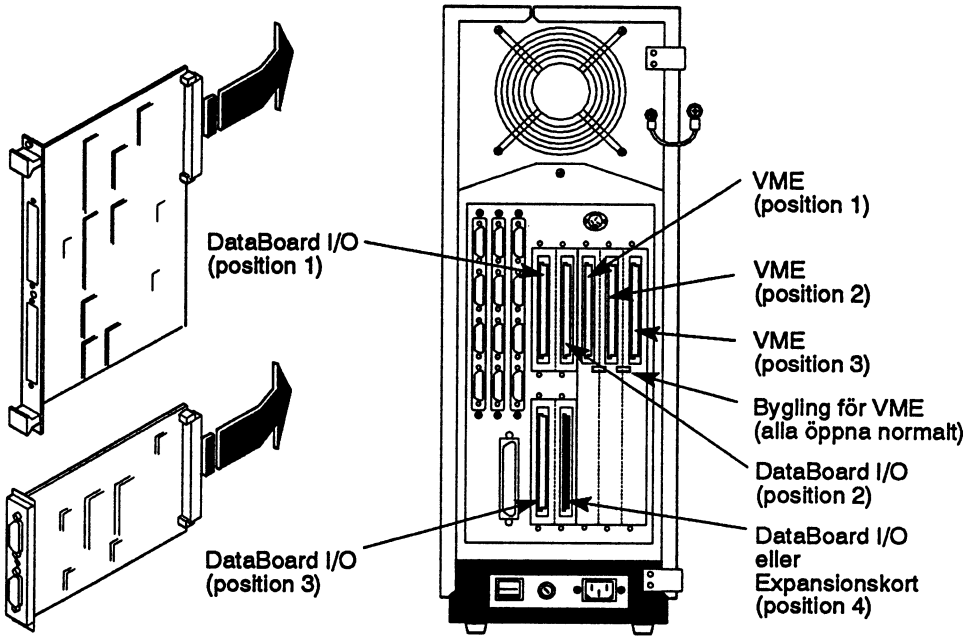


DS90-20 / 20E, DS90-21

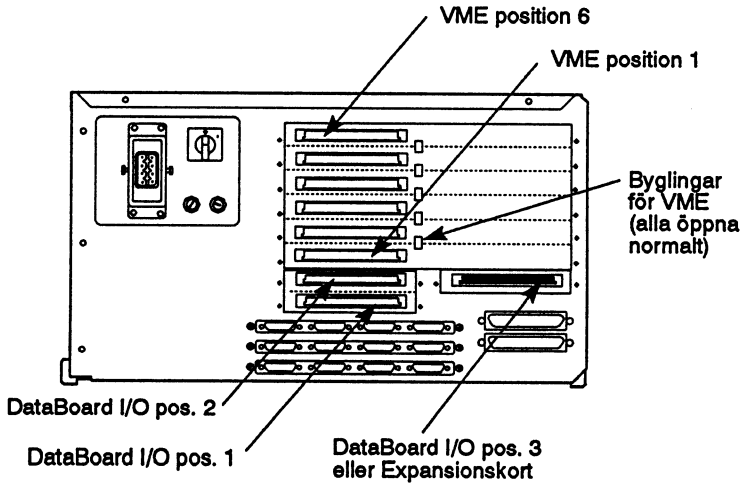
Interna racken med VME/DataBoard är ett tillval i DS90-20E.

VME-positioner: Enbart en VME-kontakt finns per position, varför kort som kräver 32-bitars kommunikation inte kan användas.

DS90-20/20E:



DS90-21:

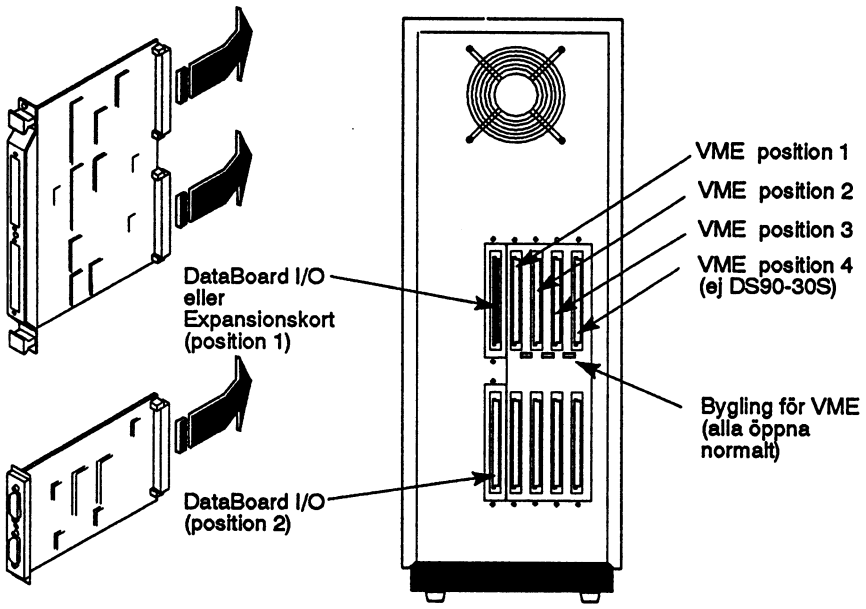


DS90-30 / 30S, DS90-31 / 41, DIABx4xx, DIABx3xx, DS101

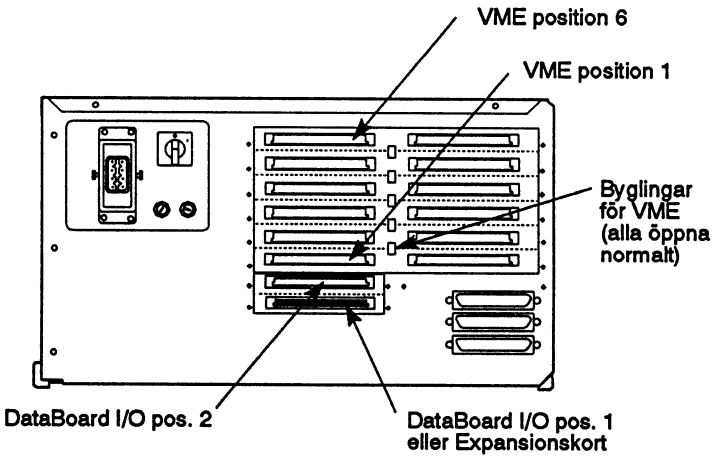
Interna racken med VME/DataBoard är ett tillval i DS90-30S, DIABxx20.

VME-positioner: Två VME-kontakter finns per position för användning av kort med upp till 32 address- och databitar.

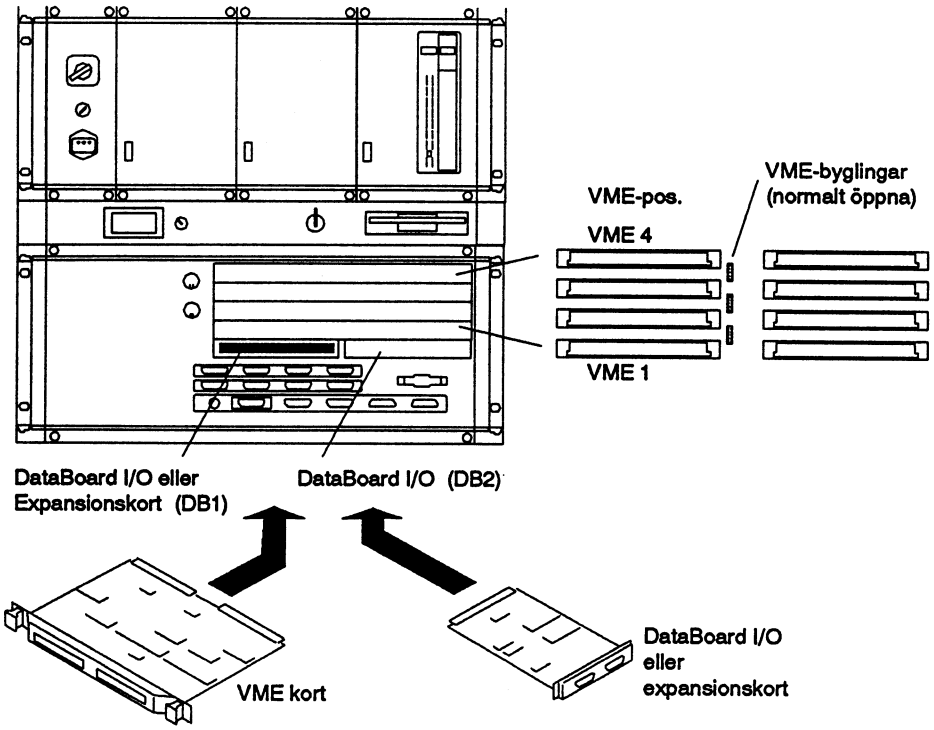
DS90-30 / 30S, DIAB1420, DIAB2420, DIAB2430



DS90-31, DS90-41:



DIAB1320:



DS101: