# RSX–11M Operating System Internals

## Tests/Exercises

**digital**

# RSX-11M Operating System Internals

## Tests/Exercises

The following are trademarks of Digital Equipment Corporation,
Maynard, Massachusetts:

| | | |
|---|---|---|
| DIGITAL | DECsystem-10 | MASSBUS |
| DEC | DECSYSTEM-20 | OMNIBUS |
| PDP | DIBOL | OS/8 |
| DECUS | EDUSYSTEM | RSTS |
| UNIBUS | VAX | RSX |
| | VMS | IAS |

# CONTENTS

## GENERAL INSTRUCTIONS

### Tests

Written tests should be taken at the completion of the indicated
module. Your instructor will provide specific instructions for
each.

### Exercises

Exercises will be performed on a live RSX-11M system. Your
instructor will provide you with a privileged account. For the
SYSGEN, you will be provided with a standalone system. Your
instructor will divide the class into small groups, each of which
will perform their own SYSGEN.

Some questions require added instructions from your instructor
(adding of a new directive to your system). These instructions
will be provided by the instructor when needed.

# Overview of RSX-11M

## TEST/EXERCISE

1. List the major design goals of RSX-11M.

2. List the major features of RSX-11M.

3. List the three major components of RSX-11M and give an example of each.

# The System Data Base

## TEST/EXERCISE

1. Use the LOOK task to perform the following:

   A. Determine the following data about the task 'LOOKnn' (where nn is your terminal number):

      Bits set in the task status words T.STS, T.ST1, and T.ST2

      The partition in which the task is executing

      The LUNs in use and the devices assigned to them

   B. Determine what devices are in the system and how many units there are of each device type.

2. What is the algorithm used to allocate space in the DSR?

## Executive Modules

## TEST/EXERCISE

Which Executive module contains each of the following:

1. Listheads and pointers used by the system to access the system database.

2. The directive dispatch table.

3. The trap and interrupt vectors.

4. Code that does context switching.

5. System entrance and exit routines.

6. Routines used to insert entries into queues.

7. The routine used to allocate POOL space.

# File System Overview

## TEST/EXERCISE

Use the LOOK task to perform the following:

1.  Determine the actual physical blocks allocated to the file
    FILE.DAT located on the system disk.

2.  Use the LOG option with the LOOK task and locate the LUN  used
    and the VCB of the log file.

3.  Access the file FILE.DAT by performing the same steps that the
    file system performs, (i.e., map to the disk, find the user
    directory in the MFD, read the header for the user  directory,
    and obtain the retrieval pointers).

# System Generation

## TEST/EXERCISE

1.  Your instructor will provide you with a  SYSGEN  kit.   Follow
    his instructions and perform Phases 1 and 2 of SYSGEN.

2.  Run VMR on the system and save the image.

# Physical and Virtual Memory

## TEST/EXERCISE

1.  Use the LOOK task to perform the following:

    A.  Determine the current subpartitions of GEN and the names of tasks using them.

    B.  Determine what tasks are attached and/or mapped to FCSRES.

    C.  Using the system map, find and diagram the system checkpoint file structure.


2.  Indicate the values loaded into the PARs for the following tasks:

    A.  Task is non privileged

        Length = 11K words

        Loaded in region starting at 00234500


    B.  Task is mapped to a PI common of length 5K words

        Length = 5K words

        Located at 01443500


    C.  Task is mapped to a non-PI library with base address 120000

        Length = 3K words

        Located at 00732100


3.  How many windows does the task have?

# Task Management

## TEST/EXERCISE

1.  Use the LOOK task to perform the following:

    A.  Install the task TASKA and find it within the STD.

    B.  Run TASKA and find its location within the ATL.

    C.  Find the TCB of the task ...PIP and determine its status and characteristics.

    D.  List the steps performed when a task terminates abnormally.


2.  Write a task which will do the following:

    A.  Prompt for input of the name of a task image file.

        Use the data contained in the task header and task label blocks to print relevant task information.

# System Synchronization

## TEST/EXERCISE

Diagram each of the following:

1.  How the system processes a software trap.

2.  How the system processes a hardware interrupt.

3.  How FORK processing works.

4.  How the system handles a switch to system state by a privileged task.

# Writing Privileged Tasks

## TEST/EXERCISE

All of the following are to be privileged tasks.

1. Modify the task POOLA to find the number of free blocks, the largest free block, and the smallest free block in POOL.

2. Write a task which lists the names of all tasks checkpointed in the system checkpoint file(s).

3. Write a task which will create a copy of its TCB in the STD with the name ABCTnn where nn is your terminal number. Task ABCTnn should not be active. Do a TAL command to check the results.

4. Write a task which lists the following for each terminal:

   Buffer size

   Terminal type

   Privilege

**System Crashes**

**TEST/EXERCISE**

Using either a CDA generated by one of your  privileged  tasks  or
one provided by your instructor, attempt to determine the cause of
the crash.  Discuss your results with the instructor in Lab.

# Directive Processing

## TEST/EXERCISE

1. Draw a diagram showing the steps needed to process a directive.

2. Write a directive processing routine to return to the user task the terminal number from which the task is run. The routine can be tested as a privileged task.

3. Write the macros needed to invoke the directive.

4. Include your directive in your system. This will require the following steps:

   Rebuild EXCOM2 to include your routine.

   Modify the DDT in DRDSP and rebuild EXCOM1.

   Run VMR to reconfigure your system.

# I/O Processing

## TEST/EXERCISE

1. Trace the flow of an I/O request through the I/O system.

2. List the three types of ACPs and indicate the principal differences between them.

3. What checks are performed within DRQIO?

## Device Driver Processing

## TEST/EXERCISE

1.  How are UNIBUS Mapping Registers used?

2.  What are the entry points to a device driver and how are  they used?

3.  Using the LOOK task, determine the status of your terminal  by examining the status bits in the UCB.

LOOK is a program that runs on RSX-11M/M-PLUS operating systems to read control structures from system pool, secondary pool, and structured disks.

The LOOK task begins with only three links to the system data structures.  These are:

     $DEVHD - Link to the first DCB
     $PARHD - Link to the first PCB
     $TSKHD - Link to the first TCB

Commands

Only the first letter of any of the keywords(Read, Scan, etc.) need be entered, but if more than one letter is entered then the whole word must be entered.


ADDRESS

Print the memory or disk address of the last block read.


EXPLODE

Turn on the explode option, which causes all status fields, protection words, and other appropriate fields to be printed bit by bit with corresponding field names and switch settings.


FORMAT="BLOCK":"FORMAT"

Change the default printout format for the specified control block type.   See Table A-1 for a list of the default format for each of the control blocks.  An acceptable new format for a control block depends on the residence of the block.  A memory-resident control block can be defaulted to any memory-resident format;  a disk-resident control block can be defaulted to any disk format; and any control block can be defaulted to the dump formats of ASCII, MIXED, OCTAL, and RAD50.


HEAD="LISTHEAD"

Read the control block at the beginning of the specified list.

LOG[:"FILENAME"]

This command creates a file and causes all terminal I/O to be
copied to the file in addition to appearing at the terminal.
Exiting from LOOK closes the log file.  The command -LOG stops
logging but does not close the file.


MAP

Allows access to disk-resident data structures.


NEXT[="BLOCK"][:"ADDRESS"]

Read the next of the specified control block.  If the block is not
specified, LOOK defaults to the next of the last block name
referenced.  If an address is given, it is used instead of the one
in LOOK's control tables.


PRINT[:"FORMAT"]

Print the control block that was just read.  If a format is given,
it is used instead of the default format, but it does not change
the default format.


READ[="BLOCK"][:"ADDRESS"]

Read the control block specified by the current pointer in control
tables.  If a block is not specified, the default is the last
control block referenced.  If address is given, then that address
is stored as the current address in the control tables.


SCAN="LIST":"NAME"[:PRINT]

Scan the memory-resident control blocks of the specified listhead
until one is found that matches the specified name.  If a matching
name is not found, a scan failure message is printed.

The print option on the end of the command causes all control
blocks read in the scan to be printed.


VERIFY

Print the current control block at the end of the execution of
each non-null command line.  The command -VERIFY turns off the
VERIFY switch.

## Table A-1  Available Data Blocks

| Block Name | | Format Name | Location |
|---|---|---|---|
| ADB | Attachment Descriptor | ADB | POOL |
| AST | Asynchronous System Trap Block | AST | POOL |
| CQCB | Clock Queue Control Block | CQCB | POOL |
| DCB | Device Control Block | DCB | POOL |
| DISK | Disk Data File | OCTAL | DISK |
| FCB | File Control Block | FCB | ACP |
| GGEF | Grp.Glbl EF Control Block | GGEF | POOL |
| HEADER | Disk File Header | HEADER | DISK |
| HOME | Disk Home Block | HOME | DISK |
| IOP | I/O Packet | IOP | POOL |
| ITB | Interrupt Transfer Block | ITB | POOL |
| LACB | Logical Assignment Control Block | LACB | POOL |
| LB | Lock Block | LB | POOL |
| MAB | Mapping Assignment Block | MAB | POOL |
| MFD | Master File Directory | MFD | DISK |
| MVL | Mounted Volume List | MVL | POOL |
| OCB | Offspring Control Block | OCB | POOL |
| PCB | Partition Control Block | PCB | POOL |
| SCB | Status Control Block | SCB | POOL |
| TCB | Task Control Block | TCB | POOL |
| THEAD | Task Header | THEAD | POOL |
| UCB | Unit Control Block | UCB | POOL |
| WB | Window Block | WB | POOL |

## Example 1

```
F11>S=TCB:...EDT
F11 -- reading TCB    from    address 35560
F11 -- reading TCB    from    address 114404
F11 -- reading TCB    from    address 113760
F11 -- reading TCB    from    address 42354
F11 -- reading TCB    from    address 116204
F11 -- reading TCB    from    address 114174
F11 -- reading TCB    from    address 57330
F11 -- reading TCB    from    address 115340
F11 -- reading TCB    from    address 115024
F11 -- reading TCB    from    address 114614
F11 -- reading TCB    from    address 114510
F11 -- reading TCB    from    address 52564
F11 -- reading TCB    from    address 40034
F11 -- reading TCB    from    address 54670
F11 -- reading TCB    from    address 57644
F11 -- reading TCB    from    address 117460
F11 -- reading TCB    from    address 117354
F11 -- reading TCB    from    address 45440
F11 -- reading TCB    from    address 116310
F11 -- reading TCB    from    address 116524
F11 -- reading TCB    from    address 115660
F11 -- reading TCB    from    address 115234
F11 -- reading TCB    from    address 114720
F11 -- reading TCB    from    address 113544
F11 -- reading TCB    from    address 114300
F11 -- reading TCB    from    address 45654
F11 -- reading TCB    from    address 52144
F11 -- reading TCB    from    address 61114
F11 -- reading TCB    from    address 61324
F11 -- reading TCB    from    address 61430
F11 -- reading TCB    from    address 61534
F11 -- reading TCB    from    address 116734
F11 -- reading TCB    from    address 51210
F11 -- reading TCB    from    address 54774
F11 -- reading TCB    from    address 56370
F11 -- reading TCB    from    address 114064
F11 -- reading TCB    from    address 115444
F11 -- reading TCB    from    address 117250
F11 -- reading TCB    from    address 43214
F11 -- reading TCB    from    address 45334
F11 -- reading TCB    from    address 57504
F11 -- reading TCB    from    address 47520
F11 -- reading TCB    from    address 47730
F11>P
    T.LNK   0
    T.PRI   101
    T.IOC   0
    T.CPCB  47230
    T.NAM   ...EDT
```

```
      T.RCVL 0,47742
      T.ASTL 0,47746
      T.EFLG 0,0
      T.UCB  0
      T.TCBL 44240
      T.STAT 1000000000000000
      T.ST2  0010000000000000
      T.ST3  0100000000000000
      T.DPRI 101
      T.LBN  0,22,1
      T.LDV  22232
      T.PCB  37070
      T.MXSZ 302
      T.ACTL 0
      T.SAST 0
      T.CHR  4
      T.TIO  120
      T.TKSZ 0
      T.ATT  0,0
      T.OFF  50014
             0
      T.SRCT 0
      T.RRFL 50020,0
      T.OCBH 0,0
      T.RDCT 33
F11>N
F11 -- reading TCB    from   address 44240
F11>P
      T.LNK  44360
      T.PRI  101
      T.IOC  1
      T.CPCB 47230
      T.NAM  EDTT15
      T.RCVL 0,44252
      T.ASTL 0,44256
      T.EFLG 1001,100000
      T.UCB  27622
      T.TCBL 44364
      T.STAT 0000000000000000
      T.ST2  0010000000000001
      T.ST3  0110100000000000
      T.DPRI 101
      T.LBN  0,22,1
      T.LDV  22232
      T.PCB  41400
      T.MXSZ 302
      T.ACTL 57504
      T.SAST 37602
      T.CHR  236
      T.TIO  102
      T.TKSZ 0
      T.ATT  0,0
```

```
      T.OFF  44324
             Ø
      T.SRCT Ø
      T.RRFL 44330,0
      T.OCBH Ø,4315
      T.RDCT 4
Fll>R=PCB
Fll -- reading PCB    from  address 41400
Fll>P
      P.LNK  Ø
      P.PRI  1Ø1
      P.IOC  Ø
      P.NAM  GEN
      P.SUB  4355Ø
      P.MAIN 3707Ø
      P.REL  1Ø212
      P.SIZE 3Ø2
      P.WAIT Ø
      P.SWSZ 3Ø2
      P.BUSY Ø
      P.TCB  4424Ø
      P.STAT ØØØ1ØØØØØØØØ1ØØØ
      P.HDR  6774Ø
      P.PRO  Ø
      P.ATT  37574
Fll>R=THEAD
Fll -- reading THEAD  from  address 6774Ø
Fll>P
      H.CSP  1156
      H.HDLN 232
      H.EFLM 2
      H.     44262
      H.CUIC 55,3Ø1
      H.DUIC 55,3Ø1
      H.IPS  1111ØØØØØØØØ1111
      H.IPC  2ØØ64
      H.ISP  1232
      H.ODVA Ø
      H.ODVL Ø
      H.TKVA Ø
      H.TKVL Ø
      H.PFVA Ø
      H.FPVA Ø
      H.RCVA Ø
      H.EFSV Ø
      H.FPSA Ø
      H.WND  7ØØ66
      H.DSW  ØØØØØØØØØØØØØØØ1
      H.FCS  27662
      H.FORT Ø
      H.OVLY Ø
      H.VEXT Ø
```

```
        H.SPRI 5
        H.NML  Ø
        H.RRVA Ø
        H.GARD 7Ø17Ø
        H.NLUN 6
        H.LUN  35446
        H.     3754Ø
        H.LUN  21222
        H.     4126Ø
        H.LUN  35446
        H.     Ø
        H.LUN  21222
        H.     4Ø14Ø
        H.LUN  35326
        H.     Ø
        H.LUN  35352
        H.     Ø
F11>R=UCB:35446
F11 -- reading UCB    from  address 35446
F11>P
        U.OWN  12
        U.DCB  35434
        U.RED  21554
        U.CTL  Ø
        U.STS  ØØØØØØØØ
        U.UNIT Ø
        U.ST2  ØØØØØØØØ
        U.CW1  ØØØ1ØØØØØØØØØØØØØ
F11>R=DCB:35434
F11 -- reading DCB    from  address 35434
F11>P
        D.LNK  Ø
        D.UCB  35446
        D.NAM  SY
        D.UNIT Ø
               Ø
        D.UCBL 12
        D.DSP  35434
        D.MSK  21554,Ø,Ø,1ØØØØ,176,35422,Ø,321Ø
        D.PCB  1
F11>R=UCB:21554
F11 -- reading UCB    from  address 21554
F11>P
        U.OWN  Ø
        U.DCB  21514
        U.RED  21554
        U.CTL  321
        U.STS  ØØØØØØØ1
        U.UNIT Ø
        U.ST2  ØØØØØ1ØØ
        U.CW1  11ØØØØØØ1ØØØ1ØØØ
        U.CW2  ØØØØ1ØØ1ØØØØØØØ1Ø
```

```
        U.CW3   0000001001100000
        U.CW4   0000001000000000
        U.SCB   22012
        U.ATT   0
        U.BUF   2571,165100
        U.CNT   52200
        U.ACP   117354
        U.VCB   370
        U.UMB   35560
        U.PRM   0,403
        U.KRB1  35514
Fll>R=DCB:21514
Fll -- reading DCB    from   address 21514
Fll>P
        D.LNK   22172
        D.UCB   21554
        D.NAM   DR
        D.UNIT  0
                3
        D.UCBL  50
        D.DSP   120102
        D.MSK   177477,70,0,177200,377,0,0,377
        D.PCB   36714
```

## Example 2

*LOO> SCAN*

```
Fll>S=UCB:DB1
Fll -- READING DCB FROM ADDRESS 20500
Fll -- READING DCB FROM ADDRESS 20540
Fll -- READING DCB FROM ADDRESS 20606
Fll>MAP A...
Fll -- READING HOME          FROM LBN 0/1
Fll -- READING HOME          FROM LBN 2/114557
Fll -- READING HOME          FROM LBN 2/114562
```

*(READ)* 
```
Fll>R=MFD:1/P  -PRINT
Fll -- READING MFD FROM LBN 2/137204          VBN 0/1
DIRECT 1,1 INDEXF     .SYS;1
DIRECT 2,2 BITMAP     .SYS;1
DIRECT 3,3 BADBLK     .SYS;1
DIRECT 4,4 000000     .DIR;1
DIRECT 754,17 RSX11    .SYS;4
DIRECT 730,21 RSX11    .SYS;3
DIRECT 720,15 RSX11    .SYS;2
DIRECT 11604,2 RSX11    .SYS;1
DIRECT 270,24 001001    .DIR;1
DIRECT 554,16 001002    .DIR;1
DIRECT 1300,31 001003    .DIR;1
DIRECT 4666,3 001006    .DIR;1
DIRECT 12150,2 001007    .DIR;1
DIRECT 12467,12 001011    .DIR;1
DIRECT 556,5 001020    .DIR;1
DIRECT 603,2 001024    .DIR;1
DIRECT 2210,6 001034    .DIR;1
DIRECT 605,2 001054    .DIR;1
DIRECT 772,26 001200    .DIR;1
DIRECT 10550,6 002054    .DIR;1
DIRECT 606,2 002200    .DIR;1
DIRECT 1035,41 005054    .DIR;1
DIRECT 647,2 006054    .DIR;1
DIRECT 474,16 007001    .DIR;1
DIRECT 1447,36 007004    .DIR;1
DIRECT 6125,36 007007    .DIR;1
DIRECT 260,2 007010    .DIR;1
DIRECT 11626,5 007021    .DIR;1
DIRECT 4057,26 007022    .DIR;1
DIRECT 7653,10 007025       .DIR;1
DIRECT 10402,1 007026    .DIR;1
```

*NEXT VIRTUAL BLOCK*
```
Fll>N=MFD/P
Fll -- READING MFD FROM LBN 2/137205          VBN 0/2
DIRECT 4077,13 007031    .DIR;1
DIRECT 4112,64 007043    .DIR;1
DIRECT 10577,3 007303    .DIR;1
DIRECT 10600,5 007305    .DIR;1
DIRECT 1313,17 007320    .DIR;1
DIRECT 12766,1 010001    .DIR;1
DIRECT 1161,3 010010    .DIR;1
```

37

```
        DIRECT 607,2 011024      .DIR;1
        DIRECT 454,16 011034     .DIR;1
        DIRECT 611,2 012024      .DIR;1
        DIRECT 613,2 013024      .DIR;1
        DIRECT 615,2 015024      .DIR;1
        DIRECT 617,2 017024      .DIR;1
        DIRECT 621,2 022024      .DIR;1
        DIRECT 623,2 023024      .DIR;1
   *    DIRECT 625,2 024024      .DIR;1
        DIRECT 627,2 025024      .DIR;1
        DIRECT 1507,3 026024     .DIR;1
        DIRECT 4772,1 031010     .DIR;1
        DIRECT 5131,1 031020     .DIR;1
        DIRECT 4120,3 031040     .DIR;1
        DIRECT 5314,5 031060     .DIR;1
        DIRECT 4153,5 042030     .DIR;1
        DIRECT 631,2 046020      .DIR;1
        DIRECT 632,2 046030      .DIR;1
        DIRECT 633,2 054020      .DIR;1
        DIRECT 634,2 054030      .DIR;1
        DIRECT 635,2 055020      .DIR;1
        DIRECT 636,2 055030      .DIR;1
        DIRECT 637,2 061020      .DIR;1
        DIRECT 640,2 061030      .DIR;1
        DIRECT 641,2 070020      .DIR;1
        F11>R=HEADER 625/P
        F11 -- READING HEADER        FROM LBN 2/115403        VBN 0/647
        H.IDOF 27
        H.MPOF 56
        H.FNUM 625
        H.FSEQ 2
        H.FLEV 401
        H.FOWN 24,24
        H.FPRO 160000
        H.UCHA 200
        H.SCHA 0
        H.UFAT 1,20,0,1,0,2,0,0,0,0,0,0,0,0,0,0
        I.FNAM 024024
        I.FTYP DIR
        I.FVER 1
        I.RVNO 0
        I.RVDT --
        I.RVTI ::
        I.CRDT 28-JUN-79
        I.CRTI 20:41:55
        I.EXDT --
        M.ESQN 0
        M.ERVN 0
        M.EFNU 0
        M.EFSQ 0
        M.CTSZ 1
        M.LBSZ 3
```

FILE NUMBER

```
M.USE   2
M.MAX   314
M.LBN   1/36114
M.LEN   Ø
F11>R=UFD:1/V   VIRTUAL BLOCK
F11 -- READING UFD FROM LBN 1/36114            VBN Ø/1
DIRECT 2751,4 BPRAST    .OBJ;1
DIRECT 2753,4 BPRCV     .OBJ;1
DIRECT 2755,4 BPRDAT    .OBJ;1
DIRECT 2757,4 BPREOJ    .OBJ;1
DIRECT 2763,4 BPRER     .OBJ;1
DIRECT 2765,4 BPRIC     .OBJ;1
DIRECT 2771,4 BPRPF1    .OBJ;1
DIRECT 2773,4 BPRPF2    .OBJ;1
DIRECT 2777,4 BPRPF3    .OBJ;1
DIRECT 3001,4 BPRPF4    .OBJ;1
DIRECT 3003,4 BPRPR     .OBJ;1
DIRECT 3007,4 BPRPRF    .OBJ;1
DIRECT 3011,4 BPRRM     .OBJ;1
DIRECT 3014,4 BPRRT     .OBJ;1
DIRECT 3017,4 BPRSPW    .OBJ;1
DIRECT 3021,4 BPRSRV    .OBJ;1
DIRECT 3763,4 BPRST     .OBJ;1
DIRECT 3765,4 BPRTOT    .OBJ;1
DIRECT 3767,4 INREQ     .OBJ;1
DIRECT 3771,4 LNPRS     .OBJ;1
DIRECT 3773,4 OUTREQ    .OBJ;1
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
DIRECT Ø,Ø          .   ;Ø
F11>R=HEADER:3773
F11 -- READING HEADER       FROM LBN 2/12Ø551     VBN Ø/4015
H.IDOF 27
H.MPOF 56
H.FNUM 3773
H.FSEQ 4
H.FLEV 4Ø1
H.FOWN 24,24
H.FPRO 16ØØØØ
H.UCHA Ø
H.SCHA Ø
H.UFAT 2,52,Ø,2,Ø,2,576,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
I.FNAM OUTREQ
I.FTYP OBJ
```

*FILE NUMBER* (handwritten annotation, left margin)

```
I.FVER 1
I.RVNO 7
I.RVDT 25-APR-79
I.RVTI 10:35:02
I.CRDT 21-FEB-78
I.CRTI 19:03:11
I.EXDT --
M.ESQN 0
M.ERVN 0
M.EFNU 0
M.EFSQ 0
M.CTSZ 1
M.LBSZ 3
M.USE  2
M.MAX  314
M.LBN  1/36204
M.LEN  1
F11>R=DISK:1
F11 -- READING DISK        FROM LBN 1/36204           VBN 0/1
OCTAL  52,1,60434,70531,0,0,140130,0
OCTAL  3000,0,127401,7624,2514,0,21612,143610
OCTAL  2100,0,21612,143660,2100,0,52,1
OCTAL  21612,151740,2100,0,21612,152010,2100,0
OCTAL  34444,3237,2100,0,34444,102716,2100,0
OCTAL  35264,71142,2100,0,52,1,35524,75013
OCTAL  2100,0,73643,76400,2100,0,75416,140700
OCTAL  2100,0,75416,144000,2100,0,75460,111504
OCTAL  2100,0,52,1,124444,7716,2100,0
OCTAL  124632,52077,2100,0,124674,3553,2100,0
OCTAL  125247,7716,2100,0,125247,63534,2100,0
OCTAL  52,1,125507,45726,2100,0,125525,100054
OCTAL  2100,0,125710,4746,2100,0,0,0
OCTAL  2450,1006,125455,77741,2150,326,12,1
OCTAL  127401,7624,1410,1,2,2,12,4
OCTAL  7,0,0,0,34,3,0,6003
OCTAL  0,1,2,30,0,0,0,0
OCTAL  0,0,0,20,4,6001,30,10002
OCTAL  124444,7716,10,326,42,3,326,12667
OCTAL  177506,122667,0,1405,22626,12701,0,167
OCTAL  0,12667,177456,12667,177454,32767,50,4
OCTAL  5017,144016,163253,7011,75460,111504,5401,11017
OCTAL  7016,173,10310,400,7003,21612,143660,5006
OCTAL  13004,124632,52077,52,3,364,0,0
OCTAL  1163,10046,10146,10246,10546,16701,177426,12700
OCTAL  0,12760,0,2,12760,0,14,10160
OCTAL  16,36,4,2002,73643,76400,3004,124674
OCTAL  3553,14001,0,16002,35264,71142,21002,125247
OCTAL  7716,52,3,432,12760,0,20,10046
OCTAL  104377,103540,12702,42,16705,177356,116700,177354
OCTAL  1441,122700,40,1425,122700,60,1416,14
OCTAL  4,3002,35524,75013,11001,42,52,3
```
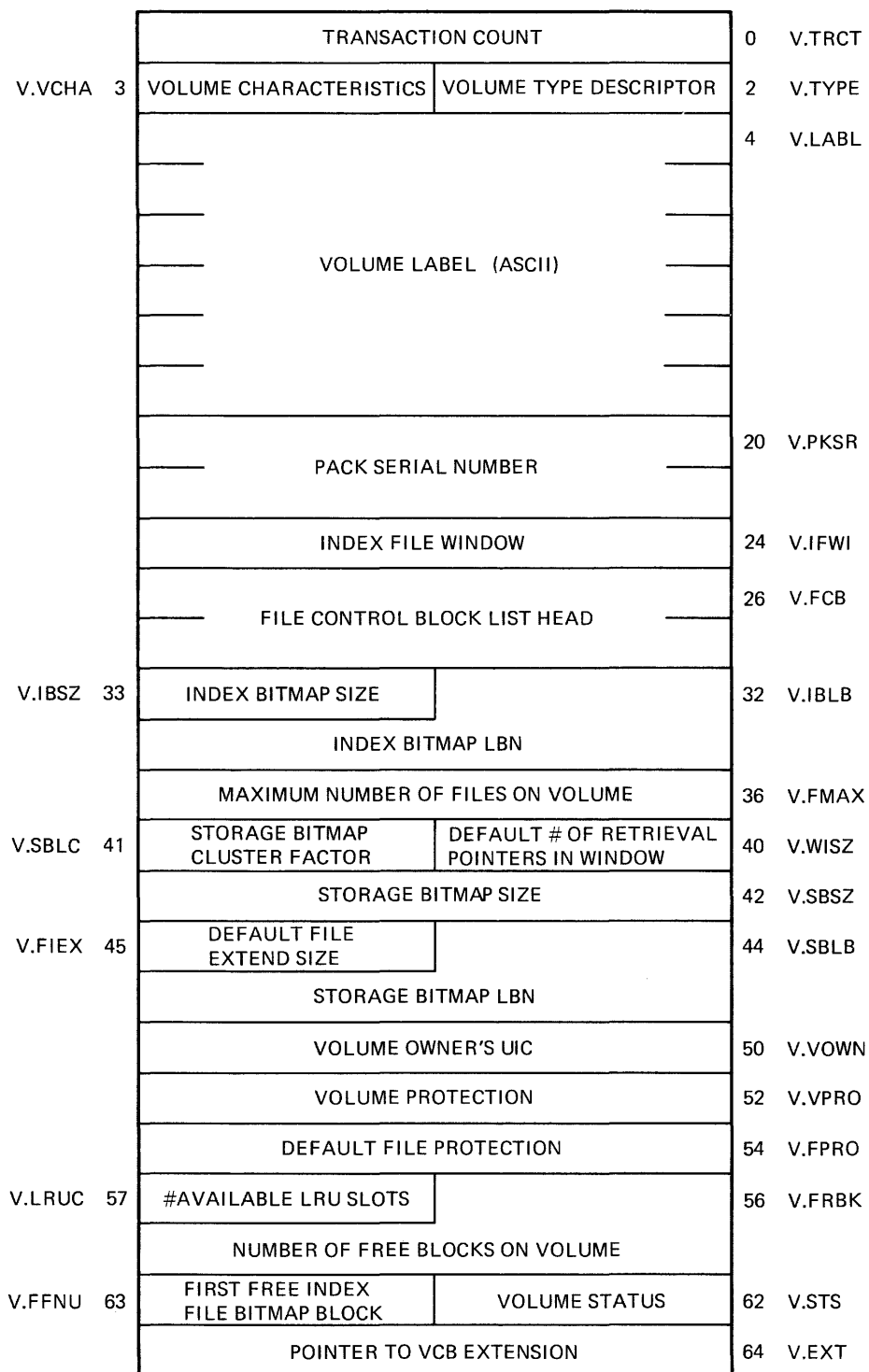
*(handwritten annotation near M.LBN / M.LEN: "DISK ADDRESS TO READ INDEX BLOCK 1")*

| | | |
|---|---|---|
| TRANSACTION COUNT | 0 | V.TRCT |
| V.VCHA 3 · VOLUME CHARACTERISTICS / VOLUME TYPE DESCRIPTOR | 2 | V.TYPE |
| | 4 | V.LABL |
| VOLUME LABEL (ASCII) | | |
| PACK SERIAL NUMBER | 20 | V.PKSR |
| INDEX FILE WINDOW | 24 | V.IFWI |
| FILE CONTROL BLOCK LIST HEAD | 26 | V.FCB |
| V.IBSZ 33 · INDEX BITMAP SIZE | 32 | V.IBLB |
| INDEX BITMAP LBN | | |
| MAXIMUM NUMBER OF FILES ON VOLUME | 36 | V.FMAX |
| V.SBLC 41 · STORAGE BITMAP CLUSTER FACTOR / DEFAULT # OF RETRIEVAL POINTERS IN WINDOW | 40 | V.WISZ |
| STORAGE BITMAP SIZE | 42 | V.SBSZ |
| V.FIEX 45 · DEFAULT FILE EXTEND SIZE | 44 | V.SBLB |
| STORAGE BITMAP LBN | | |
| VOLUME OWNER'S UIC | 50 | V.VOWN |
| VOLUME PROTECTION | 52 | V.VPRO |
| DEFAULT FILE PROTECTION | 54 | V.FPRO |
| V.LRUC 57 · #AVAILABLE LRU SLOTS | 56 | V.FRBK |
| NUMBER OF FREE BLOCKS ON VOLUME | | |
| V.FFNU 63 · FIRST FREE INDEX FILE BITMAP BLOCK / VOLUME STATUS | 62 | V.STS |
| POINTER TO VCB EXTENSION | 64 | V.EXT |

TK-7934

Figure 64   Volume Control Block (VCB)

BIT    MASK

| BIT | MASK | | |
|-----|------|---|---|
| 7 | 200 | RESERVED | |
| 6 | 100 | RESERVED | |
| 5 | 40 | RESERVED | |
| 4 | 20 | RESERVED | |
| 3 | 10 | RESERVED | |
| 2 | 4 | RESERVED | |
| 1 | 2 | VS.BMW — | STORAGE BIT MAP FILE IS WRITE ACCESSED |
| 0 | 1 | VS.IFW — | INDEX FILE IS WRITE ACCESSED |

TK-1847

Figure 65   Volume Status Byte (V.STS in VCB)

VALUE

| VALUE | | |
|-------|---|---|
| 11 | VT.UNL | - UNLABELED TAPE |
| 10 | VT.ANS | - ANSI LABELED TAPE |
| 1 | VT.SL1 | - FILES-11 STRUCTURE LEVEL 1 |

TK-7937

Figure 66   Volume Type Descriptors (V.TYPE in VCB)

BIT    MASK

| BIT | MASK | | |
|-----|------|---|---|
| 7 | 200 | RESERVED | |
| 6 | 100 | RESERVED | |
| 5 | 40 | RESERVED | |
| 4 | 20 | RESERVED | |
| 3 | 10 | VC.PUB | SET = SET DEVICE PUBLIC ON DISMOUNT<br>CLEAR = SET DEVICE NON-PUBLIC ON DISMOUNT |
| 2 | 4 | VC.DEA | - DEALLOCATE VOLUME ON DISMOUNT |
| 1 | 2 | VC.HLK | - UNLOAD VOLUME ON DISMOUNT |
| 0 | 1 | VC.SLK | - CLEAR VOLUME VALID ON DISMOUNT |

TK-7938

Figure 67   Volume Characteristics (V.VCHA in VCB)
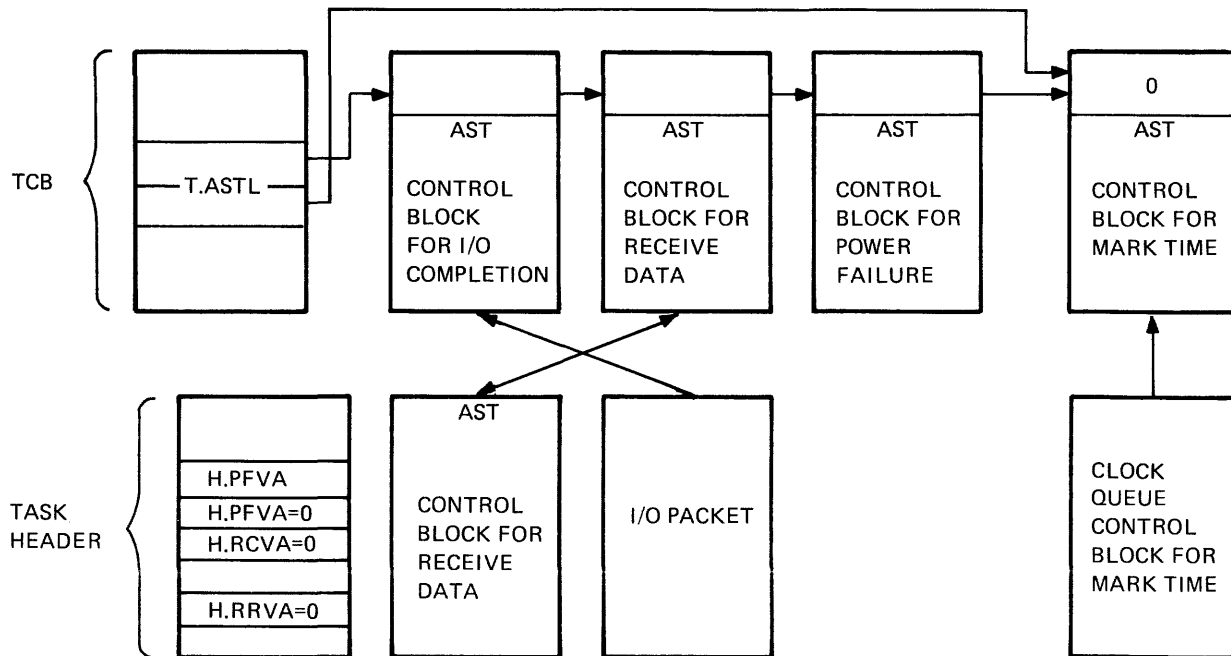
TK-1812

Figure 68   AST Queue

46
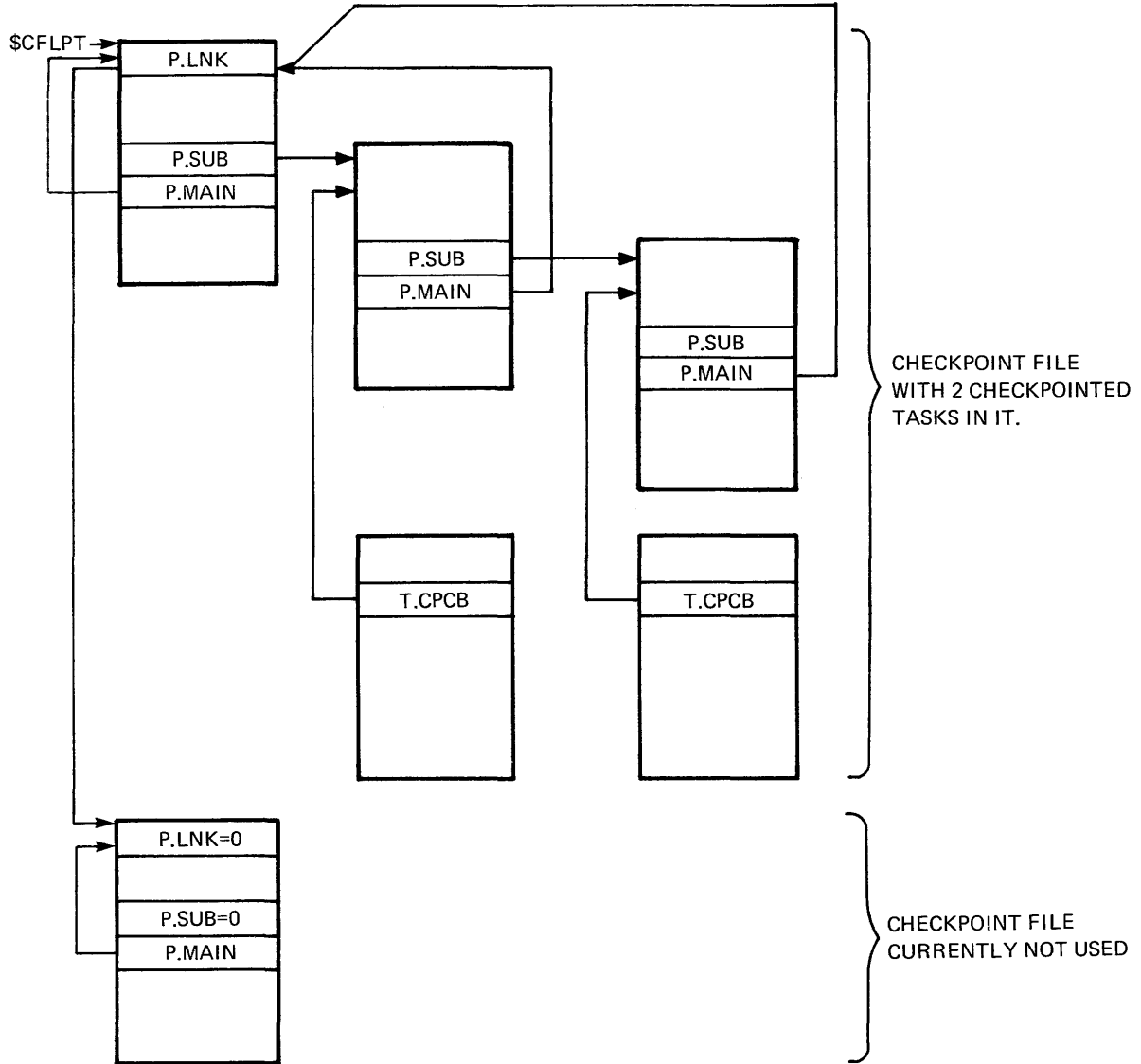
Figure 69   Checkpoint File Data Structure

POINTER TO FIRST FREE BLOCK IN DSR

$CRAVL

POINTER TO NEXT FREE BLOCK

SIZE OF FREE BLOCK (IN BYTES)

POINTER TO NEXT FREE BLOCK

SIZE OF FREE BLOCK (IN BYTES)

POINTER TO NEXT FREE BLOCK=0

SIZE OF FREE BLOCK (IN BYTES)
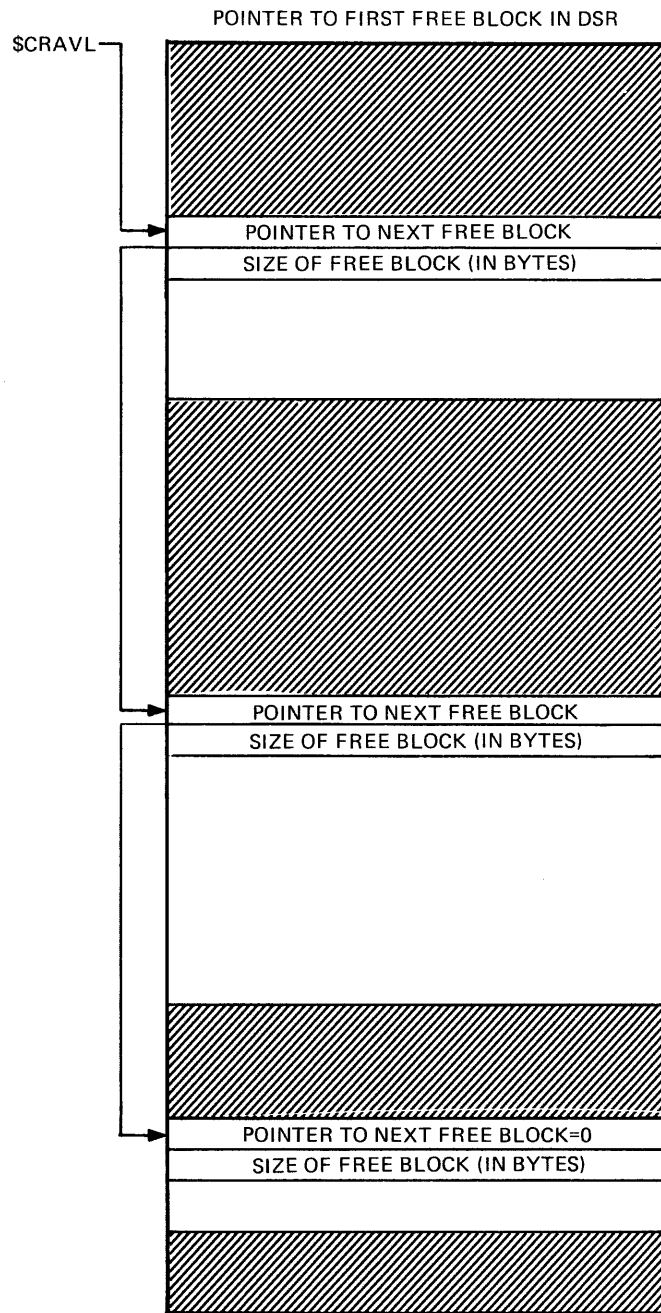
SIZE IN BYTES INCLUDING 2 BYTES FOR POINTER AND
2 BYTES FOR SIZE FIELD

TK-1800

Figure 70   Dynamic Storage Region (DSR)
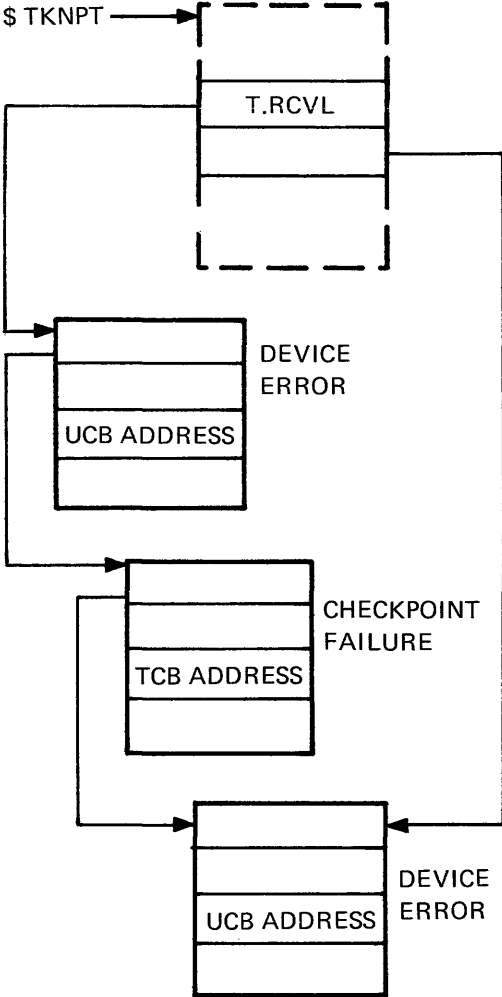
Figure 71   Error Message List
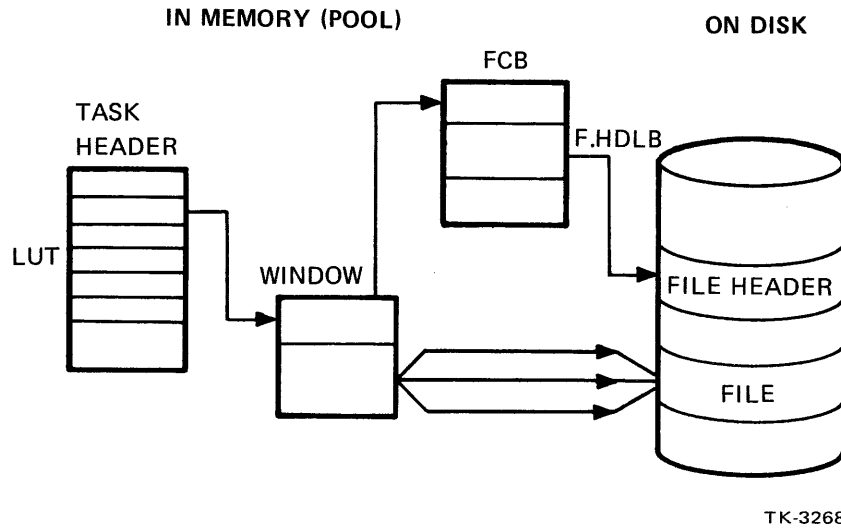
Figure 72   File Open and Close Structures
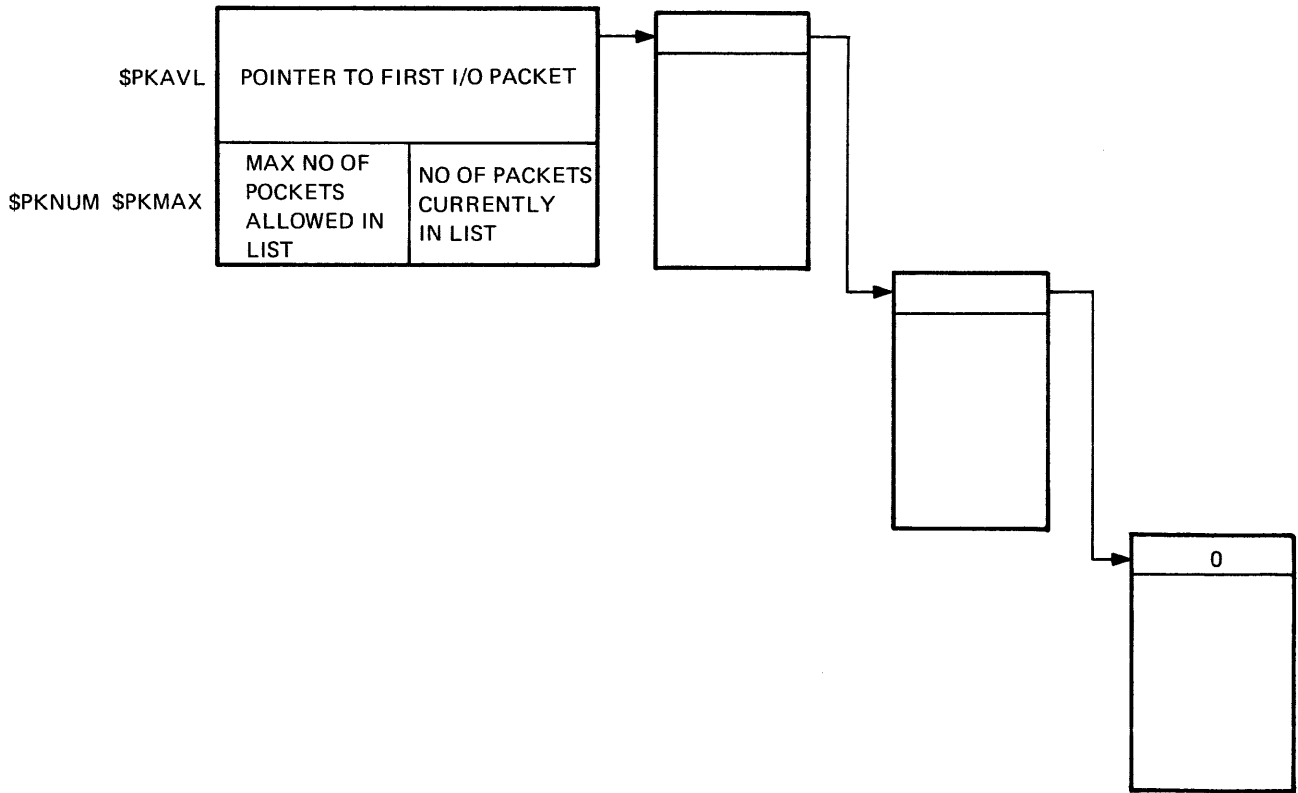


Figure 73   I/O Packets (Preallocated)

TCB'S OF TASKS TO BE LOADED OR ROLLED OUT/IN
ARE CHAINED ACCORDING TO THE RUNNING PRIORITY
OF THE TASKS IN DESCENDING ORDER.
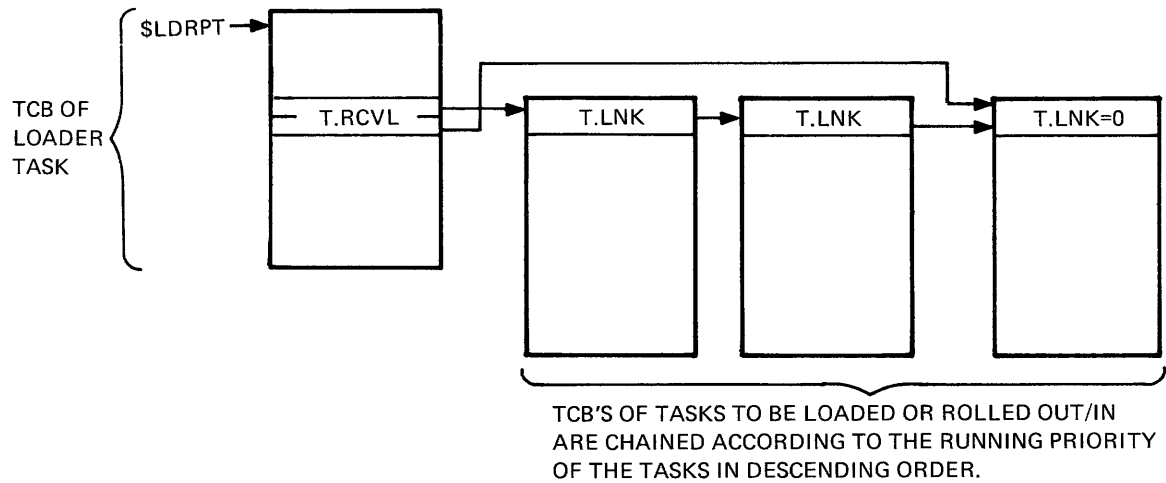
| TS.CKP | TS.OUT | |
|--------|--------|------------------|
| 0 | 0 | INVALID |
| 1 | 1 | INITIAL LOAD |
| 1 | 0 | CHECKPOINT WRITE |
| 1 | 1 | CHECKPOINT READ |

**LOADER QUEUE**

TK-1813

Figure 74   Loader Queue

51
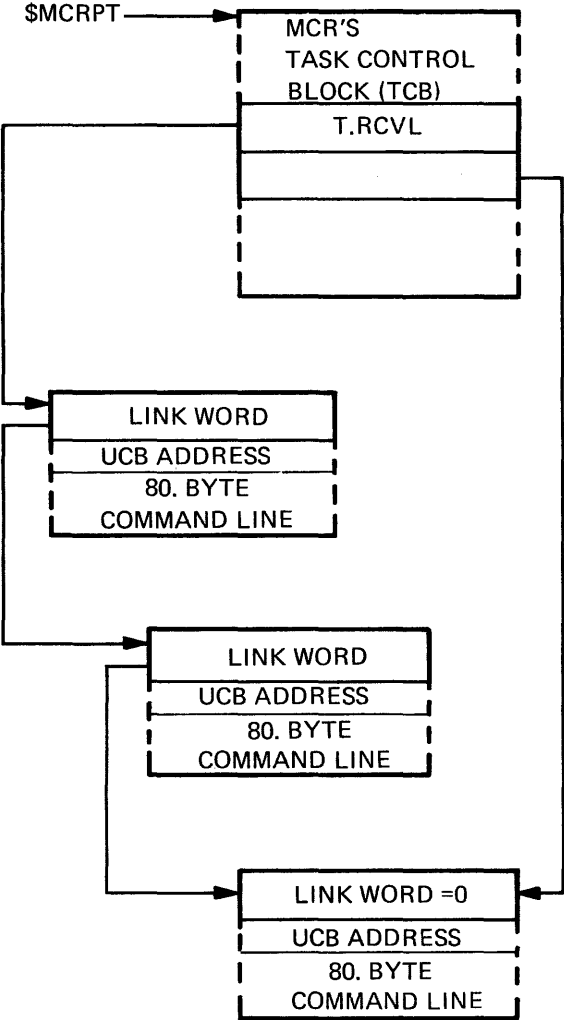
TK-1835

Figure 75   MCR Command Line List
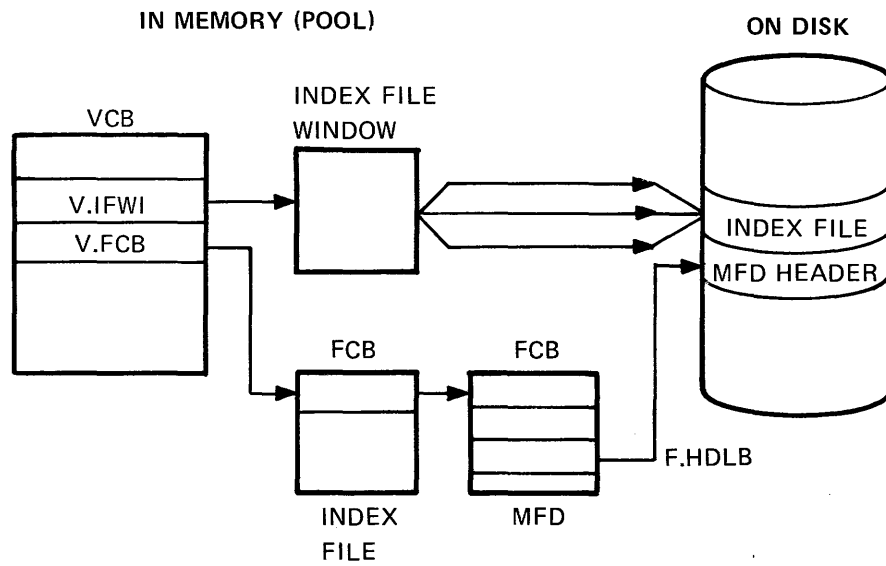
Figure 76   Mounted FILES-11 Volume
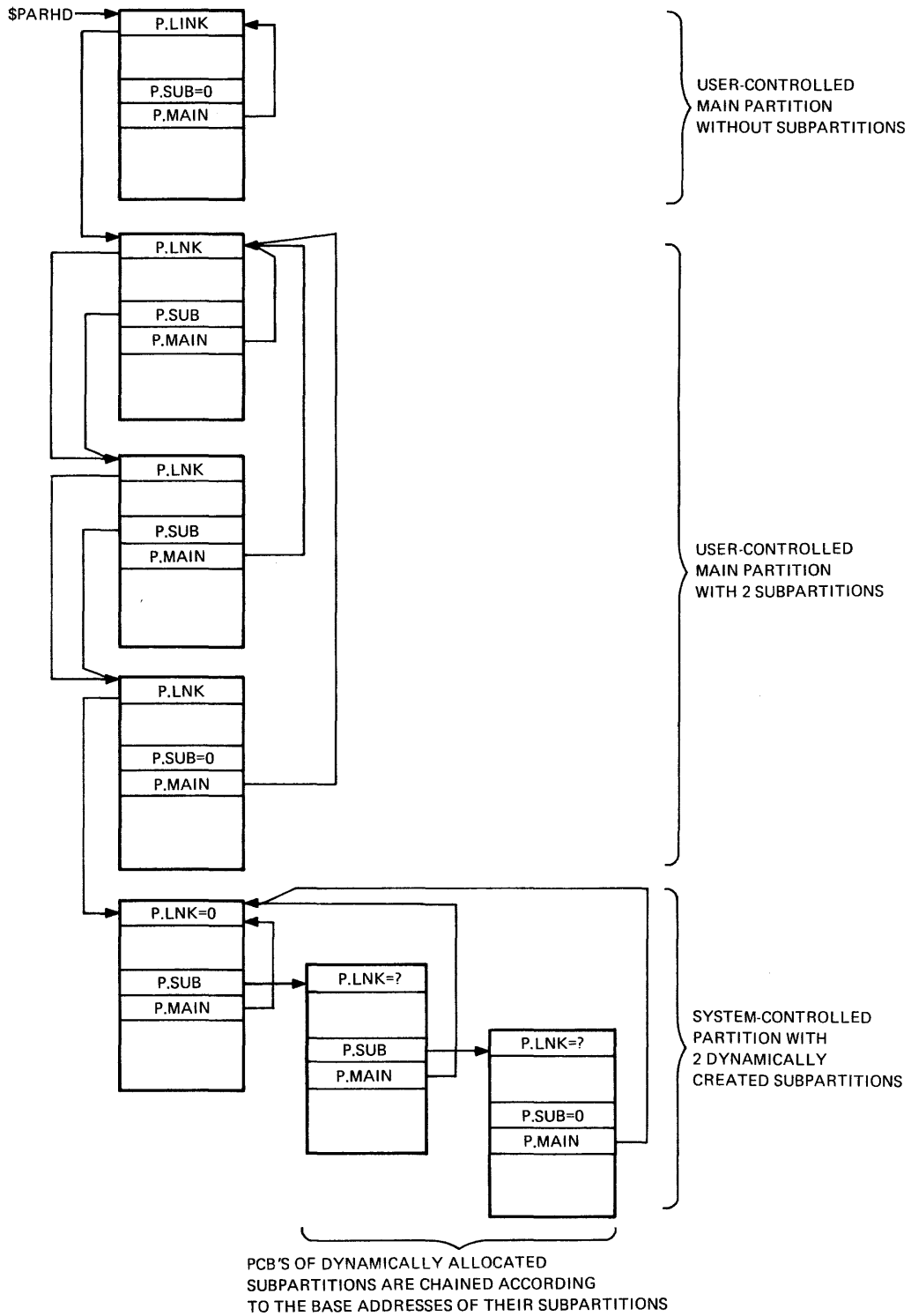
Figure 77    Partition List
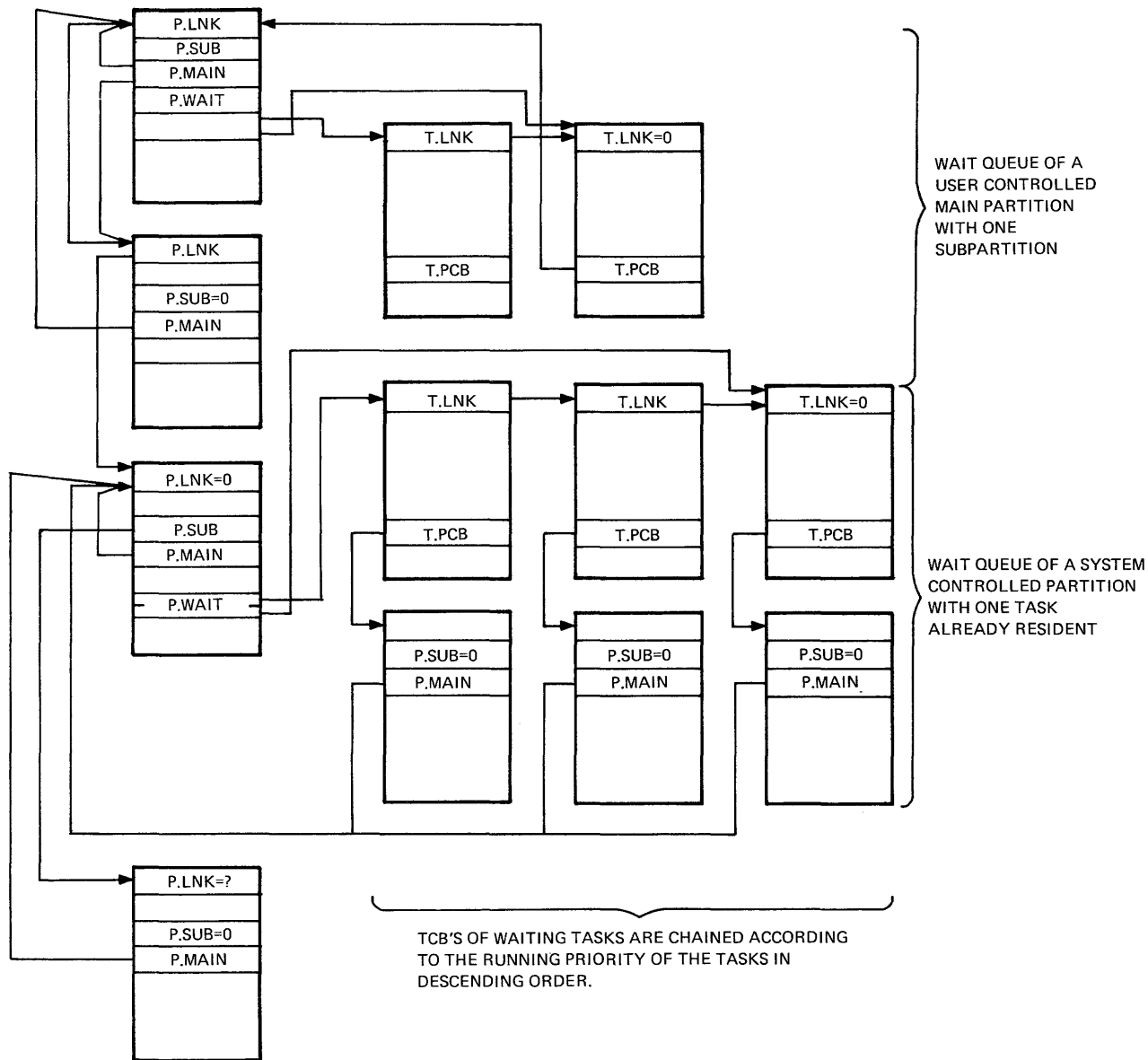
Figure 78   Partition Wait Queue
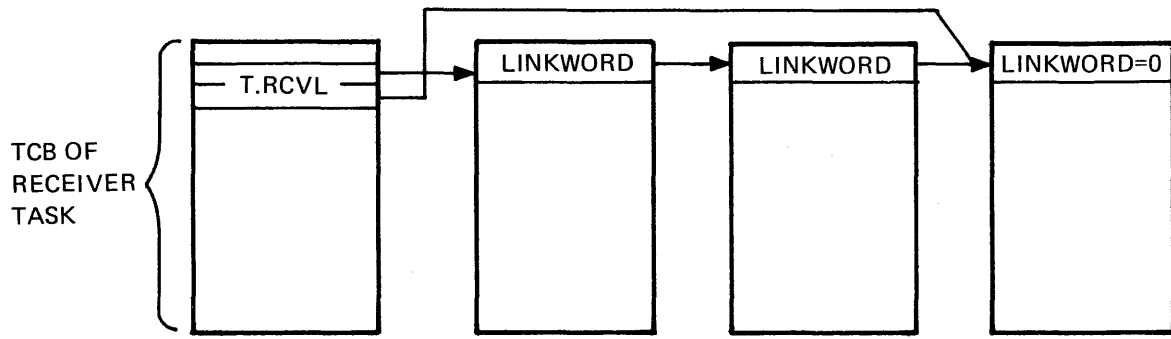
TK-1814

Figure 79   Send/Receive Data Queue
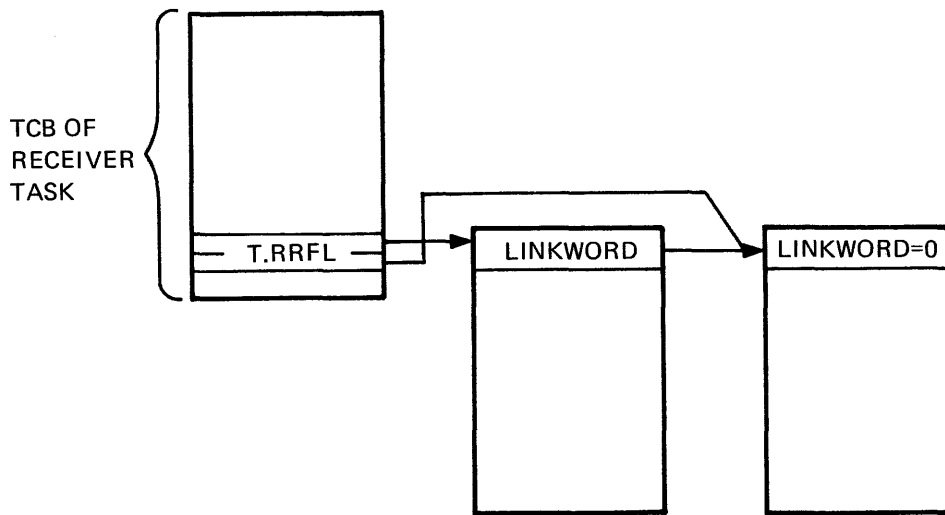


TK-1816

Figure 80   Send/Receive-by-Reference Queue
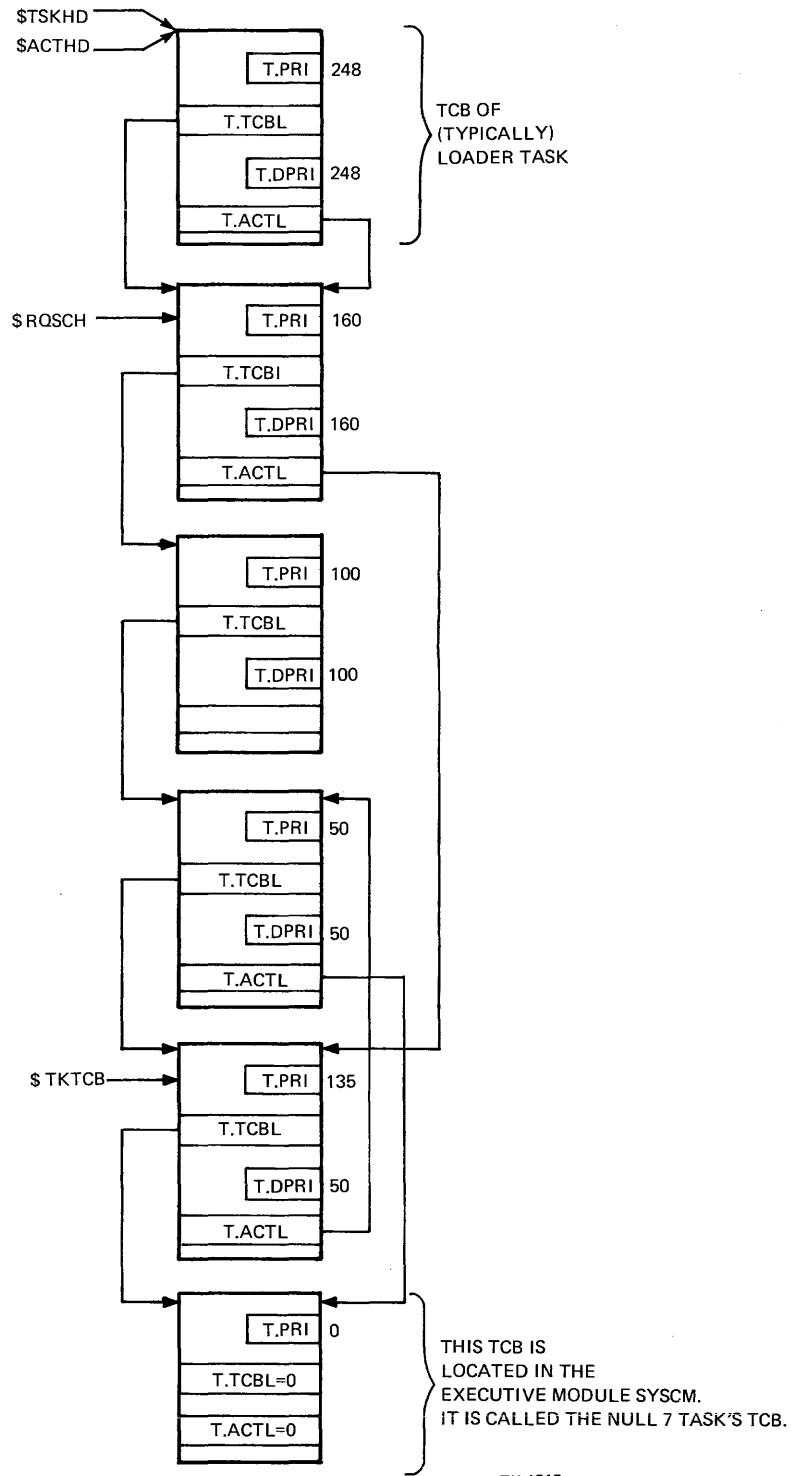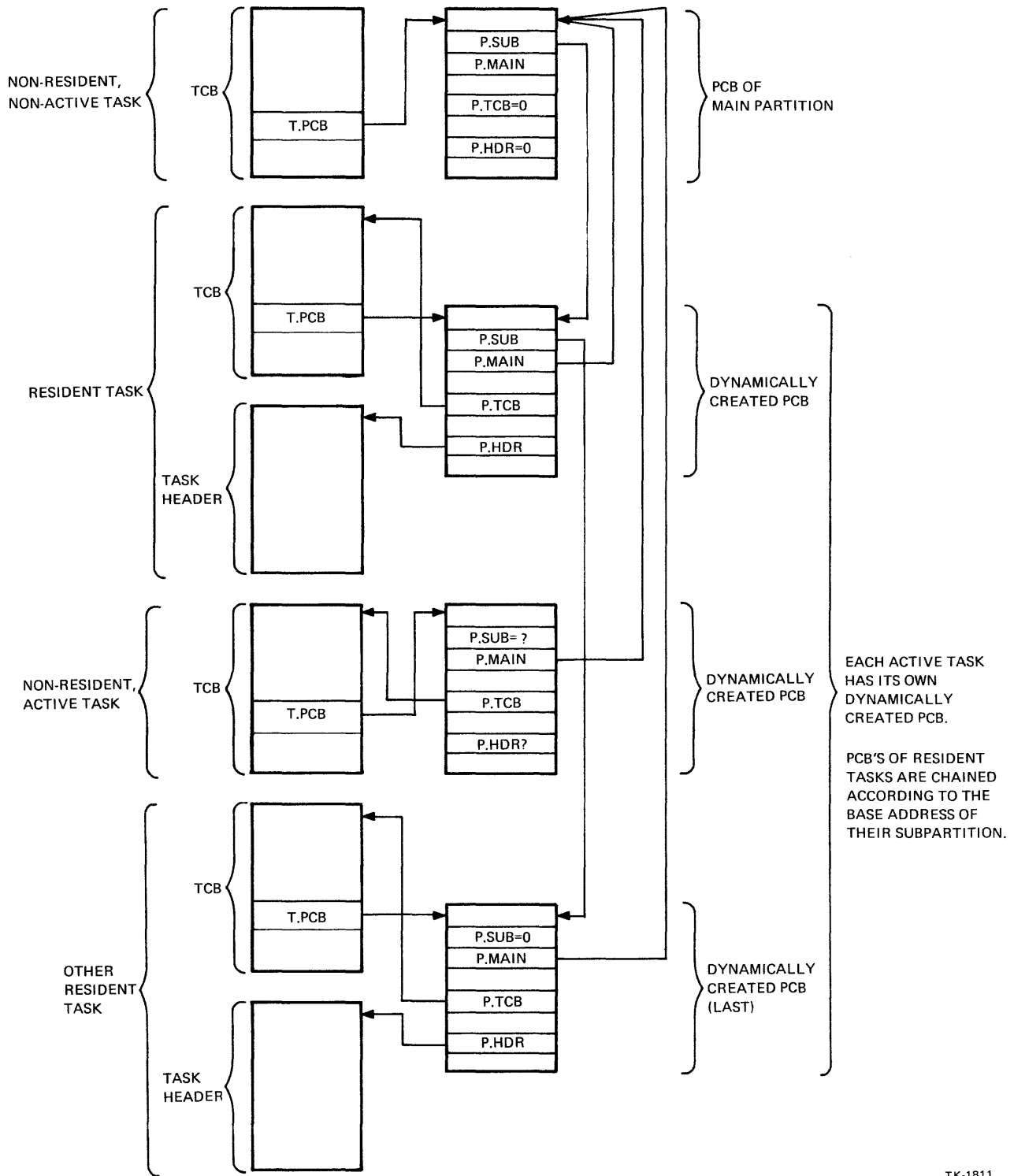
Figure 81    System Task Directory (STD) and
Active Task List (ATL)

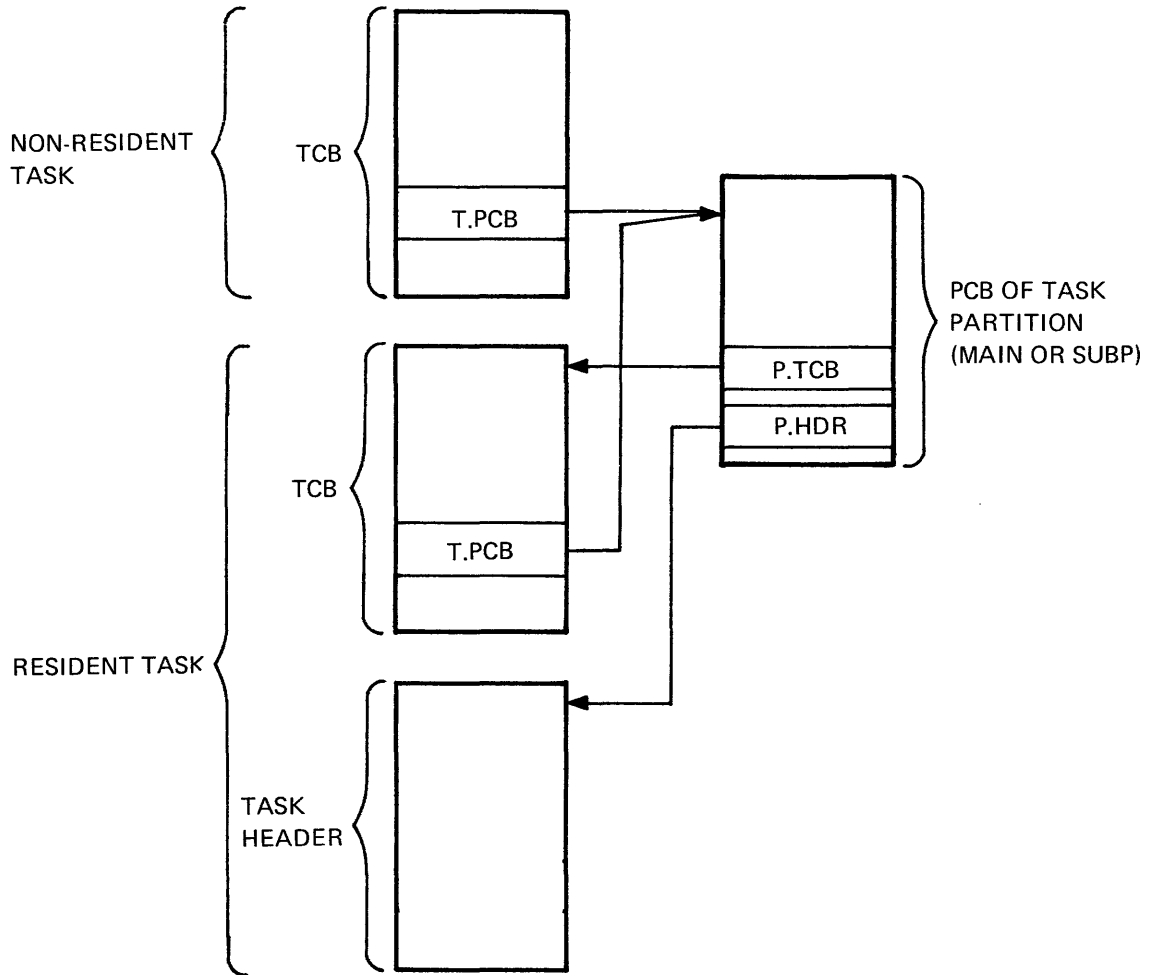Figure 82   TCB, Task Header, and PCB Interrelationship:
System Controlled Partition

TK-1795

Figure 83   TCB, Task Header, and PCB Interrelationship:
User Controlled Partition

Table 1   Macros* Which Define Data Structure Offsets

| Data Structure | Macro |
| --- | --- |
| Asynchronous System Trap Control Block | PKTDF$ |
| Attachment Descriptor Block | PCBDF$ |
| Clock Queue Control Block | CLKDF$ |
| Device Control Block (DCB) | DCBDF$ |
| Error Message Block (EMB) | EMBDF$ |
| File Control Block (FCB) | F11DF$ |
| FILES-11 Window Block | F11DF$ |
| Group Global Event Flags Block | PKTDF$ |
| Interrupt Transfer Block | ITBDF$ |
| I/O Packet | PKTDF$ |
| Logical Assignment Control Block (LCB) | LCBDF$ |
| Locked Block List Node | F11DF$ |
| Mapping Assignment Block | SCBDF$ |
| Mounted Volume List Entry (MVL) | MTADF$ |
| Offspring Control Block (OCB) | PKTDF$ |
| Partition Control Block (PCB) | PCBDF$ |
| Status Control Block (SCB) | SCBDF$ |
| Task Addressing Window Block | HDRDF$ |
| Task Control Block (TCB) | TCBDF$ |
| Task Header | HDRDF$ |
| Unmounted Volume List Node (UVL) | MTADF$ |
| Unit Control Block (UCB) | UCBDF$ |
| Volume Control Block (VCB) | F11DF$ |

*All macros are found in [1,1]EXEMC.MLB.

Digital Equipment Corporation • Bedford, MA 01730