1093

Bell Laboratories

subject: DEAD - A Program for Analyzing UNIX System Crashes

date: April 30, 1976

from: T. M. Raleigh
MF-76-8234-030

MEMORANDUM FOR FILE

Dead is a program for quickly summarizing the state of a crashed UNIX system. It produces symbolic listings of the important system tables so that a system administrator can quickly determine what activity was present on the system when it crashed and hopefully be able to eliminate a number of possible causes for the crash.

Dead was written as a result of the author's experience with intermittent hardware problems at an AT&T installation of the EADAS Network Manager[1], a UNIX based project. It was needed to analyze a large number of system crashes that occurred over several weeks and which defied Digital Equipment Corporation (DEC) hardware diagnosis. It has also turned out to be an excellent method of providing an orderly accounting of system crashes from a software standpoint, so that comparisons could be made with other crashes. The ability to compare a concise summary of a system's activity with previous crashes is particularly important for intermittent hardware problems where it is usually difficult to reproduce the conditions that caused the crash.

The most valuable contribution of dead is its ability to quickly summarize the state of a crashed UNIX system. Previously, the activity on a crashed UNIX system could only be determined after hours of tediously dumping system tables in octal. In addition to producing concise summaries of this information, dead checks many of the linkages between system tables and reports any errors so that a system administrator is now able to rule out the possibility of a software crash with more confidence.

At the present time, many UNIX projects simply reboot the system when a crash occurs without taking a dump of memory because of the amount of time required to extract information from the dump. This is unfortunate for a number of reasons. In the case of intermittent hardware failures, the conditions which produce the failure are not easily duplicated and if the failures do not occur often, they tend to be ignored. For crashes that do require action, much time is wasted in trying to surmize from the user community what

type of activity the users were engaging in when the crash occurred. Since all of the information needed to define the activity that was occurring when the system crashed is contained in a dump, efforts should be concentrated on extracting this information. It is possible that if UNIX users could specify what type of activity the system was engaged in when the crash occurred, a better guess could be made as to where DEC maintenance personnel should concentrate their efforts.

Dead was developed after some initial experience with the Pstat program written by D. M. Ritchie. Pstat produces a dump of important system tables as they appear in the crash file. No attempt is made to link together information in system tables, to check the pointers in the tables or to insure that data in the tables is consistent. Any errors in the tables must be found by laboriously checking all of the tables. An example of a Pstat printout of the Process Table is attached (see Figure 1) for comparison with the printouts produced by dead.

Pstat does eliminate the need to get a system namelist and dump tables individually, and is an important program if system tables have been badly damaged, however, for most system crashes, the systems tables are intact so that extracting information from Pstat requires a good deal of knowledge of the operating system. Dead was designed to minimize the need for this knowledge by producing system table summaries in forms which are easily understandable and familiar to most system programmers. The attached manual pages describe the various options for the dead command and what each field printed by an option represents. The various listings produced by dead are described below.

Registers

The -r option prints the contents of registers R0 to R6 (see Figure 2) and the value of Kernel Instruction Address Register 6-KISA6 (for I space only systems) or Kernel Data Address Register 6 - KDSA6 (for I and D space systems) when the system crashed. The value of KISA6 (or KDSA6) is the address (in memory blocks) of the user process that was the last to run when the system crashed. The -m option will produce a memory map so that the name of the running process can be found by looking this address up. Registers R0-R6 are usually only of interest if it is desired to do a stack trace of the crashed system.

Memory Map

The -m option prints two maps; a memory map and a swap map. These maps show the names and addresses of the processes that were in memory and on the swap device. The -I option (which indicates that the dump is from a UNIX that does not have separation of I and D space - all 11/40's) should be used in combination with the -m option to insure that the address of the text and data portions of the operating system are printed in the right order in the memory map. The -m option prints the location and size of each area followed by the name of the process (if it was in memory). Any free space in the system is marked as "Free". The -m option is essentially a merge of the Memory Map, Swap Map, Process Table and Text Table of the system and if no error diagnostics are printed when the -m option is used, the values in these three tables all produce a consistent layout of the processes in the system. If any areas are unaccounted for, or overlap any adjacent areas, dead complains (see Figure 3).

Process Table

The -p option prints a summary of all of the processes that were created in the system (see Figure 4). Processes that are in memory have their names printed (if possible) and each process is marked as being resident in memory ('M') or resident on the swap area ('D'). The priority of each process is also printed, however, the most important value printed is an interpretation of the event that a process is roadblocked on. This is a symbolic representation of the "p_wchan" entry in the Process Table and is constructed from the system namelist. To understand its significance one must be roughly familiar with the way UNIX timeshares processes [2], and know which events are commonly used to synchronize the roadblocking and unblocking of a process. The most common events that will be printed are,

tout - process sleeping (old sleep), e.g. the update process

u - process has paused waiting for the alarm clock timer.

dh[] - process waiting for teletype input, output or carrier

proc[] - process waiting for child

buf[] - process waiting for I/O

In the attached example (see [1] for more details), the author isolated a DQ11 synchronous line interface problem by noting that in every case, the system crashed when a

transmit had been initiated on the DQ (dgtxng event - see example of Figure 4) even though each crash showed a different process as the currently executing process.

I/O Buffers

The -b option searches through the buffer headers in the I/O subsystem and attempts to figure out which buffers are on each queue (see Figure 5). Care should be taken in interpreting the results printed by the -b option, since the crash may have occurred when the linkages in the buffer pool were incomplete.

For each queue, dead prints the buffers in the order that the linkages associated with a queue indicates. In addition, the -b option insures that all of the systems buffers are printed. If any buffer does not appear on at least one queue, it is printed separately on an "Unaccounted For" list.

There are five queues that are normally printed. The first two queues are system queues while the three remaining queues can be associated with each device and are therefore repeated for each device. Any queues that are empty are not printed.

1. The Free queue - This is a queue of buffers that are available to be allocated for I/O or any other purpose the system needs. The queue may contain delayed write buffers which are not available for allocation as they must have their contents written onto the device before they can be allocated.

2. Null device queue - This is a special queue for buffers that the system uses to hold data. Superblocks for each mounted file system are kept in buffers that are on this queue as are buffers being used for passing arguments between programs in an exec operation.

3. Device queue - This is a queue of buffers that contain data that has been retrieved (at some time) from or written onto the device. Buffers on this queue may also appear on the Free queue or the I/O queue.

4. Device I/O queue - This queue consists of all of those buffers that are to be read or written from a device. This queue should not normally contain many buffers. Dead prints the word "ACTIVE" if the device queue indicates that the device is active (i.e., has initiated an I/O operation). Any device that was active when the system crashed should be looked upon with suspicion. The example of Figure 5 shows that the RP03 disk has initiated I/O.

5. Raw I/O queue - This is a queue containing the buffer header used for doing physical I/O to a device. As with the other buffers, there is, unfortunately, no way of telling how recently the I/O occurred. The swap device queue is a raw I/O queue.

One use of the -b option is to quickly detect deadly embrace situations in the buffer pool. For example, on TU16 magnetic tape drives, if the tape runs off the reel doing a write operation, all of the buffers in the system will be found on the TU16 device I/O queue. (This is a DEC hardware design error in the TU16.) UNIX assumes that the device will interrupt if any error occurs and hence allows buffers to be continually added to the device I/O queue.

The pointers in each buffer header are printed in symbolic form so that they can be checked if desired, however, this requires some knowledge of the Operating System internals.

Configuration Table

The -c option (see Figure 6) produces the configuration table. There are still some systems which do not use the USG standard configuration table format. The EADAS project is one of these as the attached listing shows. The configuration table and its relation to major device numbers is a relatively critical one in any system debugging as is the location of the swap and root device, so that the ability to retrieve this table from the crash dump file and check it against the special files on the system is important.

Inode Table

The -i, -f, -M and -P options check the file systems on the crash system and hence take a long time to execute. They should only be used if the crash dump being examined is on the system from which the crash dump was taken. (It is possible to make a minor change to the dead software so that a copy of another system can be mounted and analyzed as if it were the running system. For more information, consult the comments in the software.) The long execution time occurs because dead runs the ncheck program several times to find the pathnames of the i-nodes. If an i-node name cannot be found, the message "i-number not found" is printed. This can occur if the i-node was for a temporary file or a file that was removed during a file system salvage operation.

For the -i option (see Figure 7), i-node pathnames are printed in an unambiguous form with the name of the file system on which the i-node resided preceding the actual pathname.

File Table

The File Table contains the read write pointers that a process uses to access an i-node and the access that the process is allowed to the file (read, write or read and write). The -f option prints the File Table (see Figure 8) in a manner similar to that of the Inode Table. Pipes are unnamed files and therefore have no pathname so that they will be marked as "i-number not found" in the listing, however, a 'P' will be printed in the flags field to indicate that it is a pipe.

I/O Streams

The -P option prints a list of all of the processes that are in memory followed by a list of the files that each process had open when the system crashed (see Figure 9). It is a composite of the Process Table, Inode Table, File Table and U block. It lists each process by process id and name followed by a list of the file descriptors used by that process and the file that they had open. The File Table entry which each file descriptor references is also printed.

Mount Table

The -M option lists the names of the mounted file systems and the i-node on which they are mounted (see Figure 10).

Conclusion

Dead significantly reduces the effort needed to analyze a system crash so that it is now possible for projects to keep detailed records of their system crashes and hopefully will provide local system administrators with an easy to use tool to help find hardware and software problems. Determining the cause of a crash on a UNIX system is still somewhat of a black art, however it is hoped that software tools like dead and tcmp, (see [3]) a first step can be taken to improve our diagnostic abilities on UNIX systems.

T M Raleigh

MH-8234-TMR-dh

T. M. Raleigh

Attachments

Copy to
G.L. Baldwin
Department 8234
D. Cochran
J. Carran
N. Chaffee
D. DeJager

M. Pilla
P. Kennedy
J. Malik
A. Gross
D. Jackowski
R. Farbanish
S. Arnold
D.M. Ritchie
E.A. Walker

REFERENCES

[1] Raleigh, T. M., "Minicomputer Hardware Problems at AT&T, Wayne, Pa.", to be published.

[2] Raleigh, T. M., "Introduction to Scheduling and Switching Under UNIX", TM75-8234-7, October 17, 1975.

[3] Raleigh, T. M., "TCMP - A Program for Comparing Program Text After a System Crash", MF76-8234-029, April 27, 1976.

DEAD (VIII)

DEAD (VIII)

NAME

dead — crash analysis

SYNOPSIS

dead [-sImpcrPMif] crashfile [file]

DESCRIPTION

Dead produces formatted summaries of system tables from a crash dump. Rather than just print the contents of the system's tables, it attempts to put the information into a meaningful form while checking the consistency of pointers in the tables.

Dead can produce a memory map, a swap map, a list of the i-nodes that were open in the system (with fully qualified pathnames, including device name) and a list of the file descriptors that each process has open with the pathname of the corresponding file. In addition, it will analyze the block I/O buffering system to determine which buffers are on each device queue, which buffers are queued for I/O and which block devices are active. It can also retrieve the configuration table of the system.

All addresses in the system tables are printed in symbolic terms and the options can be used individually or in combination.

Dead assumes that the namelist for the crash dump is in "/unix" unless the s option is used, in which case a *file* name may be specified. There are several options:

I indicates that the crash file is from an I space only system. This has no effect on any of the results except that it insures that the first two entries of the memory map (see m option below) are printed in correct order.

c retrieves the configuration tables (conf.c) for character and block devices from the crash file. All entries in the tables are printed in symbolic form so that the printout closely resembles the contents of conf.c. The contents of the locations which specify the root and swap devices are also printed.

r prints the registers that were saved in low core by the crash dump routine when the dump was taken. The contents of the K register (KISA6 or KDSA6) contains the address (in memory blocks) of the last process that ran.

m prints a memory map and swap map with process names. Since reentrant text is managed separately from the nonreentrant part of a process, there can be two entries for a reentrant process. Reentrant text has a T appended to the address field. The following fields are printed,

Heading	*Description*
ADDR	The address of an area in memory blocks (64 bytes).
SIZE	The size of the area in memory blocks.
PID	The process id.
NAME	The name of the process occupying the area, or if the memory space is not allocated, "Free" is printed.

The swap map is also printed with the m option in a similar manner, however, the address and size fields are in disk blocks (512 bytes).

p prints a summary of all of the processes in the crash file, their names and attempts to construct a symbolic name for the event on which a process is roadblocked.

Heading	*Description*
NO	The number (index) of the process table entry.
S	A letter encoding the state of the process:

S Sleeping
W Waiting
R Ready
I Idling
Z Zombie - process exited, parent not yet notified.

F The location of the process. It may be any of the following:

D Swapped out
M In memory
L Locked in Memory

PID The unique process number.

PRI The priority of the process; high numbers mean low priority

UID Four characters of the user's id from the password file.

EVENT A symbolic representation of the address that the process is roadblocked on (if any).

NAME The file name of the process.

M prints the names of the mounted file systems and the pathnames of the i-nodes that they are mounted on.

I prints the contents of the *inode* table including an unambiguous pathname. The following fields are printed.

Heading	*Description*
#	The number of the i-node table entry in decimal.
FLAGS	Any special flags in the i-node are printed as follows, D a directory C a character device B a block device L a large file
ACCESS	The access control permissions for the file are printed in a manner similar to the *ls* command. Permissions are printed in the same order as for *ls*, i.e., owner, group, others. s set user-ID bit on g set group-ID bit on r read permission w write permission x execute permision
INO	The number of the i-node.
UID	The name of the owner of the file, taken from the password file.
DEVICE	The pathname of the device on which the i-node resides.
PATHNAME	The pathname of the i-node if it can be found. Temporary files and pipes disappear when a system is rebooted, so some file names can not be found. Also, an i-node may be reallocated if a file is removed, so *dea* should be run relatively soon after a system is rebooted to insure that the pathnames are correct.

f prints the *file* table.

Heading	*Description*
#	The number of the file table entry.

MD	The *mode* used to open the file.
C	The number of instances of the file being open with the given *mode.*
	R read permission W write permission P a pipe
DEVICE	The pathname of the device on which the file resides.
PATHNAME	The pathname of the i-node which the *file* table references.

P prints a list of all of the processes in memory and the files that they have open. The id of the process is printed with its name, followed by a list of file descriptors that the process has open and the pathnames of the files that they reference.

Heading	*Description*
#	The file descriptor number.
F	The *mode* with which the *file* was opened (same as MD field for the f option).
IND	The number of the *file* table entry which this descriptor references.
DEVICE	The pathname of the device on which the i-node resides.
PATHNAME	The pathname of the i-node.

b prints a summary of the buffers in the I/O subsystem and the queues on which they reside. There are five classes of queues. The *Buffer Free List* is a queue containing all of the buffers that are available for allocation. The *Null Device Queue* is a queue of buffers that are allocated by the system for some special purpose and are not associated with any device (e.g., holding superblocks, holding arguments for an *exec* system call, I/O for special devices, etc.). Each device analyzed by the **b** option has at most three queues but only those queues that are nonempty are printed. The *Device Queue* links together all of the buffers that contain data that have been retrieved or written onto the given device. Buffers on this queue may also appear on the *Buffer Free List* The *Device I/O Queue* contains all of the buffers that are actually waiting to be read or written from the given device. If a block device is active when a system crashes, the I/O queue is marked as *ACTIVE.* If any physical I/O has been done on the device, the contents of the *Raw Queue* are also printed, however, there is no way of knowing how recently the physical I/O occurred. A symbolic representation of the pointers associated with each queue and each buffer on a queue is printed, however, familiarity with the I/O system is required to be able to check them and space does not permit such an explanation here. The values printed are,

Heading	*Description*
BUF	The buffer number.
FLAGS	Any of the following
	B Buffer busy D I/O complete (done) W Write R Read L Delayed write N Wanted by other processes E Error 0-3 Memory extension bits
[MAJ,MIN]	Major and minor device numbers
BLOCK	Logical block number

DEAD (VIII)

DEAD (VIII)

FILES

/unix	system namelist
/etc/passwd	password file
/usr/bin/ncheck	

SEE ALSO

tcmp(VIII), ncheck(VIII), pstat(VIII)

BUGS

For the P, M, i and f options, *dead* runs *ncheck* to find the fully qualified pathname. This takes a bit of time.

```
%pstat -p sysdump
25 processes
   LOC S    FLAG  PRI SG UID TIM   PID  PPID  ADDR SIZE  WCHAN  TEXTP  CLK
  6734 S MS       -50  0   0 127     0     0  1503   20  20276      0 0
  6762 W M         40  0   0 127     1     0  4056  110   6762      0 0
  7010 W D         40  0   0 127   266     1  1012  510   7010      0 0
  7036 W D         10  0   0 127  2720     1    47   55  24244      0 0
  7064 W M         90  0   0 127     4     1  2515   45   6464      0 0
  7112 W D         10  0   0 127  2863     1    55   55  24304      0 0
  7140 W D         10  0   0 127  2859     1    63   55  24444      0 0
  7166 W D         10  0   0 127  2866     1    71   55  24344      0 0
  7214 R M        -50  0  36   6  5301   308  4304  151      0      0 0
  7242 W D         10  0   0 127    16     1   331  110  24636      0 0
  7270 W D         10  0   0 127  2869     1    77   55  24404      0 0
  7344 W D         10  0   0 127  2502   283   105  171  22244      0 0
  7420 R M L      -50  0   0 127   271   266  4461  510      0      0 0
  7446 W D         10  0   0 127   272   266   140  457  24576      0 0
  7474 W M         40  0  36 127   308     1  4166   76   7474      0 0
  7522 W D         10  0   0 127   274   266   342  475  53234      0 0
  7550 W M L       10  0   0 127   275   266  6622  356  54200      0 0
  7576 W D         50  0   0 127   276   274   624  215  54176      0 0
  7624 W M          1  0   0 127   277   266  5171 1431  22215      0 0
  7652 R M         90 14   0 127   278   266  7200  266      0      0 0
  7700 W M          1  0   0 127   279   266  1523  772  22216      0 0
  7726 W M          1  0   0 127   280   266  2601  434  22165      0 0
  7754 W M          1  0   0 127   282   266  3235  433  22234      0 0
 10002 W D          1  0   0 127   283   266   646  174  22220      0 0
 10030 W M         10  0   0 127   285   283  3670  166  24536      0 0
```

FIGURE 1

```
%dead -r sysdump

Time of day Thu Feb  5 03:40:02 1976

REGISTERS:

R0      1342
R1      33224
R2      7214
R3      177716
R4      62
R5      177774
R6      4
KDSA6   4304
K register contains address of last running process.
```

FIGURE 2

```
%dead -m sysdump

Time of day Thu Feb  5 03:40:02 1976


                Memory Map
maxmem 2000
    ADDR     RANGE      PID      NAME
       0      0561               Unix Data
     0561     0722               Unix Text
    01503      020         0     Scheduler
    01523     0772       279     /rp4/exsys/tgmon
    02515      045         4     /etc/updat
    02562      017               Free
    02601     0434       280     /rp4/exsys/dscrmon
    03235     0433       282     /rp4/exsys/prmon
    03670     0166       285     /rp4/crt/ctrl/prets
    04056     0110         1     /etc/init
    04166      076       308     testexec
    04264      020               Free
    04304     0151      5301     pgs
    04455       04               Free
    04461     0510       271     /rp4/exsys/dpp
    05171    01431       277     /rp4/exsys/xsys
    06622     0356       275     /rp4/exsys/xpoll
    07200     0266       278     /rp4/exsys/xpb
    07466     0112               Free

                Swap Map
number of swap blocks 3554

    ADDR     RANGE      PID      NAME
     024       023               Free
     047        06      2720
     055        06      2863
     063        06      2859
     071        06      2866
     077        06      2869
    0105       020      2502
    0125       013               Free
    0140       046       272
    0206      0123               Free
    0331       011        16
    0342       050       274
    0412      0212               Free
    0624       022       276
    0646       020       283
    0666      0124               Free
   01012       051       266
   01063     02515               Free
```

FIGURE 3

```
%dead -p sysdump

Time of day Thu Feb  5 03:40:02 1976

PROCESS TABLE:

25 processes
 NO   S F    PRI    PID   UID       EVENT            NAME
  0   S M    -50     00   root       buf[ 15]        Scheduler
  1   W M     40     10   root      proc[  1]        /etc/init
  2   W D     40   2660   root      proc[  2]
  3   W D     10  27200   root      dh11[ 32]
  4   W M     90     40   root      tout             /etc/updat
  5   W D     10  28630   root      dh11[ 33]
  6   W D     10  28590   root      dh11[ 36]
  7   W D     10  28660   root      dh11[ 34]
  8   R M    -50 530136    hgf                        pgs
  9   W D     10    160   root      dh11[ 39]+032
 10   W D     10  28690   root      dh11[ 35]
 12   W D     10  25020   root      dh11
 14   R ML   -50   2710   root                        /rp4/exsys/dpp
 15   W D     10   2720   root      dh11[ 38]+032
 16   W M     40  30836    hgf       proc[ 16]        testexec
 17   W D     10   2740   root      kl11
 18   W ML    10   2750   root   dqtxng              /rp4/exsys/xpoll
 19   W D     50   2760   root   dqrslp+06
 20   W M      1   2770   root      semi+033         /rp4/exsys/xsys
 21   R M     90   2780   root                       /rp4/exsys/xpb
 22   W M      1   2790   root      semi+034         /rp4/exsys/tgmon
 23   W M      1   2800   root      semi+03          /rp4/exsys/dscrmon
 24   W M      1   2820   root      semi+052         /rp4/exsys/prmon
 25   W D      1   2830   root      semi+036
 26   W M     10   2850   root      dh11[ 37]+032    /rp4/crt/ctrl/prets
```

FIGURE 4

```
%dead -b sysdump

Time of day Thu Feb  5 03:40:02 1976

SYSTEM BUFFERS:

20 buffers
BUF FLAGS  [MAJ,MIN] COUNT    BLOCK    B_FORW      B_BACK      AV_FORW     AV_BAC

Buffer Free List:
queue pointers                                  buf[ 14]    buf[  1]    buf[ 11]    buf[

11 RD   0   [ 0, 4]   -256       616   buf[ 16]    buf[  7]    buf[  7] bfree
 7 RD   0   [ 0, 4]   -256       175   buf[ 11]    buf[  8]    buf[  4]    buf[ 1
 4 WD   0   [ 0, 4]   -256     50227 rptab         buf[ 12]    buf[  8]    buf[
 8 RD   0   [ 0, 4]   -256     54054   buf[  7]    buf[ 18]    buf[ 12]    buf[
12 RD   0   [ 0, 0]   -256      7336   buf[  4]    buf[ 10]    buf[ 10]    buf[
10 RD   0   [ 0, 0]   -256     12327   buf[ 12]    buf[ 16]    buf[ 16]    buf[ 1
16 RD   0   [ 0, 0]   -256     11506   buf[ 10]    buf[ 11]    buf[ 19]    buf[ 1
19 RD   0   [ 0, 4]   -256       156   buf[ 18]    buf         buf[ 18]    buf[ 1
18 RD   0   [ 0, 0]   -256         5   buf[  8]    buf[ 19]    buf         buf[ 1
 0 RD   0   [ 0, 0]   -256     13463   buf[ 19]    buf[  6]    buf[  6]    buf[ 1
 6 RD   0   [ 0, 4]   -256      6761   buf         buf[  9]    buf[  9]    buf
 9 RD   0   [ 0, 4]   -256     13302   buf[  6]    buf[  2]    buf[  2]    buf[
 2 RD   0   [ 0, 0]   -256       102   buf[  9]    buf[ 17]    buf[  3]    buf[
 3 RD   0   [ 0, 4]   -256     35207   buf[  5]    buf[ 15]    buf[  5]    buf[
 5 RD   0   [ 0, 4]   -256     44376   buf[ 17]    buf[  3] bfree          buf[

null device queue:
queue pointers                                  buf[ 14]    buf[  1]    buf[ 11]    buf[

14 W B  0   [-1,-1]   -256         4   buf[ 13] bfree          buf[ 15] bfree
13 W B  0   [-1,-1]   -256         1   buf[  1]    buf[ 14]    buf[ 19] bfree
 1 W B  0   [-1,-1]      0      1503 bfree         buf[ 13]    buf[  2] bfree


Raw tm Queue:
   W    0   [ 0, 0]      0         0        01          01          01          01
memory address bits 15-18   0 bits 0-14 0

rp queue:
queue pointers                                  buf[ 15]    buf[  4]    buf[ 15]      01

 5 W B  0   [ 0, 4]   -256         1   buf[  3] rptab           01          02
 3 RD   0   [ 0, 4]   -256     35207   buf[  5]    buf[ 15]    buf[  5]    buf[  2
 5 RD   0   [ 0, 4]   -256     44376   buf[ 17]    buf[  3] bfree          buf[  3
17 RDB  0   [ 0, 0]   -256       100   buf[  2]    buf[  5]    buf[  3]    0346
 2 RD   0   [ 0, 0]   -256       102   buf[  9]    buf[ 17]    buf[  3]    buf[  9
 9 RD   0   [ 0, 4]   -256     13302   buf[  6]    buf[  2]    buf[  2]    buf[  6
 6 RD   0   [ 0, 4]   -256      6761   buf         buf[  9]    buf[  9]    buf
 0 RD   0   [ 0, 0]   -256     13463   buf[ 19]    buf[  6]    buf[  6]    buf[ 18
 ) RD   0   [ 0, 4]   -256       156   buf[ 18]    buf         buf[ 18]    buf[ 16
18 RD   0   [ 0, 0]   -256         5   buf[  8]    buf[ 19]    buf         buf[ 19
 8 RD   0   [ 0, 4]   -256     54054   buf[  7]    buf[ 18]    buf[ 12]    buf[  4
 7 RD   0   [ 0, 4]   -256       175   buf[ 11]    buf[  8]    buf[  4]    buf[ 11
11 RD   0   [ 0, 4]   -256       616   buf[ 16]    buf[  7]    buf[  7] bfree
16 RD   0   [ 0, 0]   -256     11506   buf[ 10]    buf[ 11]    buf[ 19]    buf[ 10
10 RD   0   [ 0, 0]   -256     12327   buf[ 12]    buf[ 16]    buf[ 16]    buf[ 12
12 RD   0   [ 0, 0]   -256      7336   buf[  4]    buf[ 10]    buf[ 10]    buf[  8
 4 WD   0   [ 0, 4]   -256     50227 rptab         buf[ 12]    buf[  8]    buf[  7
```

FIGURE 5

```
rp I/O queue: - ACTIVE
15 W B  0   [ 0, 4]   -256        1   buf[  3] rptab          01          02

Raw rp Queue:
   RD  02   [ 8, 2]   -256    26535     01          01       buf[  6]   0404
memory address bits 15-18  07 bits 0-14 0464

Swap Queue
   RD  01   [ 0, 3]  -1184       35     01          01       buf[ 12]    025
memory address bits 15-18  02 bits 0-14 051500
```

FIGURE 5A

```
%dead -c sysdump

Time of day Thu Feb  5 03:40:02 1976


CONFIGURATION TABLE:

Standard Configuration:
Block           rk,rp,rf,tm,tc,others
Character       kl,pc,lp,dc,dh,dp,dj,dn,mem,rk,rf,rp,tm,others

Block Devices:

 MAJ d_open     d_close    d_strategyd_tab

   0 nulldev    nulldev    rpstrat    rptab
   1 tmopen     tmclose    tmstrat    tmtab
   2 nulldev    nulldev    nulldev    nulldev

Character Devices:

 MAJ d_open     d_close    d_read     d_write    d_sgtty

   0 klopen     klclose    klread     klwrite    klsgtty
   1 dhopen     dhclose    dhread     dhwrite    dhsgtty
   2 nodev      nulldev    nulldev    nulldev    nulldev
   3 dhopen     dhclose    dhread     dmawrit    dhsgtty
   4 lpopen     lpclose    nodev      lpwrite    nodev
   5 nodev      nulldev    nulldev    nulldev    nulldev
   6 cropen     crclose    crread     nodev      nodev
   7 nulldev    nulldev    mmread     mmwrite    nodev
   8 nulldev    nulldev    rpread     rpwrite    nodev
   9 dqopen     dqclose    dqread     dqwrite    dqstbf
  10 dropen     drclose    nodev      drwrite    nodev
  11 tctopen    tctclos    tctread    tctwrit    nodev
  12 tmopen     tmclose    tmread     tmwrite    nodev
  13 nodev      nulldev    nulldev    nulldev    nulldev


Rootdev      0
Swapdev     03
Nswap    03554
Swplo      024
```

FIGURE 6

```
%dead -i sysdump
                sysdump:

Time of day Thu Feb  5 03:40:02 1976


INODE TABLE:

  #  FLAGS  ACCESS     INO    UID  DEVICE        PATHNAME
  0  D      rwxr-xr-x    1    bin  /dev/rp0      /
  1  C      rw--w--w-  550   root  /dev/rp0      /dev/ttye
  2         rw-rw-rw-  593   root  /dev/rp0      /etc/rc
  3  D      rwxrwxrwx  801   root  /dev/rp0      /nems
  4  C      rw--w--w-  414   root  /dev/rp0      /dev/ttya
  5  D      rwxrwxrwx  930   root  /dev/rp0      /rp4
  6         rw-rw-rw-  747    hgf  /dev/rp1      747 not found
  7         rw-rw-rw- 1685    hgf  /dev/rp4      /crt/image/cn12
  8  C      rw--w--w-  549   root  /dev/rp0      /dev/ttyd
  9  C      rw--w--w-  721   star  /dev/rp0      /dev/tty8
 11  C      rw--w--w-  546   root  /dev/rp0      /dev/ttyb
 12  C      rw--w--w-  524   root  /dev/rp0      /dev/ttyc
 13         rw-rw-rw- 1741    hgf  /dev/rp4      /doc/crtpbkg/cn12
 14  C      rw-rw-rw- 1038   root  /dev/rp0      /dev/regdta
 16  C      rw-rw-rw- 1043   root  /dev/rp0      /dev/rp2
 17  C      rw-rw-rw-  949   root  /dev/rp0      /dev/wall
 19  D      rwxr-xr-x 1007    bin  /dev/rp0      /dev/tct0
 20   L     rw-rw-rw- 1549    hgf  /dev/rp4      /tmp/combuf
 21  D      rwxrwxrwx  537   root  /dev/rp1      /hgf1
 23  C      rw-rw-rw-  820   root  /dev/rp0      /dev/discrt
 24  C      rw-rw-rw-  522   root  /dev/rp0      /dev/ttym
 26  C      rw-rw-rw-  470   root  /dev/rp0      /dev/ets0
 27  C      rw-rw-rw-  537   root  /dev/rp0      /dev/ttyl
 28  C      rw-rw-rw-  969   root  /dev/rp0      /dev/mesage
 32  C      rw-rw-rw-  440   root  /dev/rp0      /dev/pbc0
```

FIGURE 7

```
%dead -f sysdump
                sysdump:

Time of day Thu Feb  5 03:40:02 1976
```

FILE TABLE

#	MD	C	DEVICE	PATHNAME
0	R	1	/dev/rp0	/etc/rc
1	RW	2	/dev/rp0	/dev/ttye
2	R	1	/dev/rp1	747 not found
3	RW	2	/dev/rp0	/dev/ttyd
4	R	2	/dev/rp0	821 not found
5	R	1	/dev/rp4	/crt/image/cn12
6	RW	2	/dev/rp0	/dev/ttyb
7	RW	28	/dev/rp0	/dev/tty8
8	RW	13	/dev/rp0	/dev/rp2
9	RW	2	/dev/rp0	/dev/ttyc
10	RW	2	/dev/rp0	/dev/ttya
11	W	1	/dev/rp0	/dev/wall
12	RW	1	/dev/rp0	/dev/rp2
13	RW	1	/dev/rp0	/dev/discrt
14	W	1	/dev/rp0	/dev/ttym
15	R	2	/dev/rp0	/dev/rp2
16	W	1	/dev/rp4	/doc/crtpbkg/cn12
17	RW	1	/dev/rp0	/dev/rp2
18	W	1	/dev/rp0	/dev/ttyl
19	RW	2	/dev/rp0	/dev/mesage
20	R	1	/dev/rp0	/dev/rp2
21	RW	1	/dev/rp0	/dev/rp2
22	RW	1	/dev/rp0	/dev/rp2
23	R	1	/dev/rp0	/dev/rp2
24	RW	1	/dev/rp0	/dev/rp2
25	RW	1	/dev/rp0	/dev/rp2
26	W	1	/dev/rp0	/dev/wall
27	RW	1	/dev/rp0	/dev/rp2
28	W	1	/dev/rp0	/dev/ttym
29	W	1	/dev/rp0	/dev/wall
30	RW	1	/dev/rp0	/dev/pbc0
31	RW	3	/dev/rp4	/tmp/combuf
32	RW	1	/dev/rp0	/dev/regdta
33	RW	4	/dev/rp0	/dev/ttyd
34	RW	1	/dev/rp4	/tmp/combuf
38	RW	1	/dev/rp4	/tmp/combuf
39	R	1	/dev/rp0	/dev/ets0

FIGURE 8

```
%dead -P sysdump
            sysdump:

Time of day Thu Feb  5 03:40:02 1976


FILE DESCRIPTORS:

  PID       PROCESS

    0 M?(????4????4?C????????2%? C?????x???


    1 /etc/init


    4 /etc/updat

#   F    IND  DEVICE          PATHNAME
0   R      0 /dev/rp0         /etc/rc


 5301 pgs

#   F    IND  DEVICE          PATHNAME
0   R      4 /dev/rp0         821 not found
1   RW    33 /dev/rp0         /dev/ttyd
2   RW    33 /dev/rp0         /dev/ttyd
3   R      2 /dev/rp1         747 not found
4   R      5 /dev/rp4         /crt/image/cn12
5    W    16 /dev/rp4         /doc/crtpbkg/cn12


  271 /rp4/exsys/dpp

#   F    IND  DEVICE          PATHNAME
0   RW     7 /dev/rp0         /dev/tty8
1   RW     7 /dev/rp0         /dev/tty8
2   RW     8 /dev/rp0         /dev/rp2
3   RW    12 /dev/rp0         /dev/rp2
4    W    11 /dev/rp0         /dev/wall
5   RW    13 /dev/rp0         /dev/discrt


  308 testexec

#   F    IND  DEVICE          PATHNAME
0   R      4 /dev/rp0         821 not found
1   RW    33 /dev/rp0         /dev/ttyd
2   RW    33 /dev/rp0         /dev/ttyd


  275 /rp4/exsys/xpoll

#   F    IND  DEVICE          PATHNAME
0   RW     7 /dev/rp0         /dev/tty8
1   RW     7 /dev/rp0         /dev/tty8
2   RW     8 /dev/rp0         /dev/rp2
3   RW    22 /dev/rp0         /dev/rp2
4   RW    32 /dev/rp0         /dev/regdta
```

28

FIGURE 9

277 /rp4/exsys/xsys

	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
1	RW	7	/dev/rp0	/dev/tty8
2	RW	8	/dev/rp0	/dev/rp2
	RW	24	/dev/rp0	/dev/rp2
	W	26	/dev/rp0	/dev/wall

278 /rp4/exsys/xpb

#	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
	RW	7	/dev/rp0	/dev/tty8
	RW	8	/dev/rp0	/dev/rp2
3	RW	27	/dev/rp0	/dev/rp2
4	RW	30	/dev/rp0	/dev/pbc0

279 /rp4/exsys/tgmon

#	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
1	RW	7	/dev/rp0	/dev/tty8
2	RW	8	/dev/rp0	/dev/rp2
3	RW	25	/dev/rp0	/dev/rp2
	W	29	/dev/rp0	/dev/wall

280 /rp4/exsys/dscrmon

#	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
1	RW	7	/dev/rp0	/dev/tty8
2	RW	8	/dev/rp0	/dev/rp2
3	RW	21	/dev/rp0	/dev/rp2

282 /rp4/exsys/prmon

#	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
1	RW	7	/dev/rp0	/dev/tty8
2	RW	8	/dev/rp0	/dev/rp2
3	R	20	/dev/rp0	/dev/rp2
4	W	28	/dev/rp0	/dev/ttym

285 /rp4/crt/ctrl/prets

#	F	IND	DEVICE	PATHNAME
0	RW	7	/dev/rp0	/dev/tty8
	RW	7	/dev/rp0	/dev/tty8
2	RW	8	/dev/rp0	/dev/rp2
3	RW	31	/dev/rp4	/tmp/combuf
4	RW	34	/dev/rp4	/tmp/combuf
5	W	14	/dev/rp0	/dev/ttym

23

FIGURE 9A

```
%dead -M sysdump
                sysdump:

Time of day Thu Feb  5 03:40:02 1976

MOUNT TABLE:

  DEVICE                INODE

/dev/rp0            /
/dev/rp1            /nems
/dev/rp4            /rp4
```

FIGURE 10