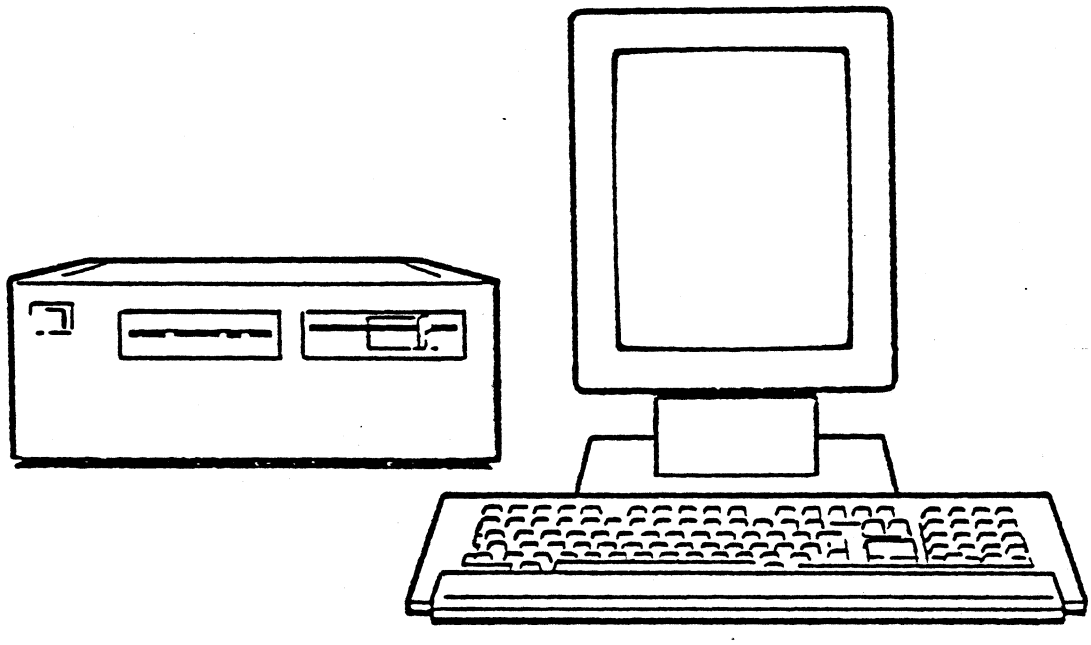
LUXOR Datorer

ABC 1600

ABCenix-kommandon
(Preliminär)



Art.nr. 66 78400-11

Förord

I denna handbok beskrivs alla kommandon till operativsystemet ABCenix som är implementerat på datasystemen ABC 1600 och ABC 9000. Operativsystemet ABCenix är kompatibelt med standard UNIX system III och innehåller samma funktioner. Men en del kommandon har flera nya optioner för att underlätta en del arbetsmoment.

Denna handbok gör inte anspråk på att ge grunderna inom UNIX och därmed ABCenix utan vi rekommenderar att dessa kunskaper hämtas från boken - "A user guide to the UNIX system" av Rebecca Thomas PhD och Jean Yates.

Handboken är upplagd i tre delar:
Del 1 Formatet på kommandon
Del 2 Beskrivning av kommandon
Del 3 Kommandosammanfattning

I del 1 beskrivs vilket format som valts för funktionsbeskrivningen av kommandon och den generella kommandosyntaxen. Dessutom förklaras en del av de uttryck som är speciella för ABCenix.

I del 2 beskrivs alla kommandon ingående i alfabetisk ordning.

I del 3 slutligen finns en kortfattad sammanfattning av alla kommandon.

Motala i juni 1985
Luxor Datorer AB

Innehållsförteckning

1 Formatet till kommandon

Kommandon inom ABCenix världen är lite speciella eftersom de med få undantag är korta och inte säger användaren vad de gör. I de flesta fall är kommandot ett sammandrag av flera ord, tex ls - list current directory.

I ABcenix skrivs alla kommandon med små bokstäver. Detta betyder att ett kommando som skrivits med versaler - stora bokstäver - inte kan behandlas av systmet.

Varje kommando beskrivs i dennna handbok med hjälp av en eller flera av följande underrubriker:

Namn	Anger namnet på kommandot och en mycket kortfattad beskrivning.
Syntax	Visar hur kommandot används. Här anges också möjliga optioner och argument.
Beskrivning	Ingående beskrivning av hur kommandot fungerar.
Option	Beskrivning över vilka optioner som finns och deras funktion.
Filer	Namn på de standard filer som påverkas eller kan påverkas.
Hänvisning	Refererar till snarlika kommandon.
Felmeddelande	Beskriver i vilka fall som felmeddelanden skrivs ut.
Anmärkning	Anger om det är något speciellt att tänka på.

Under syntax används följande konventioner:

- hakparenteser runt ett argument, visar att det inte är obligatoriskt att ange detta argument.
- Ordet namn anger att det här kan anges både ett filnamn eller ett bibliotek.
- När ett argument kan upprepas hur många gånger som helst betecknas detta med "...".

1.1 Kommandosyntax

Det vanliga formatet på ett kommando består av en sekvens av ord, där orden är skilda med ett eller flera blanktecken. Varje kommando avslutas med RETURN.

Första ordet är kommandot, de övriga är argument till kommandot. Argumentet får inte innehålla några blanktecken men före, efter och mellan argumenten får det vara hur många blanktecken som helst. Om ett argument innehåller blanktecken, måste något av tecknen " eller ' sättas runt argumentet.

Argumentet består i de flesta fall av tre saker:

Filnamn
Bibliotek — Namnet på den fil som kommandot ska manipulera på något sätt. Filnamnet kan bestå av upp till 14 tecken.

Option — Består av en eller två bokstäver, i de flesta fall föregående av ett minustecken, tex -al. Optionera ändrar kommandot eller specificerar exakt hur kommandot ska arbeta.

Uttryck — Ett uttryck består av en teckensträng som kommandot kan använda.

Den normala ordningen för angivna kommandon är:

kommando optioner uttryck filnamn/bibliotek

Men för att få reda på den exakta ordningen hänvisas till respektive kommando.

1.2 Speciella uttryck

Det finns en del specicella uttryck som är unika för ABC-enix. Vi har valt att behålla det engelska uttrycket i handboken och ger därför en förklarning här.

1.2.1 Super-User

En privilegierad användare som har obegränsad åtkomst till alla delar av systemet.

1.2.2 Standard Output

Systemet skickar resultatet av ett kommando till en fil som heter Standard output fil. Normalt är denna fil bunden till terminalen.

1.2.3 Standard Input

Standard Input är den plats varifrån programmet förväntar sig indata. Normalt får programmet standard input från tangentbordet.

1.2.4 Directory

Ett directory är en datafil som innehåller namn och pekare till platsen för varje fil i minnet. Man kan likna ett directory vid ett bibliotek.

I ABCenix finns det många nivåer med directories. Ett directory som ligger i ett annat directory kallas för ett subdirectory.

1.2.5 Special filer

Det är special filerna som står i förbindelse med de yttre enheterna så som terminal, radskrivare, massminne. Man kommer åt filerna på samma sätt som vanliga filer men med hänsyn tagen till de yttre enheteras restriktioner.

2 Beskrivning av ABCenix-kommandon

2.1 Följande ABCenix-kommandon finns:

basename	basename	Tar bort bibliotek ur sträng
cat	catenate	Skriver ut filer
chgrp	change group	Ändra gruppidentitet
chmod	change mode	Ändra tillståndsmod
chown	change mode	Ändra ägare
clock *	clock	Visar en analog klocka på bildskärmen
cmp	compare	Jämför två filer
copy	copy files	Kopiera grupper av filer
cp	copy	Kopiera filer eller bibliotek
date	print current time and date	Skriver eller sätter datum och tid
demo *	demonstrationsprogram	Kör ett litet demonstrations-program
dd	dd	Konventerar och kopierar en fil
df	disk free	Anger hur mycket ledig plats det finns på disken
du	disk usage	Information om använd diskarea
echo	echo arguments	Ekar argumenten
expr	expr	Kör argument som uttryck
false	false	Ger utgångsvärde skilt från noll
find	find	Letar upp filer
format	formatter för floppy disk	formatterar flexskivor
fsck	filesystem check	Kontroll av filsystem
fscl	filesystem clean	Testar om filsystemet på en disk är intakt
haltsys	halt system	Stänger filsystemen och stoppar CPUn
isamin	isaminitialize	Initieringsprog. för isam
kill	stop background processes	Avslutar bakgrundsprocesser
ln	make link	Skapa länk mellan filer
login	login	Inloggning till systemet
lpr	line printer spooler	Spooler för skrivare
ls	list current directory	Lista info om filer

mkdir	make a directory	Skapar nya bibliotek
mkfs	make filesystem	Konstruerar ett ABCenix-filsystem
mknod	make node	Skapar en nod ("deivce")
mkuser	make user	Lägger till en ny användare
mntchk	mount check	Mount på specificerad enhet
mount	mount filesystem	Lägger till ett filsystem
mv	move files	Flyttar och/eller byter namn på filer och bibliotek
ncu *	net call UNIX	Nätverksanrop i ABCenix
netman *	Network transport manager	..... på ABCenix-system
nice	put command in background	Kör komandon i bakgrunden och ändrar prioritet
nohup	no hang up	Kör i bakgrunden, no hang up
od	octal dump	Oktal dump
passwd	password	Ändrar lösenord
pr	print	Formatterar textfiler
ps	process status	Process status
pwd	print working directory	Listar väg från root till aktuellt bibliotek
rm	remove files	Ta bort filer
rmdir	remove directories	Ta bort tomma directories
rmuser	remove user	Ta bort en användare
rx *	remote execut	Kör ett kommando på en annan maskin
settimezon	set time zon	Sätter korrektion i förhållande till GMT
setspeed	setpeed	Sätter baudrate på tex printer printer kanal
setup	setup	Definierar funktionstangenter
sh	shell	Anropar kommandotolken i shell
shutdown	shutdown	Avslutar alla processer
sleep	suspend execution	Uppskjuter exekvering viss tid
stty	set tty	Sätter terminaloptioner
su	become super-user	Ändra användare eller bli superuser
sync	synchronization	Uppdatering av super-block
tar	tape archiver	Spara filer på arkivformat
test	test	Provar villkor
time	time a command	Tid att köra ett kommando
touch	update date of a file	Uppdatera filernas modifierigstid
true	true	Ger utgångsvärdet noll
tty	tty	Ger terminalens namn

umount	mount filesystem	Mountar filsystem som lagts till
wait	await completion of process	Väntar på att processen ska avslutas
wall	write to all users	Skriv till alla inloggade användare
who	who is using system	Vem är inloggad?
write	write	Skriver till annan användare

* = endast ABC 1600

Namn: basename - tar bort biblioteknamn ur pathname

Syntax: basename sträng (suffix)

Funktion: basename tar bort alla prefix som slutar med / och suffixet (om det finns något) i strängen och skriver resultatet på standard output. Resultatet är filens "basnamn", dvs filens namn utan föregående bibliotek och utan extension. Det används satt mellan e e inom shellprocedurer för att konstruera nya filnamn.

Det liknande kommandot dirname tar bort det sista elementet i strängen och skriver den resulterande pathen på standard output.

Hänvisning: dirname(C), sh(C)

Exempel: 1/ basename /usr/tf/brev.brv .brv

brev

Resultatet blir brev eftersom trunkering av suffixet .brv begärdes vid kommandoanropet

2/ basename /usr/tf/brev.brv

brev.brv

Namn: cat (catenate) - skriver ut filer

Syntax: cat (-s) (-u) filnamn ...

Funktion: Kommandot cat läser de filer som specificerats med filnamn och skriver ut dem på standard output i tur och ordning. Om en fil inte existerar kommer ett felmeddelande att skrivas ut och körningen av cat att avbrytas.

cat kan
. skapa nya filer; en i taget
. sammanfoga gamla filer
. lägga till en eller flera filer till en gammal fil, tillägget sker i slutet av filen
. skriva in ett helt nytt innehåll till en gammal fil eller lägga till det sist i filen
. man kan skapa en fil med ett helt nytt innehåll, från standard input, skriva över en gammal fil eller lägga till sist i en gammal fil

Varje tillägg som sker via standard input måste avslutas med ctrl D och detta tecken måste ges först på en ny rad.

Option:

-u Med option -u satt skrivs data ut direkt till önskad fil. I de fall där optionen inte är satt kommer det som skrivs via standard output att lagras i en 512 bytes buffer. Data skrivs inte ut till filen förrän denna buffer är full.

-s Med optionen satt skrivs inga varningar ut för filer som ej kan läsas.

Exempel:

1/ cat

Allt som skrivs på tangentbordet skrivs ut på standard output, dvs i normala fall terminalen. För att avsluta kommandot måste tecknet ctrl D anges först på en ny rad

2/ cat > prodikl

Allt som skrivs på tangentbordet, skrivs till filen prodikl. Varje rad kommer att skrivas in i prodikl exakt som den skrivs på tangentbordet. Men inget skrivs ut på filen förrän bufferten på 512 byte är full. Ibland kan det vara önskvärt att skriva direkt till filen, i detta fall ska option -u användas.

3/ cat prodikl

Innehållet i filen prodikl skrivs ut på standard output.

4/ cat prodikl kbesriv > tprkbtt

Sammanfogar innehållet i fil prodikl och fil kbesriv och skriver in resultatet på fil tprkbtt. Obs! Om det sedan tidigare finns något skrivit i tprkbtt, raderas detta, innan prodikl och kbesriv skrivs in.

5/ cat kbesriv < tprkbtt

Innehållet i filen tprkbtt skrivs in i filen kbesriv.

6/ cat prodikl kbesriv >> tprkbtt

Sammanfogar innehållet i filen prodikl och filen kbesriv och skriver in detta sist i filen tprkbtt, dvs innehållet i prodikl och kbesriv sammanfogas med innehållet i filen tprkbtt.

Namn: chgrp (change group) - ändra grupptillhörighet

Syntax: chgrp grupp fil ...

Funktion: Med kommandot chgrp ändras gruppidentiteten för en eller flera filer. Namnet kan antingen vara ett gruppnamn eller också en numerisk gruppidentifikation,båda finns i gruppidentifikationsfilen, /etc/group. Den numeriska gruppidentifikationen finns också i password-filen, /etc/passwd. Superuser bestämmer hur gruppnamn och gruppidentifikation ska se ut. När den numeriska gruppidentifikationen ändras, ändras den i både gruppid.-filen och i password-filen.

OBS! Endast super-user och ägaren kan ändra gruppidentifikationen på filer.

Filer: /etc/group /etc/passwd

Exempel: 1/ chgrp gh postf brevgh brevgg

Gruppnamnet till filerna postf, brevgh och brevgg ändras till gh, detta innebär att gruppen gh har rätt att använda de specificerade filerna.

2/ chgrp 20 postkl postad brevdt

Gruppidentifikationen till filerna postkl, postad och brevdt ändras till 20, detta innebära att gruppen 20 har rätt att använda de specificerade filerna.

Namn: chmod (change mode) - ändra behörighetsmod

Syntax: chmod mode filnamn ...

Funktion: Med detta kommando ändras behörighetsmod (the permission mode) för en eller flera filer eller bibliotek.

Tillståndsmoden för en användare, en grupp och alla andra kan ändras oberoende av varandra.

Ändringen av moden kan ske på två olika sätt, antingen i absolut eller symbolisk mod.

Mod: oktal siffra (absolut) eller <vem><vad> <vilket> ... (symbolisk)

Den symboliska moden är uppbyggd enligt följande:

<vem> a all. Definitionen på "alla" beror på användarens umask (se umask(C))
. Anges inte <vem> på kommandoraden är a default
. u user, dvs ägaren av filen g group, dvs övriga som tillhör samma grupp som ägaren o others, dvs alla övriga

<vad> + addera privilegier, utan att ändra något annat för <vem> - tag bort privilegier, utan att ändra något annat för <vem> = sätt privilegier för <vem>

<vilket> r read,tillstånd att läsa filen w write, tillstånd att skriva/radera filen x execute, tillstånd att exekvera filen tillstånd att söka i bibliotek s set uid/gid, sätt användarid (uid)/gruppid (gid) till användarens vid exekvering. OBS! <vem> måste vara satt till u (användare) eller g (grupp). u tar nuvarande fil/biblioteks u privilegier g tar nuvarande fil/biblioteks g privilegier. o tar nuvarande fil/biblioteks o privilegier.

Namn: chown (change owner) - ändra ägare

Syntax: chown ägare namn ...

Funktion: Med kommandot chown ändras ägaren till filen/-filerna. Ägaren är antingen ett logi-namn eller en numerisk ägaridentifikation, båda finns i passwordfilen, /etc/passwd.

OBS! Endast super-user och ägaren kan ändra ägare för en fil.

Filer: /etc/passwd

Exempel: 1/ chown rp postf

Ägaren av filen postf blir användaren rp.

2/ chown 108 beda oskar

Personen med ägare-id 108 blir ny ägare till filerna beda och oskar.

3/ chown gh *

På alla filer i aktuellt bibliotek ändras ägaridentifikationen till gh.

Namn: clock - visar en analog klocka på skärmen genom fönsterhanteraren.

Syntax: clock opitoner

Funktion. Lägger upp en analog klocka i ett eget fönster på skärmen. Som standard antas fönsterrubriken "Stockholm" och klockan tid = datorn tid. Programmet sköter all nödvändig signalhantering och omritning. Signal 16 används för omritning, signal 17 som terminering.

Optioner:

-h Fönsterrubrik. Ett sammanhängande ord. Standard är Stockholm

-t Tidszon. Anger tidsförskjutningen i timmar från datorns interna tid, som klockan ska visa

-n Startar klockan utan att den skapar sig ett eget fönster

Exempel:

1/ clock -h New York -t -6

Startar en klocka med rubriken New York och tiden förskjuten med 6 timmar före datorn tid.

2/ clock -h Helsinki -t 1

Startar en klocka med rubriken Helsinki och tiden förskjuten med en timme efter datorns tid.

Namn: cmp (compare) - jämför två filer

Syntax: cmp (-l) (-s) fil1 fil2

Funktion: cmp jämför två filer. Är de olika visas radnummer och byte för skillnaderna. Om fil1 sätts till -, används standard input.

Optioner:

-l Skriver bytenummer (decimalt) och skiljaktiga bytes (oktalt) för varje skillnad.

-s Ger endast en utgångskod. 0 för identiska filer, 1 för olika filer och 2 om någon fil saknas eller inte är tillgänglig. Detta kommando bör endast användas för att jämföra binärfiler; använd diff(C) eller diff3(C) för att jämföra textfiler.

Hänvisning: comm(C), diff(C), diff3(C) (Dessa ingår i Xenix Prógramutvecklings paket, Microsoft USA).

Namn: copy - kopiera grupper av filer

syntax: copy (-alnomradvtu) namn ... dest

copy (-a -l -n -o -m -r -a -d -v -t -u) namn ... dest

Funktion: Innehållet i ett eller flera bibliotek kopieras till ett annat bibliotek. Hela filsystem kan kopieras eftersom de bibliotek eller de specialfiler som ev. inte finns, skapas allt eftersom de behövs. De bibliotek och specialfiler som skapas på detta sätt, sätts i samma mod och får samma flaggor satta som orginalet. De bibliotek som redan finns vid destinationen (dest) behåller sin mod och sina satta flaggor.

Observera att kopiering kan ske från flera bibliotek på en gång, effekten är precis den samma som om kopieringen gjorts med ett bibliotek itaget.

Argumentet "namn" kan vara en fil, ett bibliotek eller en specialfil men den måste existera. Om "namn" inte är ett bibliotek kommer resultatet av copy att vara precis det samma som om kommandot cp användts.

"Dest" (destinationen) måste antingen vara en fil eller ett bibliotek som är skild ifrån "namn". Om både "namn" och "dest" är bibliotek kommer kommandot copy att kopiera en fil itaget till destinations biblioteket enligt de flaggor som är satta.

Option:

-a Innan kopieringen utförs ställs en fråga, denna måste besvaras med ett 'y' för att kopiering ska ske. Optionen -ad sätts automatiskt när -a sätts.

-l Kopiering sker endast i de fall där det inte är möjligt att använda länkar, dvs om filerna inte ligger på samma filsystem. Specialfiler och bibliotek kan inte länkas utan de kopieras.

-n Destinationsfilen måste vara ny. Om så inte är fallet, kan ingen kopiering ske. Optionen -n finns inte för bibliotek. För specialfiler är -n optionen alltid satt.

-o Denna option får endast super-user använda. När optionen sätts kommer filens ägaroch gruppidentifikation att ändras till samma som orginalfilens.

-m Varje kopierad fil kommer att få samma access och modifieringstid som orginal-filen. Om optionen inte är satt kommer modifieringstiden att sättas till den tid då kopieringen utfördes.

-r Alla bibliotek som påträffas kommer att undersökas. Om optionen inte är satt kommer alla bibliotek som påträffas att ignoreras.

-ad När ett bibliotek påträffas, får användaren en fråga om optionen -r ska anses vara satt, dvs ska det bibliotek som påträffats undersökas . Om frågan inte besvaras med ett 'y' i början ignoreras biblioteket.

-v När ett kommando körs med denna option satt kommer namnet på varje fil som kopieras att skrivas ut på terminalen.

-t Bevarar trädstrukturen hos de filer/bibliotek som kopieras. OBS! EJ UNIX KOMPATIBEL.

Exempel:

1/ copy -a /ds/frpt /ds/opqr

Innan varje fil kopieras, ställs en fråga. Om denna fråga besvaras med 'y' som första bokstav kommer filen ds/frpt att kopieras till filen /ds/opqr, annars kommer ingen kopiering att utföras.

2/ copy -n /ds/alfa /ds/beta

De filer som finns i båda biblioteken /ds/alfa och /ds/beta kommer inte att kopieras. De övriga filerna i /ds/alfa kommer att kopieras till /ds/beta.

3/ copy -r * tmp

Om tmp inte finns kommer först ett bibliotek att skapas med namnet tmp. Därefter kommer alla filer/ bibliotek som finns i nuvarande bibliotek (betecknat med *) att kopieras. När ett bibliotek påträffas kommer dess innehåll att kopieras till tmp, men något bibliotek skapas inte.

4/ copy -rt * tmpl

Om tmpl inte finns kommer först ett bibliotek att skapas med namnet tmpl. Därefter kommmer alla filer/bibliotek som finns i nuvarande bibliotek (betecknat med *) att kopieras. tmpl kommer att bli en exakt kopia av nuvarande bibliotek, *.

Namn: demo - ett fönsterdemonstrationsprogram

Syntax: Startas från fönsterhanterarmeny eller från directory "/usr/window/demo", programnamn: demo. Observera att man måste stå på /usr/window/demo när programmet körs!

Funktion: Kör ett fönsterdemonstrationsprogram som visar något av datorns grafiska kapacitet.

Filer: /usr/window/demo/vt Directory med bildfiler

Namn: cp (copy) - kopiera filer (eller bibliotek)

Syntax:

1 cp filnamnl filnamn2

2 cp filnamnl filnamn2 ... bibliotek

Funktion: Med syntax variant 1, skapar detta kommando en backup kopia, filnamn 2, av filnamn 1. Filen fill kommer inte att påverkas av kopieringen. Om filnamn 2 redan existerar kommer dess innehåll att raderas innan innehållet i filnamn 1 kopieras. Filnamn 2 behåller i detta fall sin gamla mod och ägaridentitet.

Om filnamn 2 inte finns skapas den och den kommer att få samma mod och ägaridentitet som filnamn 1.

Med syntax variant 2, kopieras en eller flera filer med sina orginalnamn till det specificerade biblioteket, detta bibliotek måste existera

Det går inte att kopiera från en fil, t ex mankpp, till sig själv, dvs mankpp.

Hänvisning: mv

EXEMPEL

Exempel:

1/ cp /ds/addmsk /ds/betamy

Om filen betamy redan existerar kommer dess innehåll att raderas innan filen addmsk kopieras. Skulle däremot filen betamy inte finns kommer denna fil att först skapas innan en kopiering kan ske. I båda fallen kommer innehållet i filerna att vara exakt lika efter kopieringen.

2/ cp addmsk datamsk rptydr atpydr /addera

Filerna kommer att kopieras till biblioteket /addera och filerna kommer att behålla sina orginalnamn.

3/ cp /gre/chapt* /hty/kap21

Alla filer i underbiblioteket /gre som börjar på chapt kommer att kopieras till underbiblioteket /hty/kap21

Namn: date - skriver eller sätter datum och tid

Syntax: date (-u) (mmddttmm(åå)) (+format)
date (-u) (ååmmddttmm(.ss)) (+format)

Funktion: Om bara kommandot date anges, skrivs det löpande datumet och tiden ut.
Med argumenten mmddttmmåå sätts datum och/eller tid. Systemet arbetar med GMT.

OBS! Det är bara super-user som får ändra datum.

Förklaring till argumenten,

åå = de två sista siffrorna i årtalet, default nuvarande år mm = månadens nummer, default nuvarande månad.

dd = dagens datum, default nuvarande dag. tt = timmar (24 -timmars system). mm = minuter/timme.

Format: Formatet styr hur utskriften skall se ut. Alla tecken utom "%" skrivs ut som i formatet. Skriver man en sekvens bestående av "%" följt av ett av nedanstående tecken, svarar date med dess motsvarighet. Date avslutar alltid med ny rad.

Sekvens	Date's motsvarighet
%%	%
%h	Ny rad
%t	Tabulering
%m	Månad - 01 till 12
%d	Datum - 01 till 31
%y	Två sista siffrorna i årtalet - 00 till 99
%D	Datum i formatet "mm/dd/yy"
%H	Timme - 00 till 23
%M	Minut - 00 till 59
%S	Sekund - 00 till 59
%T	Tid i formatet "hh:mm:ss"
* %g	Tidszon i formatet "GMT+-hh.mm"
%j	Julianskt datum - 001 till 366
%w	Dag i vecka - 0 (söndag) till 6 (lördag)
%a	Förkortad veckodag - Sun till Sat
%h	Förkortad månad - Jan till Dec
%r	Tid i formatet "hh:mm:ss AM/PM" - hh
* %G	Använd i fortsättningen GMT
* %L	Använd i fortsättningen lokaltid
* %z	Förkortad tidszon

(* = tillägg) Anges ej %G eller %L, används GMT.

Anges varken tid eller format skrivs tiden ut i formatet "mm/dd/yy hh/mm/ss".

För att sätta datum och tid från systemklockan används "date 'mudate'".

Option: -u Tillåts fortfarande, men programmet arbetar endast i GMT.

Felmeddelande: Felmeddelande skrivs ut om någon som inte är super-user försöker ändra datum eller om argumenten är felskrivna.

Exempel:

1/ date
Nuvarande datum och tid skrivs ut på skärmen, t ex
Wed Dec 7 18:41:36 GMT +1:00 1983

2/ Datum och tid sätts på följande sätt:
date 8407010800

3/ date '+ Lokaltid: %L %D %T %g %n GMT: %G %D %T'ger:
Lokal tid: 12/07/83 18:41:36 GMT+01:00 GMT: 12/07/83 17:41:36

Namn: dd - konverterar och kopierar en fil.

Syntax: dd (option=värde) ...

Funktion: dd kopierar den angivna inputfilen till angiven outputfil med eventuella konverteringar. Som default används standard input och standard output. Blockstorlek vid input och output kan väljas för att utnyttja rent fysisk kapacitet vid I/O.

Optioner:

Option	Värde
if=file	Input filnamn; standard input som default
of=file	Output filnamn; standard output som default
ibs=n	Blockstorlek n bytes på input (default är 512)
obs=n	Blockstorlek på output (default 512)
bs=n	Sätter blockstorlek på både in- och output och har högre dignitet än ibs och obs. Synnerligen effektivt om ingen omvandling anges, då ingen intern kopia behöver göras.
cbs=n	Bufferstorlek för konvertering
skip=n	Hoppar över n input records innan kopieringen påbörjas
seek=n	Söker n records från början av outputfilen innan kopieringen påbörjas
count=n	Kopierar bara n input records
conv=ascii	Omvandlar EBCDIC till ASCII
conv=ebcdic	Omvandlar ASCII till EBCDIC
conv=ibm	Något annorlunda omvandling från ASCII till EBCDIC
conv=lcase	Omvandlar bokstäver till gemena
conv=ucase	Omvandlar bokstäver till versaler

conv=swap	skiftar varje bytepar
conv=noerror	Avbryter inte exekveringen vid ett fel
conv=sync	Blankutfyller alla input records till ibs
conv="...,..."	Flera omvandlingar separerade med komman.

Där storlekar anges, förväntas ett antal bytes. Ett tal kan avslutas med k, b eller w för att ange multiplikation med 1024, 512 eller 2; ett talpar kan separeras med x för att ange en produkt.

ebs används enbart om omvandling till ascii eller ebcdic angivits. I det första fallet placeras cbs tecken i omvandlingsbufferten, omvandlas till ASCII, avslutande blanktecken trimmas bort och newline läggs till innan raden skickas till output. I det senare fallet läses ASCIItecken in i omvandlingsbuffern, omvandlas till EBCDIC och blanktecken läggs till för att fylla en output record av cbs storlek.

Efter utförandet ger dd antal hela och delvis fyllda in- och outputblock.

Hänvisning: copy(C), cp(C), tar(C)

Exempel:

1/ dd if=myfile of=newfile

Kopierar myfile till newfile. jmfr cp

2/ dd if=myfile of=newfile conv=ucase

Kopierar myfile till newfile samt konverterar alla gemena bokstäver till VERSALER.

3/ dd if=/dev/sf0 of=xfile count=1970

Kopierar från devicet /dev/sf0 till filen xfile, count indikerar att man ska läsa 1970 records med default värdet 512 bytes/-record.

4/ dd if=xfile of=/dev/sf0

Kopierar filen xfile till devicet /dev/sf0 Läsning sker tills dess att slut på filen nås.

5/ dd if /dev/sf0 of=xfile ibs=256 count=1970

Läser 1970 st record från devicet /dev/sf0 med record storleken 256 bytes/record till filen xfile med default värdet 512 bytes/record.

dd avslutar med att säga hur många record den läst in och hur många den skrivit.

1970+0 records in 985+0 records out

Observera att antalet skrivna är hälften av de lästa pga ibs=256.

6/ dd if=/dev/rmt0 of=outfile ibs=800 cbs=80 conv=ascii,lcase

Kommandot läser en EBCDIC tape, i block om tio 80-bytes EBCDICblock per record, till ASCIIfilen outfile.

Namn: df - anger antal fria diskblock

Syntax: df (-t) (-f) (filsystem ...)

Funktion: df skriver ut antal fria block tillgängliga för on-line filsystem genom att undersöka räknarna i superblocken. Ett eller fler filsystem kan ges som argument (t.ex. /dev/hd0 eller /dev/usr). Om argumentet filsystem icke anges, rapporteras allt fritt utrymme i samtliga anslutna filsystem till standard output. Lista på anslutna filsystem finns i /etc/mnttab.

Optioner: -t ger även totala antalet tilldelade block

-f ger endast fria block, dvs fria inoder visas ej

Filer: /dev/* /etc/mnttab

Hänvisning: fsck(C), mnttab(F)

Anmärkning: Se Anm. under mount(C).

Exempel: 1/ df

Listar hur mycket ledigt diskutrymme det finns på samtliga i systemet ingående filsystem.

2/ df /sf0

Listar hur mycket diskutrymme det finns ledigt på filsystemet /sf0.

Namn: dirname - ger bibliotekdelen av ett pathname

Syntax: dirname sträng

Funktion: dirname ger allt utom den sista delen av pathname i strängen och skriver resultatet på standard output. Om pathname bara har en komponent skrivs bara en punkt. Det används vanligtvis inom é é i shellprocedurer.

Det besläktade kommandot basename tar bort alla prefix som slutar med /, samt eventuella suffix, från strängen och skriver resultatet på standard output.

Exempel: 1/ NAME =édirname /usr/src/cmd/cat.cé

Sätter shellvariabeln NAME till /usr/src/cmd

2/ dirname /a/b/c/d

Skriver /a/b/c på standard output

3/ dirname fil.ext

Skriver en punkt på standard output

Hänvisning: basename(C), sh(C)

Namn: du (disk usage) - information om hur diskarean används, i 512-bytes block

Syntax: du (-s -a -r) (namn ...)
du (-sar) (namn ...)

Funktion: Kommandot du ger en summering över det antal block diskarea som aktuellt bibliotek upptar, detta inkluderar alla underbibliotek i det aktuella biblioteket. Om en fil eller bibliotek anges vid namn, kommer kommandot du att ge en summering över de antal block diskarea som denna fil/ bibliotek upptar. Anges inget namn andvänds aktuellt bibliotek.

De filer som är länkade till varandra kommer bara räknas en gång.

Option:
-s Anger det total antalet block, för alla filer.
-a Anger antalet block för var och en av filerna.
-r Normal undertryckning av felmeddelnade upphävs.

Anmärkning: Om det i ett bibliotek finns många länkar till filerna kommer kommandot du att räkna de accessade filerna flera gånger.

Exenpel: 1/ du /usr/hty/kap5
Anger det totala antalet block diskarea som underbiblioteket /usr/hty/kap5 upptar.

Namn: echo - ekar det som skrivs som argument

Syntax: echo (-n)(-e) (--) arg ...

Funktion: Kommmandot echo skriver ut de angivna argumenten med ett blankt tecken emellan och avslutar med en ny rad. Kommandot kan användas när felmeddelande skall produceras i shell program eller för att skriva ett konstant dataflöde till pipelines.

Option: -n Ingen ny rad efter utskrift. -e Skriver argumenten på standard error output. -- Skriver argumenten exakt så att ett argument som börjar med streck (t.ex. -n eller -e) kan anges.

Format: I argumentet kan man ange vissa specifikationer. Dessa består av ett "Ö" (versalt) följt av en av nedanstående sekvenser:

Sekvens	Utskrift
Öb	Backa markören ett steg
Öc	Samma funktion som optionen "-C"
Öf	Ny sida
Ön	Ny rad
Ör	Flyttar markören till början på raden
Öt	Tabulerar
ÖÖ	"Ö"
Ö0xxx	xxx är noll, ett, två eller tre tecken valda ur tecknen 0 till 7. Echo skriver ut det tecken vars ASCII-värde motsvaras av det oktala talet xxx (eller 0, om xxx ej anges). Observera att sekvenserna måste placeras mellan accenterna för att kommandohanteraren inte ska "förstöra" dem.

Exempel:

1/ echo Detta är en test,för att se hur echo fungerar.
Texten kommer att skrivas ut på skärmen och därefter ställer sig markören på ny rad.

2/ echo -n Denna option kommer att påverka markörens placering på skärmen.
Texten kommer att skrivas ut på skärmen. Markören ställer sig inte på en ny rad utan ställer sig direkt efter det skrivna meddelandet.

Namn: expr - utvärderar argument som uttryck

Syntax: expr argument

Funktion: Argumenten tas som ett uttryck. Efter utvärderingen skrivs resultatet på standard output. Delarna i uttrycket måste separeras med blanktecken. Tecken reserverade i shell måste "escapas". Notera att noll ges för att ange nollvärde och inte en nullsträng. Strängar innehållande blanktecken eller andra speciella tecken måste sättas inom apostrofer. Argument med integervärden kan föregås av ett gemensamt minustecken. Internt hanteras integers som 32-bitars tvåkomplementära tal.

Operatorer och nyckelord listas nedan. Uttryck ska sättas inom apostrofer i shell, eftersom många av tecknen som har särskild betydelse i shell också har särskild betydelse i expr. Listan är i rangordning, med operatorer med samma rang grupperade inom parenteser.

uttryck - uttryck Ger första uttrycket om det inte är null eller 0, annars ges andra uttrycket.

uttryck & uttryck Ger första uttrycket om inget av dem är null eller 0, ger annars 0.

uttryck (=, >, >=, <, <=, !=) uttryck Ger resultatet av en taljämförelse om båda argumenten är integers, ger annars resultatet av en alfabetisk jämförelse.

uttryck (+, -) uttryck Addition eller subtraktion av integer-argument.

uttryck (*, /, %) uttryck Multiplikation, division eller rest av integer-argument

uttryck : uttryck Jämförelseoperatorn : jämför första argumentet med det andra, som måste vara ett regelrätt uttryck; regelrätt syntax är samma som i ed(C), med det undantaget att alla uttryck ska "förankras" (dvs. börja med en caret (Ü)) och därför är caret inte något särskilt tecken i detta sammanhang (lägg märke till att i shell har caret samma betydelse som piping-symbolen (!).) Vanligtvis ger jämförelseoperatorn det antal tecken som är lika. Alternativt kan mönstret %(. . . . %) användas för att ge en del av det första argumentet.

Exempel: 1/ l a='expr ¤a + 1'

Adderar 1 till shellvariabeln a

2/ # 'For ¤a equal to either "/usr/abc/file" or just "/fil" '
'expr ¤a : '.*/%(.*%)' ! ¤a'

Ger sista segmentet av pathname (dvs. fil). Se upp för slashen ensam som argument: expr tar den som divisionsoperator (se anm. nedan).

3/ expr ¤VAR : '.*'

ger antalet tecken i VAR

Hänvisning: ed(C), sh(C)

Anmärkning: När argumentet har behandlats av shell, kan expr inte skilja på operander och operatorer på annat sätt än deras värden. Om ¤a är ett likhetstecken (=), ser kommandot

expr ¤a = '='

ut som

expr = = =

när det skickas till expr, där alla tas som operatorn =. Omskrivning enligt nedan tillåter jämförelse av likhetstecken.

expr X¤a = X=

Namn: false - ger utgångsvärde skilt från noll.

Syntax: false

Funktion: false gör inget annat än att ge ett utgångsvärde skilt från noll. falses motsats, true(C), gör inget annat än att ge noll som utgångsvärde. false används typiskt i shellprocedurer som

```
until false do command done
```

Hänvisning: sh(C), true(C)

Namn: find - letar upp filer

Syntax: find pathname-lista uttryck

Funktion: find går rekursivt igenom hela bibliotekshierarkin för varje pathname i pathnamelistan (dvs ett eller flera pathname) och söker filer som matchar ett Boolskt uttryck enligt listan nedan. I beskrivningen används argumentet n som ett decimalt heltal. +n betyder mer än n, -n mindre än n och n exakt n.

-name file	Sann om file matchar aktuellt filnamn. Vanlig shell argument syntax kan användas om den "citeras" (se upp för kolonet (:), frågetecknet (?) och stjärnan (*)).
-perm onum	Sann om filens tillståndsflaggor matchar det oktala talet onum exakt (se chmod(C)). Om onum har ett minustecken som prefix, blir fler bitar signifikanta (0177777, se stat(S)) och flaggorna jämförs: (flags&onum)==onum
-type x	Sann om filtypen är x, där x kan vara b för blockspecialfil, c för teckenspecialfil, d för ett bibliotek, p för en namngiven pipe, f för en vanlig fil eller n för en semafor eller delad datafil.
-links n	Sann om filen har n länkar.
-user uname	Sann om filen tillhör användaren uname.Om uname är numeriskt och inte finns som loginnamn i filen /etc/passwd antas det vara user ID.
-group gname	Sann om filen tillhör gruppen gname. Om gname är numeriskt och inte finns i filen /etc/group, antas det vara group ID.
-size n	Sann om filen är n block lång (512 bytes per block)
-atime n	Sann om filen accessats de senaste n dagarna.

-mtime n	Sann om filen modifierats de senaste n dagarna.
-ctime n	Sann om filen ändrats de senaste n dagarna.
-exec cmd	Sann om det exekverade cmd ger ett värde noll som utgångsstatus. Slutet på cmd måste förses med "escapat" semikolon. Kommandoargument "åå" ersätts av aktuellt pathname.
-ok cmd	Som -exec men den genererade kommandoraden skrivs ut med ett frågetecken först, och exekveras bara om användaren svarar med att skriva y.
-print	Alltid sann; skriver ut aktuellt pathname.
-newer file	Sann om den aktuella filen har modifierats senare än argumentfilen.
(uttryck)	Sann om uttrycket inom parenteser är sant (parenteser är speciella för shell och måste "citeras").

Ovanstående uttryck kan kombineras genom att använda följade operatorer (ordnade i sjunkande rang):

negation. Att negera ett uttryck görs med utropstecknet (!) som ICKE-operator

AND AND-operationen åstadkommes genom att placera uttrycken efter varandra.

OR OR-operationen åstadkommes genom att placera operatorn -o mellan uttrycken.

Exempel: find / %(-name a.out -o -name '*.o' %) -atime +7 -exec rm :: %;

Alla filer med namnen a.out eller *.o som inte använts den senaste veckan tas bort.

Filer: /etc/passwd /etc/group

Hänvisning: cpio(C), sh(C), test(C), stat(S), cpio(F)

Namn:	format - formatterar flexskivor
Syntax:	fprmat enhet
Funktion:	Formatterar, dvs skapar spår, sektorer mm på flexskivan, så att den blir användbar.
Exempel:	1/ /etc/foramt /dev/mfs
Observera:	Allt tidigare innehåll på skivan förstörs!
Anmärkning:	5,25-tums flexskivor måste formatteras före användning. 8-tums flexskivor är normalt formatterade vid leverans.

Namn: fsck (file system check) - kontroll av filsystemet

Syntax: fsck (-y -n -s -t) filsystem ...

Funktion: Kommandot fsck granskar de angivna filsystemen och rättar till de villkor som inte är korrekta interaktivt. fsck ignorerar villkorsflaggan 'filsystemet är rent'. Denna flagga sätts om kommandot fsck avslutas på ett framgångsrikt sätt.

För de filsystem som är korrekta kommer en utskrift att göras på antalet filer som ingår, antalet block som används och antalet fria block. Innan en ändring görs i ett filsystem som inte är korrekt, måste operatören konfirmera denna ändring. I de flesta ändringar som görs förloras data men dessa rapporteras. Defaultvärdet för fsck när ingen skrivning är tillåten är -n.

Om inget filsystem anges i kommandot fsck, kommer fsck att som defaultvärde läsa den lista på filsystem som finns i filen /etc/checklist.

Filsystem kontrolleras efter följande lista:

1. Volymbeskrivningsblocket integritet och blockstorlek.
2. Block som mer än en inod gör anspråk på.
3. Block som en inod utanför filsystemets område gör anspråk på.
4. Felaktig räkning av länkar.
5. Storlekskontroll: Felaktigt antal block i en fil. Directory-storleken är inte en multipel av 16 bytes.
6. Dåligt inod-format.
7. Block som inte har någon tillhörighet.
8. Directory-kontroll: Fil som pekar på en inod som inte är allokerad. Inod-numret ligger inte inom gränserna.
9. Volymkontroll: Mer än 65536 stycken inoder. Det finns fler inodsblock än det ingår i filsystemet.

10. Dålig bitmap.

11. Beräkningen av totala antalet fria block och/eller antalet fria inoder stämmer ej.

Nya filer och directories som är allokerade men inte har någon referens, kan med operatörens tillåtelse, åter läggas till filsystemet genom att de läggs i 'lost + found' directory. Det namn som används i detta fall är inodnumret. Enda restriktionen är att directory 'lost + found' ska vara placerati roten i det filsystem som kontrolleras och att det måste finnas ledigt utrymme där entries kan göras. Detta kan uppnås genom att innan kommandot fsck utförs, kopiera ett antal filer m h a 'losty + found' till directory och därefter ta bort dem.

Att kontrollera det råa devicet är alltid snabbare.

Optioner:

-y Antar att alla frågor besvaras med ja.
-n Antar att alla frågor besvaras med nej.
-s Ignorerar den aktuella bitmappen och konstruerar en ny. Filsystemet som undersöks bör inte finnas i systemet när denna option är satt eller också måste filsystemet "gömmas" på någon plats i systemet. Direkt efter att kommandot är utfört måste systemet bootas igen. Denna procedur är till för att ingen gammal, dålig kopia av bitmappen ska fortsätta att användas eller skrivas till det nya filsystemet.
-t Denna option anger att nästa argument ska användas som namn på en arbetsfil. Filen används om kommandot fsck inte har tillräckligt minnesutrymme för sina tabeller. Filen bör inte befinna sig på det filsystem som undersöks. När kommandot fsck är fullbordat kommer arbetsfilen att förstöras om det är en specialfil eller om filen inte existerade sedan tidigare. Om inte optionen -t är angiven och kommandot fsck behöver en arbetsfil, kommer den att begära att få en.

Filer:

/etc/checklist defaultlista över de filsystem som ska kontrolleras.

lost + found plats för nya filer

Hänvisning: mount
/etc/rc startupp-rutinen vilken används ofta av fsck.

Anmärkning: Inodnumren för . och .. bör kontrolleras om de är giltiga, för varje directory.

kontrolleras

lost + found plats för nya filer.

Namn: fscl (filesysten clean) - Testar om filsystemet på en disk är intakt

Syntax: fscl special

Funktion: Kommandot fscl kontrollerar om file-system-clean flaggan är uppdaterad på enheten special. Filsystemet på den angivna enheten förutsätts vara intakt om flaggan är uppdaterad. Kommandot returnerar status koder med följande betydelse:

0 = Flaggan file-system-clean är uppdaterad
1 = Flaggan file-system-clean är inte uppdaterad
2 = fscl kan inte öppna och/eller läsa enheten special

Exempel:

1/ fscl /dev/sf0
Om devicet /dev/sf0 är tillagt (mounted) och filsystemet är OK erhålls följande:
fscl: File system on device /dev/sf0 is ok! Är filsystemet inte ok erhålls:
fscl: File system on device /dev/sf0 is not ok! Om devicet inte är tillagt erhålls följande:
SA:drive offline fscl: Error at read

2/ fscl /dev/sil00
/dev/sil00 är ett device som inte är inlänkat i systemet, då erhålls följande:
fscl: Cannot open device /dev/sil00.

Namn: haltsys - stänger filsystemen och stoppar CPUn

Syntax: /etc/haltsys

Funktion: haltsys stänger filsystemen och stoppar processorn. haltsys får effekt omedelbart, så användarprocesser bör avslutas först. shutdown(C) rekommenderas för normal systemstängning: det varnar användarna, rensar upp saker och ting och kallar slutligen på haltsys. Använd haltsys direkt bara om något systemproblem hindrar körning av shutdown.

Hänvisning: shutdn(S), shutdown(C)

Namn: isamin - initieringsprogram för isam

Syntax: isamin

Funktion: isamin är ett initieringsprogram att användas när en ny ISAM datastruktur skapas. Programmet är interaktivt och frågar anvädaren om data för datastrukturen och fältbeskrivningar.

Filnamnet som ges som första svar kommer att förlängas med de tre tilläggen .DAT, .ISM och .STR varvid motsvarande filer skapas.

OBS! Om det filnamn som ges redan existerar som namn på en ISAM-struktur, kommer denna att raderas.

Index-record-längden bör sättas till enhetens (devicets) blockstorlek, för att erhålla snabbast möjliga läsning/ skrivning.

Exempel: ** Create ISAM-files Ver 1.0 **

* Create files *

Isam file name, without extension (max 8 characters) ? tst Data record length ? 20 Size of index records, recommended is same as block size on device (ex. 1024) ? 1024 Definition of fields in data record, The following types exist:

Code	Type	Sorting	Fieldsize
B	Binary	direct binary	selectable
A	Ascii	file 'sortorder.tab'	selectable
S	Int short	integer with sign	2 bytes
L	Int long	integer with sign	4 bytes
F	Float	floating point	4 bytes
D	Double	floating point	8 bytes

* Describe field nr 1 *

Field name (max 8 characters) ? name Starting at 1 Field type (B,A,S,L,F,D) ? a Field size ? 13 Should the field be indexed (y/n) ? y Duplicates allowed (y/n) ? y * Describe field nr 2 *

Field name (max 8 characters) ? phone Starting at 14 Field type (B,A,S,L,F,D) ? a Field size ? 7 Should the field be indexed (y/n) ? n

Namn: kill (stop background processes) - avslutar bakgrundsprocesser

Syntax: kill (-signal) processid ...

Funktion: Kommandot kill avslutar en process som körs i bakgrunden. Kommandot skickar, som defaultvärde, signal nummer 15 till den specificerade processen. En del processer ignorerar denna signal , dessa processer kan avslutas med signal -9, det säkraste sättet att ta död på en process.

Processid är det nummer som systemet skriver ut vid uppstarten av processen. Detta nummer är den enda identifikation som finns till processen. Processid kan fås fram med hjälp av kommandot ps. Om 0 anges som processid kommer alla medlemmar i processgruppen, dvs alla processer som är aktiverade under nuvarande login, att signaleras.

De processer som avslutas måste tillhöra användaren om inte användaren är super-user.

OBS! Användarens egen loginprocess kan avslutas med kill, detta innebär att användaren kommer att bli utloggad ut ur systemet. Men denna process kan bara avslutas med signal nummer 9.

Exempel:

1/ kill 1076
När bakgrundsprocessen startades upp, meddelade systemet att denna process har processid 1076. I och med att detta processid anges, kommer denna bakgrundsprocess att avslutas.

2/ kill -9 107
Bakgrundsprocessen med nummret 107 kan inte avslutas med enbart kill, därför måste signal -9 skickas till processen för att den ska kunna avslutas..

Namn: ln (make link) - skapar en länk mellan filerna

Syntax: ln filnamn1 filnamn2

Funktion: Kommandot ln skapar en länk till den existerande filen som angetts vid filnamn1. Länken benäms med det namn som angetts vid filnamn2 eller om detta inte har angivits, får länken samma namn som sista delen av filnamn1 och länken placeras i det aktuella biblioteket.

En länk är en entry till ett bibliotek som refererar till en fil. En fil kan ha flera länkar till sig. All information om filen kommer att behållas, såsom storleken på filen, moden den arbetar i, etc. Det går inte att särskilja en länk från orginal entry till biblioteket.

Alla förändringar av filens innehåll är helt oberoende av om filen har ingen, en eller flera länkar och därmed flera namn.

Information om hur många länkar som finns och vilka filer som är länkade med varandra kan fås med ls kommandot.

OBS! Det är förbjudet att göra en länk till ett bibliotek eller att skapa länkar mellan olika filsystem.

Hänvisning: rm

Exempel:

1/ ln /pal/hty/chap1
Skapar en länk mellan aktuellt bibliotek och filen chap1 i biblioteket /pal/hty. Den nya filen får namnet chap 1.

2/ ln /pal/hty/chap5 kap5
Skaper en länk mellan aktuellt bibliotek och filen chap5 i biblioteket /pal/hty samtidigt som ett andra namn skapas till filen chap5. Det nya namnet är kap5. Båda namnen har samma status och båda namnen kan användas.

Namn: login - inloggning till systemet

Syntax: login (användarnamn)

Funktion: Detta kommando används av en användare för att logga in sig i systemet eller när helst en användaren vill byta till en annan. Om inget argument anges frågar login efter användarnamnet. När rätt login och password har angetts, är användaren inne i systemet.

Vid inloggning uppdateras alla beräkningsfiler och användaren informeras om det finns någon post eller meddelande.

login kommandot initierar användaroch gruppid. och det bibliotek som tillhör användaren. Därefter exekveras antingen processen shell (/bin/sh) som kommandointepretator eller den process som finns angiven i passwordfilen (/etc/passwd).

Filer:

/etc/utmp	konto
/usr/adm/wtmp	konto
/usr/mail/*	post
/etc/motd	dagens meddelande
/etc/passwd	password filen

Felmeddelande: Felmeddelande skrivs ut om felaktigt passwordnamn anges.

Exempel: 1/ login
När inget namn anges efter login, frågar systemet efter det genom att skriva: login:. Därefter kommer systemet att fråga efter password.

Namn: lpr (line printer spooler) - spooler för skrivare

Syntax: lpr (-r -c -m -n) filnamn

Funktion: Kommandot lpr ställer filer på kö för utskrift. Om ingen fil anges läser kommandot från standard input.

Option:

-r Tar bort filen efter det att den har ställts i kö.
-c Kopiera filen för att förhindra att några ändringar sker innan utskriften.
-m Rapportera via mail när utskriften är klar.
-n Rapportera inte när utskriften är klar, default.
-tty Speficerar vilen sorts skrivare som ska användas. Om det inte specificeras kommer default att användas.
-pn Minskar skrivarprioriteten med n, för super-user kan n bli negativt.
-hn Anger antalet försättsblad.

Exempel:

1/ lpr - r myfile
Skickar filen myfile till spoolkön, när det är klart tas filmen myfile bort ur aktuellt bibliotek.

2/ lpr -c myfile
Skapar en kopia av myfile till spoolkön, detta sker för att inga ändringar ska ske i dokumentet före utskrift. Optionen -c är default i DNIX.

3/ ls -l ö lpr
Skickar resultatet av ls via en pipe till lpr som i sin tur lägger det på spoolkön.

Namn: ls (list current directory) - listar information om filerna och innehållet i biblioteken.

Syntax: ls (-e -l -t -a -s -d -r -u -c -i -f -g) namn...

ls (-eltasdrucifg) namn ...

Funktion: Listar innehållet i de bibliotek som anges som argument och/eller den begärda informationen om de filer som anges som argument. Utskriften listas i alfabetisk ordning om ingen option anges. Om kortformen l andvänds kommer informationen alltid att listas som om ls -l hade angets (långt format).

Option:

-e Listar alla underbibliotek. -l Listar på långt format, dvs följande information kommer att skrivas ut:
* mod
* antal länkningar
* grupp
* storlek i bytes
* tid för senaste uppdatering
* directory/filnamn

Om filen är en specialfil, kommer det i storleksfältet istället att anges det största och minsta numret på den yttre enheten.

-o Samma som -l, förutom att grupp ej anges.
-g Samma som -l, förutom att ägare ej anges.
-t Listar efter uppdateringstiden, senaste först.
-a Listar alla entries. Normalt listas ej filer vars namn börjar med en punkt (.).
-s Anger antalet 512-bytes block för varje fil/bibliotek, inkl. antalet indirekta block.
-d Namnet på biblioteket skrivs ut (används ofta tillsammans med option -l för att få statusen på filen/biblioteket). Innehållet listas ej.
-r Listar i omvänd ordning.
-u Listar efter accesstiden, senaste först.
-c Listar efter i-nodens sista modifieringstid istället för efter filens sista modifieringstid.
-i Skriver filens i-nummer i första kolumnen på utskriften.
-f Varje argument som anges uppfattas som ett bibliotek. Skriver ut filnamnen i den ord-

ning som de förekommer i biblioteket.
Optionerna -l,-t,-s och -r tas bort och optionen -a sätts.

-g Anger gruppidentitet istället för ägaridentitet vid listning med -l .

Filer: /etc/passwd Användarid.
/etc/group Gruppid.

Exempel:

1/ ls -l
Listar innehållet för innevarande bibliotek på långt format.

```
drwxrw-r-- 3 karl    45  Nov 10 09:22
                         Manual
-rwxr-xr-x 1 oskar   189 Jan 18 11:09 Kap1
-r-xr-xr-x 1 putte   256 Jan 22 14:32 Kap2
```

Förklaring:

drwxrw-r--	d anger att det är ett bibliotek (i denna position anges teckenet - om det är en fil).
-rwxrw-r--	anger den mod som biblioteket arbetar i. rwx anger ägarprivilegier, rw- anger gruppprivilegier och r-- anger alla övrigas privilegier.
3	Antalet länkningar till filen, i detta fall Manual.
karl	Ägaren till filen Manual.
45	Antalet tecken i Manual.
Nov 10	Datum för senaste uppdatering.
09:22	Tid för senaste uppdatering.
Manual	Filnamn.

2/ ls -lt
Listar innehållet efter senaste uppdateringstid för innevarande bibliotek på långt format.

```
-rwxr-xr-x 1 oskar  189  Jan 18 11:09 Kap1
-r-xr-xr-x 1 putte  256  Jan 22 14:32 Kap2
drwxr-xr-x 3 karl    45  Nov 10 09:22
                         Manual
```

3/ ls -a

Alla entries till innevarande bibliotek listas, även de dolda, på följande sätt.

```
.               ( . = nuvarande bibliotek)
..              (.. = föräldrar bibliotek)
                Manual Kap1 Kap2
```

4/ ls -ld
Skriver ut statusen på det specificerade biblioteket, om inget namn anges tar systemet nuvarande bibliotek.

drwxr-xr-x 3 karl 45 Nov 10 09:22 Manual

5/ ls -i

1946 myfile
Ger bibliotekets/filens i-nummer. i-nummer är filens index-nummer som pekar ut var den startar på disken.

6/ ls -u /Manual
Listar filer och bibliotek efter accesstiden. Den senast accessade filen/biblioteket skrivs först.

7/ ls -f myfile
myfile
myfile: total 0:
Försöker att lista myfile som ett bibliotek, om myfile skulle ha innehållit något, görs ett försök att lista det som medlemmar i biblioteket.

8/ ls -gl
-rw-r--r-- 1 other 58 Aug 7 09:58 myfile
Ger ett långt format av ls, med ägaren till biblioteket/ filen utbytt till vilken grupp biblioteket/filen tillhör.

Namn: mknod - gör speciella filer

Syntax: /etc/mknod namn (c) (b) primär sekundär

/etc/mknod namn p

Funktion: mknod gör plats i bibliotek och motsvarande inod för en specialfil. Första argumentet är filens namn. I det första fallet är det andra argumentet b om specialfilen är av blocktyp (diskar, tape) eller c om den är av charactertyp. De sista argumenten är tal som anger primär enhetstyp och sekundär enhetstyp (t.ex unit, drive eller linjenummer) och kan vara både decimala och oktala.

Tilldelningen av enhetsnummer är specifik för varje system.

mknod kan också användas för att skapa namngivna pipes med optionen p.

Bara super-user kan använda den första formen av syntax.

Hänvisning: mknod(S)

Namn: mntchk - mount på specificerad enhet

Syntax: mntchk (fhnd) special bibliotek

Funktion: Kommandot mntchk lägger till ett filsystem till ett bibliotek med hjälp av en filhanterare.

Argumentet fhnd specificerar den filhanterare som ska användas tillsammans med kommandot. Vid special anges den enhet som filsystemet befinner sig på och vid bibliotek till vilket bibliotek som filsystemet ska flyttas.

Mntchk kontrollerar med hjälp av kommandot fscl om filsystemet är intakt. Om så inte är fallet körs fsck innankommandot mntchk körs.

Exempel:

1/ /etc/mntchk /hnd/fh00 /dev/sf0 /sf0 Kommandot lägger till ett filsystem med filhanteraren fh00 i devicet /dev/sf0 via biblioteket /sf0.

2/ /etc/mntchk
Kommandot ger en listning av vilka filsystem som är tillagda i DNIX. Detta kommando kan vara lämpligt att göra innan man lägger till ett filsystem eller ett kommando som ska använda devicet för att kontrollera att det är ledigt.

3/ /etc/mntchk /hnd/abcfh /dev/sf0 /sf0
Kommandot lägger till ett filsystem med filhanteraren för ABC-datorer i devicet /dev/sf0 via biblioteket /sf0.

Namn: mkdir (make a directory) - skapar nya bibliotek

Syntax: mkdir bibliotek ...

Funktion Med kommandot mkdir skapas ett eller flera bibliotek. För varje bibliotek som skapas sätter systemet en entry för '.' och en för '..'. Namnet '.' är en pseudonym för biblioteket självt och namnet '..' är en pseudonym för föräldrarbiblioteket. Alla bibliotek innehåller dessa entries och det är bara super-usern som har tillåtelse att ta bort dessa.

De skapade biblioteken får en mod beroende av senaste mask satt av umask. OBS! Super-usern kan ändra detta defaultvärde.

OBS! mkdir kräver tillstånd att skriva i föräldrar-'directory'.

Hänvisning: rmdir, umask

Felmeddelande: mkdir ger exitstatus skilt från noll och skriver ut ett felmeddelande om något eller några bibliotek inte kan skapas, annars ges exitstatus noll.

Exempel: 1/ mkdir Progex
Skapar ett nytt underbibliotek till nuvarande bibliotek. Om detta inte är möjligt kommer ett felmeddelande att skrivas ut, i övrigt kommer promten att skrivas ut när systemet är klart. (För att förvissa sig om att biblioteket har skapas, skriv ls Progex.)

Namn: mkfs (make filesystem) - konstruera ett DNIX filsystem

Syntax mkfs optioner ... ofile

Funktioner: Kommandot mkfs konstruerar ett filsystem genom att den skriver till filen ofile enligt de specificerade optionerna.

mkfs lägger alltid till följande filer:

lost + found
directory som används av kommandot fsck.

systemfiles
directory som innehåller följande entries:
bitmap: volym bitmap
badspots
inodlista: fil för volymens inodlista
swaparea: systemet swap area (om det finns någon)
volumedescr: volym beskrivningsfilen

Användningen av dessa filer är begränsad.

Option: Nedanstående optioner är tillåtna.

- adress namn Kopiera filens namn till den angivna adressen. Det utrymmet som krävs reserveras i bitmappen. Denna option kan användas t ex vid boot-procedurer.

- b bstorlek Sätter volymens blockstorlek till bstorlek. Bstorleken måste vara en multipel av 2 i området 256 - 16384.

- d Sätter debug mod. Alla interna aktiviteter kommer att skrivas ut på standard output.

- n nmax Allokerar bara nmax utrymme för inoder på volymen. Default är 1/32 av volymstorleken. Detta segment är garanterat kontinuerligt. Om fler inoder behövs kommer inodlistan att expanderas på precis samma sätt som en vanlig fil. Sammanlagt kan maximalt 65500 I-noder allokeras.

- r directory I det nya filsystemet sätts directory till root directory. Alla filer och directories som tillhör kopieras rekursivt till det nya filsystemet. Filerna kopieras med sin originalstatus och modifieringstider, allt utom /dev/files, vilken

innehåller device parametrar för värdsystemet. Den nya device-statusen kommer att läggas till efter kopieringen. gen.

- s sstorlek Allokerar en kontinuerlig swaparea på volymen med storleken sstorlek. Denna area behövs endast på systemvolymen.

- v vstorlek Sätter volym storlek till det i vstorlek specificerat antal block.

- x xfil Läser från xfilen badspot definitionstabellen. Varje badspot beskrivs som en entry på formen m, n , där m är startblocket och n är längden (i block) på badspot.

Filer: Inga

Hänvisning: fsck

Anmärkning: Blockstorleken har defaultvärdet 512 bytes. Volymstorleken och målfilen måste specificeras, på ena eller andra sättet. Filen ofile får inte tillhöra den kopierade hierarkin.

Exempel: 1/ /etc/mkfs -d -v 980 -b 1024 /dev/sf0
/etc finns med därför att kommandot finns under det biblioteket.
/mkfs kommandot självt

-d Alla händelser ska loggas på standard output.

-v 980 OBS! blankslaget, detta är storleken på floppyn i antal block.

-b 1024 OBS! blankslaget, detta är den blockstorlek som ska användas vid initieringen.

/dev/sf0/ är det device där floppyn sitter som ska initieras.

Namn: mkuser (make user) - lägger till en användare

Synatx: mkuser> (andvändarnamn)

Funktion: mkuser lägger till en ny användare genom att uppdatera filerna /etc/passwd och /etc/group samt att skapa ett hembibliotek och en .profile fil för den nya användaren. Kommandot finns under biblioteket /etc, därför stå i eller ladda det från biblioteket /etc när du ska lägga till en användare.

OBS! Kommandt kan endast köras av super user (root).

Kommandot kommer att ställa ett antal frågor som är beskrivna nedan, frågorna som ställs är här understrukna.

Användar-namn: - Namnet på den nya användaren (user name). Om detta redan är använt ställs frågan om och man får ge ett nytt användarnamn. Slår man bara en return vid denna fråga avbryts kommandot. Namnet kan anges direkt på kommandoraden, varvid denna fråga hoppas över.

Användar-id: - Den nya användarens användar-indetifikation (user id). Detta måste vara ett heltal större än eller lika med 1. Om den givna indentifikationen redan är andvänd kommer frågan att ställas om. Genom att endast trycka på return fås första lediga indentifiakations-nummer.

Kodord: - Kodord (password) som ska andvändas vid "login". Endast return gör att inget kodord behöver anges vid "login".

Grupp-namn: - Namnet på den grupp som användaren ska tillhöra (group name). Genom att enbart trycka på return så antas grupp-namnet "other". För att lista alla använda grupper, svara med ett "?".

Grupp-id: - Grupppens grupp-identifikation (group-id). Funktionen är den samma som vid inmatningen av användar-id. Om enbart return har slagits vid föregående fråga kommer denna fråga att hoppas över och ges grupp-id 1.

Kodord: - Kodord som ska användas vid byte av grupp-tillhörighet. Endast return gör att inget kodord behöver anges vid byte av grupp. Om endast return angets vid frågan om Grupp-namn kommer denna fråga att hoppas över.

Extra användar-beskrivning: - Godtycklig sträng som dock ej får innehålla kolon(:).

Start bibliotek: - Det bibliotek (directory) som ska vara användarens hem-bibliotek. Komplett namn (full pathname) måste anges t.ex /usr/olle. Om bibloteket redan finns får man verifiera sitt svar. Om man endast anger return kommer kommandot att avbrytas.

Start-program: - Det program som andvändaren ska hamna i direkt efter inloggning. Om man endast trycker på return så kommer /bin/sh att andvändas. I annat fall, om det givna programmet ej existerar får man verifiera sitt svar. OBS ett komplett namn på programmet måste anges.

Terminal-typ (vt100): - Den terminaltyp som andvändaren förvänatas arbeta vid. I .profile filen kommer variablen TERM att sättas till angivet värde, om enbart return sätts den till värdet inom paranteserna. sättas till i .profile filen.

Filer: /etc/group /etc/passwd

Exempel:

1/ /etc/mkuser

Alla ovan beskrivna frågor måste besvaras.

2/ cd /etc ; mkuser olle

Lägger till en ny användare med namnet olle. Alla frågor måste besvaras utom den första.

Namn: mount, umount (mount filesystems, demount filesystems) - lägga till eller ta bort filsystem

Syntax: /etc/mount (thnd) special namn (-r)
/etc/umount special

Funktion: Om inget argument anges efter mount, skrivs en lista ut på de yttre enheter som är anslutna till systemet.

Kommandot mount informerar kärnan i D-NIX att det existerar filsystem som ska inkorporeras med det accessbara filsystemet. Med argumentet special anges namnet på den yttre enhet som innehåller det filsytem som ska läggas till. Det andra argumentet namn anger namnet på det bibliotek till vilket filsystemet ska kopplas. Detta argument måste vara ett bibliotek, det måste redan existera och vara tomt. Enda undantaget från denna regel är om rooten till det filsystem som ska flyttas, inte är ett bibliotek. I detta fall kommer rooten att få samma namn som det filsystem har som läggs till.

OBS! När ett filsystem har lagts till med kommandot mount måste kommandot umount användas innan den yttre enheten kan tas bort.

Kommandot umount talar om för systemet att det filsystem som lagts till från den yttre enheten, ska tas bort. Ett filsystemet kan inte flyttas om någon fil är öppen eller om någon arbetar i något av biblioteken.

En flexskiva eller en magnettape som innehåller ett filsystem som inte är ett giltligt ABCenix filsystem kan inte läggas till men ABCenix har tillgång till filsystemet..

Option: -r Det filsystem som flyttas, ska bara vara läsbart.

Filsystem som ligger på magnettape eller fysiskt skrivskyddade enheter måste läggas till som endast läsbara. I annat kommer fel att genereras när accesstiden ska uppdateras, oavsett om ett skrivförsök har gjorts eller inte.

Filer: /etc/mtab mount lista

Hänvisning: mntchk

Anmärkning: Följande exitkoder skickas tillbaka till systemet efter det att kommandona är utförda:

0 Flyttningen gick att genomföra.
1 Flyttningen misslyckades.
2 Försök till att flytta en oren struktur.

OBS! Systemet kraschar om ett filsystem fullt av skräp flyttas.

Exempel:

1/ /etc/mount /dev/shl /usr
Det filsystem som ligger på enheten shl ska flyttas och läggas till under biblioteket usr. OBS! Innan enheten tas bort måste kommandot umount anges.

/etc/umount /dev/shl

2/ /etc/mount /dev/sh3 /usr/kkp/rst -r
Det filsystem som ligger på enheten sh3 ska flyttas och läggas till under biblioteket /usr/kkp/rst. Det nya filsystem ska bara vara läsbart. OBS! Innan enheten tas bort måste kommandot umount anges.

/etc/umount /dev/sh3

Namn: mv (move files) - flyttar och/eller byter namn på filer och bibliotek

Syntax

1 mv filnamnl filnamn2

2 mv filnamn ... bibliotek

Funktion: Kommandot mv har två funktioner, det kan antingen användas till att byta namn på filer eller till att flytta filer till andra bibliotek.

Syntax variant 1 används när en fil ska byta namn. Om båda filerna som anges befinner sig i samma bibliotek kommer filnamn 1 att byta namn till filnamn 2, och fill förstörs. Befinner sig de angivna filerna i olika filsystem, flyttas fill till filnamn 2's bibliotek och fill förstörs. Om filnamn 2 redan finns, i båda dessa fall, kommer innehållet i filnamn 2 att förstöras innan en flyttning görs av fill's innehåll. Om filnamn 2's privilegier inte tillåter skrivning, avbryts exekveringen och filnamn och filens privilegier skrivs ut på standard output. Systemet väntar därefter på ett svar, innan den kan fortsätta. Exekveringen utav mv kommandot fortsätter om svaret börjar på 'y', i annat fall avbryts den helt.

Generellt sätt kan inte ett bibliotek flyttas till ett annat med det finns ett undantag, flyttningen går att utföra om biblioteken har samma förälderbibliotek.

Används den andra syntax varianten, flyttas en eller flera filer med sina orginalnamn ,till det specificerade biblioteket.

Det går inte att med detta kommando flytta en fil till sig själv.

Hänvisning: cp, copy, chmod, rm

Anmärkning: Om fill och fil2 befinner sig på olika filsystem kommer mv att kopiera filen och därefter ta bort orginalet. I detta fall kommer ägarnamnet att ändras till det som den kopierande processen har och alla länkar till andra filer kommer att gå förlorade.

Exempel:

1/ mv temp temp2
Om filen temp2 inte finns kommer den att skapas, finns den kommer dess innehåll att förstöras. Därefter flyttas innehållet i temp till temp2. Filen temp förstörs. Detta går att förvissa sig om genom att använda kommandot ls, både före och efter användningen utav kommandot mv.

2/ mv Kapl Kap2 Kap3 Kap4 /Manual
Filerna Kapl-4 flyttas till biblioteket Manual. Alla fyra filerna kopieras med sina orginalnamn och orginalen förstörs.

3/ mv ornew ornewbib
Om ornew är ett bibliotek, kan mv kommandot exekveras om ornewbibd är ett bibliotek och de har samma förälderbibliotek.

Namn: ncu - nätverksanrop i DNIX

Syntax: ncu Ä option . . . Å systemid

Funktion: ncu kallar ett annat DNIX-system via ett nätverk. Det hanterar en interaktiv dialog med möjlighet till filtransfer. Systemid är maskinens namn. En kopia av ncu startas på den avlägsna maskinen och denna kopia startar en loginprocess och agerar terminal till denna avlägsna login. På den kallande sidan körs ncu som två processer: sendprocessen läser standard input och skickar det mesta till den avlägsna maskinen; receiveprocessen läser från det avlägsna systemet och skickar alla data till standard output. ncu använder stty/gtty--möjligheterna i den lokala terminalen och skickar data bara när en läsning kan lyckas. Detta medför vissa kosekvenser:

- Linjeredigering görs lokalt.

- Eftersom data överförs som om det skulle läsas kan 8-bitars binärdata (exekverbara program t.ex.) flyttas med file transfer möjligheten.

- Läses tomma tecken överförs de som sådana, INTE som specialtecken.

Sendprocessen tolkar en rad som börjar med 'ü' som följer:

ü . avsluta konversationen
üEOT avsluta konversationen .C .....
ü<fil skicka innehållet i filen till den avlägsna filen, som om den skrevs på terminal.

ü! kalla interaktivt shell på det lokala systemet.

ü!kommando kör kommandot på det lokala systemet (via sh -c) .

üü . . . skickar raden ëü . . . '.

Filer kan läggas upp på det avlägsna systemet med följande system (när inloggning på det avlägsna systemet har gjorts, behövs det avslutande EOT som en tom läsning till cat och därmed avslutar kopieringen till destinationen):

```
cat >remotedest ü<localsource EOT .C
..é.....>
```

Filer kan tas från det avlägsna system genom att först kalla det systmet genom att använda ncu och sedan (efter inloggning) kalla det lokala systemet. Efter återinloggning på det lokala systemet ges kommandon som följer (Den dubbla tilden används som citerad tilde på den lokala maskinen för att få den avlägset startade ncu att ta sitt input från önskad fil):
.C ..... cat >localdest üü<remotesource EOT

Option: Optioner används huvudsakligen för debugging:

-s I stället för systemid ges användarnamn. Defaultvärdet på login server prefix 'L' undertrycks.

-B Detta kommando används av netman för att starta B-sidan. Ska inte ges explicit.

-mnamn Får ncu att använda den network manager som anges av namn.

-xnn sätter debugnivån till nn (om DEBUG kompilerats in).

Filer:

/netphys	ncu använder netman ansluten här.
/tmp/ncuXXXXXX	B-sidan av ncu ansluter sin pseudokosol här.
/bin/login	Startas på B-sidan.
/bin/sh	Kör ü!-kommandon.

Hänvisning: netman(C), rx(C), ncu(S)

Namn: netman - network transport manager på DNIX--system

Syntax: netman - Ä option ... Å

Funktion: Network managern är en process som hanterar trafik till och från DNET och startar processer för att hantera inkommande uppkallningar.

Service för att kontakta det egna systemet eller ett avlägset system identifieras med namn (Symbolic Service String, SSS) enligt DNET.

Network managern ansluter sig själv till filsystemet (normalt under namnet /netphys) för att leta efter utgående uppkallningar; den ansluter också till network controllern och letar efter inkommande uppkallningar.

Optioner används huvudsaklingen för debugging:

-cnamn Får netman att öppna namn som network controller. Som default provas först /dev/n4004, sedan /dev/n4204.

-tnn Sätter korttyp till -I nn. Används för att välja filer att boota och metod för booten. Default är 4004 eller 4204 om ingen -c switch givits beroende på vilken öppning som lyckats.

-dnamn eller -d Får netman att ladda debuggern i stället för parameterprogram. Får också drivern att ignorera timeout när den kommunicerar med kort. Om namn givits används det som pathname åt debuggern, annars används /usr/lib/net/dbg4004 eller /usr/lib/net/dbg4204.

-pxnamn Kör över defaultvalen av filer för boot. x anger vilket program som ska ersättas: 0 betyder preboot, 1 betyder kontrollprogrammet, 2 betyder parameter/debuggerprogram. namn är det pathname som ska användas i stället för defaultvärdet.

-mnamn Får netman att ansluta sig vid namn.

-xnn Sätter debugnivån till nn (om DEBUG har kompilerats in)

-l Får -I netman att simulera controllerns uppförande lokalt. Naturligtvis är enbart lokala tjänster tilgängliga. (Denna switch stöds bara om DEBUG kompilerats in)

Inkommande uppkallningar:

Lokala tjänster beskrivs i filen /usr/lib/net/servtab. Varje tjänst beskrivs på en rad som innehåller:

(1) ett minustecken eller tjänstens nummer (2) tjänstens namn (3) pathname till ett program som börjar ta emot den inkommande uppkallningen.

Tjänstens nummer används sällan direkt i tillämpningar eftersom tjänsterna är tillgängliga med namn och ett minustecken här anger att netman ska välja nummer automatiskt. Vissa tjänster har dock kända nummer (se dokumentation: SSS/04-06-28) och i så fall kan numret tvingas in här.

Tjänstens namn används av tillämpningar som önskar utnyttja tjänsten. Om strängen i tjänstetabellsfilen innehåller '¤S' så ersätts '¤S' av systemnamnet som finns i /etc/systemid så att samma fil kan utnyttjas i olika system. Om en tjänst är identisk på olika system så att det inte spelar någon roll till vilket system anropet går kan namnet vara helt identiskt på olika system: normalt är dock (för avlägsen login och liknade) att tjänstens namn innehåller systemnamnet med en prefixsträng för att identifiera typen av tjänst.

Pathname används för att starta en process som hanterar inkommande anrop. När ett inkommande anrop anländer startar en process med användarid och gruppid tagna från filern uid och gid. Processen ges "-B" och tjänstens namn som parametrar. Processen får också ett virtuellt anrop fd som dess fd #0. Alla andra DNIX-fd stängs och aktuellt bibliotek sätts till "/".

Utgående anrop.

Utgående anrop genom att öppna till det anslutna namnet (dvs /netphys/tjänstnamn). Om öppningen går lyckligt skickas ett virtuellt anrop fd tillbaka och en server har startat i något system som också har ett virtuellt anrop fd anslutet iandra änden av det virtuella anropet. Kan tjänsten inte nås misslyckas öppningen ENOENT. .C ..... Utgående anrop kan också genomföras med numerisk adress genom att använda ett namn som består av numret

föregånget av ett nummertecken (#). Numret används direkt i anropspaketet på DNET som bara kontrollerar att det är <= 15 siffror. DNET mjukvara skickar tal med 1-2 siffror till ursprungliga systemet med tjänstnummer lika med det anropade numret. Tal med 5-7 skickas till det system vars 4004-kort har ett serienummer som motsvarar de fem första siffrorna, varvid tjänstens nummer tas från de resterande siffrorna. Numerisk adressering tillhandahålls huvudsakligen för debugging.

Filer:

/etc/systemid	maskinens namn. /usr/lib/net/-servtab Tjänsttabell för lokala tjänster.
/netphys	Default anslutningsplats.
/tmp/netlog	Här skrivs felmeddelanden.
/tmp/netswapXXX	Används för att spara buffrar när många buffrar används. /usr/lib/net/pre4004 Default preboot för 4004-kortet. /usr/-lib/net/ctl4004 Default kontrollprogram för 4004-kort. /usr/lib/net/par4004 Default parameterprogram för 4004-kort. /usr/lib/net/dbg4004 Default debugger för 4004-kortet. /usr/lib/net/ctl4204 Default kontrollprogram för 4204-kort. /usr/lib/net/par4204 Default parameterprogram för 4204-kort. /usr/lib/net/dbg4204 Default debugger för 4204-kortet.

Hänvisning: netman(S), ncu(C), rx(C)

Anmärkning: netman ska endast startas från root. Som det är idag ska det inte sättas till uid root.

Namn: od (octal dump) - oktal dump

Syntax: od (-bcdox) (filnamn) (+) offset (.) (b)

od -b -c -d -o -x filnamn + offset . b

Funktion: Kommandot od producerar en dump av den specificerade filen, eller standard input om ingen fil har angivits. Filen kan dumpas i flera olika format, oktala ord (default), oktala bytes , ASCII bytes, hexadecimala ord eller decimala ord. De olika formaten kan produceras tillsammans eller separat. Normalt dumpas filen med start från början om inte en offset har angivits.

Som första argumentet anges vilken fil som ska dumpas. Om inget filnamn anges ,används 'standard input'.

Offset som anges som andra argument används till att kontrollera avståndet från början av filen till den position där dumpningen ska börja. Om inget filnamn är specificerat måste offset föregås av tecknet +. Normalt tolkas offset som oktala ord. Läggs tecknet '.' till offset tolkas den som decimala bytes. Om b läggs till efter offset tolkas den i 512 bytes block.

Dumpningen fortsätter till end-of-file.

Option: -b Tolkar oktala bytes.

-c Tolkar ASCII bytes. Några icke-grafiska tecken förekommer :

0	null
b	backspace
f	formfeed
n	newline
r	return
t	tab

de övriga förekommer som oktala 3-siffriga nummer.

-d Tolkar decimala ord.

-o Tolkar oktala ord, default.

-x Tolkar hexadecimala ord.

Exempel: 1/ od -c Kapl
Filen Kapl dumpas med ASCII bytes.

2/ od /dev/shl +1024
Disken som specificeras som shl undersöks med start på 1024 bytes från början av filen.

3/ od /dev/shl +2b
Disken som specificeras som shl undersöka med start vid 3:e blockets början, dvs 2 x 512 = 1024 bytes, räknat från början av filen.

4/ od -bc myfile
Ger en listning på standard output av innehållet i myfile på två sätt, -b som anger oktal listning bytevis och -c som anger ascii listning bytevis.

5/ od -bcx myfile
Som ex 1 med tillägget för optionen -x som ger en listning hexadecimalt ordvis (16 bits ord). Skifta -x med -o eller -d, eller lägg till -o och -d. Optionen -o ger en oktal listning ordvis och -d en decimal listning ordvis.

od -bcxod myfile

Visar samtliga listningssätt på en gång.

Namn: passwd (password) - ändra lösenordet

Syntax: passwd (namn)

Funktion: Kommandot passwd används till att ändra lösenordet som används vid inloggningen. OBS! Det är endast användaren eller super-usern som kan ändra lösenordet.

Efter det att kommandot har angivits, frågar programmet först efter gamla lösenordet. Detta förhindrar att någon ändrar ditt lösenord utan ditt tillstånd. Därefter frågar programmet efter det nya lösenordet, två gånger, detta för att lösenorder inte ska bli fel utav misstag.

Ett lösenord ska vara åtminstonde 4 tecken långt om både stora och små tecken används och 6 tecken långt om antingen stora eller små tecken används. Programmet accepterar kortare lösenord, för det krävs två försök i rad att skriva in det nya lösenordet.

En säkerhetskopia av passwordfilen (/etc/-passwd) skapas varje gång kommandot används.

Filer: /etc/passwd

Hänvisning: login

Exempel: 1/ passwd
Följande skrivs ut av systemet:

changing password for (här anges loginnamnet)
old password:
Systemet väntar på att det gamla lösenordet ska anges. Därefter frågas efter det nya lösenordet, 2 ggr.

Namn: pr (print) - skriver ut filer

Syntax: pr - n,h,wn,ln,t,sc,m,b ,+n filnamn ...

pr -n +n -h -wn -ln -t -sc -m -b filnamn ...

Funktion: Kommandot pr delar in text filer i sidor och lägger till sidhuvud och sidfot. Överst pa varje sida skrivs datum, namnet på filen eller det specificerade sidhuvudet, och sidnummret.

Om inget filnamn anges läser kommandot pr standard input.

Option: De optioner som anges gäller för alla de filer som specificerats, om de inte ändras mellan filerna. OBS! Optionerna kan inte nollställas mellan filnamnen men nya optioner kan läggas till.

-n Skriver ut kolumnen n

+n Börjar skriva på sidan n.

-h Tar nästa argument som ett sidhuvud.

-wn Ändrar pappersbredden till n tecken, default är 80 tecken.

-ln Ändrar längden på sidan till n rader, default är 72.

-t Skriv inte ut sidhuvudet eller sidfoten på 5 rader.

-sc Separerar kolumnerna med bokstaven c istället för med ett blanktecken.

-m Skriver alla filer samtidigt, i varsin kolumn.

-b Sidorna separeras med ett 'formfeed' tecken, normalt separeras sidorna med ett antal tecken för ny rad.

Filer: /dev/tty*

Hänvisning: cat

Exempel:

1/ pr kap3
På standard output skrivs först ett 5-radigt sidhuvud ut, innehållande två blanka rader, en rad med datum, tid och sidnummer och två blanka rader. Därefter skrivs innehållet i filen ut. Om innehållet är längre än 72 rader skrivs först den 5-radiga sidfoten ut, därefter skrivs sidan 2 ut med sidhuvud.

2/ pr -t kap3
På standard output skrivs filens innehåll ut utan något sidhuvud eller sidfot. Men ny sida indikeras efter 72 rader.

3/ pr kap1 > kap1.pr
Text filen kap1 delas in i sidor och både sidhuvud och sidfot läggs till, resultatet placeras i filen kap1.pr.

Namn: ps - processtatus

Syntax: ps axl

Funktion: ps skriver viss information om pågående processer. Med optionen a fås information om alla processer med terminaler (normalt visas bara ens egna processer); x visar även processer utan terminaler; l ger lång lista. Den långa listan innehåller:

UID	Processägarens användarid
TTY	Kontrollerande terminal
PID	Processens id PPID Föräldraprocessens id
STATE	Switchar med anknytning till processen på format xxx:s T - Processen spåras av annan process L - Processen låst till kärnan S - Processen är swapped S-switchen visar processens status: R - Processen är körbar S - Processen vilar Z - Processen i mellanläge P - Processen har stannat
PRI	Processens prioritet; höga tal betyder låg prioritet
SIZE	Processens storlek i kärnan på format 'N1+N2' N1 innehåller delbar kod N2 innehåller icke delbar kod
UTIME	Kumultativ användartid för processen
STIME	Kumultativ systemtid för processen
CMD	Kommandot som startade processen

Den korta listan innehåller UID, TTY, PID, PPID och CMD.

Exempel:

1/ ps
Listar information om användarens egna processer.

2/ ps -a
Utökar informationen till att visa alla processer som har en terminaltillhörighet.

ps -x
-x innebär alla processer, även såna processer som inte har terminaltillhörighet.

3/ ps -lx
-lx innebär att man listar informationen om alla processerna i långt format.

ps -lx är lika med ps -lax

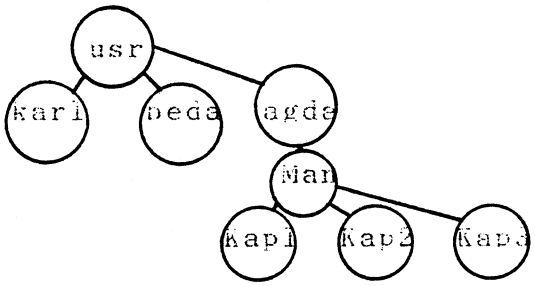
Namn: pwd (print working directory) - anger vilket bibliotek som är aktuellt.

Syntax: pwd

Funktion: Kommandot pwd skriver ut från root till det nuvarande biblioteket.

Hänvisning: cd(C)

Exempel: 1/ Följande trädstruktur finns i ett filsystem.



Om man befinner sig i Man och kommandot pwd anges blir resultatet /usr/agda/Man.

Namn: rm, rmdir (remove files, remove directories) - tar bort filer, tar bort bibliotek.

Syntax: rm (-fri) filnamn ...

rmdir bibliotek ...

Funktion: Kommandot rm tar bort en eller flera filer från filsystemet. Om en fil har flera länkar, förstörs inte filen förrän den sista länken till filen tas bort. Om ett bibliotek anges istället för en fil, skrivs ett felmeddelande ut.

För att det ska vara möjligt att ta bort en fil, måste användaren ha tillåtelse att skriva i det bibliotek som filen befinner sig i. Däremot behöver användaren inte ha tillåtelse att varken skriva eller läsa i filen.

När kommandot används på filer som det inte är tillåtet att skriva till och om standard output är en terminal, kommer programmet att interaktivt fråga användaren om filen skall förstöras. Kommandot väntar därefter på ett svar. Om den första bokstaven i svaret är ett 'y' kommer filen att förstöras.

Kommandot rmdir tar bort bibliotek. Dessa bibliotek kan bara tas bort om de är tomma.

OBS! Det är förbjudet att ta bort '..' .

Option:

-f Denna flagga gäller bara filer som är skrivskyddade. Ingen fråga ställs innan filen tas bort.

-r Inget meddelande skrivs ut om den angivna filen är ett bibliotek. Allt innehåll i biblioteket kommer att förstöras liksom biblioteket själv.

-i Frågar interaktivt om filen ska förstöras.

-ir Frågar om varje bibliotek som påträffas ska undersökas.

Exempel: 1/ rm /Manual/kap8

Filen kap8 tas bort ur biblioteket Manual.

2/ rm *

Tar bort alla filer i nuvarande bibliotek.

3/ rm -i kap*

Kommandot frågar interaktivt om alla filer som börjar på kap i nuvarande bibliotek skall tas bort.På standard output skrivs namnet på varje fil, som i det här fallet börjar på kap, åtföljt av ?ÄylnÅ: frågetecken. Om svaret börjar med ett'y' kommer filen att tas bort. I alla andra fall kommer filen att behållas.

Namn: rmuser (remove user) - tar bort en användare

Syntax: rmuser (-nr) (andvändarnamn)

Funktion: rmuser tar bort en andvändare genom att uppdatera filerna /etc/passwd och /etc/group samt att ta bort användarens hembibliotek (home directory). Om användarnamnet (user name) ej anges frågar kommandot efter det. Användarens hem-bibliotek tas bort endast om det är tomt, detta kan dock kringgås om man använder optinen -r. Kommandot finns under biblioteket /etc, därför stå i eller ladda det från biblioteket /etc när du ska ta bort en användare.

OBS! Kommandot kan endast super-user (root) köra.

Option:

-r Ta bort användarens hem-bibliotek även om det ej är tomt. Allt innehåll i biblioteket kommer att försvinna även under bibliotek.

-n Ta ej bort bort användarens hem-bibliotek.

Filer: /etc/group /etc/passwd

Exempel:

1/ /etc/rmuser

Kommandot frågar efter användar-namn, varefter användaren kommer att tas bort. Även hem-biblioteket tas bort om det är tomt.

2/ cd /etc ; rmuser -n olle

Användaren olle tas bort. Dock kommer olles hem-bibliotek att lämnas kvar.

Namn: rx - kör ett kommando på en annan maskin

Synatx: rx Ä option ... Å systemid!kommando Ä parametrar ... Å

Fuktion: rx loggar automatiskt in på en annan maskin och exekverar ett kommando där. Om det avlägsna programmet läser standard input kopieras det lokala rx-kommandots standard input via nätverket och matas som standard input till det avlägsna kommandot. Det avlägsna kommandots standard output och standard error kopieras på det lokala rx-kommandots standard output och standard error. Om det avlägsna kommandot avslutar normalt (med utgång(2)) avslutar rx med samma kod. Om det avlägsna kommandot avbryts avslutar rx med kod 1.

Normalt letar rx i aktuellt HOME-bibliotek efter filen .netkey för att hitta information om inloggningsrutinerna på det avlägsna systemet. Filen .netkey innehåller rader med tre poster på varje:

- ett maskinnamn eller '*' för att ange vilken maskin som helst.
- uid att använda på den maskinen.
- ett nyckel som ska matcha mot den avlägsna maskinen.

Ursprungssidan genomsöker den lokala .netkey-filen efter ett namn som matchar den främmande maskinen och plockar sedan uid och nyckel från den posten. På den avlägsna sidan startas en annan kopia av rx som först letar i det systemets passordsfil efter angivet uid och sedan i dess ¤HOME/.netkey-fil efter systemid och jämför nycklar. Om nycklarna är lika utförs den avlägsna exekveringen.

Eftersom filen .netkey söker efter destinationens systemid på båda sidor, kan en användare ha identiska filer på flera maskiner, även om hennes/hans uid och/eller nyckel varierar från maskin till maskin. Hon/han kan hantera sin .netkey-fil efter behag utan att ha rot-privilegier.

Som en säkerhetsåtgärd ändrar rx skyddet till 0600 på alla .netkey-filer det använder, ifall användaren skulle glömma göra sin nyckel hemlig.

Följande signaler fångas av rx och överförs till det avlägset exekverade kommandot: SIGINT, SIGQUIT, SIGPIPE och SIGTERM. Går linjen ner får det avlägsna kommandot SIGHUP. Andra signaler på lokalsidan får rx att avsluta och därmed går linjen ner SIGHUP ges på andra sidan.

Omgivningen kopieras till den avlägsna maskinen med två undantag:

HOME sätts till loginbiblioteket på den andra maskinen.

PATH modifieras för att återge det nya HOME: på alla ställen där home-biblioteket förekommer i vägvalen ersätts de med HOME-biblioteket på den andra maskinen.

Option:

-luid Ä :passord Å Kör över den automatiska inloggningen och loggar in som uid med passord på den andra maskinen. Detta passord är det normala som används av login(1) och kräver inte närvaron av en .netkey-fil på den avlägsna maskinen.

-ddir Ändrar aktuellt biblioteket till dir innan kommandot exekveras. Normalt exekveras kommandon i loginbiblioteket på den avlägsna maskinen.

-B Denna option används av netman för att starta B-sidan. Ska aldrig användas explicit.

-mnamn Får rx att använda en network manager ansluten vid namn.

-xnn Sätter debugnivån till nn (om DEBUG har kompilerats in)

Filer:

/netphys	rx använder netman ansluten här
/tmp/rxXXXXXX	B-sidan av rx ansluter sin pseudokonsol här
/etc/systemid	maskinnamn

Hänvisning: netman(C), ncu(C), rx(S)

Namn. setspeed (set speed) - ändrar överförningshastighet till exempelvis en skrivare.

Syntax: setspeed (-s -t -r) (enhet)

Funktion: Kommandot setspeed sätter överföringshastigheten på ett device tex lp.

Option: -s överföringshastighet i bit/s

-t sändningshastighet vid split speed

-r mottagningshastighet vid split speed

Anmärkning: Kommandot skapar en process som lägger sig att sova varje gång det används. Detta upptar alltså en tnod, vilket innebär att antalet körbara processer i systemet minskar med ???? (se ps -lax)

Exempel: 1/ /etc/setspeed -s2400 /dev/lp

2/ /etc/setspeed -t75 -r1200 /dev/tty03

Namn: setup, setupinv - definierar funktionstangenter

Syntax:
setup (fil)
setupinv (fil)

Funktion: Med programmet setup kan du skapa och editera filer som definierar tangentbordet ABC99s funktionstnagenter PF1 - PF15, markörplaceraren samt HELP, STOP, PRINT, INS och tabulatortangenterna. Gemensamt för alla funktionstangenter gäller att varje PF-tangent kan defineras med upp till 60 tecken. setup kan förses med ett argument, där argumentet står för den fil som ska läsas, dvs den uppsättning PF-funktioner som man vill ha definierad. Om inget argument ges läser programmet bara av aktuella definitioner.

Genom att ändra environmentvariabeln LANGUAGE = swedish erhålls svensk text i menyerna. Default gäller LANGUAGE = english.

Programmet setupinv är i första hand avsett för att köras vid inloggningen i systemet. Anropet som också kan förses med ett argument, placers lämpligen i .profile. Om inget argumnet ges letar systemet efter filen .fnkey i logginbiblioteket. Finns den inte där letar systemet i /etc. Det innebär att man kan ha en privat uppsättning PF-funktioner i sitt logginbibliotek i filen .fnkeys och en allmän uppsättning i /etc/.fnkeys.

Huvudmenyn i programmet setup innehåller 9 st rubriker:

```
1  PF-tangent
2  PF-tangent (SHIFT)
3  PF-tangent (CTRL)
4  PF-tangent (SHIFT+CTRL)
5  Andra tangenter
6  Läsa på fil
7  Spara på fil
8  Nollställ PF-tangenterna
0  Slut

   Välj
```

"1 PF-tangent - 4 PF-tangent (SHIFT+CTRL)" visar följande på skärmen:

```
PF1:
PF2:
PF3:
PF4:
PF5:
PF6:
PF7:
PF8:
PF9:
PF10:
PF11:
PF12:
PF13:
PF14:
PF15:

Ange PF-nummer.
```

Skriv in de tecken som önskas (max 60 st.) Vissa karaktärer tex CTRL I, CTRL H kräver att en sk. forcing caharcter "CTRL Ö" skrivs in före karaktären. Avsluta med return.

"5 - Andra tangenter" ger följande meny på skärmen:

```
1 HELP:
2 STOP:
3 PRINT:
4 ALT:
5 INS:
6 --->:
7 <---:

MARKÖRPLACERARE                 MARKÖRPLACERARE (SHIFT)
8 UPP:                          16 UPP:
9 UPP-HÖGER:                    17 UPP-HÖGER:
10 HÖGER:                       18 HÖGER:
11 NER - HÖGER:                 19 NER - HÖGER:
12 NER:                         20 NER:
13 NER - VÄNSTER:               21 NER - VÄNSTER:
14 VÄNSTER:                     22 VÄNSTER:
15 UPP - VÄNSTER:               23 UPP - VÄNSTER:

Ange nummer:
```

Gör som i föregående exempel. Observera att för nummer 8-23 skrivs endast 20 tecken ut på skärmen pga utrymmesbrist.

"6 - Läsa från fil" innebär att du kan läsa en fil som innehåller en uppsättning definitioner av PFtangenterna. Med fullständigt pathname kan man läsa en fil var som helst i filsystemet.

"7 - Spara på fil" innebär att du kan spara en uppsättning definitoner av PF-tangenter på en fil. Med fullstädigt pathname kan du skapa och skriva en fil var som helst i filsystmet.

"8 - Nollställning av PF-tangenterna" innebär att samtliga funktioner nollställs.

Anmärkning: Programmen setup och setupinv placeras under /usr/bin och en fil som innehåller en standraduppsättning av PF-funktioena placeras under/etc i filen .fnkeys.

Namn: sh - anropar kommandotolken i shell

Synatx: sh (-ceiknrstuvx) (argument)

Funktion: Shell ("skalet") är det standard-programspråk som utför kommandon lästa från terminal eller fil. Se ANROP nedan för förklaring av argu-ments betydelse för shell.

Kommandon: Ett enkelt kommando är en rad ickeblanka ord separerade av blanktecken (ett blanktecken är ett mellanslag eller annat märke). Det första ordet anger kommandots namn. Förutom vad som sägs nedan, skickas resterande ord som argu-ment till det kallade kommandot. Kommandonam-net skickas som argument 0 (se exec(S)). Ett enkelt kommandos värde är dess utgångsvärde om det avslutas normalt, eller 200+status (ok-talt) om det inte avslutas normalt, dvs om om felet ger en kärnfil. Se signal(S) för en lis-ta på statusvärden.

En pipeline är en sekvens av ett eller flera kommandon åtskilda av ett utrops tecken (!). (En caret (^) har samma effekt.) Standard out-put från varje kommando, utom det sista, ans-luts av pipe(S) till standard input på nästa kommando. Varje kommando körs som en separat process; shell väntar på att det sista komman-dot ska avslutas.

En lista är en sekvens av en eller flera pipe-lines åtskilda av ;, &, && eller !! , och eventuellt avslutade med ; eller &. Av dessa fyra symboler har ; och & lika rang, vilken är lägre än rangen för && och !! . && och !! har också lika rang. Ett semikolon (;) ger sekven-tiell exekvering av föregående pipeline; teck-net & ger asynkron exekvering av föregående pipeline (dvs shell väntar icke på att den pipelinen skall avslutas). Symbolen && (!!) anger att listan efter exekveras endast om den föregående pipelinen ger en nolla (skild från noll) som utgångsstatus. Ett valfritt antal nyrad får förekomma i en lista i stället för semikolon, för att avgränsa kommandon.

Ett kommando är antingen ett enkelt kommando eller något av de följande kommandona. Om ing-et annat anges är det värde kommandot ger det som det sista enkla kommandot i kommandot ger:

for namn (in ord . . .) do lista done Varje gång ett for-kommando exekveras, sätts namn till nästa ord taget från ordlistan. Om in

word utelämnas, exekverar for-kommandot do lista en gång för varje positionsparameter som har satts (se Parameter Substitution nedan). Exekveringen avslutas när inga ord återstår i ordlistan.

case ord in (mönster (! mönster) . . . é<)> list-
a ;;). Ett case-kommando exekverar listan knuten till det första mönster som matchar ord. Mönstrets form är densamma som används för att skapa filnamn (se Skapa Filnamn nedan).

if lista then lista (elif lista then lista) ... (else lista) Listan efter if exekveras och om den ger utgångsstatus noll exekveras listan efter det första then. Annars exekveras listan efter elif och, om den ger status noll, exekveras listan efter nästa then. Skulle inte så vara fallet exekveras listan efter else. Om ingen elselista eller then-lista exekverats, ger if-kommandot utgångsstatus noll.

while lista do lista done Ett while-kommando upprepar exekveringen av while-listan tills status på sista kommandot i listan är noll. Då exekveras do-listan, varefter while-kommandot ger utgångsstatus noll; för att negera testen i slutet av loopen används until i stället för while.

lista Exekverar listan i ett subshell

:lista;: Exekverar helt enkelt listan

De följande orden känns endast igen som första ord i ett kommando och när de inte placeras inom apostrofer.

if then else elif fi case esac for while until do done : :

Kommantarer: Ett ord som börjar med ": ", och alla följande ord till newline,ignoreras. .C .....

Kommandosubstitutioin: Standard output från ett kommando omslutet av ett par (#..#) nummer tecken kan användas som del av ord eller som helt ord: newlines på slutet tas bort.

Parametersubstitution: Tecknet α används för att införa substituerbara parametrar. Positionsparametrar kan tilldelas värden med set. Variabler kan sättas genom att skriva:

namn=värde (namn=värde) . . .

Mönster-matchning görs inte på värdet.

¤:parameter: En parameter är en följd bokstäver, siffror, understreck (ett namn), en siffra eller något av tecknen *, É, #, -, ¤, och !. Parameterns värde, om den har något, substitueras. Parenteserna behövs bara när parametern följs av en bokstav, siffra eller understreck som inte får tolkas som en del av dess namn. Ett namn måste börja med en bokstav eller understreck Om parametern är en siffra så är den en positionsparameter. Om parametern är * eller É, substitueras alla positionsparametrar, med början ¤1. Parameter ¤0 sätts från argument noll när shell anropas.

¤:parameter-ord): Om parametern har satts, substituera dess värde; annars substitueras ordet.

¤:parameter=ord: Om parameter inte satts, sätts dess värde till ord; parameterns värde substitueras därefter. Positionsparametrar kan inte tilldelas på detta sätt.

¤:parameter?ord: Om parametern har satts, substitueras dess värde; annars skrivs ordet och shell lämnas. Om ordet utelämnas skrivs meddelandet "parameter null or not set".

¤:parameter+ord: Om parametern har satts, substitueras ordet; annars substitueras ingenting.

I det ovanstående utvärderas inte ordet om det inte ska användas som den substituerade strängen. I följande exempel exekveras pwd endast om d inte satts:

echo ¤:d-#pwd#:

Om kolon (:) sätts in i uttrycken före -,=,? och + kommer shell även att kontrollerar parametern mot värdet null (om shell körs i engelskt läge).

Följande parametrar sätts automatiskt av shell:

#	Antal positionsparametrar i decimala tal
-	Switchar ges till shell vid anrop eller av kommandot set
?	Det decimala värdet av det senaste synkront exekverade kommandot

¤ Detta shells processnummer

! Det senast kallade backgroundkommandots processnummer

Följande parametrar används av shell:

HOME Defaultargument (home directory) till kommandot cd

PATH Sökväg för kommandon (se Exekvering nedan)

MAIL Sätts denna parameter till namnet på mailfilen kommer shell att på angiven rad informera användaren när post anländer.

PS1 Primär promptsträng, default "¤ "

PS2 Sekundär promptsträng, default "> "

IFS Interna fältseparatorer, normalt space, tab och newline

Shell ger defaultvärden till PATH, PS1, PS2 och IFS. HOME och MAIL sätts inte ens av shell (däremot sätts HOME av login(M)).

Tolkning av "blanka"

Efter parameter- och kommandosubstitution genomsöks resultaten efter interna fältseparatorer (dvs de som återfinns i IFS), och delning till entydiga argument görs där sådana separatorer finns. Explicita nullargument ("" eller '') behålls. Implicita nullargument (dvs sådana som är resultatet av parametrar utan värde) tas bort.

Skapande av filnamn:

Efter substitutionen sökes i alla kommandoord tecknen *, ?
och . Finns något av dessa tecken betraktas ordet som ett mönster. Ordet ersätts då med alfabetiskt sorterade filnamn som matchar mönstret. Om inget matchande filnamn hittas, lämnas ordet orört. Tecknet . i början av ett filnamn, eller omedelbart efter ett /, liksom tecknet /, måste matchas explicit. Matchningstecknen och deras mönster ser ut som följer:

* Matchar alla strängar, även nullsträng.

? Matchar alla enkla tecken

: . . . : Matchar var och en av de ingående tecknen. Ett par tecken åtskilda av - matchar alla tecken alfabetiskt

mellan tecknen (inklusive tecknen själva).

Citering: Följande tecken har särskild mening i shell och förorsakar avslutning av ord om de icke "citeras":

: & ( ) ! Ü < > newline space tab

Ett tecken kan citeras (dvs stå för sig självt) genom att man låter ett % föregå tecknet. Paret %newline ignoreras. Alla tecken som placeras mellan apostrofer (’’ ; single quote på engelska), utom apostrofen, citeras. Inom citationstecken ("") (double quote) sker parameter- och kommandosubstitution och % citerar tecknen %, " och ¤. "¤*" är detsamma som "¤1 ¤2 . . . ", medan "¤É" är detsamma som "¤1" "¤2" . . . .

Promter: När shell används interaktivt visas värdet av PS1 som prompt innan kommandot kan läsas in. Om newline någon gång trycks och vidare indata krävs för att fullfölja ett kommando, visas den sekundära prompten (dvs värdet av PS2).

Input !output: Innan ett kommando exekveras, kan dess input och output omdirigeras med en särskild notation som tolkas av shell. Det följande kan förekomma var som helst i ett enkelt kommando eller kan föregå eller följa efter ett kommando. De skickas inte vidare till det kallade kommandot; substitution sker innan ord eller siffra används:

ord Använd filen ord som standard input (filbeskrivare 0)

ord Använd filen ord som standard output (filbeskrivare 1). Om filen inte existerar så skapas den: annars trunkeras den till längden noll.

ord Använd filen ord som standard output. Om filen existerar läggs output till på slutet av filen (genom att söka upp end-of-file); annars skapas filen.

: - :ord Shell input läses fram till en rad som är likadan som ord, eller till en end-of-file markering. Det resulterande dokumentet blir standard input. Om något tecken i ordet är citerat, görs ingen tolkning av tecknen i dokumentet; annars utförs parameteroch kommandosubstitution, %newline ignoreras och % måste användas

för att citera tecknen %, o och det första tecknet i ordet. Om - laggs till << tas alla inledande tabbar bort från ordet och dokumentet.

&siffra Standard output kopieras från filbeskrivare siffra (se dup (S)). Motsvarande gäller standard output om man använder >.

&- Stänger standard input. Motsvarande för standard output om man använder >.

Om något av de ovanstående uttrycken föregås av en siffra, får filbeskrivaren det värde som anges av siffran (i stället för defaultvärdena 0 och 1). Exempel:

. . . 2>&1

Filbeskrivare 2 skapas som en kopia av filbeskrivare 1.

Om ett kommando följs av & är default standard input för kommandot den tomma filen /dev/null. Annars innehåller omgivningen för kommandots exekvering filbeskrivningar från kallande shell, eventuellt modifierade av input/outputspecifikationer.

Omgiving: Omgivningen (se environ(M)) är en lista med namn-värdepar som skickas till ett exekverat program på samma sätt som en vanlig argumentlista. Shell påverkar/påverkas av omgivningen på flera olika sätt. När det kallas går shell igenom omgivningen och skapar en parameter för varje namn som hittas, och ger den ett motsvarande värde. Exekverade kommandon "ärver" samma omgivning. Om användaren ändrar värdena på dessa parametrar, eller skapar nya, påverkar detta inte omgivningen om inte kommandot export används för att koppla shells parameter till omgivningen. Den omgivning som det exekverade kommandot ser består således av de icke modifierade namn-värdeparen som ursprungligen ärvdes av shell, plus eventuella modifieringar eller tillägg, som alla måste anges i exportkommandon.

Omgivningen till ett enkelt kommando kan förstärkas genom att föregå det med en eller flera tilldelningar till parameterar. Sålunda är

TERM=450 kommando argument och (export TERM; TERM=450; kommando argument)

helt likvärdiga (vad exekveringen av kommandot beträffar).

Om switchen -k har satts, placeras alla nyckelords argument i omgivningen, även om de förekommer efter kommandonamnet.

Signaler: Signalerna INTERRUPT och QUIT för ett kallat kommando ignoreras om kommandot följs av &; annars har signaler de värden som shell ärvde av sin förälder, förutom signal 11 (men se även kommandot trap nedan).

Körning: Varje gång ett kommando exekveras, utförs ovannämnda substitutioner. Förutom när det gäller SpecialKommandona nedan, skapas en ny process och ett försök att exekvera kommandot görs via exec(S).

Shellparametern PATH anger sökvägen i biblioteket som innehåller kommandot. Alternativa biblioteksnamn skiljs med kolon (:). Defaultväg är :/bin:/usr/bin (anger aktuellt bibliotek, /bin, och /usr/bin, i den ordningen). Lägg märke till att aktuellt bibliotek anges som ett nullpathname, som kan placeras omedelbart efter likhetstecknet eller mellan kolondelimiters var som helst i väglistan. Innehåller kommandonamnet en / används inte sökvägen. Annars genomsöks alla bibliotek på vägen efter en exekverbar fil. Om filen får exekveras men inte är en a.out fil, antas den innehålla shellkommandon. Ett subshell (dvs en separat process) tilldelas för att läsa det. Ett kommando exekveras också i ett subshell.

Specialkommandon: Följande kommandon exekveras i shellprocessen och ingen omdirigering av input/output tillåts om så inte särkilt anges.

: Ingen effekt; kommandot gör ingenting. Utgångskod noll.

. fil Läser och exekverar kommandon från filen och återvänder. Sökvägen definierad i PATH används för att hitta det bibiliotek som innehåller filen.

break (n) Lämnar for- eller while-loop om sådan genomlöps. Är n angiven bryter kommandot n nivåer.

continue (n) Återupptar nästa iteration i den genomlöpta for- eller while-loopen. Om n anges återupptas exekveringen i den n:te loopen.

cd (arg) Ändrar aktuellt bibliotek till arg. Shellparametern HOME är default arg.

eval (arg . . .) Argumenten läses som input till shell och de resulterande kommandona exekveras.

exec (arg . . .) Kommandot angivet av argumenten exekveras istället för detta shell utan att skapa en ny process. Input/outputargument får förekomma och om inga andra argument anges ändras då shell input/output.

exit (n) Låter shell avsluta med utgångsstatus n. Om n utelämnas blir utgångsstatus samma som från det sist exekverade kommandot (end-of-file avslutar också shell).

export (namn . . .) Det givna namnet märks för automatisk export av därefter följande kommandon till omgivningen. Om inget argument anges, visas en lista med alla namn som exporteras.

newgrp (arg . . .) Likvärdig med exec newgrp arg . . . .

read (namn . . .) En rad läses från standard input och det första ordet tilldelas det första namnet, det andra ordet det andra namnet osv. Överblivna ord tilldelas det sista namnet. Utgångsvärdet är noll under förutsättning att ingen end-of-file påträffas.

readonly (namn . . .) Det givna namnet märks readonly och dess värde kan inte ändras av efterföljande tilldelningar. Om inget argument anges, visas en lista med alla readonly namn.

set (-ekntuvx (arg . . .))

-e Avslutar omedelbart, om shell inte är interaktivt, om ett kommando avslutas med utgångsstatus skild från noll.

-k Placerar alla nyckelords argument i omgivningen till ett kommando, inte bara de som föregår kommandot.

-n Läser kommandon men exekverar dem ej.

-t Avslutar efter att ha läst och exekverat ett kommando. Bara avsett att användas med C-program; inte användbart interaktivt.

-u Behandlar icke satta variabler som fel vid substitueringen.

-v Skriver ut shells inputrader allteftersom de läses.

-x Skriver ut kommandon och deras argument allteftersom de läses.

-- Ändrar inga switchar; användbar för att sätta $1 till -.

Genom att använda + i stället för - stänger av dessa switchar. Switcharna kan också användas när shell kallas. De aktuella switcharna finns i $-. De återstående argumenten är positionsparametrar och är tilldelade, i ordning, till $1, $2, ... Om inga argument ges skrivs värdena till alla namn.

shift Positionsparametrarna från $2 . . . döps om till
$1 . . . .

test Utvärderar villkorliga uttryck. Se test(C) för användning och beskrivning.

times Skriver ackumulerad användar- och systemtid för processer körda från shell.

trap (arg) (n) . . . Arg är ett kommando som ska läsas och exekveras när shell får signalen (signalerna) n. (Lägg märke till att trap läses en gång när fällan sätts och en gång när den aktiveras.) Trapkommandon exekveras i signalnummerordning. Det högsta tillåtna signalnumret är 16. Varje försök att sätta trap på en signal som ignorerades vid ingången i shell är utan verkan. Försök att fånga signal 11 (minnesfel) ger fel. Om arg saknas kommer alla trap n att sättas åter till ursprungliga värden. Om arg är en nullsträng ignoreras denna signal av shell och de kommandon det kallar. Om n är 0 exekveras kommandot arg när shell lämnas. Kommandot trap utan argument skriver en lista på kommandon tillhörande respektive signalnummer.

umask (ooo) User file-creation mask sätts till ooo, där o är en oktal siffra (se umask(C)). Om ooo utelämnas skrivs maskens aktuella värde ut.

wait Inväntar avslut på alla underprocesser och meddelar utgångsstatus. Om n inte anges inväntas alla underprocesser. Utgångsstatus är alltid 0 för detta kommando.

Anrop: Om shell kallas genom exec(S) och första tecknet i argument 0 är -, läses kommandon först från /etc/profile och sen från $HOME/.profile, om dessa existerar. Därefter läses kommandon enligt nedan, vilket också görs om shell kallas som /bin/sh. Switcharna nedan tolkas bara av shell vid uppkallningen: lägg märke till att om inte switch -c eller -s satts, tolkas det första argumentet som namn på en fil innehållande kommandon, och resterande argument skickas till denna kommandofil som positionsparametrar:

-c sträng Om switch -c satts läses kommandon från strängen

-s Om switch -s satts eller om inga argument återstår, läses kommandon från standard input. Kvarvarande argument anger positionsparametrar. Shell output skrivs till filbeskrivare 2.

-i Om switch -i satts eller om shell input och output anslutits till terminal, är detta shell interaktivt. I så fall ignoreras TERMINATE (så att kill 0 inte förstör ett interaktivt shell) och INTERRUPT fångas och ignoreras (så att wait kan avbrytas). I alla fall ignorerar shell QUIT.

-r Om switch -r satts är shell ett restricted shell (se rsh(C)).

Övriga switchar beskrivs i samband med kommandot set ovan.

Exit Status Fel som upptäcks av shell, såsom syntaxfel, får shell att lämna en utgångsstatus skild från noll. Används inte shell interaktivt överges exekveringen av shellfilen. Annars lämnar shell utgångsvärdet från det senast exekverade kommandot (se även kommandot exit ovan).

Filer: /etc/profile $HOME/.profile /tmp/sh* /dev/null

Hänvisning: cd(C), env(C), login(M), newgrp(C), rsh(C), test(C), umask(C), dup(S), exec(S), fork(S), pipe(S), signal(S), umask(S), wait(S), a.out(F), profile(M), environ(M)

Anmärkning: Kommandot readonly (utan argument) ger samma output som kommandot export.

Om << används för att förse en asynkron fil, kallad av &, med standard input, kommer shell att trassla till namngivningen av inputdokumentet: en skräpfil /tmp/sh* skapas och shell klagar på att den inte återfinns under något annat namn.

Namn: shutdown - Avslutar alla processer

Synatx: /etc/shutdown (tid) (su)

Funktion: shutdown är en del av DNIX operationsproceduer. Dess primära funktion är att avsluta alla processer som körs, på ett ordnat och försiktigt sätt. Argumentet tid om hur många minuter shutdown ska göras; default är fem minuter. Argumentet su, som är option, låter användaren bli single-user, utan att stänga systemet helt. Däremot stängs systemet för användning som multiuser. Shutdown går igenom följande steg: Först uppmanas alla inloggade användare att logga ut i ett utsänt meddelande. Alla filsystems super-block uppdateras innan systemet stannas (se sync(C)). Detta måste göras innan systemet laddas igen, för att hålla filsystemet intakt.

Hänvisning: sync(C), umount(C), wall(C)

Anmärkning: När shutdown en gång har kallats måste det tillåtas köra färdigt. Det får INTE avbrytas med BREAK eller DEL.

Namn: sleep - uppskjuter körningen

Syntax: sleep tid

Funktion: Kommandot sleep uppskjuter körningen med det antal sekunder som är angivna med argumentet tid.

Kommandot används antingen till att köra ett kommando efter en viss tid eller att köra ett kommando upprepade gånger.

Exempel: 1/ sleep 105; cat fopgk

Kommandot cat kommer att köras efter 105 s.

2/ while true do echo Kommandot sleep sleep 37 done

Kommandot echo ska köras upprepade gånger, fördröjningstiden är 37 s.

Anmärkning: Den angivna tiden måste vara mindre än 65535 sekunder.

Namn: stty - sätter terminaloptioner

Syntax: stty (-a) (-g) (optioner)

Funktion: stty sätter vissa I/O-optioner till det device/den enhet som är aktuell standard input; utan argument meddelar det hur optioner är satta; med optionen -a gäller meddelandet alla optioner; med optionen -g ger den samma information på ett format som kan användas som argument till ett annat stty-kommando. Ingående uppgifter om moderna i de fem första grupperna nedan finns i tty(M). Optioner i den sista gruppen är implementerad genom användning av optioner i de föregående grupperna. Optionerna väljs ur följande lista:

Kontrollmoder: parenb (-parenb) Sätter på (av) generering och spårning av parity.

parodd (-parodd) Väljer udda (jämn) parity.

cs5 cs6 cs7 cs8 Väljer teckenstorlek (se tty(M)).

0 Ringer av omedelbart.

50 75 110 134 150 200 300 600 1200 1800 2400 4800 9600 exta Sätter, om möjligt, terminalens baudrate till angivet nummer.

hupcl (-hupcl) Ringer av (ringer inte av) vid sista close.

hup (-hup) Samma som hupcl (-hupcl).

cstopb (-cstopb) Använder två (en) stoppbitar per tecken.

cread (-cread) Sätter på (stänger av) mottagaren.

clocal (-clocal) Antar en linje utan (med) modemkontroll.

Inputmoder: ignbrk (-ignbrk) Ignorerar (ignorerar inte) break on input.

brkint (-brkint) Ger (ger inte) signal INTR vid break.

ignpar (-ignpar) Ignorerar (ignorerar inte) paritetsfel.

parmrk (-parmrk) Märker (märker inte) paritetsfel (se tty(M)).

inpck (-inpck) Sätter på (stänger av) paritetskontroll vid input.

istrip (-istrip) Strippar (strippar inte) inputtecken till 7 bitar.

inlcr (-inlcr) Mappar (mappar inte) NL till CR vid input.

igncr (-igncr) Ignorerar (ignorerar inte) CR vid input.

icrnl (-icrnl) Mappar (mappar inte) CR till NL vid input.

iuclc (-iuclc) Mappar (mappar inte) versala bokstäver till gemena vid input.

ixon (-ixon) Sätter på (stänger av) START/STOP outputkontroll. Output stoppas genom att skicka ASCII DC3 och startas genom att skicka ASCII DC1.

ixany (-ixany) Tillåter alla tecken (bara DC1) att starta om output.

ixoff (-ixoff) Begär att systemet skickar (inte skickar) START/STOPtecken när inputkön är nästan tom/full.

Outputmoder: opost (-opost) Efterbehandlar output (efterbehandlar inte output; ignorerar alla andra outputmoder).

olcuc (-olcuc) Mappar (mappar inte) gemena bokstäver till versala vid output.

onlcr (-onlcr) Mappar (mappar inte) NL till CR-NL vid output.

ocrln (-ocrln) Mappar (mappar inte) CR till NL vid output.

onocr (-onocr) Sätter inte (sätter) output CR i kolumn noll.

onlret (-onlret) På terminalen utför NL (utför NL inte) CR-funktionen.

ofill (-ofill) Använder utfyllnadstecken (använd timing) för fördröjningar.

ofdel (-ofdel) Utfyllnadstecken är DEL (NUL).

cr0 cr1 cr2 cr3 Väljer typ av fördröjning för radmatningar (carriage returns)(se tty(M)).

bs0 bs1 Väljer typ av fördröjning för backspace (se tty(M)).

ff0 ff1 Väljer typ av fördröjning för form feed (se tty(M)).

vt0 vt1 Väljer typ av fördröjning för vertikala tabar (se tty(M)).

Lokalmoder:

isig (-isig) Sätter på (stänger av) kontrollen av tecken mot de speciella kontrolltecknen INTR och QUIT.

icanon (-icanon) Sätter på (stänger av) kanonisk input (ERASE och KILL processer).

xcase (-xcase) Kanonisk (oprocessad) presentation av upper/lowercase.

echo (-echo) Ekar (ekar inte) varje skrivet tecken.

echoe (-echoe) Ekar (ekar inte) ERASE-tecken som strängen backspace-mellanslag-backspace. OBS: denna mod raderar det tecken som ERASEas på många CRT-terminaler; däremot håller den inte reda på kolumnpositioner och kan därför förvilla vid escapetecken, tabar och backspace.

echok (-echok) Ekar (ekar inte) NL efter KILL-tecken.

echonl (-echonl) Ekar (ekar inte) NL.

noflsh (-noflsh) Stänger av (sätter på) flush efter INTR och QUIT.

Kontroll-
till-
delnigar:

nm control -tecken -C Sätter kontrolltecknet till C, där kontrolltecknet är erase, kill, intr, quit, eof, eol. Om C föregås av en caret (^) (escaped från shell), används värdet av motsvarande CNTRLtecken (dvs "^D" är CNTRL-D); "^?" tolkas som DEL och "^-" tolkas som odefinierad.

min i , time i (0<i <127) När icanon inte satts accepteras inte läsning förrän minst min tecken mottagits eller tidsgränsen time löpt ut. Se tty(C).

line i Sätter linjediciplinen till i (0<i<127). Det finns för närvarande inga linjedicipliner implementerade.

Kombinationsmoder:

evenp eller parity Sätter parenb och cs7.

oddp Sätter perenb, cs7 och parodd.

-parity, -evenp eller -oddp Stänger av parenb, sätter cs8

raw (-raw eller cooked) Sätter på (stänger av) rå input och output (ingen ERASE, KILL, INTR, QUIT, EOT eller output efterbehandling).

nl (-nl) Tar bort (sätter) icrnl, onlcr. Dessutom tar -nl bort inlcr, igncr, ocrnl och onlret.

lcase (-lcase) Sätter (tar bort) xcase, iuclc och olcuc.

LCASE (-LCASE) Samma som lcase (-lcase).

tabs (-tabs eller tab3) Sparar (expanderar till mellanslag) tabar vid skrivning.

ek Sätter ERASE- och KILL-tecken till de normala CNTRL-H och CNTRL-U.

sane Sätter alla moder till rimliga värden. Användbar när en terminals sättning blivit hopplöst rörig.

term Sätter alla moder passande för terminaltyp term, där term är en av tty33, tty37, vt05, tn300, ti700 eller tek

Hänvisning: ioctl(S), tty(M)

Anmärkning: Många kombinationer är meningslösa, men ingen kontroll utförs.

Namn: su (switch-user) - ändra användare eller bli super-user

Syntax: su (-) (namn (argument ...))

Funktion: Med kommandot su är det möjligt att under pågående körning byta användare utan att logga ut. Den nye användaren anges vid argumentet namn. Om inget namn anges kommer användaren att logga in sig som super-user.

När en användare har accepteras som super-user kommer system att skriva ut en annan prompt, vanligen , som en indikation på att det är andra privilegier som gäller.

Efter det att kommandot angivits måste rätt password anges, om användaren inte redan är super-user. När rätt password har angivits kommer kommandot att köra en ny shell med den nya användaridentiteten. För att återgå till den normala användaridentiteten, använd ctrl D dvs end-of-file skrivs till nya shell.

Alla argument som anges, skickas till shell, vilket tillåter super-usern att köra shell-procedurer med begränsade privilegier. När argument anges används alltid shell i /bin/sh. Om inga argument anges använder kommandot su det shell som specificeras i password-filen.

Om första argumentet är ett minustecken ('-') fås samma resultat som om användaren loggade in igen. Detta görs genom att invokera sh med första argumentet satt till -su, så att .profile-filen i den nya användarens hembibliotek utförs. Annars ändras endast variabeln $PATH till /bin:/etc:/usr/bin om den nya användaren är root. Observera att .profile-filen kan kontrollera om den startades med -sh eller -su.

Filer:

/etc/passwd	password filen
&HOME/.profile	användarens profile

Exempel: 1/ su

Efter att kommandot har angivits, skrivs följande ut på standard output:

Password: När användaren har angivit rätt password kommer systemet att skriva ut en ny promt, vanligen . När användaren är klar

som super-user, återvänder systemet till den normala användaridentiteten genom att trycka ctrl d.

2/ su oskar

Användaren önskar att tillfälligt logga in som användaren oskar, detta är möjligt om användaren känner till oskars password. När användaren är klar mitt sitt arbete loggar användaren ut ur oskar genom att trycka ctrl d.

Namn: sync (synchronization) - uppdatering av super-block

Syntax: sync>

Funktion: Kommandot sync tömmer den residenta informationen i skivbuffrarna ut till den fysiska enheten. Detta sker oavsett om buffrarna är fulla eller ej. Kommandot kan användas för att tvinga ut data i filsystemen. I och med att sync kommandot har genomförts är det ingen garanti för att informationen har skrivits ut till den fysiska enheten, även om den har inventerats.

Kommandot kan med fördel användas innan systemet stängs av. Men om kommandot fsck har använts ska inte sync kommandot användas innan systemet stängs av, därför att information på den fysiska enheten är mer exakt den som ligger resident i minnet.

Namn: test - provar villkor

Syntax: test uttryck (uttryck)

Funktion: test utvärderar uttrycket och returnerar en nolla som utgångsstatus om uttrycket är sant: annars ges en utgångsstatus skild från noll om argument saknas. Följande stammar kan användas för att konstruera uttrycket:

-r fil Sann om filen finns och är läsbar (readable)

-w fil Sann om filen finns och är skrivbar (writeable).

-x fil Sann om filen finns och är körbar.

-f fil Sann om filen finns och är en vanlig fil.

-d fil Sann om filen finns och är ett bibliotek.

-c fil Sann om filen finns och är en teckenspecialfil (character)

-b fil Sann om filen finns och är en blockspecialfil.

-u fil Sann om filen finns och dess set-user-bit har satts.

-g fil Sann om filen finns och dess set-group-bit har satts.

-k fil Sann om filen finns och dess sticky bit har satts.

-s fil Sann om filen finns och har en storlek större än noll.

-t Ä filbes Á Sann om den öppna fil vars filbeskrivarnummer är filbes (default är 1) är ansluten till terminaldevice.

-z s1 Sann om strängen s1 har längden noll.

-n s1 Sann om strängen s1 har en längd skild från noll.

s1 = s2 Sann om strängarna s1 och s2 är identiska.

s1 != s2 Sann om strängarna s1 och s2 inte är identiska.

s1 Sann om s1 inte är nullsträng.

n1 -eq n2 Sann om integer n1 och n2 är algebraiskt lika. Vilken som helst av jämförelserna -ne, -gt, -ge, -lt och -le kan användas i stället för -eq.

Tillägg: (Nytt i ABCenix)

Istället för n1 och/eller n2 kan uttrycket -l s1 användas, vilket ger längden av strängen s1.

Dessa stammar kan kombineras med följande operatorer:

! Negationsoperator.

-a Binär AND-operator.

-o Binär OR-operator (-a har högre rang än -o).

(uttryck) Parenteser för gruppering.

Notera att alla operatorer och flaggor är separata argument till test. Lägg också märke till att parenteser har betydelse för shell och således måste "escapas".

Hänvisning: find(C), sh(C)

Varning: Används den andra formen av kommandot (dvs den som använder Ä Å) i stället för ordet test, MÅSTE hakparenteserna åtskiljas av mellanslag.

Namn: tar (tape archive) - sparar filer i bandarkiv

Synatx: tar nyckel (namn ...)

Funktion: Kommandot tar sparar och återsparar filer på magnetband. Med det är det möjligt att skapa ett bandarkiv på en annan diskett. Tar kommandot dumpar filer till en speciel enhet på ett speciellt format, med headers, checksumma, etc. Kommandot skriver ut en trovärdig representation av biblioteksstrukturen, filer, åtkomstflaggor etc., så att ett senare tar kommando kan rekonstruera hela eller delar av de dumpade filerna.

Vad exakt kommandot tar gör bestäms helt av första argumentet, nyckeln. Detta består av en teckensträng med åtminstone en funktionsbokstav och oftast en eller flera funktionsmodifierare.

I andra argumentet specificeras vilken eller vilka filer eller bibliotek som ska dumpas eller återsparas. Om ett bibliotek anges betyder det att hela bibliteket ska dumpas dvs alla dess filer och underbibliotek.

Option: Nyckeln består av två delar, dels en funktionsbokstav, specificerad av bokstäverna r,x,t,u,c, och dels av funktionsmodifierare, 0,...,7, v w,f,b,l,m,k,n.

- Funktionsbokstav

  r De specificerade filerna sparas på slutet av bandet eller disketten.

  x De specificerade filerna hämtas från enheten. Om ett namn motsvarar ett bibliotek vars innehåll har sparats på bandet, hämtas rekursivt detta bibliotek. Ägaren, modifieringstiden och åtkomstflaggorna återsparas om möjligt. Om inget filnamn anges hämtas hela innehållet från enheten.

  Anm: Om flera entries till en och samma fil finns på enheten, skriver den sista över alla tidigare.

  t De specificerade filerna listas varje gång de förekommer på enheten. Om inga filnamn anges listas alla filer på enheten.

u De specificerade filerna sparas på enheten om de inte redan finns där eller om de har modifierats sedan de sist sparades.

c Anger för kommandot tar att detta är ett nytt arkiv. Börjar skriva i början av bandet, istället för efter sista filen. Funktionsbokstaven r är underförstådd.

- Funktionsmodifierare

0,...,7 Specificerar den yttre enheten. Default lika med 1.

v Normalt arbetar kommandot tar i tysthet. Om modifierare v sätts, sker en utskrift av varje fil och dess funktionsbokstav när den behandlas. Om funktionen t också är satt, skrivs det ut mer information om filen än bara namnet.

w Skriver ut filnamnet och vilken åtgärd som ska vidtagas. Väntar därefter på att användaren ska konfirmera. Om svaret börjar med ett 'y', utförs åtgärden annars inte.

f Kommandot tar använder nästa argument som namn på arkivet istället för /dev/mt?. Om filnamnet är '-', skriver tar till standard output eller läser från standard input, vilket som är lämpligast. I och med detta är det möjligt att använda tar som start eller avslut på en filterkedja. Kommandot tar kan också användas till att flytta hierarkier t ex

cd fromdir; tar cf - . (cd todir; tar xf -)

b Kommandot tar använder det efterföljande talet som blockfaktor för posterna på bandet. På andra yttre enheter kommer tar att buffra I/O på en buffer lika stor som det specificerade talet.(Ett tar block är 512 bytes). Default är 1 och maximum är 20.

Denna modifierare ska bara användas när råa magnetband och flexskivearkiv används, se modifierare f ovan. Poststorleken bestäms automatiskt när banden läses, funktionsbokstav x och t.

l Kommandot tar meddelar om det inte kan lösa upp alla länkar till filerna som ska dumpas. Om denna modifierare inte är

specificerad kommer inga felmeddelande att skrivas ut.

m Kommandot tar kommer inte att återspara modifieringstiden. Istället för modifieringstiden kommer tiden då hämtningen gjordes att skrivas in.

k Kommandot tar kommer att tolka nästa argument som storleken på en arkivvolym i kilobytes. Minvärdet är 250. Denna modifierare kan användas när arkivet ska läggas upp på något annat media, än magnetband, som har en bestämd storlek t ex en flexskiva. Mycket stora filer delas och läggs på olika volymer. När dessa ska återskapas kommer kommandot tar att begära en ny volym så länge som filen inte är fullständig.

n Indikerar att arkivenheten inte är ett magnetiskt band. Modifieraren k är underförstådd. Listning och hämtning av ett arkiv kan snabbas upp genom att kommandot tar vid sökning kan hoppa över en del filer. Storleken skrivs i kilobytes istället för i antal poster på bandet.

Filer: /dev/mt? /tmp/tar*

Felmeddelande: Felmeddelande skrivs ut om

- felaktig nyckeloption ges - om ett fel uppstår vid läsning/- skrivning på bandet - om inte tillräckligt minne finns för länktabellen.

Hänvisning: copy

Anmärkning: Det går inte att på något sätt fråga efter den n-te förekomsten av filen. Bandfel behandlas på ett mycket klumpigt sätt. Optionen kan vara mycket långsam. Vid uppdatering (r eller u option) eller nyskapande av ett tar arkiv får inte råa magnetband eller option b användas. Filnamnen får inte vara längre än 100 tecken.

Exempel: 1/ tar cf /dev/shl *
Alla filer som ligger i ett bibliotek på en flexskiveenhet som kallas /dev/shl, ska dumpas. Optionen c talar om för tar att det är ett nytt arkiv som ska läggas upp. Optionen f talar om för tar att nästa argument i kommandosträngen anger namnet på den fil var arkivet ska placeras.

2/ tar tf /dev/shl

Med optionen t satt skriver kommandot tar ut vad som placerats i arkviet, blockstorlek och en lista över de filer som finns i arkivet.

3/ tar xf /dev/fl0 oskar

Kommandot tar återskapar alla filer som finns på /dev/fl0 och placerar dem i biblioteket oskar.

4/ tar cvf /dev/sf0 usr/myfile

c talar om att det är ett nytt arkiv som ska skapas, v innebär lista vad som sker på standard output, f nästa argument är namnet på arkivet dit biblioteket eller filen usr/myfile ska kopieras.

5/ tar tvf /dev/sf0

t innebär att man listar innehållet i arkivet som f optionens argument är namnet på.

tv innebär att man får ut en mer fullständig information om det som finns i arkivet än om man bara använder t.

6/ tar xvf /dev/sf0 myfile

x innebär läs från arkivet som funktionsmodifieraren f argumentet är namnet på. v lista vad som sker, myfile är biblioteket/filen som ska kopieras från arkivet. Om man inte anger någon fil/bibliotek kopieras hela arkivet.

7/ tar cvfbk /dev/sf0 20 970 *

Kommandot skapar ett nytt arkiv. Modifierarna bk säger att det ska komma två argument, här 20 och 970 vilket är blocknings-faktorn respektive storleken på arkivvolymen i kbytes. Modifieraren k innebär också att när/om arkivvolymen blir full ber tar att man ska montera en ny volym. Viktigt är att ordningen på argumenten följer funktionsmodifierarnas ordning.

cvfbk /dev/sf0 20 970
är lika med
cvkbf 970 20 /dev/sf0

8/ tar rvf /dev/sf0 myfile

Biblioteket/filen myfile kommer att läggas till på slutet av arkivet oberoende av om det redan finns där och är oförändrad.

9/ tar uvf /dev/sf0 myfile

Biblioteket/filen myfile kommer att läggas till på slutet av arkivet endast om det saknas eller är förändrat. Inga filer tas någonsin bort från arkivet med detta kommando, dvs den gamla myfile finns kvar men kommer att kopieras över vid en återläsning av det nya myfile som ligger i slutet på arkivet.

Namn: time (time a command) - exekveringstid för ett kommando

Syntax: time kommando

Funktion: Kommandot time skriver ut den tid som det tar för det specificerade kommandot att exekveras. Tre tider skrivs ut, den första anger den totala tiden för exekvering utav kommandot, den andra anger tiden för exekveringen utav kommandot självt och den tredje tiden anger systemtiden som kommandot tog i anspråk. Tiderna anges i sekunder.

Körningstiden är beroende av i vilken typ av minne som programmet hamnar.

Exempel: 1/ time ps

Kommandot time anger tiden för exekvering av kommandot ps.

2/ time who who-file

Kommandot time anger den tid det tar för kommandot who att placera sin utdata i filen who-file.

3/ time ls -l

- lista från ls -

real 8.0 user 0.5 sys 2.3

Exemplet beskriver hur man får reda på hur lång tid det tar att exekvera kommandot ls -l.

Namn: touch - uppdatera modifieringstid. och användningstid på filer

Syntax: touch (-amc) (mmddttmm(åå)) filnamn...

Funktion: Modifieringstiden och användningstiden för filer uppdateras med kommandot touch genom att touch läser ett tecken från filen och därefter skriver tillbaka det.

Om filen inte finns kommer det att göras ett försök att skapa den.

Option:

-c Om filen inte finns kommer det inte att göras något försök att skapa den.

-a Ändrar bara användningstiden.

-m Ändrar bara modifieringstiden (Default är -am).

Anmärkning: Utgångsstatus från touch är det antal filer vars tider ej kunnat ändras plus det antal filer som ej fanns och ej skapades.

Exempel:

1/ touch kapl

Modifieringstiden för filen kapl uppdateras.

2/ touch k02int.f,k12int.f,k22int.f

Modifieringstiderna uppdateras för filerna k02int.f, k12int.f och k32int.f.

Namn: true - ger utgångsvärdet noll.

Syntax: true

Funktion: true gör inget annat än att ge ett utgångsvärde noll. trues motsats, false(C), gör inget annat än att ett utgångsvärde skilt från noll. true används typiskt i shellprocedurer som

while true do command done

Hänvisning: sh(C), false(C)

Namn: tty - ger terminalens namn

Syntax: tty (-s)

Funktion: Kommandot tty skriver användarens terminals pathname på standard output. Optionen -s undertrycker skrivningen, så att enbart utgångskoden testas.

Utgångskoder: 0 om standard input är en terminal, annars 1.

Anmärkning: "not a tty" skrivs på standard output om optionen -s ej givits och standard input ej är en terminal.

Namn: wait - väntar på att processen ska avslutas

Synatx: wait

Funktion: Kommandot wait väntar på att alla processer som har startas upp i bakgrunden ska avslutas och rapporterar om någon process avslutats på ett felaktigt sätt.

På grund av att systemanropet wait(S) måste exekveras av föräldrarprocessen, exekveras kommandot wait av shell själv, utan att en ny process skapas.

OBS! lla 3- eller större flerstegs pipelines är barn av shell och kan inte väntas på.

Namn: wall (write to all users) - skriver ut meddelande till alla inloggade användare

Syntax wall

Funktion: Kommandot wall läser från standard input till end-of-file. Därefter sänds meddelandet ut till alla inloggade användare. Meddelandet föregås av texten "Broadcast Message..." som talar om att det är ett meddelande till alla. Sändaren bör vara super-user för att gå förbi de skydd som varje enskild användare eventuellt kan ha anropat.

Anmärkning: wall bör endast användas för att varna användare t ex om avstängning av systemet.

Felmeddelande: Felmeddelande om någon användares tty fil inte kan öppnas.

Exempel: 1/ wall

Systemet kommer att tas ner fredagen 24/11 kl 12.00. Detta meddelande som är skrivet på terminalen kommer att skickas ut till alla användare. Detta kan se ut på följande sätt:

Brodcast Message ... Systemet kommer att tas ner fredag 24/11 kl 12.00.

Namn: who (who is using system) - vem är inloggad på systemet

Syntax: who (who-file) (am I)

Funktion: I filen /etc/utmp sparas information om vilka användare som är inloggad. Där finns också information om terminalnamn och om hur lång tid varje användare varit inloggad. När kommandot who anges skrivs denna fil ut.

Who-filen, /usr/adm/wtmp, innehåller information om alla inloggningar som gjorts sedan filen skapades. Om who-file anges i kommandot, skrivs alla login, logout och krascher ut som gjorts sedan filen wtmp skapades.

Med två argument angivna, who am I eller who are you, talar who om vem du är inloggad som.

Filer: /etc/utmp /usr/adm/wtmp

Exempel:

1/ who

Listar ut alla användare som är inloggade för tillfället.

2/ who who-file

En lista över alla in-, utloggningar och krascher sen denna fil skapades, skrivs ut på standard output. Filen kan se ut på följande sätt:

3/ who am I

Skriver ut vem du är inloggad som.

Namn: write - skriver till annan användare.

Syntax: write user (tty)

Funktion: write kopierar rader från Din terminal till en annan användares terminal. När det kallas skickar det först meddelandet

Message from ditt-loginnamn din-tty

Mottagaren ska skriva ett svar vid denna punkt. Kommunikationen fortsätter tills en end-of-file läses från terminal eller ett interrupt skickas. Då skriver write

(end of message)

på den andra terminalen och avslutar.

Vill Du skriva till en användare som loggat in mer än en gång, kan tty-argumentet användas för att ange lämplig terminal.

Tillstånd att skriva kan nekas eller godtas med kommandot mesg(C). Från början är skrivning tillåten. Vissa kommandon, särskilt nroff(CT) och pr(C), förbjuder skrivning för att förebygga rörig utskrift.

Påträffas tecknet ! i början av en rad kallar write på shell för att exekvera resten av raden som ett kommando.

Följande protokoll föreslås vid användandet av write: när du skrivit till en annan användare, vänta tills hon eller han skriver tillbaks innan Du börjar sända. Båda deltagarna bör avsluta respektive meddelanden med en tydlig signal ((o) som i "over" är konvention i engelskspråkiga länder: (k) som i "kom" på svenska?) varvid den andra kan svara: likaledes bör man kunna visa att konversationen är slut: (oo) för "over and out" på engelska; (ks)' för "klart slut" på svenska?. Det viktiga är att man är överens om signalsystemet.

Filer: etc/utmp för att hitta användare

/bin/sh för att exekvera !

Hänvisning: mail(C), mesg(C), who(C)

3 Kommandosammanfattning

basename - basename

basename sträng (suffix)

Tar bort biblioteksnamn ur sträng.

cat - catente

cat -u filnamn...

Skriver ut filer.

-u Data skrivs ut direkt till filen.

chgrp - change group

chgrp grupp namn...

Ändrar gruppidentitet.

grupp Gruppnamn eller gruppid.

chmod - change mode

chmod mod filnamn...

Vem
u Login ägare
g Grupp
o Övriga

Vad
+ Addera privilegier
- Subtrahera privilegier
= Sätt privilegier

Vilket

r Läsa
w Skriva
x Exekvera
s Sätt uid/gid till an vändarens vid exekvering
t Programkoden sparas i swaparean
u Användarprivilegier från nuvarande mod.
g Gruppprivilegier från nuvarande mod.
o Övrigas privilegier från nuvarande mod.

chown - change owner

chown ägare namn...

Ändra tillstånd till fil/bibliotek.

ägare Namnet på ägaren eller ägareid.

clock - clock

clock optioner

Visar en analog klocka på bildskärmen

-h Fönsterrubrik, standard Stockholm
-t Tidszon, anger tidsförskjutning i timmar
-n Startar klockan utan eget fönster

cmp - compare

cmp -ls filnamnl filnamn2

Jämför två filer.

-l Skriver bytenummer och skriver ut de bytes som skiljer.
-s Ger endast en utgångskod.

copy - copy files

copy -alnomradv namn... dest -t

Kopiera grupper av filer.

-a Innan kopieringen sker ställs en fråga.
-l Om ej länkningen går sker kopiering.
-n Dest. filen måste vara ny.
-o Filens ägare- och gruppid. ändras till orginalfilens.
-m Filen får samma access- och modifieringstid som orginal-
    filen.
-r Biblioteken kommer att undersökas.
-ad Frågar om optionen
-r skall anses vara satt.
-v Namnet på den fil som körs skrivs ut på terminalen.
-t Bevarar trädstrukturen på de filer/bibliotek som körs.

cp - copy

cp filnamnl filnamn2

cp filnamnl filnamn2... bibliotek

Kopierar filer och bibliotek.

date - date

date -u ååmmdd ttmm .ss

Skriver eller sätter datum och tid.

-u Avser GMT tid.

dd - dd

dd option=värdet...

Konverterar och kopierar en fil.

if=fil	Input filnamn of=fil Output filnamn
ibs=n	Blockstorlek input
obs=n	Blockstorlek output
bs=n	Blockstorlek in-/output
cbs=n	Buffertstorlek för konvertering skip=n Hoppar över n records innan kopiering seek=n Söker n records innan kopiering count=n Kopierar n records
conv=ascii	Omvandlar EBCDIC till ASCII conv=ebcdic Omvandlar ASCII till EBCDIC conv=ibm Annorlunda omvandling ASCII till EBCDIC conv=Icase Omvandlar bokstäver till gemena

demo - ett fönsterdemonstrationsprogram

Kör ett fönsterdeomonstrationsprogram som visar datorns grafiska kapacitet.

df - diskfree

df -tf (filsystem)

Anger hur mycket ledig palts det finns på disken

-t Ger totala antalet tilldelade block
-f Ger antalet fria block

dirname - directory name

dirname sträng (suffix)

Skriver ut biblioteksnamn utan suffix.

du - disk usage

du (-sar) (namn...)

Information om hur diskarean används.

-s Anger totala antalet block
-a anger antalet block per fil
-r Undertryckning av felmeddelande upphävs

echo - echo arguments

echo -n argument...

Ekar det som skrivs som argument.

-n Ingen ny rad efter utskrift

expr - expression

expr argument

Utvärderar argument som uttryck.

false - false

false

Ger utgångsvärde skilt från noll.

find - find

find pathname-list uttryck

Letar upp filer

-name fil Sann om fil matchar aktuellt filnamn
-perm onum Sann om fils tillståndsflaggor matchar oktala
            talet onum exakt
-type x Sann om filtyp är x, x=

b blockspecialfil
c teckenspecialfil
d bibliotek
f fil
n semafor eller delad datafil
p namngiven pipe

-links n Sann om filen har n länkar
-user uname Sann om filen tillhör användaren uname
-group gname Sann om filen tillhör gruppen gname
-size n Sann om filen är n block

-atime n Sann om access gjorts till fil de senaste n dagarna
-mtime n Sann om fil modifierats de senaste n dagarna
-ctime n Sann om fil ändras de senaste n dagarna
-exec cmd Sann om körda cmd ger utgångsvärde noll som utgångsstatus
-ok cmd Som exec men innan körning ställs en fråga
-print Alltid sann;skriver ut aktuellt pathname
-newer fil Sann om aktuell fil modifierats senare än argumentsfil
(uttryck) Sann om uttryck är sann

fsck - file system check

fsck -y -n -s -t filsystem...

Kontroll av filsystemen.

-y Frågorna antas besvarade med ja
-n Frågorna antas besvarade med nej --
-s Ignorerar gamla bitmappen, konstruerar en ny
-t Nästa argument används som namn på arbetsfilen

format - formatterar skivor

Formatterar en flexskiva så att den blir användbar

fscl - filesystem clean

fscl special

Testar om filsystemet på en disk är intakt.

haltsys - halt system

/etc/haltsys

Stänger filsystemen och stoppar CPUn.

isamin - isam init.program

isamin

Initieringsprogram för isam.

kill - stop background processes

kill (-signal) processid...

Avslutar processer som går i bakgrunden.

ln - make link

ln filnamnl filnamn2

Skapar en länk mellan filer

login - login

login (användarnamn)

Inloggning till systemet

lpr - line printer spooler

lpr (-r -c -m -n) filnamn

Spooler för skrivare.

-r Filen ställs i kö, tas sedan bort
-c Kopierar filen
-m Rapporterar via mail när utskriften är klar
-n Rapporterar inte när utskriften är klar, default

l,ls - list current directory

ls -eltasdrucifg namn...

Listar information onm filer och innehållet i biblioteken.

-e Listar alla subbibliotek.
-l Listar på långt format
-t Listar efter uppdateringstid
-a Listar alla entries
-s Anger antalet 512-bytes block i varje fil/bibliotek
-d Namnet på biblioteket skrivs ut
-r Listar i omvänd ordning
-u Listar after accesstiden
-c Listar efter i-nodens sista modifieringstid
-i Skriver filens i-nummer på utskriften
-f Argumenten tolkas som bibliotek.

Optionerna -ltsr tas bort och option -a sätts -g Anger grup-
pid vid listning med -l

mkdir - make a directory

mkdir bibliotek...

Skapar nya bibliotek.

mkfs - make file system

mkfs optioner ... ofile

Konstruerar ett filsystem.

-adress namn	Kopierar filens namn till adress
-b bstorlek	Volymens blockstorlek sätts till bstorlek
-d	Sätter debug mod
-n nmax	Allokerar utrymmet nmax för inoderna
-r directory	Nya filsystemet får directory till n:t root
-s sstorlek	Allokerar en kont. swaparea med storlek sstorlek
-v vstorlek	Sätter volymens storlek till vstorlek (antal block)
-x xfil	Läser från badspot definitionstabell

mknod - make nod

/etc/mknod namn (c)(b) primär sekundär
/etc/mknod namn p

Gör speciella filer.

-c	Special av teckentyp
-b	Specialfil av blocktyp
-p	Skapar namngivna pipes

mkuser - make user

mkuser (användarnamn)

Lägger till en användare.

mntchk - mount check

mntchk (fhnd) special bibliotek

Mount på specificerad enhet.

mount - mount file system

mount fhnd special -r

Lägger till ett filsystem.

-r Filsystemet som flyttas är endast läsbart

mv - move files

mv fill fil2

mv fill ... bibliotek

Flyttar och byter namn på filer.

ncu - net call UNIX

ncu option ... systemid

Nätverksanrop i ABCenix.

-s Användarnamn ges istället för systemid
-B Används av netman för att starta B-sidan
-mnamn netman använder den network manager som anges av namn
-xnn Sätter debuggnivån till nn

netman - netman

netman - Äoption ...Å

Network transport manager på D-NIX-system.

-cnamn Öppnar namn som network controller
-tnn Sätter korttyp till -l nn
-dnamn eller -d Laddar debuggern
-pxnamn Kör över defaultvalen av filer till boot
-mnamn Netman ansluter sig vid namn
-xnn Sätter debuggnivån till nn -l Netman simulerar controllerns uppförande lokalt

nice - put command in background

nice -nummer kommando

Kör kommando med lägre prioritet i bakgrunden, och ändrar prioritet.

-nummer Värde som prioriteten räknas ned med

nohup - no hang up

nohup kommandog up

Kör de kommando som inte känner av 'hangup' och stoppsignaler från terminalen.

od - octal dump

od -bcdox filnamn + offset . b

Dumpar filer.

filnamn	Default standard input
offset	Offset hos fil som dumpas, oktala bytes
+	Föregår offset om inget filnamn anges
.	Offset tolkas som decimala bytes
b	Offset tolkas i 512-bytes block

passwd - password

passwd namn

Ändrar lösenordet.

pr - print

pr +n -n -h -wn -ln -t -sc -m -b filnamn

Formatterar textfiler.

+n	Skriv ut kolumn n
-n	Börja skriva på sid n
-h	Nästa argument är ett sidhuvud
-wn	Ändra papperbredden till n tecken, default 80
-ln	Ändra längen på sidan till n rader, default 72
-t	Skriv inte sidhuvudet eller sidfoten på 5 rader
-sc	Separera kolumnera med c
-m	Skriver alla filer samtidigt i varsin kolumn
-b	Sidorna separeras med formfeed tecken

ps - process status

ps Ä lax Å

Processtatus.

-l	Listar på långt format
-a	Information om alla processer med terminaler
-x	Information om alla processer utan terminaler

pwd - print working directory

pwd

Anger vilket bibliotek som är aktuellt.

rm - remove files

rm -f -r -i -v filnamn...

Tar bort filer.

-f Ingen fråga innan filen tas bort, gäller skrivskyddade
-r Inget meddelande om filen är ett bibliotek
-i Frågar om filen är ett bibliotek
-ir Frågar om biblioteket ska undersökas

rmdir - remove directories

rmdir bibliotek...

Tar bort tomma bibliotek.

rmuser - remove user

rmuser (-nr) (användarnamn)

Tar bort en användare.

-n Ta ej bort användarens hembibliotek
-r Ta bort användarens hembibliotek även om det ej är tomt

rx - remote execute

rx Äoption...Å systemid!kommando

Äparametrar...Å

Kör ett kommando på en annan maskin

luid Ä :lösenord Å Loggar in som uid med lösenord på den andra maskinen
-ddir Ändrar aktuellt bibliotek till dir innan kommandot körs
-B Används av netman att starta B-sidan
-mnamn rx använder en network manager ansluten vid namn
-xnn Sätter debuggnivån till nn

setspeed - set speed

setspeed (-s -t -r) (enhet)

Sätter överföringshastigheten på ett device tex lp

-s överföringshastighet i bit/s
-t sändningshastighet vid split speed
-r mottagnigshastighet

settimezon - swt time zon

Sätt korrektion i förhållande till GMT

sh - shell

sh (-ceiknrstuvx) (argument)

Anropar kommandotolken shell.

-c sträng Kommandot läses från sträng
-e Avslutar omedelbart om ett kommando avslutas med utgångsstatus skild från noll
-i Shell interaktivt
-k Placerar alla nyckelordsargument i omgivningen till ett kommando
-n Läser kommando men kör dem ej
-r Shell är restricted shell
-s Kommando läses från standard input
-t Avslutar efter att ha läst och kört ett kommando
-u Behandlar icke satta variabler som fel vid substitueringen
-v Skriver ut input rader från shell allt eftersom de läses
-x Skriver ut kommando och dess argument ellt eftersom de läses

shutdown - shutdown

/etc/shutdown (tid) (su)

Avslutar alla processer.

sleep - suspend execution

sleep tid

Uppskjuter exekvering.

stty - set tty

Sätter terminaloptioner

su - switch-user

su (-) (namn (argument...))

Ändra användare eller bli super-user.

sync - synchronization

sync

Uppdatering av superblock.

tar - tape archiver

tar nyckel namn...

Sparar filer på arkivformat.

Nyckel består av två delar:

1. r Filerna sparas i slutet av arkivet
   x Filerna hämtas från arkivet om inget filnamn anges, hämtas hela arkivet
   t Filerna listas varje gång de förekommer i arkivet
   u Filerna sparas om de inte finns i arkivet eller om de modifierats
   c Skapar ett nytt arkiv

2. 0-7 Specifierar yttre enhet.
   v Utskrift av varje fil när den behandlas
   w Frågar om den speciella åtgärden skall utföras.
   f Tar nästa arg. som namn på arkivet
   b Tar nästa arg.
   n som blockningsfaktor för band record
   l Signalerar om inte alla länkar till filerna kan lösas upp
   m Modifieringstiden sätts till den tid då hämtning skedde
   k Tolkar nästa arg. som storleken på en erkivvolym i kilobytes
   n Arkivenheten är ej ett magn.band

test - test

test uttryck

Provar villkor.

-r fil	Sann om fil existerar och är läsbar
-w fil	Sann om fil existerar och är skrivbar
-x fil	Sann om fil existerar ovh är exekverbar
-f fil	Sann om fil existerar och är vanlig fil
-d fil	Sann on fil existerar och är bibliotek
-c fil	Sann om fil existerar och ör teckenspecialfil
-b fil	Sann om fil existerar och är blockspecialfil
-u fil	Sann om fil existerar och dess set-user-bit har satts
-g fil	Sann om fil existerar och dess set-group bit har satts
-k fil	Sann inm fil existerar och dess sticky bit har satts
-s fil	Sann om fil existerar och har storlek större än noll

```
-t ÄfilbesÅ Sann om fil vars filbeskrivarnummer är filbes är
           ansluten till terminal device
-z sl      Sann om sträng har längden noll
-n sl      Sann om sträng sl har längd skild från noll
sl=s2      Sann om strängarna identiska
sl !=s2    Sann om strängarna inte är identiska
sl         Sann om sl inte är nullsträng
nl -eq n2  Sann om integer nl och n2 är algebraiskt lika.

           OBS!Istället för nl och/eller n2 kan -l sl
           användas, ger längden av sl
```

time - time a command

time kommando

Exekveringstid för ett kommando

touch - update date of a file

touch (-amc)(mmddttmm(åå)) fil...

Uppdaterar modifierings- och användningstid på filer

-a Ändrar bara användningstiden
-c Om filen ej finns kommer inget försöka att skapa den att göras
-m Ändrar bara modifieringstiden

true - true

true

Ger utgångsvärdet noll.

tty - tty

tty (-s)

Väntar på att processen skall avslutas.

-s Undertrycker skrivning enbart utgångskoden testas

umount - unmount file systems

/etc/umount special

Tar bort filsystem

wait - await completion of process

wait

Väntar på att processen skall avslutas.

wall - write to all users

wall

Skriver till alla inloggade användare.

who - who is using system

who (who-file) (am I)

Vem är inloggad på systemet.

write - write

write user (tty)

Skriver till annan användare.