**Bell Laboratories** Cover Sheet for Technical Memorandum



Title- **The MINI-UNIX System**

Date- **January 3, 1977**

TM- **77-1352-1**

Other Keywords- **UNIX**
**Operating Systems**
**PDP11**

| Author | Location | Extension | Charging Case- **39394** |
|---|---|---|---|
| **H. Lycklama** | **MH 7C-211** | **6170** | Filing Case- **39394-11** |

*ABSTRACT*

The MINI-UNIX Operating System is basically the UNIX Operating System re-written to run on a PDP-11 processor without a segmentation unit. The system supports all of the standard system calls of UNIX with the exception of: ptrace, pipe, prof, getgid and setgid. The entire system resides in 12K words of memory and is written in the C language. An emulation package is included for those machines which do not support the extended instruction set (e.g. mul, div, ash, etc.). The system will support up to 4 users using a simple round-robin time-slice scheduling algorithm. It provides an inexpensive software development system in a UNIX time-sharing environment for those installations with insufficient hardware to support the full standard UNIX Operating System.

| Pages Text 5 | Other 3 | Total 8 | |
|---|---|---|---|
| No. Figures 0 | No. Tables 0 | No. Refs. 4 | |

E-1932-U (6-73) SEE REVERSE SIDE FOR DISTRIBUTION LIST

DISTRIBUTION
(REFER GEI 13.9-3)

MPLETE MEMORANDUM TO

RRESPONDENCE FILES

OFFICIAL FILE COPY
PLUS ONE COPY FOR
EACH ADDITIONAL FILING
CASE REFERENCED

DATE FILE COPY
(FORM E-1328)

10 REFERENCE COPIES

LES,H G
DERSON,ROBERT V
DERSON,W A
DIS,R B
THURS,E
ERY,J R
YER,D L
RGERON,DAVID N
YD,GARY D
YLE,W S
ADLEY,M HELEN
ANDT,RICHARD B
EECE,H T,3RD
EEN,ROBERT N
OWN,W STANLEY
RNS,HARRY S
MLET,J V,JR
REY,J E
RRATO,R C
RISTENSEN,C
OGSTON,ALBERT M
HEN,AARON S
NNOLLY,C V
PP,DAVID H
ASZAR,MARYANN
TLER,C CHAPIN
NNIS,T M
LAN,MARIE T
WD,P G
IO,W H
RRELL,JAMES W
DER,J
ORE,RHODA J
SCHER,W C
EENY,STAN L
LCOMER,E J,JR
LLETTE,DEAN
ORDANO,PHILIP P
LDENBERG,H R
ANGER,ROBERT H
THERY,S B
GELBARGER,DAVID
HNER,MRS IRENE A
NNAY,N B
NSELL,P D
USE,A DICKSON
NRY,J B
RSCH,D
DUM,HARRY C
NT,C J
HNSON,T W

COMPLETE MEMORANDUM TO

KAISER,J F
KEEFAUVER,W L
<KEESE,W M
KIKOLSKI,H J,JR
KOGELNIK,H W
KOLL,V G
KUBIK,P S
KUSTKA,G J
LAGGY,W J
LAX,M
LEPPALA,R A
LIMB,JOHN O
+LINK,G L
LONG,P F
LOZIER,JOHN C
LUCKY,R W
LUDERER,GOTTFRIED W R
<LYCKLAMA A NYEHOLT,H
MACCHIONE,WILLIAM
MALEK,G
MARANZANO,J F
MARKEVICH,J A
MAUZEY,P
MAXSON,H S,JR
<MC DONALD,H S
MCILROY,M DOUGLAS
MEADORS,HOWARD C,JR
MEIRI,A ZVI
MILLER,STEWART E
MORGAN,SAMUEL P
MUELLER,KURT H
MYER,D J
NINKE,WILLIAM H
ORTEL,WILLIAM C G
PALUMBO,N P
PECK,R J
PENZIAS,A A
PEREZ,CATHERINE D
PINSON,ELLIOT N
PREVITE,R A
+PRIM,R C
RALEIGH,T M
RITACCO,J E
<ROBERTS,CHARLES S
ROBINSON,H E
SCHROEDER,H C
SEIBEL,RICHARD R
SHERBON,W E
SIPEK,S J
SLICHTER,W P
SWARTZWELDER,J C
SWIFT,R E
TAGUE,BERKLEY A
TERRY,M E
<TEWKSBURY,S K
<THOMPSON,J S
TONG,S Y
TOWNLEY,RAY C
TREMOULET,PAUL C
TUKEY,JOHN W
UNION,R F
USAS,ALAN M
VOGEL,GERALD C

COMPLETE MEMORANDUM TO

WAGNER,MARY R
WALDHUTER,JEFFREY S
WALTON,J RANDOLPH
WEBB,D A
WELLER,D R
WELLIVER,B S
WOLF,W J,JR
WOLONTIS,V MICHAEL
<YAMIN,ELAINE E
ZEIGLER,ROBERT M
124 NAMES

COVER SHEET ONLY TO

CORRESPONDENCE FILES

4 COPIES PLUS ONE
COPY FOR EACH FILING
CASE

ACKERMAN,A FRANK
AHO,ALFRED V
ALBERTS,BARBARA A
ALEXIS,A D,JR
ALLISON,C E,JR
AMITAY,N
AMOSS,JOHN J
ANDERSON,C R
ANTONELLI,PHYLLIS
ARNOLD,GEORGE W
ARNOLD,S L
ARTIS,H P
BACCASH,JEANNE M
BACKUS,C F,SR
BAKER,BRENDA S
<BALDWIN,GARY L
BALDWIN,GEORGE L
BATTAGLIA,FRANCES
BAUER,HELEN A
<BAUGH,C R
BECKETT,J T
BEL BRUNO,KATHLEEN A
BERGERON,R F,JR
BERNSTEIN,DANIELLE R
<BEYER,JEAN-DAVID
BIANCHI,M H
BIRCHALL,R H
BIREN,IRMA B
BITTRICH,MARY E
BLAZIER,S D
BLINN,J C
BLUM,MARION
BONANNI,L E
BOTHUR,R H
BOURNE,STEPHEN R
BOWRA,JAMES W
BOWYER,L RAY
BOYLE,GERALD C
BRADFORD,EDWARD G
BRAINARD,R C
BRANTLEY,JOAN T

COVER SHEET ONLY TO

BROWNE,PAUL N
BROWN,C W
BROWN,JAMES W
BURES,C
BURGESS,JOHN T,JR
BURNETTE,W A
BURNET,ROSE M
BURROWS,THOMAS A
<BUTZIEN,PAUL E
CALLAHAN,R L
CAMPBELL,JERRY H
CANADAY,RUDD H
CANDY,J C
CANNON,T B
CARDOZA,WAYNE M
CARRAN,JOHN H
CARR,DAVID C
CASEY,J L
CATO,H E
CHAFFEE,N F
CHARLES,JOYCE
CHEE,T
CHEN,STEPHEN
CHERRY,LORINDA L
CHIANG,T C
CHODROW,M M
CHRIST,C W,JR
CLOUTIER,J
CLYMER,J C
COHEN,HARVEY
COLDREN,L A
<COLE,LOUIS M
<COLLIER,ROBERT J
COMMRADE,GERALDINE
COOK,T J
COREY,D A
COSTELLO,PETER E
COSTON,W P
<COURTNEY-PRATT,J S
CRAGUN,DONALD W
CRANE,R P
CROWE,MARGARET M
CUNNINGHAM,S J,JR
DAVIDSON,CHARLES LEWIS
DAVIS,D R
DAVIS,R DREW
DE GRAAF,D A
DE JAGER,D S
DE PAOLA,THELMA T
DENISON,JOANNA
DERMOND,FAITH L
DEVLIN,N V
DICK,G W
DOBLMAIER,A H
DOLOTTA,T A
DONKIN,A E
DONOFRIO,LOUIS,J
DRENNING,C R
DRUMMOND,R E
DSTEFAN,D J
DUDICK,ANTHONY L
DUDLEY,ELIZABETH H
DWYER,T J

COVER SHEET ONLY TO

EDDY,MICHAEL R
EDELSON,DAVID
EDMUNDS,T W
EITELBACH,DAVID L
ELLIOTT,RUBY J
EL-SHABAZZ,A M
ELY,T C
EPLEY,ROBERT V
ERRICHIELLO,PHILIP M
ESTES,VIRGINIA DANIELLE
ESTOCK,RICHARD G
EVENSON,E K
FACTOR,ROBERT M
FASCIANO,V
FEINBERG,HENRY R
FELTON,WILLIAM A
FERRER,NANCY L
<FLANAGAN,J L
FLANDRENA,R
FOLEY,G
FONTENOT,SHARON M
FORTNEY,V J
FOUNTOUKIDIS,A
FRANKLIN,DANIEL L
FRANKS,RICHARD L
FRANK,H GREGORY
FRANZ,ANN M
FRASER,A G
FREEDMAN,MARVIN I
FREEMAN,R DON
GANGAWARE,BERNICE C
GANNON,T F
<GATES,G W
GELBER,CHERON L
GIBSON,H T,JR
GIBSON,T A
GILLON,ALEX C
GILL,R J
GINSBERG,N
<GLASSER,ALAN L
GODWIN,R E
GOETZ,FRANK M
GOGUEN,N H
<GORMAN,J E
GORTON,D R
<GRAHAM,R L
GRANT,GEORGE E
GRAVEMAN,R F
GREENBAUM,HOWARD J
GRESHAM,J P
GROSS,ARTHUR G
GUIDI,PIER V
GUTT,DONALD J
HAGGERTY,JOSEPH P
HAIGHT,R C
HALL,J T
HALPIN,T
HAMILTON,LINDA L
HAMILTON,PATRICIA A
HANCOCK,J DAVID
<HANNAH,JUDY R
HANSEN,LYNN D
HAYDEN,DONALD F,JR
...
462 TOTAL

NAMED BY AUTHOR > CITED AS REFERENCE < REQUESTED BY READER (NAMES WITHOUT PREFIX
ERE SELECTED USING THE AUTHOR'S SUBJECT OR ORGANIZATIONAL SPECIFICATION AS GIVEN BELOW)

CURY SPECIFICATION.....

PLETE MEMO TO:
135-DPH 13-DIR 11-EXD 15-EXD 16-EXD 127-SUP 135-SUP 10-EXD 3114 1352 1356 1359
34

ER SHEET TO:
135-MTS 9152-MTS 1271 1273

OS# = UNIX/OPERATING SYSTEM

---

LYCKLAMA,H
MH 7C211

TM-77-1352-1
TOTAL PAGES 8

GET A COMPLETE COPY:

BE SURE YOUR CORRECT ADDRESS IS GIVEN ON THE OTHER SIDE.
FOLD THIS SHEET IN HALF WITH THIS SIDE OUT AND STAPLE.

CIRCLE THE ADDRESS AT RIGHT. USE NO ENVELOPE.

PLEASE SEND A COMPLETE COPY TO THE ADDRESS SHOWN ON THE OTHER SIDE
NO ENVELOPE WILL BE NEEDED IF YOU SIMPLY STAPLE THIS COVER SHEET TO THE COMPLETE COPY.
IF COPIES ARE NO LONGER AVAILABLE PLEASE FORWARD THIS REQUEST TO THE CORRESPONDENCE FILES.

**Bell Laboratories**

Subject: **The MINI-UNIX System**
Case- 39394 -- File- 39394-11

date: **January 3, 1977**

from: **H. Lycklama**

TM: **77-1352-1**

*MEMORANDUM FOR FILE*

## 1. Introduction

The MINI-UNIX Operating System was written to run on all PDP-11 processors without the memory management unit available on the PDP-11/40, 11/45 and 11/70 processors and is therefore restricted to a 28K word address space. The operating system itself (MX) is basically a modified version of the Version 6 UNIX Operating System (1) and as such supports most of the standard UNIX "system calls". The system resides in 12K words of memory and is written in the C language. An instruction emulation package is included in the system for those machines which do not support the extended instruction set (e.g. mul, div, ash, etc.). The system supports up to four users using a simple round-robin time-slicing algorithm. It supports all of the UNIX user programs unmodified. User programs which have been slightly modified are discussed in a later section. MINI-UNIX thus provides an inexpensive software development system in a UNIX time-sharing environment for those installations with a minimum amount of hardware which is insufficient to support the full Version 6 UNIX Operating System.

Other software tools are also available for easing the transition from software written under the DEC DOS Operating System. This includes a macro assembler and linker.

## 2. Hardware

The MINI-UNIX system runs on any PDP-11 processor with 28K words of memory. The PDP-11 computer is a 16-bit word mini-computer with a UNIBUS for interfacing DEC peripherals to the CPU. The typical configuration consists of a PDP-11/10 CPU with 28K words of memory, a console terminal and an RK05 moving-head disk controller with two removable disk cartridges for swapping and file system storage. Each RK05 disk pack has 2.5 Megabytes (8-bit byte) of storage. However, the MX system also supports the RF fixed-head disk (1 Megabyte) and the RP03 and RP04 moving-head disk controllers with 40M bytes and 80M bytes, respectively. Other peripherals supported include line-printer, Dectape, magtape and various asynchronous and synchronous interface units.

The system is normally configured to be 12K words in size. This includes an emulation package for interpreting the 10 extended instructions normally performed by the EIS hardware available as an option on some PDP-11 processors and standard on the PDP-11/45 processor. A minimum system has room for 6 or 7 system buffers. As new drivers are added to the system, the number of system buffers must be decreased if the system size is maintained at 12K words. Thus it is recommended that for some applications it may be appropriate to add the drivers for only a few peripherals on any one version of the system and thus maintain a few versions of the system, one for each set of drivers desired concurrently in the system. This keeps the system size at 12K words in order to be able to support all of the user software of Version 6 UNIX.

## 3. System Features

The Operating System itself is written in the high-level language, C (2) and as such bears a strong resemblance to the standard UNIX Operating System which runs on the PDP-11/40, 11/45 and 11/70 computers. Because of the memory address space limitation due to the lack of a segmentation unit, the system size is generally kept to 12K words in size. Thus its capabilities are somewhat less than those of the standard UNIX system, especially in the area of interprocess communication and interactive debugging. The number of processes (and users) which MINI-UNIX (MX) can comfortably support is in general much less than for the standard UNIX system. No more than four users are recommended for MX. Currently the number of processes allowed is thirteen. This is a 'sysgen' parameter. The address space available to a user program is 16K words. This enables almost all user programs which run under Version 6 UNIX to run unmodified under MX. The exceptions are noted in a later section. The capabilities of MX fall somewhere in between those of standard Version 6 UNIX and those of the LSX operating system (3). LSX is a single-user UNIX-compatible system for the LSI-11 microprocessor using floppy disks as secondary storage.

MX supports all of the UNIX system calls of the Version 6 UNIX operating system with the exception of: ptrace, pipe, prof, setgid and getgid. For the sake of completeness the status of implementation of the system calls are summarized in an Appendix. Thus MX supports all UNIX user programs with the exception of programs which use the above mentioned system calls. User programs are compiled and relocated to start at address 060000 and may occupy up to 16K words of memory. Up to 13 processes may exist at any one time, although only one process may be in core and running. This should be sufficient to handle up to four simultaneous users. Since no memory relocation is available, the complete user program image must be swapped out to bring in a new user program. Hence no sharing of text is possible. No software memory management is required. Scheduling is done on a simple round-robin basis with each process in the run state receiving a two second time slice.

The file system supported by MX is identical to that provided by the UNIX time-sharing system. The structure of the super-block and of the file inodes is maintained. MX supports the identical file system hierarchical structure and makes the same distinctions between ordinary files, directories and special files. Removable file systems are supported as well. Hence 'mount' and 'umount' system requests are treated identical to those in standard UNIX. Under MX, a file's size is limited to one megabyte. Large files are supported, but huge files (two level indirect block) are not. Groups are not supported in MX. Thus a file has only a given owner user ID and no group ID. Read, write and execute permission bits are available for both owner and non-owner of a file. The set-user ID bit is also supported in MINI-UNIX.

No interactive debugging is possible in MX since the 'ptrace' system call is not implemented. One may still use the C debugger 'cdb' but cannot plant breakpoint traps in the running image of a child process. It may only be used for post-mortem debugging on core images. The profiling of a process to determine where it is spending its time is also not permitted. Pipes have not been implemented in the MX system for two reasons. One is that it requires too much code in the system which is already butting its head against the top. Two is that the overhead required in switching from the process writing the pipe to the process reading the pipe involves a process swap. Filters, however are simulated at the command level as discussed in the next section.

Some other features have been stripped out of the system in MX in the interest of minimizing system address space. Upon reading a file block, "read-ahead" is not invoked. This contributes only slightly to a loss in throughput. Physical I/O has also not been implemented. This precludes the ability to read large contiguous pieces of disk directly into the user's address space without system side-buffering. The system buffering scheme is much simpler than in standard UNIX but a maximum of only eight buffers may be allocated in a minimum system configuration.

## 4. UNIX User Program Modifications

Since the MX system uses no segmentation unit, all user programs must be compiled and relocated to start at address 060000. Under MX, the C programs are relocated automatically using the 'ld' program which has been modified as described below. For assembly-language source programs, the 'a.out' program may be relocated either using the 'reloc' program or the 'ld' program. The various user programs which have been modified from the standard Version 6 UNIX or new programs which are supported but not mentioned in the UNIX Programmer's Manual (4) are described below.

### 4.1. ar

The archive program supported is a version which is updated from that described in the UNIX Programmer's Manual. It is written in the C language and has some new keys added. However the documentation still applies.

### 4.2. bc

The 'bc' command works as in the standard UNIX system but cannot be used interactively since pipes are not implemented in the system. In generating the source for 'bc' using the 'yacc' compiler-compiler, the source must be edited slightly to make it simulate the use of pipes.

### 4.3. check

The **check** program is not described in the UNIX Programmer's Manual. However, it combines most of the individual features of **dcheck, icheck** and **ncheck** as described in the manual. A flag of *-l* will do the equivalent of **ncheck** with no flags. All other flags described for **dcheck, icheck** and **ncheck** apply.

### 4.4. Debugger

The debugger **db** has been slightly modified in order to ease debugging under the MINI-UNIX system. The default relocation address of all programs is assumed to be 60000(8). If the flag '-a' is given when the debugger is invoked, the relocation address is assumed to be zero. This is useful for debugging the system.

The default address can be changed by editing 'db1.s' and setting the variable 'uorg' equal to the new relocation address. Then the source code should be re-assembled. If none of the changes are desired, the variable *mx* in *db1.s* should be set to 0 and the source code re-assembled.

### 4.5. Kdmp

This program is used to extract a dump of the complete system (28K words) from the swap area on disk. The core image produced in the file "kore" may then be debugged post-mortem. Consult system source code for the actual disk tracks used. If a system crash occurs and the core image of the system is written out to disk, the system should be booted up single-user. Immediately, one should execute "kdmp" (preferably on an uncorrupted file system) before the swap area is over-written.

### 4.6. Ld

The **ld** program has been modified to relocate the program origin to 060000 automatically. This is to facilitate the compilation of C programs under the MINI-UNIX system. Note that all assembly programs, after being assembled, must be relocated to 060000 either by using the link editor **ld** or the relocation program **reloc**. The relocation origin may be changed from 060000 by changing the value of the TOPSYS parameter in *ld.c* to the appropriate value.

Using the *'-a'* option flag with **ld** turns off the relocation of a program's origin. The program is then assumed to start execution at location 0. This is useful for link editing the MINI-UNIX system itself or any other program which starts execution at location 0.

### 4.7. Mkpt

The **mkpt** program constructs a prototype file given a specification file for direct input to the **mkfs** program. Run "man 8 mkpt" for details.

### 4.8. Ps

The process status command outputs basically the same information as that of the Version 6 UNIX **ps** command. It has been modified to take into consideration the different process table layout and the different swapping technique used in MINI-UNIX.

### 4.9. Reloc

This Program is used to relocate all relocatable symbols in a program. Thus "reloc a.out 60000" will relocate all relocatable symbols in "a.out" and relocate the starting address of the program (absolute zero) to 060000. It must be used with all assembler output programs if the link-editor is not required.

### 4.10. Sh

Since the MINI-UNIX system does not support pipes, the shell has been modified to simulate pipes through the use of disk files. When a command line which requires the use of a pipe is detected, a disk file is created and opened for reading and writing. The file is immediately unlinked, so that the name is available for another pipe right away. The file is called '.__pf'. Hopefully this name will not conflict with any user file names. Thus, the symbol '|' in a shell command line becomes equivalent to '> .__pf ; < .__pf'. The command:

```
% prog1 | prog2
```

translates into:

```
% prog1 > .__pf ; prog2 < .__pf
```

The process writing on the pipe writes everything into the file, and when it exits, the reader process is swapped in. It reads what the writer has written on the pipe file. The only danger in this type of implementation is due to the limitation of space on the file system being used. A very large amount of information going through a pipe could fill up the disk. Except for this, the pseudo-pipe code is transparent to the user.

The **sync** command has been added to the shell. Thus the system call is made directly from the shell and no other process is spawned.

### 4.11. Typo

This program checks for the most likely spelling errors in a document. It has been modified somewhat to enable it to fit in 16K words of memory.

### 4.12. Yacc

The source for the **yacc** compiler has been edited to change some table sizes to make it run in 16K words of memory. Further editing of symbols at the beginning of *y0.c* may be required to make up a **yacc** compiler with different table sizes for a particular application.

## 5. Summary

Since MX uses no segmentation unit, no protection is provided for the user program. Thus new user programs must be debugged carefully. In practice, the use of the C language limits the user's program's use of the program counter and stack pointer thus limiting damage and usually causing a bus error before anything drastic happens. The lack of a segmentation unit does have its advantages. It means the user can directly access all I/O registers on the UNIBUS and does not have to write special device drivers interfaced with the file system to control the special peripherals. Thus in cases where a real-time program is to be run, one may disable the system clock to inhibit unwanted clock interrupts and also swapping of processes. One may also catch clock interrupts during the running of user programs if the clock is to be used for user program timing control. The clock should then be restored to system control upon exit from the user program.

There is also another set of user programs available under MX which may be used to ease transition from DOS, the DEC operating system, to UNIX for those installations now using the DEC DOS operating system on a PDP-11 CPU. This package consists of a macro-assembler and a linker-loader for assembling programs written under DOS for the DEC macro-assembler. The result is a UNIX 'a.out' file.

The normal configuration for MX includes a PDP-11/10 CPU with 28K words of memory and two RK05 disk cartridges for secondary. The PDP-11/10 processor is slower than the PDP-11/40 processor and does not have the full instruction set of the PDP-11/40 processor, thus requiring the emulation of the missing instructions. A typical C compilation requires about twice the total time of that required on the equivalent PDP-11/40 configuration. However, response to the editor commands is not significantly longer than on a more powerful CPU. The cost of a minimum configuration:

PDP-11/10 CPU
28K words memory
2 RK05 disk drives
KL11 interface to control console
DL11E interface to dial-up line
60 cycle clock

is of the order of $20,000 at today's prices (December 1976). This provides an inexpensive tool for software development in a UNIX time-sharing environment for those configurations which have insufficient hardware to support a full Version 6 UNIX system.

H. Lycklama

**H. Lycklama**

MH-1352-HL

Atts.
References
Appendix

*References*

1. K.Thompson and D. M. Ritchie, "The UNIX Time-Sharing System", Comm. ACM 17, (July 1974), p365.
2. D. M. Ritchie, "C Reference Manual", TM-74-1273-1.
3. H. Lycklama, "The LSI-UNIX System", TM-76-1352-4.
4. K. Thompson and D. M. Ritchie, "UNIX Programmer's Manual - 6th Edition", May, 1975.

## Appendix - MX System Calls

```
0, &nullsys,   /*  0 = indir */
0, &rexit,     /*  1 = exit */
0, &fork,      /*  2 = fork */
2, &read,      /*  3 = read */
2, &write,     /*  4 = write */
2, &open,      /*  5 = open */
0, &close,     /*  6 = close */
0, &wait,      /*  7 = wait */
2, &creat,     /*  8 = creat */
2, &link,      /*  9 = link */
1, &unlink,    /* 10 = unlink */
2, &exec,      /* 11 = exec */
1, &chdir,     /* 12 = chdir */
0, &gtime,     /* 13 = time */
3, &mknod,     /* 14 = mknod */
2, &chmod,     /* 15 = chmod */
2, &chown,     /* 16 = chown */
1, &sbreak,    /* 17 = break */
2, &stat,      /* 18 = stat */
2, &seek,      /* 19 = seek */
0, &getpid,    /* 20 = getpid */
3, &smount,    /* 21 = mount */
1, &sumount,   /* 22 = umount */
0, &setuid,    /* 23 = setuid */
0, &getuid,    /* 24 = getuid */
0, &stime,     /* 25 = stime */
3, &nullsys,   /* 26 = ptrace */
0, &alarm,     /* 27 = alarm */
1, &fstat,     /* 28 = fstat */
0, &pause,     /* 29 = pause */
1, &nullsys,   /* 30 = smdate; inoperative */
1, &stty,      /* 31 = stty */
1, &gtty,      /* 32 = gtty */
2, &nosys,     /* 33 = access, not implemented */
0, &nice,      /* 34 = nice */
0, &sslep,     /* 35 = sleep */
0, &sync,      /* 36 = sync */
1, &kill,      /* 37 = kill */
0, &getswit,   /* 38 = switch */
0, &nosys,     /* 39 = x */
0, &nosys,     /* 40 = x */
0, &dup,       /* 41 = dup */
0, &nosys,     /* 42 = pipe, pseudo-pipe in Shell */
1, &times,     /* 43 = times */
4, &nosys,     /* 44 = prof, not implemented */
0, &nosys,     /* 45 = x */
0, &nullsys,   /* 46 = setgid, not implemented */
0, &nullsys,   /* 47 = getgid, not implemented */
2, &ssig,      /* 48 = sig */
0, &nosys,     /* 49 = x */
```

```
0, &nosys,      /* 50 = x */
0, &nosys,      /* 51 = x */
0, &nosys,      /* 52 = x */
0, &nosys,      /* 53 = x */
0, &nosys,      /* 54 = x */
0, &nosys,      /* 55 = x */
0, &nosys,      /* 56 = x */
0, &nosys,      /* 57 = x */
0, &nosys,      /* 58 = x */
0, &nosys,      /* 59 = x */
0, &nosys,      /* 60 = x */
0, &nosys,      /* 61 = x */
0, &nosys,      /* 62 = x */
0, &nosys       /* 63 = x */
```