Unix Programmer's Manual
Stanford University Systems Programmer's Version

This manual consists of selections from the Unix Programmer's Manual that are likely to be of use to systems programmers at Stanford. Details pertinent to system administration are not included here; they are available in the System Administrator's version. As a guide to the organization of this manual, the organization of the complete manual is outlined below. The table of contents of each volume or section is included in full, so that the reader can determine what additional material is available; portions actually included in this version are checked (except in Volume 1).

The Unix Programmer's Manual provided by Berkeley has been augmented to contain documentation of additional software used at Stanford. The complete manual consists of two volumes. Volume 1 contains brief "manual pages" describing the commands and features provided by the system. There are nine sections:

1. Commands
2. System calls
3. Subroutines
4. Special files
5. File formats and conventions
6. Games
7. Miscellaneous
8. Maintenance commands and procedures
9. PUP library routines

This systems programmer's manual contains all of Volume 1, except Section 8.

Volume 2 contains documents that supplement the manual pages in Volume 1. These are mostly articles, tutorials or manuals on specific programs, commands or systems. There are five sections:

2a and 2b Provided by Bell Laboratories.
2c Provided by Berkeley.
User Contributed Software Provided by users whose software is distributed together with Unix.
Additional Material Not part of the Berkeley manual.

This systems programmer's manual contains a variety of articles, including complete Emacs and MH manuals.
Getting Started

The following material in this manual is particularly useful for obtaining an overview of 4.2 Unix and for finding one's way around the manual:

• For users unfamiliar with Unix, the introduction to Volume 1.
• For users familiar with 4.1 BSD Unix, the documents “Changes from 4.1 BSD to 4.2 BSD Vax Unix at Stanford University”, at the start of Additional Material, and “Bug fixes and changes in 4.2 BSD”, at the start of Volume 2c.
• The tables of contents at the start of Volumes 1, 2, 2c, User Contributed Software and Additional Material.
• The permuted index at the start of Volume 1.

Since changes are made to the system periodically, the most reliable way to locate up-to-date documentation based on keyword is to use the command `apropos(1)` online.
UNIX PROGRAMMER'S MANUAL

4.2 Berkeley Software Distribution
Virtual VAX—11 Version

August, 1983

Computer Science Division
Department of Electrical Engineering and Computer Science
University of California
Berkeley, California 94720
This update to the 4.1 distribution of June 1981 provides support for the VAX 11/730, full networking and interprocess communication support, an entirely new file system, and many other new features. It is certainly the most ambitious release of software ever prepared here and represents many man-years of work. Bill Shannon (both at DEC and at Sun Microsystems) and Robert Elz of the University of Melbourne contributed greatly to this distribution through new device drivers and painful debugging episodes. Rob Gurwitz of BBN wrote the initial version of the code upon which the current networking support is based. Eric Allman of Britton-Lee donated countless hours to the mail system. Bill Croft (both at SRI and Sun Microsystems) aided in the debugging and development of the networking facilities. Dennis Ritchie of Bell Laboratories also contributed greatly to this distribution, providing valuable advise and guidance. Helge Skrivervik worked on the device drivers which enabled the distribution to be delivered with a TU58 console cassette and RX01 console floppy disk, and rewrote major portions of the standalone i/o system to support formatting of non-DEC peripherals.

Numerous others contributed their time and energy in organizing the user software for release, while many groups of people on campus suffered patiently through the low spots of development. As always, we are grateful to the UNIX user community for encouragement and support.

Once again, the financial support of the Defense Advanced Research Projects Agency is gratefully acknowledged.

S. J. Leffler
W. N. Joy
M. K. McKusick

Preface to the Fourth Berkeley distribution

This update to the fourth distribution of November 1980 provides support for the VAX 11/750 and for the full interconnect architecture of the VAX 11/780. Robert Elz of the University of Melbourne contributed greatly to this distribution especially in the boot-time system configuration code; Bill Shannon of DEC supplied us with the implementation of DEC standard bad block handling. The research group at Bell Laboratories and DEC Merrimack provided us with access to 11/750’s in order to debug its support.

Other individuals too numerous to mention provided us with bug reports, fixes and other enhancements which are reflected in the system. We are grateful to the UNIX user community for encouragement and support.

The financial support of the Defence Advanced Research Projects Agency in support of this work is gratefully acknowledged.

W. N. Joy
R. S. Fabry
K. Sklower

Preface to the Fourth Berkeley distribution

This manual reflects the Berkeley system mid-October, 1980. A large amount of tuning has been done in the system since the last release; we hope this provides as noticeable an improvement for you as it did for us. This release finds the system in transition; a number of facilities have been added in experimental versions (job control, resource limits) and the implementation of others is imminent (shared-segments, higher performance from the file system, etc.). Applications which use facilities that are in transition should be aware that some of the system calls and library routines will change in the near future. We
have tried to be conscientious and make it very clear where this is likely.

A new group has been formed at Berkeley, to assume responsibility for the future development and support of a version of UNIX on the VAX. The group has received funding from the Defense Advanced Research Projects Agency (DARPA) to supply a standard version of the system to DARPA contractors. The same version of the system will be made available to other licensees of UNIX on the VAX for a duplication charge. We gratefully acknowledge the support of this contract.

We wish to acknowledge the contribution of a number of individuals to the system.

We would especially like to thank Jim Kulp of IIASA, Laxenburg Austria and his colleagues, who first put job control facilities into UNIX; Eric Allman, Robert Henry, Peter Kessler and Kirk McKusick, who contributed major new pieces of software; Mark Horton, who contributed to the improvement of facilities and substantially improved the quality of our bit-mapped fonts, our hardware support staff: Bob Kridie, Anita Hirsch, Len Edmondson and Fred Archibald, who helped us to debug a number of new peripherals; Ken Arnold who did much of the leg-work in getting this version of the manual prepared, and did the final editing of sections 2-6, some special individuals within Bell Laboratories: Greg Chesson, Stuart Feldman, Dick Haight, Howard Katseff, Brian Kernighan, Tom London, John Reiser, Dennis Ritchie, Ken Thompson, and Peter Weinberger who helped out by answering questions; our excellent local DEC field service people, Kevin Althaus and Frank Chargois who kept our machine running virtually all the time, and fixed it quickly when things broke; and, Mike Accetta of Carnegie-Mellon University, Robert Elz of the University of Melbourne, George Goble of Purdue University, and David Kashtan of the Stanford Research Institute for their technical advice and support.

Special thanks to Bill Munson of DEC who helped by augmenting our computing facility and to Eric Allman for carefully proofreading the “last” draft of the manual and finding the bugs which we knew were there but couldn’t see.

We dedicate this to the memory of David Sakrison, late chairman of our department, who gave his support to the establishment of our VAX computing facility, and to our department as a whole.

W. N. Joy
O. Babaoglu
R. S. Fabry
K. Sklower

Preface to the Third Berkeley distribution

This manual reflects the state of the Berkeley system, December 1979. We would like to thank all the people at Berkeley who have contributed to the system, and particularly thank Prof. Richard Fateman for creating and administrating a hospitable environment, Mark Horton who helped prepare this manual, and Eric Allman, Bob Kridie, Juan Porcar and Richard Tuck for their contributions to the kernel.

The cooperation of Bell Laboratories in providing us with an early version of UNIX/32V is greatly appreciated. We would especially like to thank Dr. Charles Roberts of Bell Laboratories for helping us obtain this release, and acknowledge T. B. London, J. F. Reiser, K. Thompson, D. M. Ritchie, G. Chesson and H. P. Katseff for their advice and support.

W. N. Joy
O. Babaoglu
Preface to the UNIX/32V distribution

The UNIX† operating system for the VAX*-11 provides substantially the same facilities as the UNIX system for the PDP*-11.

We acknowledge the work of many who came before us, and particularly thank G. K. Swanson, W. M. Cardoza, D. K. Sharma, and J. F. Jarvis for assistance with the implementation for the VAX-11/780.

T. B. London
J. F. Reiser

Preface to the Seventh Edition

Although this Seventh Edition no longer bears their byline, Ken Thompson and Dennis Ritchie remain the fathers and preceptors of the UNIX time-sharing system. Many of the improvements here described bear their mark. Among many, many other people who have contributed to the further flowering of UNIX, we wish especially to acknowledge the contributions of A. V. Aho, S. R. Bourne, L. L. Cherry, G. L. Chesson, S. I. Feldman, C. B. Haley, R. C. Haight, S. C. Johnson, M. E. Lesk, T. L. Lyon, L. E. McMahon, R. Morris, R. Muha, D. A. Nowitz, L. Wehr, and P. J. Weinberger. We appreciate also the effective advice and criticism of T. A. Dolotta, A. G. Fraser, J. F. Maranzano, and J. R. Mashey; and we remember the important work of the late Joseph F. Ossanna.

B. W. Kernighan
M. D. McIlroy

† UNIX is a trademark of Bell Laboratories.
*VAX and PDP are Trademarks of Digital Equipment Corporation.
INTRODUCTION TO VOLUME 1

This volume gives descriptions of the publicly available features of the UNIX/32v† system, as extended to provide a virtual memory environment and other enhancements at U. C. Berkeley. It does not attempt to provide perspective or tutorial information upon the UNIX operating system, its facilities, or its implementation. Various documents on those topics are contained in Volume 2. In particular, for an overview see ‘The UNIX Time-Sharing System’ by Ritchie and Thompson; for a tutorial see ‘UNIX for Beginners’ by Kernighan, and for an guide to the new features of this virtual version, see ‘Getting started with Berkeley Software for UNIX on the VAX’ in volume 2C.

Within the area it surveys, this volume attempts to be timely, complete and concise. Where the latter two objectives conflict, the obvious is often left unsaid in favor of brevity. It is intended that each program be described as it is, not as it should be. Inevitably, this means that various sections will soon be out of date.

The volume is divided into eight sections:

1. Commands
2. System calls
3. Subroutines
4. Special files
5. File formats and conventions
6. Games
7. Macro packages and language conventions
8. Maintenance commands and procedures

Commands are programs intended to be invoked directly by the user, in contradistinction to subroutines, which are intended to be called by the user’s programs. Commands generally reside in directory /bin (for binary programs). Some programs also reside in /usr/bin, or in /usr/ucb, to save space in /bin. These directories are searched automatically by the command interpreters.

System calls are entries into the UNIX supervisor. The system call interface is identical to a C language procedure call; the equivalent C procedures are described in Section 2.

An assortment of subroutines is available; they are described in section 3. The primary libraries in which they are kept are described in intro(3). The functions are described in terms of C, but most will work with Fortran as well.

The special files section 4 discusses the characteristics of each system ‘file’ that actually refers to an I/O device. The names in this section refer to the DEC device names for the hardware, instead of the names of the special files themselves.

The file formats and conventions section 5 documents the structure of particular kinds of files; for example, the form of the output of the loader and assembler is given. Excluded are files used by only one command, for example the assembler’s intermediate files.

Games have been relegated to section 6 to keep them from contaminating the more staid information of section 1.

† UNIX is a trademark of Bell Laboratories.
Section 7 is a miscellaneous collection of information necessary to writing in various specialized languages: character codes, macro packages for typesetting, etc.

The maintenance section 8 discusses commands and procedures not intended for use by the ordinary user. The commands and files described here are almost all kept in the directory /etc.

Each section consists of a number of independent entries of a page or so each. The name of the entry is in the upper corners of its pages, together with the section number, and sometimes a letter characteristic of a subcategory, e.g. graphics is 1G, and the math library is 3M. Entries within each section are alphabetized. The page numbers of each entry start at 1; it is infeasible to number consecutively the pages of a document like this that is republished in many variant forms.

All entries are based on a common format, not all of whose subsections will always appear.

The name subsection lists the exact names of the commands and subroutines covered under the entry and gives a very short description of their purpose.

The synopsis summarizes the use of the program being described. A few conventions are used, particularly in the Commands subsection:

**Boldface** words are considered literals, and are typed just as they appear.

Square brackets [ ] around an argument indicate that the argument is optional. When an argument is given as 'name', it always refers to a file name.

Ellipses '...' are used to show that the previous argument-prototype may be repeated.

A final convention is used by the commands themselves. An argument beginning with a minus sign '–' is often taken to mean some sort of option-specifying argument even if it appears in a position where a file name could appear. Therefore, it is unwise to have files whose names begin with '–'.

The description subsection discusses in detail the subject at hand.

The files subsection gives the names of files which are built into the program.

A see also subsection gives pointers to related information.

A diagnostics subsection discusses the diagnostic indications which may be produced. Messages which are intended to be self-explanatory are not listed.

The bugs subsection gives known bugs and sometimes deficiencies. Occasionally also the suggested fix is described.

At the beginning of the volume is a table of contents, organized by section and alphabetically within each section. There is also a permuted index derived from the table of contents. Within each index entry, the title of the writeup to which it refers is followed by the appropriate section number in parentheses. This fact is important because there is considerable name duplication among the sections, arising principally from commands which exist only to exercise a particular system call.

**HOW TO GET STARTED**

This section sketches the basic information you need to get started on UNIX how to log in and log out, how to communicate through your terminal, and how to run a program. See 'UNIX for Beginners' in Volume 2 for a more complete introduction to the system.

Logging in. You must call UNIX from an appropriate terminal. Almost any ASCII terminal capable of full duplex operation and generating the entire character set can be used. You must also have a valid user name, which may be obtained, together with necessary telephone numbers, from the system administration. After a data connection is established, the login procedure depends on what kind of terminal you are using and local system conventions. The following examples are typical.
**300-baud terminals:** Such terminals include the GE Terminet 300, and most display terminals run with popular modems. These terminals generally have a speed switch which should be set at '300' (or '30' for 30 characters per second) and a half/full duplex switch which should be set at full-duplex. (This switch will often have to be changed since many other systems require half-duplex). When a connection is established, the system types 'login:': you type your user name, followed by the 'return' key. If you have a password, the system asks for it and turns off the printer on the terminal so the password will not appear. After you have logged in, the 'return', 'new line', or 'linefeed' keys will give exactly the same results.

**1200- and 150-baud terminals:** If there is a half/full duplex switch, set it at full-duplex. When you have established a data connection, the system types out a few garbage characters (the 'login:' message at the wrong speed). Depress the 'break' (or 'interrupt') key; this is a speed-independent signal to UNIX that a different speed terminal is in use. The system then will type 'login:,' this time at another speed. Continue depressing the break key until 'login:' appears in clear, then respond with your user name. From the TTY 37 terminal, and any other which has the 'newline' function (combined carriage return and linefeed), terminate each line you type with the 'new line' key, otherwise use the 'return' key.

**Hard-wired terminals.** Hard-wired terminals usually begin at the right speed, up to 9600 baud; otherwise the preceding instructions apply.

For all these terminals, it is important that you type your name in lower-case if possible; if you type upper-case letters, UNIX will assume that your terminal cannot generate lower-case letters and will translate all subsequent upper-case letters to lower case.

The evidence that you have successfully logged in is that a shell program will type a prompt ('$' or '%') to you. (The shells are described below under 'How to run a program.')

For more information, consult `tset(1)`, and `stty(1)`, which tell how to adjust terminal behavior, `getty(8)`, which discusses the login sequence in more detail, and `tty(4)`, which discusses terminal I/O.

**Logging out.** There are three ways to log out:

- By typing an end-of-file indication (EOT character, control-d) to the Shell. The Shell will terminate and the 'login: ' message will appear again.
- You can log in directly as another user by giving a `login` command.
- If worse comes to worse, you can simply hang up the phone; but beware — some machines may lack the necessary hardware to detect that the phone has been hung up.

Ask your system administrator if this is a problem on your machine.

**How to communicate through your terminal.** When you type characters, a gnome deep in the system gathers your characters and saves them in a secret place. The characters will not be given to a program until you type a return (or newline), as described above in Logging in.

UNIX terminal I/O is full-duplex. It has full read-ahead, which means that you can type at any time, even while a program is typing at you. Of course, if you type during output, the printed output will have the input characters interspersed. However, whatever you type will be saved up and interpreted in correct sequence. There is a limit to the amount of read-ahead, but it is generous and not likely to be exceeded unless the system is in trouble. When the read-ahead limit is exceeded, the system throws away all the saved characters (or beeps, if your prompt was a %).

The character '@' in typed input kills all the preceding characters in the line, so typing mistakes can be repaired on a single line. Also, the character '#' erases the last character typed. (Most users prefer to use a backspace rather than '#', and many prefer control-U instead of '@'; `tset(1)` or `stty(1)` can be used to arrange this.) Successive uses of '#' erase characters back to, but not beyond, the beginning of the line. '@' and '#' can be transmitted to a program by preceding them with '\'. (So, to erase '\', you need two '#s').

The 'break' or 'interrupt' key causes an interrupt signal, as does the ASCII 'delete' (or 'rubout') character, which is not passed to programs. This signal generally causes whatever program you
are running to terminate. It is typically used to stop a long printout that you don't want. However, programs can arrange either to ignore this signal altogether, or to be notified when it happens (instead of being terminated). The editor, for example, catches interrupts and stops what it is doing, instead of terminating, so that an interrupt can be used to halt an editor printout without losing the file being edited. Many users change this interrupt character to be "C (control-C) using stty(1).

It is also possible to suspend output temporarily using "S (control-s) and later resume output with "Q. In a newer terminal driver, it is possible to cause output to be thrown away without interrupting the program by typing "O; see tty(4).

The quit signal is generated by typing the ASCII FS character. (FS appears many places on different terminals, most commonly as control-\ or control-\.) It not only causes a running program to terminate but also generates a file with the core image of the terminated process. Quit is useful for debugging.

Besides adapting to the speed of the terminal, UNIX tries to be intelligent about whether you have a terminal with the newline function or whether it must be simulated with carriage-return and line-feed. In the latter case, all input carriage returns are turned to newline characters (the standard line delimiter) and both a carriage return and a line feed are echoed to the terminal. If you get into the wrong mode, the reset(1) command will rescue you.

Tab characters are used freely in UNIX source programs. If your terminal does not have the tab function, you can arrange to have them turned into spaces during output, and echoed as spaces during input. The system assumes that tabs are set every eight columns. Again, the tset(1) or stty(1) command will set or reset this mode. Tset(1) can be used to set the tab stops automatically when necessary.

How to run a program; the shells. When you have successfully logged in, a program called a shell is listening to your terminal. The shell reads typed-in lines, splits them up into a command name and arguments, and executes the command. A command is simply an executable program. The Shell looks in several system directories to find the command. You can also place commands in your own directory and have the shell find them there. There is nothing special about system-provided commands except that they are kept in a directory where the shell can find them.

The command name is always the first word on an input line; it and its arguments are separated from one another by spaces.

When a program terminates, the shell will ordinarily regain control and type a prompt at you to indicate that it is ready for another command.

The shells have many other capabilities, which are described in detail in sections sh(1) and csh(1). If the shell prompts you with "$", then it is an instance of sh(1) the standard Bell-labs provided shell. If it prompts with '%' then it is an instance of csh(1), a shell written at Berkeley. The shells are different for all but the most simple terminal usage. Most users at Berkeley choose csh(1) because of the history mechanism and the alias feature, which greatly enhance its power when used interactively. Csh also supports the job-control facilities; see csh(1) or the Csh introduction in volume 2C for details.

You can change from one shell to the other by using the chsh(1) command, which takes effect at your next login.

The current directory. UNIX has a file system arranged in a hierarchy of directories. When the system administrator gave you a user name, he also created a directory for you (ordinarily with the same name as your user name). When you log in, any file name you type is by default in this directory. Since you are the owner of this directory, you have full permission to read, write, alter, or destroy its contents. Permissions to have your will with other directories and files will have been granted or denied to you by their owners. As a matter of observed fact, few UNIX users protect their files from perusal by other users.
To change the current directory (but not the set of permissions you were endowed with at login) use cd(1).

Path names. To refer to files not in the current directory, you must use a path name. Full path names begin with ‘/’, the name of the root directory of the whole file system. After the slash comes the name of each directory containing the next sub-directory (followed by a ‘/’) until finally the file name is reached. For example, /usr/lex refers to the file lex in the directory lex; lex is itself a subdirectory of us; us springs directly from the root directory.

If your current directory has subdirectories, the path names of files therein begin with the name of the subdirectory with no prefixed ‘/’.

A path name may be used anywhere a file name is required.

Important commands which modify the contents of files are cp(1), mv(1), and rm(1), which respectively copy, move (i.e. rename) and remove files. To find out the status of files or directories, use Is(1). See mkdir(1) for making directories and rmdir (in rm(1)) for destroying them.

For a fuller discussion of the file system, see ‘The UNIX Time-Sharing System,’ by Ken Thompson and Dennis Ritchie. It may also be useful to glance through section 2 of this manual, which discusses system calls, even if you don’t intend to deal with the system at that level.

Writing a program. To enter the text of a source program into a UNIX file, use the editor ex(1) or its display editing alias vi(1). (The old standard editor ed(1) is also available.) The principal languages in UNIX are provided by the C compiler cc(1), the Fortran compiler ft(1), the Pascal compiler pc(1), and interpreter pr(1) and px(1), and the Lisp system lisp(1). User contributed software in the latest release of the system supports APL, the Functional Programming language, and Icon. Refer to apl(1), fp(1), and icon(1), respectively for more information about each. After the program text has been entered through the editor and written on a file, you can give the file to the appropriate language processor as an argument. The output of the language processor will be left on a file in the current directory named ‘a.out’. (If the output is precious, use mv to move it to a less exposed name soon.)

When you have finally gone through this entire process without provoking any diagnostics, the resulting program can be run by giving its name to the shell (‘$’ or ‘%’) prompt.

Your programs can receive arguments from the command line just as system programs do, see execve(2).

Text processing. Almost all text is entered through the editor ex(1) (often entered via vi(1)). The commands most often used to write text on a terminal are: cat, pr, more and nroff, all in section 1.

The cat command simply dumps ASCII text on the terminal, with no processing at all. The pr command paginates the text, supplies headings, and has a facility for multi-column output. Nroff is an elaborate text formatting program. Used naked, it requires careful forethought, but for ordinary documents it has been tamed; see me(7) and ms(7).

Troff prepares documents for a Graphics Systems phototypesetter or a Versatec Plotter; it is very similar to nroff, and often works from exactly the same source text. It was used to produce this manual.

Script(1) lets you keep a record of your session in a file, which can then be printed, mailed, etc. It provides the advantages of a hard-copy terminal even when using a display terminal.

More(1) is useful for preventing the output of a command from zipping off the top of your screen. It is also well suited to perusing files.

Status inquiries. Various commands exist to provide you with useful information. w(1) prints a list of users presently logged in, and what they are doing. date(1) prints the current time and date. ls(1) will list the files in your directory or give summary information about particular
files.

Surprises. Certain commands provide inter-user communication. Even if you do not plan to use them, it would be well to learn something about them, because someone else may aim them at you.

To communicate with another user currently logged in, `write(1)` is used; `mail(1)` will leave a message whose presence will be announced to another user when he next logs in. The write-ups in the manual also suggest how to respond to the two commands if you are a target.

If you use `csh(1)` the key `Z` (control-Z) will cause jobs to "stop". If this happens before you learn about it, you can simply continue by saying "fg" (for foreground) to bring the job back.

When you log in, a message-of-the-day may greet you before the first prompt.

CONVERTING FROM THE 6TH EDITION

There follows a catalogue of significant, mostly incompatible, changes that will affect old users converting from the sixth edition on a PDP-11. No attempt is made to list all new facilities, or even all minor, but easily spotted changes, just the bare essentials without which it will be almost impossible to do anything.

Addressing files. Byte addresses in files are now long (32-bit) integers. Accordingly `seek` has been replaced by `lseek(2)`. Every program that contains a `seek` must be modified. `Stat` and `fstat(2)` have been affected similarly, since file lengths are now 32- rather than 24-bit quantities.

Assembly language. This language is dead. Necromancy will be severely punished.

Stty and gtty. These system calls have been extensively altered, see `ioctl(2)` and `tty(4)`.

C language, lint. The syntax for initialization requires an equal sign `=` before an initializer, and brackets `[]` around compound initial values; arrays and structures are now initialized honestly. Assignment operators such as `+=` and `=-` are now written in the reverse order: `+=`, `-=`. This removes the possibility of ambiguity in constructs such as `x-=2`, `y=*p`, and `a=-*/b`. You will also certainly want to learn about

- long integers
- type definitions
- casts (for type conversion)
- unions (for more honest storage sharing)
- `#include <filename>` (which searches in standard places)

The program `lint(1)` checks for obsolete syntax and does strong type checking of C programs, singly or in groups that are expected to be loaded together. It is indispensable for conversion work.

Fortran. The old `fc` is replaced by `f77`, a true compiler for Fortran 77, compatible with C. There are substantial changes in the language; see ‘A Portable Fortran 77 Compiler’ in Volume 2.

Stream editor. The program `sed(1)` is adapted to massive, repetitive editing jobs of the sort encountered in converting to the new system. It is well worth learning.

Standard I/O. The old `fopen`, `getc`, `putc` complex and the old `-lp` package are both dead, and even `getchar` has changed. All have been replaced by the clean, highly efficient, `stdio` package, `intro(3S)`. The first things to know are that `getchar(3)` returns the integer `EOF` (-1) (which is not a possible byte value) on end of file, that 518-byte buffers are out, and that there is a defined FILE data type.

Make. The program `make(1)` handles the recompilation and loading of software in an orderly way from a ‘makefile’ recipe given for each piece of software. It remakes only as much as the modification dates of the input files show is necessary. The makefiles will guide you in building your new system.
Shell, chdir. F. L. Bauer once said Algol 68 is the Everest that must be climbed by every computer scientist because it is there. So it is with the shell for UNIX users. Everything beyond simple command invocation from a terminal is different. Even chdir is now spelled cd. If you wish to use sh (as opposed to csh) then you will want to study sh(1) long and hard.

C shell. Csh(1), developed at Berkeley, has features comparable to sh. It includes a history mechanism that saves you from retyping all or part of previous commands, as well as an efficient aliasing (macro) mechanism. The job control facilities of the system, which make the system much more pleasant to use, are currently available only with csh. See csh(1) for a description. These features make csh pleasant to use interactively. Csh programs have a syntax reminiscent of C, while sh command programs have a syntax reminiscent of ALGOL-68.

Debugging. Sdb is a far more capable replacement for the debugger cdb, and debugs C and Fortran at the source level. For machine language debugging, adb replaces db. The first-time user should be especially careful about distinguishing / and ? in adb commands, and watching to make sure that the x whose value he asked for is the real x, and not just some absolute location equal to the stack offset of some automatic x. You can always use the ‘true’ name, _x, to pin down a C external variable.

Dsw. This little-known, but indispensable facility has been taken over by rm -ri.

Boot procedures. Needless to say, these are all different. See section 8 of this volume, and the other documentation you should have received with your tape.

CONVERTING FROM THE DECEMBER, 1979 BERKELEY DISTRIBUTION

There have been a number of significant changes and improvements in the system. This list just gives the bare essentials:

C language changes. The C compiler now accepts and checks essentially arbitrary length identifiers and preprocessor names. There is a new type available in type casts: void which signifies that a value is to be ignored. It is useful in keeping lint happy about values which are not used (especially values returned from procedures). Finally, the language has been changed so that field names need not be unique to structures; on the other hand, the compiler insists that you be more honest about types involved in pointer constructs or it will warn you.

Object file format. The object file format has been changed to include a string table, so that language compilers may have names longer than 8 characters in their resulting a.out files. Old .o files must be recreated. A.out files will still run on both this and the December 1979 version of the system; only the symbol tables are incompatible.

Archive format and table of contents. The archive format has been changed to one which is portable between the VAX and other machines (e.g. the PDP-11). Old VAX archives should be converted with arcv(8); loader archives should just be recreated since the object files are also obsolete. Loader archives should have table-of-contents added by ranlib(1); if they dont the loader will gripe when they are used.

New tty driver, job control facilities and csh. Hand in hand are new job control facilities, a new tty driver and a new version of the C shell which supports and uses all of this. See tty(4) and csh(1) for a quick introduction.

Pascal compiler. There is a true Pascal compiler, pc(1) which allows separate compilation as well as mixing in of FORTRAN and C code.

Error analyzer. There is an error analyzer program error(1), which takes a set of error message and merges them back into the source files at the point of error. It can be used interactively to avoid inserting errors which are uninteresting. This program eliminates once and for all making lists of errors on small scraps of paper.

Mail forwarding. The system now provides mail forwarding and distribution facilities. Group and aliases are defined in the file /usr/lib/aliases see aliases(5). If you change this file you will have to rerun newaliases(1). For any particular system a table in the source of the delivermail postman program may have to be changed so that it knows about the gateways on the local
machine.

System bootstrap procedures. These are totally changed; the system performs automatic reboots and preens the disks automatically at reboot. You should reread the appropriate pages in section 8 if you deal with system reboots.

CONVERTING FROM THE JUNE, 1981 BERKELEY DISTRIBUTION

Many many changes have been made. This list indicates those which are most visible to users.

Directory format. Directory entries are no longer fixed length. This forces user programs which read directories to be modified to use the directory(3) package.

Signals. A new signal package has replaced the previous signal mechanism as well as the “jobs library”. When using the compatible signal(3C) interface routine, the two most important changes are: signal handlers are not reset to SIG_DFL when a process receives a signal, and while a signal handler is processing a signal, that signal is blocked until the handler returns. This has implications, in particular, for programs which process the suspend character typed at the terminal. Refer to sigvec, sigblock, sigpause, sigstack, and sigsetmask(2) for information about the new signal facilities.

File and path names. File names may now be up to 255 characters in length. Path names are restricted to be at most 1024 characters. These two constants are provided as MAXNAMLEN and MAXPATHLEN in <sysdir.h> and <sys/param.h>, respectively.

System time. System time is provided in microsecond precision with 10 millisecond accuracy. The new system call gettimeofday(2) supplants the old time(3) call which is now a library routine. The major impact of this change is that programs are now written in a fashion which is independent of the line clock frequency.

Groups. A user may now be in many groups simultaneously. This has obviated the need for the newgrp command. See getgroups(2) for more information.

Stat and fstat return value. The structure returned by the stat and fstat system calls is now larger. This is due to inode numbers growing to 32-bits, time stamps expanding to 64-bits and other information being included in the return value. Consult stat(2) for more information.

Mail forwarding. The system now provides general internetwork mail forwarding and distribution facilities. The sendmail(8) program replaces the old delivermail facility.

Debuggers. The previous C source language debugger, sdb, has been replaced by a new one, dbx(1). adb(1) has been extended to simplify debugging of the operating system.

Networking support. Many new user programs provide access to the networking facilities. The rlogin(1C) and rsh(1C) programs are intended for communicating between UNIX systems. The telnet(1C) and ftp(1C) programs support the DARPA Internet standard protocols. The netstat(1) program is useful in watching network activity.
# TABLE OF CONTENTS

1. Commands and Application Programs

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>introduction to commands</td>
</tr>
<tr>
<td>adb</td>
<td>debugger</td>
</tr>
<tr>
<td>addbib</td>
<td>create or extend bibliographic database</td>
</tr>
<tr>
<td>allusers</td>
<td>print list of all authorized users</td>
</tr>
<tr>
<td>altolode</td>
<td>load files from an Alto F/TP</td>
</tr>
<tr>
<td>ansi</td>
<td>read and write ANSI format magnetic tapes</td>
</tr>
<tr>
<td>apl</td>
<td>apl interpreter</td>
</tr>
<tr>
<td>apply</td>
<td>apply a command to a set of arguments</td>
</tr>
<tr>
<td>apropos</td>
<td>locate commands by keyword lookup</td>
</tr>
<tr>
<td>ar</td>
<td>archive and library maintainer</td>
</tr>
<tr>
<td>arptab</td>
<td>show contents of kernel ARP table</td>
</tr>
<tr>
<td>as</td>
<td>VAX-11 assembler</td>
</tr>
<tr>
<td>as68</td>
<td>.a68 -&gt; .b assembler component of cc68</td>
</tr>
<tr>
<td>at</td>
<td>execute commands at a later time</td>
</tr>
<tr>
<td>awk</td>
<td>pattern scanning and processing language</td>
</tr>
<tr>
<td>backup</td>
<td>make a backup version copy of a file</td>
</tr>
<tr>
<td>basename</td>
<td>strip filename affixes</td>
</tr>
<tr>
<td>bboard</td>
<td>bulletin board reading program</td>
</tr>
<tr>
<td>bc</td>
<td>arbitrary-precision arithmetic language</td>
</tr>
<tr>
<td>bibtex</td>
<td>make a LaTeX bibliography</td>
</tr>
<tr>
<td>biff</td>
<td>be notified if mail arrives and who it is from</td>
</tr>
<tr>
<td>binmail</td>
<td>send or receive mail among users</td>
</tr>
<tr>
<td>boise</td>
<td>send files to the HP2680a printer using TCP</td>
</tr>
<tr>
<td>btroff</td>
<td>troff to the ImPrint printer</td>
</tr>
<tr>
<td>buildmake</td>
<td>preprocessor to provide extended syntax for makefiles</td>
</tr>
<tr>
<td>cal</td>
<td>print calendar</td>
</tr>
<tr>
<td>calen</td>
<td>print large-format calendar</td>
</tr>
<tr>
<td>calendar</td>
<td>reminder service</td>
</tr>
<tr>
<td>cat</td>
<td>catenate and print</td>
</tr>
<tr>
<td>catboise</td>
<td>convert C/A/T files to DVI format and print on Boise</td>
</tr>
<tr>
<td>catdvi</td>
<td>convert C/A/T files to DVI format</td>
</tr>
<tr>
<td>cb</td>
<td>C program beautifier</td>
</tr>
<tr>
<td>cc</td>
<td>C compiler</td>
</tr>
<tr>
<td>cc68</td>
<td>C compiler for the MC68000</td>
</tr>
<tr>
<td>ccom68</td>
<td>.c -&gt; .s translator component of cc68</td>
</tr>
<tr>
<td>cd</td>
<td>change working directory</td>
</tr>
<tr>
<td>checknr</td>
<td>check nroff/troff files</td>
</tr>
<tr>
<td>chfn</td>
<td>change finger entry</td>
</tr>
<tr>
<td>chgrp</td>
<td>change group</td>
</tr>
<tr>
<td>chmod</td>
<td>change mode</td>
</tr>
<tr>
<td>chsh</td>
<td>change default login shell</td>
</tr>
<tr>
<td>ci</td>
<td>check in RCS revisions</td>
</tr>
<tr>
<td>clear</td>
<td>clear terminal screen</td>
</tr>
<tr>
<td>cmp</td>
<td>compare two files</td>
</tr>
<tr>
<td>cnest</td>
<td>check for nested comments in C code</td>
</tr>
<tr>
<td>co</td>
<td>check out RCS revisions</td>
</tr>
<tr>
<td>col</td>
<td>filter reverse line feeds</td>
</tr>
<tr>
<td>colcrt</td>
<td>filter nroff output for CRT previewing</td>
</tr>
<tr>
<td>colrm</td>
<td>remove columns from a file</td>
</tr>
<tr>
<td>comm</td>
<td>select or reject lines common to two sorted files</td>
</tr>
<tr>
<td>compact</td>
<td>compress and uncompress files, and cat them</td>
</tr>
<tr>
<td>congraph</td>
<td>plot connectivity of a graph</td>
</tr>
<tr>
<td>cp</td>
<td>copy</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>cp</td>
<td>cp - add parentheses to C expressions</td>
</tr>
<tr>
<td>cref</td>
<td>cross reference program</td>
</tr>
<tr>
<td>crypt</td>
<td>encode/decode</td>
</tr>
<tr>
<td>csh</td>
<td>a shell (command interpreter) with C-like syntax</td>
</tr>
<tr>
<td>ctags</td>
<td>create a tags file</td>
</tr>
<tr>
<td>cxref</td>
<td>cross reference C source files</td>
</tr>
<tr>
<td>cz</td>
<td>convert files to press format and print them on the Dover.</td>
</tr>
<tr>
<td>dbaio</td>
<td>load the data i/o prom programmer</td>
</tr>
<tr>
<td>date</td>
<td>print and set the date</td>
</tr>
<tr>
<td>st</td>
<td>list contents of an Emacs data base</td>
</tr>
<tr>
<td>dbx</td>
<td>print an entry from an Emacs database</td>
</tr>
<tr>
<td>dc</td>
<td>desk calculator</td>
</tr>
<tr>
<td>deat</td>
<td>convert troff phototypesetter output files to press format and print them on the Dover.</td>
</tr>
<tr>
<td>dd</td>
<td>convert and copy a file</td>
</tr>
<tr>
<td>ddt68</td>
<td>symbolic debugger for 68000</td>
</tr>
<tr>
<td>deroft</td>
<td>remove nroff, troff, tbl and eqn constructs</td>
</tr>
<tr>
<td>detex</td>
<td>remove TeX constructs</td>
</tr>
<tr>
<td>df</td>
<td>disk free</td>
</tr>
<tr>
<td>diction</td>
<td>print wordy sentences; thesaurus for diction</td>
</tr>
<tr>
<td>diff</td>
<td>differential file and directory comparator</td>
</tr>
<tr>
<td>diff3</td>
<td>3-way differential file comparison</td>
</tr>
<tr>
<td>d168</td>
<td>b.out -&gt; .dl downloader component of c68</td>
</tr>
<tr>
<td>dx</td>
<td>download with error correction 68000 Sun1 monitor</td>
</tr>
<tr>
<td>dpq</td>
<td>prints the Dover printer queue</td>
</tr>
<tr>
<td>dpr</td>
<td>dover printer spooler</td>
</tr>
<tr>
<td>dprm</td>
<td>remove a file from the Dover printer queue</td>
</tr>
<tr>
<td>dtreec</td>
<td>print directory tree structures</td>
</tr>
<tr>
<td>dtroff</td>
<td>troff to the Dover</td>
</tr>
<tr>
<td>du</td>
<td>summarize disk usage</td>
</tr>
<tr>
<td>dumpfonts</td>
<td>show what Press fonts are available in fonts,widths</td>
</tr>
<tr>
<td>dviboise</td>
<td>send DVI files to the HP2680a printer using TCP</td>
</tr>
<tr>
<td>dviimp</td>
<td>convert DVI files to impress format</td>
</tr>
<tr>
<td>dvip</td>
<td>convert a dvi (TeX output) file to press format.</td>
</tr>
<tr>
<td>dviypress</td>
<td>convert dvi (TEX output) files to press format and print them on the Dover.</td>
</tr>
<tr>
<td>echo</td>
<td>echo arguments</td>
</tr>
<tr>
<td>ed</td>
<td>text editor</td>
</tr>
<tr>
<td>elf</td>
<td>Extended Fortran Language</td>
</tr>
<tr>
<td>eftpcrec</td>
<td>receive-only PUP/ElFTP file transfer program with routing</td>
</tr>
<tr>
<td>eftpsend</td>
<td>send-only PUP/ElFTP file transfer program with routing</td>
</tr>
<tr>
<td>emacs</td>
<td>a screen editor</td>
</tr>
<tr>
<td>eqn</td>
<td>typeset mathematics</td>
</tr>
<tr>
<td>error</td>
<td>analyze and disperse compiler error messages</td>
</tr>
<tr>
<td>etherport</td>
<td>show status of ethernet minor devices</td>
</tr>
<tr>
<td>ex</td>
<td>text editor</td>
</tr>
<tr>
<td>exlog</td>
<td>extract data from system load log file</td>
</tr>
<tr>
<td>expand</td>
<td>expand tabs to spaces, and vice versa</td>
</tr>
<tr>
<td>explain</td>
<td>explain, diction - print wordy sentences; thesaurus for diction</td>
</tr>
<tr>
<td>expr</td>
<td>evaluate arguments as an expression</td>
</tr>
<tr>
<td>cyacc</td>
<td>modified yacc allowing much improved error recovery</td>
</tr>
<tr>
<td>f77</td>
<td>Fortran 77 compiler</td>
</tr>
<tr>
<td>false</td>
<td>provide truth values</td>
</tr>
<tr>
<td>fed</td>
<td>font editor</td>
</tr>
<tr>
<td>file</td>
<td>determine file type</td>
</tr>
<tr>
<td>find</td>
<td>find files</td>
</tr>
<tr>
<td>fing</td>
<td>front end for finger</td>
</tr>
<tr>
<td>finger</td>
<td>user information lookup program</td>
</tr>
</tbody>
</table>
Table of Contents

fmt .......................................................... simple text formatter
fold .......................................................... fold long lines for finite width output device
fp ........................................................ Functional Programming language compiler/interpreter
fpr ........................................................ print Fortran file
from ......................................................... who is my mail from?
fsplit ....................................................... split a multi-routine Fortran file into individual files
ftp ........................................................ file transfer program
gcore ....................................................... get core images of running processes
gprof ....................................................... display call graph profile data
graph ......................................................... draw a graph
grep ......................................................... search a file for a pattern
gripe ......................................................... mail a local system bug report
groups ...................................................... show group memberships
head ........................................................ give first few lines
host ........................................................ print IP host names and addresses
hostid ....................................................... set or print identifier of current host system
hostname ................................................ set or print name of current host system
ident ....................................................... identify files
imprint ..................................................... imprint - print text files on Imprint-10
include .................................................... search for and print header (include) files
indent ...................................................... indent and format C program source
ingroup ..................................................... show membership in a specified group
install ..................................................... install binaries
iostat ....................................................... report I/O statistics
iphostid .................................................... set or print Internet Protocol (IP) identifier of current host
iprint ....................................................... iprint - convert text files to DVI format
itroff ...................................................... troff to the ImPrint printer
join ........................................................ relational database operator
kill ........................................................ terminate a process with extreme prejudice
last ........................................................ indicate last logins of users and teletypes
lastcomm ..................................................... show last commands executed in reverse order
latex ....................................................... TeX with a macro package preloaded
id .............................................................. link editor
ld68 ........................................................ .b -> b.out linker for the MC68000
learn ....................................................... computer aided instruction about UNIX
leave ........................................................ remind you when you have to leave
lex ............................................................ generator of lexical analysis programs
linelen ..................................................... print line lengths for a text file
lint ........................................................ a C program verifier
lisp .......................................................... lisp interpreter
liszt ........................................................ compile a Franz Lisp program
ln .............................................................. make links
loadavg ..................................................... average load log data on a weekly basis
loadlog ..................................................... log the current time, number of users, and load average
locate ...................................................... location and owner of Pup network hosts
lock ........................................................ reserve a terminal
login ........................................................ sign on
look ........................................................ find lines in a sorted list
lookbib ..................................................... build inverted index for a bibliography, find references in a bibliography
lorder ....................................................... find ordering relation for an object library
lorder68 ..................................................... find ordering relation for an MC68000 object library
lower ........................................................ lower the case of a filename
lpq .......................................................... spool queue examination program
lpr .......................................................... off line print
lprm ........................................................ remove jobs from the line printer spooling queue
ls ............................................................ list contents of directory
pxref .................................................. Pascal cross-reference program
quota .......................................................... display disc usage and limits
ranlib .......................................................... convert archives to random libraries
ratfor .......................................................... rational Fortran dialect
rcp .......................................................... remote file copy
rcs .......................................................... change RCS file attributes
rcsdiff .......................................................... compare RCS revisions
resintro .......................................................... resintro - introduction to RCS commands
rcsmerge .......................................................... merge RCS revisions
rdist .......................................................... remote file distribution program
refer .......................................................... find and insert literature references in documents
remote .......................................................... Remote command execution
reset .......................................................... reset the teletype bits to a sensible state
rev .......................................................... reverse lines of a file
rev68 .......................................................... reverse byte order in 68000 .b and .68 (b.out) files
rl68 .......................................................... print relocation commands in a .b file for the 68000
rlog .......................................................... print log messages and other information about RCS files
rlogin .......................................................... remote login
rm .......................................................... remove (unlink) files or directories
rmail .......................................................... handle remote mail received via uucp
rmdir .......................................................... remove (unlink) directories or files
roffbib .......................................................... run off bibliographic database
rsh .......................................................... remote shell
rtar .......................................................... remote tape manipulation programs
rftime .......................................................... show host status of local machines
rwho .......................................................... who's logged in on local machines
screen .......................................................... repeatedly display output of command on terminal screen
script .......................................................... make typescript of terminal session
sed .......................................................... stream editor
sendbug .......................................................... mail a system bug report to 4bsd-bugs
sh .......................................................... command language
shar .......................................................... produce shell-script archives
size .......................................................... size of an object file
size68 .......................................................... prints sizes of segments in a .b or .68 file
sleep .......................................................... suspend execution for an interval
soclim .......................................................... eliminate .so's from mroff input
sort .......................................................... sort or merge files
sortbib .......................................................... sort bibliographic database
spell .......................................................... find spelling errors
spline .......................................................... interpolate smooth curve
split .......................................................... split a file into pieces
strings .......................................................... find the printable strings in a object, or other binary, file
strip .......................................................... remove symbols and relocation bits
struct .......................................................... structure Fortran programs
sty .......................................................... set terminal options
style .......................................................... analyze surface characteristics of a document
su .......................................................... substitute user id temporarily
sum .......................................................... sum and count blocks in a file
symchk .......................................................... check for bad symbolic links
symorder .......................................................... rearrange name list
sysline .......................................................... display system status on status line of a terminal
tabs .......................................................... set terminal tabs
tail .......................................................... deliver the last part of a file
talk .......................................................... talk to another user	angle .......................................................... convert web file into pascal file, tex file
tar .......................................................... tape archiver
# Table of Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tbl</td>
<td>format tables for nroff or troff</td>
</tr>
<tr>
<td>tc</td>
<td>phototypesetter simulator</td>
</tr>
<tr>
<td>tee</td>
<td>pipe fitting</td>
</tr>
<tr>
<td>telnet</td>
<td>user interface to the TELNET protocol</td>
</tr>
<tr>
<td>test</td>
<td>condition command</td>
</tr>
<tr>
<td>texp</td>
<td>text formatting and typesetting</td>
</tr>
<tr>
<td>tftp</td>
<td>trivial file transfer protocol</td>
</tr>
<tr>
<td>time</td>
<td>time a command</td>
</tr>
<tr>
<td>timecheck</td>
<td>checks and sets Pup network time</td>
</tr>
<tr>
<td>tip</td>
<td>connect to a remote system</td>
</tr>
<tr>
<td>tk</td>
<td>pagination for the Tektronix 4014</td>
</tr>
<tr>
<td>top</td>
<td>display and update information about the top cpu processes</td>
</tr>
<tr>
<td>touch</td>
<td>update date last modified of a file</td>
</tr>
<tr>
<td>tr</td>
<td>manipulate tape archive</td>
</tr>
<tr>
<td>trman</td>
<td>translate version 6 manual macros to version 7 macros</td>
</tr>
<tr>
<td>troff</td>
<td>translate version 6 manual macros to version 7 macros</td>
</tr>
<tr>
<td>true</td>
<td>text formatting and typesetting</td>
</tr>
<tr>
<td>tset</td>
<td>provide truth values</td>
</tr>
<tr>
<td>sort</td>
<td>terminal dependent initialization</td>
</tr>
<tr>
<td>ttime</td>
<td>topological sort</td>
</tr>
<tr>
<td>tty</td>
<td>measure terminal output rate</td>
</tr>
<tr>
<td>ul</td>
<td>get terminal name</td>
</tr>
<tr>
<td>undump</td>
<td>do underlining</td>
</tr>
<tr>
<td>unifdef</td>
<td>convert a core dump to an executable a.out file</td>
</tr>
<tr>
<td>uniq</td>
<td>remove ifdef'ed lines</td>
</tr>
<tr>
<td>units</td>
<td>report repeated lines in a file</td>
</tr>
<tr>
<td>unpcnt</td>
<td>conversion program</td>
</tr>
<tr>
<td>unscribe</td>
<td>remove lines beginning with % from a file</td>
</tr>
<tr>
<td>uptime</td>
<td>remove Scribe constructs</td>
</tr>
<tr>
<td>users</td>
<td>show how long system has been up</td>
</tr>
<tr>
<td>uucp</td>
<td>compact list of users who are on the system</td>
</tr>
<tr>
<td>uuencode</td>
<td>encode/decode a binary file for transmission via mail</td>
</tr>
<tr>
<td>usend</td>
<td>send a file to a remote host</td>
</tr>
<tr>
<td>uux</td>
<td>unix to unix command execution</td>
</tr>
<tr>
<td>v6c</td>
<td>version changing program for Pascal sources</td>
</tr>
<tr>
<td>vfontinfo</td>
<td>inspect and print out information about UNIX fonts</td>
</tr>
<tr>
<td>vgrind</td>
<td>grind nice listings of programs</td>
</tr>
<tr>
<td>vi</td>
<td>screen oriented (visual) display editor based on ex</td>
</tr>
<tr>
<td>vlp</td>
<td>Format Lisp programs to be printed with troff, vtroff, or troff</td>
</tr>
<tr>
<td>vmstat</td>
<td>report virtual memory statistics</td>
</tr>
<tr>
<td>vnews</td>
<td>read news articles</td>
</tr>
<tr>
<td>vpr</td>
<td>raster printer/plotter spooler</td>
</tr>
<tr>
<td>vtroff</td>
<td>troff to a raster plotter</td>
</tr>
<tr>
<td>vwidth</td>
<td>make troff width table for a font</td>
</tr>
<tr>
<td>w</td>
<td>who is on and what they are doing</td>
</tr>
<tr>
<td>wait</td>
<td>await completion of process</td>
</tr>
<tr>
<td>wall</td>
<td>write to all users</td>
</tr>
<tr>
<td>wc</td>
<td>word count</td>
</tr>
<tr>
<td>what</td>
<td>show what versions of object modules were used to construct a file</td>
</tr>
<tr>
<td>whatis</td>
<td>describe what a command is</td>
</tr>
<tr>
<td>whereami</td>
<td>report name of terminal</td>
</tr>
<tr>
<td>whereis</td>
<td>locate source, binary, and or manual for program</td>
</tr>
<tr>
<td>which</td>
<td>locate a program file including aliases and paths (csh only)</td>
</tr>
<tr>
<td>who</td>
<td>who is on the system</td>
</tr>
<tr>
<td>whoami</td>
<td>print effective current user id</td>
</tr>
</tbody>
</table>
2. System Calls

- intro: introduction to system calls and error numbers
- accept: accept a connection on a socket
- access: determine accessibility of file
- acct: turn accounting on or off
- bind: bind a name to a socket
- brk: change data segment size
- chdir: change current working directory
- chmod: change mode of file
- chown: change owner and group of a file
- chroot: change root directory
- close: delete a descriptor
- connect: initiate a connection on a socket
- creat: create a new file
- dup: duplicate a descriptor
- execl: execute a file
- exit: terminate a process
- fcntl: file control
- flock: apply or remove an advisory lock on an open file
- fork: create a new process
- fsync: synchronize a file's in-core state with that on disk
- getdtablesize: get descriptor table size
- getegid: get group identity
- getgroups: get group access list
- gethostname: get/set unique identifier of current host
- gethostid: get/set name of current host
- getitimer: get/set value of interval timer
- getpagesize: get system page size
- getpeername: get name of connected peer
- getpgid: get process group
- getpid: get process identification
- getpriority: get/set program scheduling priority
- getrlimit: control maximum system resource consumption
- getrusage: get information about resource utilization
- getsockname: get socket name
- getsockopt: get and set options on sockets
- gettimeofday: get/set date and time
- getuid: get user identity
- ioctl: control device
- kill: send signal to a process
- killpg: send signal to a process group
- link: make a hard link to a file
- listen: listen for connections on a socket
- lseek: move read/write pointer
- mkdir: make a directory file
- mkstemp: make a special file
- mount: mount or remove file system
- open: open a file for reading or writing, or create a new file

whois: ask the ARPA Internet NIC about a user
write: write to another user
xsend: secret mail
xstr: extract strings from C programs to implement shared strings
yacc: yet another compiler-compiler
yapp: yet another pretty printer
yes: be repetitively affirmative
### Table of Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pipe</td>
<td>create an interprocess communication channel</td>
</tr>
<tr>
<td>profil</td>
<td>execution time profile</td>
</tr>
<tr>
<td>ptrace</td>
<td>process trace</td>
</tr>
<tr>
<td>quota</td>
<td>manipulate disk quotas</td>
</tr>
<tr>
<td>read</td>
<td>read input</td>
</tr>
<tr>
<td>readlink</td>
<td>read value of a symbolic link</td>
</tr>
<tr>
<td>reboot</td>
<td>reboot system or halt processor</td>
</tr>
<tr>
<td>recv</td>
<td>receive a message from a socket</td>
</tr>
<tr>
<td>rename</td>
<td>change the name of a file</td>
</tr>
<tr>
<td>rmdir</td>
<td>remove a directory file</td>
</tr>
<tr>
<td>select</td>
<td>synchronous I/O multiplexing</td>
</tr>
<tr>
<td>send</td>
<td>send a message from a socket</td>
</tr>
<tr>
<td>setgroups</td>
<td>set group access list</td>
</tr>
<tr>
<td>setpgid</td>
<td>set process group</td>
</tr>
<tr>
<td>setquota</td>
<td>enable/disable quotas on a file system</td>
</tr>
<tr>
<td>setregid</td>
<td>set real and effective group ID</td>
</tr>
<tr>
<td>setreuid</td>
<td>set real and effective user ID's</td>
</tr>
<tr>
<td>shutdown</td>
<td>shut down part of a full-duplex connection</td>
</tr>
<tr>
<td>sigblock</td>
<td>block signals</td>
</tr>
<tr>
<td>sigpause</td>
<td>atomically release blocked signals and wait for interrupt</td>
</tr>
<tr>
<td>sigsetmask</td>
<td>set current signal mask</td>
</tr>
<tr>
<td>sigstack</td>
<td>set and/or get signal stack context</td>
</tr>
<tr>
<td>sigvec</td>
<td>software signal facilities</td>
</tr>
<tr>
<td>socket</td>
<td>create an endpoint for communication</td>
</tr>
<tr>
<td>socketpair</td>
<td>create a pair of connected sockets</td>
</tr>
<tr>
<td>stat</td>
<td>get file status</td>
</tr>
<tr>
<td>swapon</td>
<td>add a swap device for interleaved paging/swapping</td>
</tr>
<tr>
<td>symlink</td>
<td>make symbolic link to a file</td>
</tr>
<tr>
<td>sync</td>
<td>update super-block</td>
</tr>
<tr>
<td>syscall</td>
<td>indirect system call</td>
</tr>
<tr>
<td>truncate</td>
<td>truncate a file to a specified length</td>
</tr>
<tr>
<td>umask</td>
<td>set file creation mode mask</td>
</tr>
<tr>
<td>unlink</td>
<td>remove directory entry</td>
</tr>
<tr>
<td>utimes</td>
<td>set file times</td>
</tr>
<tr>
<td>vfork</td>
<td>spawn new process in a virtual memory efficient way</td>
</tr>
<tr>
<td>vhangup</td>
<td>virtually &quot;hangup&quot; the current control terminal</td>
</tr>
<tr>
<td>wait</td>
<td>wait for process to terminate</td>
</tr>
<tr>
<td>write</td>
<td>write on a file</td>
</tr>
</tbody>
</table>

### 3. C Library Subroutines

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>introduction to library functions</td>
</tr>
<tr>
<td>abort</td>
<td>generate a fault</td>
</tr>
<tr>
<td>abs</td>
<td>integer absolute value</td>
</tr>
<tr>
<td>atof</td>
<td>convert ASCII to numbers</td>
</tr>
<tr>
<td>bstring</td>
<td>bit and byte string operations</td>
</tr>
<tr>
<td>crypt</td>
<td>DES encryption</td>
</tr>
<tr>
<td>ctime</td>
<td>convert date and time to ASCII</td>
</tr>
<tr>
<td>ctype</td>
<td>character classification macros</td>
</tr>
<tr>
<td>directory</td>
<td>directory operations</td>
</tr>
<tr>
<td>ecvt</td>
<td>output conversion</td>
</tr>
<tr>
<td>end</td>
<td>last locations in program</td>
</tr>
<tr>
<td>except</td>
<td>C exception handling</td>
</tr>
<tr>
<td>exec</td>
<td>execute a file</td>
</tr>
<tr>
<td>exit</td>
<td>terminate a process after flushing any pending output</td>
</tr>
<tr>
<td>frexp</td>
<td>split into mantissa and exponent</td>
</tr>
<tr>
<td>getbanner</td>
<td>get system login banner string</td>
</tr>
</tbody>
</table>
Fortran Library

3F.

Table of Contents

getenv .............................................................. value for environment name
getgrent ............................................................ get group file entry
getlogin ............................................................. get login name
getpass ............................................................... read a password
getpwent ............................................................. get password file entry
getwd ................................................................. get current working directory pathname
insque ................................................................. insert/remove element from a queue
malloc ............................................................... memory allocator
mktemp ............................................................... make a unique file name
monitor .............................................................. prepare execution profile
nlist ................................................................. get entries from name list
 perror ............................................................. system error messages
popen ................................................................. initiate I/O to/from a process
psignal .............................................................. system signal messages
qsort ................................................................. quicker sort
random .............................................................. better random number generator; routines for changing generators
regex ............................................................... regular expression handler
scandir .............................................................. scan a directory
setjmp .............................................................. non-local goto
setuid .............................................................. set user and group ID
sleep ............................................................... suspend execution for interval
strmpfold ........................................................... case-folded string comparison operations
string ............................................................... string operations
swab ................................................................. swap bytes
syslog ............................................................... control system log
system .............................................................. issue a shell command
ttyname ............................................................ find name of a terminal
valloc ............................................................... aligned memory allocator
varargs .............................................................. variable argument list

3F. Fortran Library

intro ............................................................... introduction to FORTRAN library functions
abort ................................................................. terminate abruptly with memory image
access .............................................................. determine accessibility of a file
alarm ............................................................... execute a subroutine after a specified time
bessel .............................................................. of two kinds for integer orders
bit ................................................................. and, or, xor, not, rshift, lshift bitwise functions
chdir ............................................................... change default directory
chmod ............................................................... change mode of a file
ctime ............................................................... return elapsed execution time
exit ................................................................. terminate process with status
ftime ............................................................... return date and time in an ASCII string
fmin ............................................................... return extreme values
flush ............................................................... flush output to a logical unit
fork ............................................................... create a copy of this process
fseek ............................................................... reposition a file on a logical unit
getarg ............................................................... return command line arguments
getc ............................................................... get a character from a logical unit
getcwd ............................................................. get pathname of current working directory
getenv ............................................................. get value of environment variables
getlog ............................................................. get user's login name
getpid ............................................................. get process id
getuid ............................................................. get user or group ID of the caller
hostnm ............................................................ get name of current host
idate ............................................................... return date or time in numerical form
index .............................................................. tell about character objects
### Table of Contents

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ioinit</code></td>
<td>change f77 I/O initialization</td>
</tr>
<tr>
<td><code>kill</code></td>
<td>send a signal to a process</td>
</tr>
<tr>
<td><code>link</code></td>
<td>make a link to an existing file</td>
</tr>
<tr>
<td><code>loc</code></td>
<td>return the address of an object</td>
</tr>
<tr>
<td><code>long</code></td>
<td>integer object conversion</td>
</tr>
<tr>
<td><code>perror</code></td>
<td>get system error messages</td>
</tr>
<tr>
<td><code>putc</code></td>
<td>write a character to a fortran logical unit</td>
</tr>
<tr>
<td><code>qsort</code></td>
<td>quick sort</td>
</tr>
<tr>
<td><code>rand</code></td>
<td>return random values</td>
</tr>
<tr>
<td><code>rename</code></td>
<td>rename a file</td>
</tr>
<tr>
<td><code>signal</code></td>
<td>change the action for a signal</td>
</tr>
<tr>
<td><code>sleep</code></td>
<td>suspend execution for an interval</td>
</tr>
<tr>
<td><code>stat</code></td>
<td>get file status</td>
</tr>
<tr>
<td><code>system</code></td>
<td>execute a UNIX command</td>
</tr>
<tr>
<td><code>time</code></td>
<td>return system time</td>
</tr>
<tr>
<td><code>topen</code></td>
<td>f77 tape I/O</td>
</tr>
<tr>
<td><code>traper</code></td>
<td>trap arithmetic errors</td>
</tr>
<tr>
<td><code>trapov</code></td>
<td>trap and repair floating point overflow</td>
</tr>
<tr>
<td><code>trpfppe</code></td>
<td>trap and repair floating point faults</td>
</tr>
<tr>
<td><code>ttynam</code></td>
<td>find name of a terminal port</td>
</tr>
<tr>
<td><code>unlink</code></td>
<td>remove a directory entry</td>
</tr>
<tr>
<td><code>wait</code></td>
<td>wait for a process to terminate</td>
</tr>
</tbody>
</table>

### IV. Math Library

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>intro</code></td>
<td>introduction to mathematical library functions</td>
</tr>
<tr>
<td><code>exp</code></td>
<td>exponential, logarithm, power, square root</td>
</tr>
<tr>
<td><code>floor</code></td>
<td>absolute value, floor, ceiling functions</td>
</tr>
<tr>
<td><code>gamma</code></td>
<td>log gamma function</td>
</tr>
<tr>
<td><code>hypot</code></td>
<td>Euclidean distance</td>
</tr>
<tr>
<td><code>j0</code></td>
<td>bessel functions</td>
</tr>
<tr>
<td><code>sin</code></td>
<td>trigonometric functions</td>
</tr>
<tr>
<td><code>sinh</code></td>
<td>hyperbolic functions</td>
</tr>
</tbody>
</table>

### V. Internet Network Library

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>intro</code></td>
<td>introduction to network library functions</td>
</tr>
<tr>
<td><code>byteorder</code></td>
<td>convert values between host and network byte order</td>
</tr>
<tr>
<td><code>gethostent</code></td>
<td>get network host entry</td>
</tr>
<tr>
<td><code>getnetent</code></td>
<td>get network entry</td>
</tr>
<tr>
<td><code>getprotoent</code></td>
<td>get protocol entry</td>
</tr>
<tr>
<td><code>getservent</code></td>
<td>get service entry</td>
</tr>
<tr>
<td><code>inet</code></td>
<td>Internet address manipulation routines</td>
</tr>
</tbody>
</table>

### VI. C Standard I/O Library Subroutines

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>intro</code></td>
<td>standard buffered input/output package</td>
</tr>
<tr>
<td><code>fclose</code></td>
<td>close or flush a stream</td>
</tr>
<tr>
<td><code>ferror</code></td>
<td>stream status inquiries</td>
</tr>
<tr>
<td><code>fopen</code></td>
<td>open a stream</td>
</tr>
<tr>
<td><code>freuald</code></td>
<td>buffered binary input/output</td>
</tr>
<tr>
<td><code>fseek</code></td>
<td>reposition a stream</td>
</tr>
<tr>
<td><code>getc</code></td>
<td>get character or word from stream</td>
</tr>
<tr>
<td><code>gets</code></td>
<td>get a string from a stream</td>
</tr>
<tr>
<td><code>printf</code></td>
<td>formatted output conversion</td>
</tr>
<tr>
<td><code>putc</code></td>
<td>put character or word on a stream</td>
</tr>
<tr>
<td><code>puts</code></td>
<td>put a string on a stream</td>
</tr>
<tr>
<td><code>scanf</code></td>
<td>formatted input conversion</td>
</tr>
</tbody>
</table>
setbuf ........................................ assign buffering to a stream
ungetc ....................................... push character back into input stream

3X. Other Libraries
intro ........................................ introduction to miscellaneous library functions
assert ......................................... program verification
curses ....................................... screen functions with "optimal" cursor motion
dbm ........................................... data base subroutines
getdisk ...................................... get disk description by its name
getsent ...................................... get file system descriptor file entry
initgroups .................................... initialize group access list
lib2648 ..................................... subroutines for the HP 2648 graphics terminal
plot .......................................... graphics interface
rcmd .......................................... routines for returning a stream to a remote command
rexec ........................................ return stream to a remote command
termcap ..................................... terminal independent operation routines

3C. Compatibility Library Subroutines
intro ........................................ introduction to compatibility library functions
alarm .......................................... schedule signal after specified time
getpwn ....................................... get name from uid
nice .......................................... set program priority
pause ........................................ stop until signal
rand .......................................... random number generator
signal ........................................ simplified software signal facilities
stty ........................................... set and get terminal state (defunct)
time .......................................... get date and time
times ........................................ set file times
utime .......................................... get process times
vlimit ........................................ control maximum system resource consumption
vtimes ........................................ get information about resource utilization

4. Special Files
intro ........................................ introduction to special files and hardware support
acc ............................................. ACC 1.1/1DH IMP interface
ad ............................................. Data Translation A/D converter
arp ............................................ Address Resolution Protocol
autoconf .................................... diagnostics from the autoconfiguration code
bk ............................................. line discipline for machine-machine communication (obsolete)
cons ........................................... VAX-11 console interface
css ............................................. DEC IMP-11A 1.1/1DH IMP interface
cet ............................................ phototypesetter interface
de ............................................. DEC DRIUNA 10 Mb/s Ethernet interface
dh ............................................. DH-11/DM-11 communications multiplexer
dmc ........................................... DEC DMC-11/DMR-11 point-to-point communications device
dmf ........................................... DMF-32, terminal multiplexor
dn ............................................. DN-11 autocall unit interface
dr ............................................. DR11-B/DR11-W interface
drb ............................................ DR11-B/DR11-W general purpose user device interface
drum ........................................... paging device
dz ............................................. DZ-11 communications multiplexer
ec ............................................. 3Com 10 Mb/s Ethernet interface
en ............................................. Xerox 3 Mb/s Ethernet interface
enet .......................................... ethernet packet filter
fl ............................................. console floppy interface
5. File Formats

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.out</td>
<td>assembler and link editor output</td>
</tr>
<tr>
<td>acct</td>
<td>execution accounting file</td>
</tr>
<tr>
<td>aliases</td>
<td>aliases file for sendmail</td>
</tr>
<tr>
<td>ar</td>
<td>archive (library) file format</td>
</tr>
<tr>
<td>core</td>
<td>format of memory image file</td>
</tr>
<tr>
<td>dir</td>
<td>format of directories</td>
</tr>
<tr>
<td>disktab</td>
<td>disk description file</td>
</tr>
<tr>
<td>dump</td>
<td>incremental dump format</td>
</tr>
<tr>
<td>fs</td>
<td>format of file system volume</td>
</tr>
<tr>
<td>fstab</td>
<td>static information about the filesystems</td>
</tr>
<tr>
<td>gettytab</td>
<td>terminal configuration data base</td>
</tr>
<tr>
<td>group</td>
<td>group file</td>
</tr>
<tr>
<td>hosts</td>
<td>host name data base</td>
</tr>
<tr>
<td>mtab</td>
<td>mounted file system table</td>
</tr>
<tr>
<td>networks</td>
<td>network name data base</td>
</tr>
</tbody>
</table>

Grinnell Systems display
RK6-11/RK06 and RK07 moving head disk
MASSBUS disk interface
Network Systems Hyperchannel interface
Ikonas frame buffer, graphics device interface
Interlan 10 Mb/s Ethernet interface
1822 network interface
IMP raw socket interface
Internet protocol family
Broadcasting Internet Protocol packets
K.I.-11/DL-11W line clock
Software loopback network interface
Line printer
Main memory
TM-03/TE-16, TU-45, TU-77 MASSBUS magtape interface
UNIX magtape interface
UNIST magtape interface
DEC CSS PCI-11 B Network Interface
Evans and Sutherland Picture System 2 graphics device interface
Pseudo terminal driver
Xerox PUP-1 protocol family
Raw PUP socket interface
DEC RX02 floppy disk interface
Internet Transmission Control Protocol
TM-11/TE-10 magtape interface
TS-11 magtape interface
General terminal interface
VAX-11/730 and VAX-11/750 TU58 console cassette interface
UDA-50 disk controller interface
Internet User Datagram Protocol
Ungermaa-Bass interface
Unibus storage module controller/drives
UNIBUS TU45 tri-density tape drive interface
TU58/DJ:Taue II UNIBUS cassette interface
Benson-Variant interface
Versatec interface
Proteon proNET 10 Megabit ring

August 1983 - xxvi - Fourth Berkeley Distribution
## Table of Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newsrc</td>
<td>information file for readnews(1) and checknews(1)</td>
</tr>
<tr>
<td>passwd</td>
<td>password file</td>
</tr>
<tr>
<td>phones</td>
<td>remote host phone number data base</td>
</tr>
<tr>
<td>plot</td>
<td>graphics interface</td>
</tr>
<tr>
<td>printcap</td>
<td>printer capability data base</td>
</tr>
<tr>
<td>protocols</td>
<td>protocol name data base</td>
</tr>
<tr>
<td>resfile</td>
<td>format of RCS file</td>
</tr>
<tr>
<td>remote</td>
<td>remote host description file</td>
</tr>
<tr>
<td>services</td>
<td>service name data base</td>
</tr>
<tr>
<td>stab</td>
<td>symbol table types</td>
</tr>
<tr>
<td>tar</td>
<td>tape archive file format</td>
</tr>
<tr>
<td>termcap</td>
<td>terminal capability data base</td>
</tr>
<tr>
<td>tp</td>
<td>DEC/mag tape formats</td>
</tr>
<tr>
<td>ttys</td>
<td>terminal initialization data</td>
</tr>
<tr>
<td>ttytype</td>
<td>data base of terminal types by port</td>
</tr>
<tr>
<td>types</td>
<td>primitive system data types</td>
</tr>
<tr>
<td>utmp</td>
<td>login records</td>
</tr>
<tr>
<td>uuencode</td>
<td>format of an encoded uuencode file</td>
</tr>
<tr>
<td>vfont</td>
<td>font formats for the Benson-Varian or Versatec</td>
</tr>
<tr>
<td>vgrindefs</td>
<td>vgrind’s language definition data base</td>
</tr>
</tbody>
</table>

### 6. Games

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>yet another exploration game</td>
</tr>
<tr>
<td>adventure</td>
<td>an exploration game</td>
</tr>
<tr>
<td>arithmetric</td>
<td>provide drill in number facts</td>
</tr>
<tr>
<td>backgammon</td>
<td>the game</td>
</tr>
<tr>
<td>banner</td>
<td>print large banner on printer</td>
</tr>
<tr>
<td>bcd</td>
<td>convert to antique media</td>
</tr>
<tr>
<td>boggle</td>
<td>play the game of boggle</td>
</tr>
<tr>
<td>canfield</td>
<td>the solitaire card game canfield</td>
</tr>
<tr>
<td>chess</td>
<td>the game of chess</td>
</tr>
<tr>
<td>ching</td>
<td>the book of changes and other cookies</td>
</tr>
<tr>
<td>cribbage</td>
<td>the card game cribbage</td>
</tr>
<tr>
<td>doctor</td>
<td>interact with a psychoanalyst</td>
</tr>
<tr>
<td>fish</td>
<td>play “Go Fish”</td>
</tr>
<tr>
<td>fortune</td>
<td>print a random, hopefully interesting, adage</td>
</tr>
<tr>
<td>hangman</td>
<td>Computer version of the game hangman</td>
</tr>
<tr>
<td>mille</td>
<td>play Mille Bournes</td>
</tr>
<tr>
<td>monop</td>
<td>Monopoly game</td>
</tr>
<tr>
<td>number</td>
<td>convert Arabic numerals to English</td>
</tr>
<tr>
<td>quiz</td>
<td>test your knowledge</td>
</tr>
<tr>
<td>rain</td>
<td>animated raindrops display</td>
</tr>
<tr>
<td>rogue</td>
<td>Exploring The Dungeons of Doom</td>
</tr>
<tr>
<td>snake</td>
<td>display chase game</td>
</tr>
<tr>
<td>trek</td>
<td>trekkie game</td>
</tr>
<tr>
<td>worm</td>
<td>Play the growing worm game</td>
</tr>
<tr>
<td>worms</td>
<td>animate worms on a display terminal</td>
</tr>
<tr>
<td>wump</td>
<td>the game of hunt-the-wumpus</td>
</tr>
<tr>
<td>zork</td>
<td>the game of dungeon</td>
</tr>
</tbody>
</table>

### 7. Miscellaneous

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>miscellaneous useful information pages</td>
</tr>
<tr>
<td>ascii</td>
<td>map of ASCII character set</td>
</tr>
<tr>
<td>environ</td>
<td>user environment</td>
</tr>
<tr>
<td>eqnchar</td>
<td>special character definitions for eqn</td>
</tr>
</tbody>
</table>

Fourth Berkeley Distribution - xxvii - August 1983
### Table of Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hier</td>
<td>file system hierarchy</td>
</tr>
<tr>
<td>mailaddr</td>
<td>mail addressing description</td>
</tr>
<tr>
<td>man</td>
<td>macros for formatting papers</td>
</tr>
<tr>
<td>me</td>
<td>conventional names for terminals</td>
</tr>
<tr>
<td>ms</td>
<td>text formatting macros</td>
</tr>
<tr>
<td>term</td>
<td></td>
</tr>
</tbody>
</table>

8. System Maintenance

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>introduction to system maintenance and operation commands</td>
</tr>
<tr>
<td>750rom</td>
<td>details of Vax-11/750 boot ROMs</td>
</tr>
<tr>
<td>ac</td>
<td>login accounting</td>
</tr>
<tr>
<td>addrfmt</td>
<td>IP/ICMP Address Format Request user program</td>
</tr>
<tr>
<td>adduser</td>
<td>procedure for adding new users</td>
</tr>
<tr>
<td>analyze</td>
<td>Virtual UNIX postmortem crash analyzer</td>
</tr>
<tr>
<td>arcv</td>
<td>convert archives to new format</td>
</tr>
<tr>
<td>arff</td>
<td>archiver and copier for floppy</td>
</tr>
<tr>
<td>arp</td>
<td>address resolution display and control</td>
</tr>
<tr>
<td>bad144</td>
<td>read/write dec standard 144 bad sector information</td>
</tr>
<tr>
<td>badsect</td>
<td>create files to contain bad sectors</td>
</tr>
<tr>
<td>breathlife</td>
<td>breath-of-life server for bootloading 3mb Altos</td>
</tr>
<tr>
<td>bugfiler</td>
<td>file bug reports in folders automatically</td>
</tr>
<tr>
<td>buildnetdir</td>
<td>build binary-format Pup Network Directory</td>
</tr>
<tr>
<td>catman</td>
<td>create the cat files for the manual</td>
</tr>
<tr>
<td>chown</td>
<td>change owner</td>
</tr>
<tr>
<td>cli</td>
<td>clear i-node</td>
</tr>
<tr>
<td>comsat</td>
<td>biff server</td>
</tr>
<tr>
<td>config</td>
<td>build system configuration files</td>
</tr>
<tr>
<td>crash</td>
<td>what happens when the system crashes</td>
</tr>
<tr>
<td>cron</td>
<td>clock daemon</td>
</tr>
<tr>
<td>dcheck</td>
<td>file system directory consistency check</td>
</tr>
<tr>
<td>ddacct</td>
<td>Dump Dover Accounting</td>
</tr>
<tr>
<td>diskpart</td>
<td>calculate default disk partition sizes</td>
</tr>
<tr>
<td>dmesg</td>
<td>collect system diagnostic messages to form error log</td>
</tr>
<tr>
<td>drtest</td>
<td>standalone disk test program</td>
</tr>
<tr>
<td>dump</td>
<td>incremental file system dump</td>
</tr>
<tr>
<td>dumpfs</td>
<td>dump file system information</td>
</tr>
<tr>
<td>edquota</td>
<td>edit user quotas</td>
</tr>
<tr>
<td>enstat</td>
<td>print enet (packet filter) information</td>
</tr>
<tr>
<td>expire</td>
<td>remove outdated news articles</td>
</tr>
<tr>
<td>fastboot</td>
<td>reboot/halt the system without checking the disks</td>
</tr>
<tr>
<td>filtime</td>
<td>tell minutes since file (access, modification) time</td>
</tr>
<tr>
<td>fingd</td>
<td>network finger server</td>
</tr>
<tr>
<td>format</td>
<td>how to format disk packs</td>
</tr>
<tr>
<td>fck</td>
<td>file system consistency check and interactive repair</td>
</tr>
<tr>
<td>fckblks</td>
<td>print alternate super block numbers for fck -b</td>
</tr>
<tr>
<td>fstat</td>
<td>filter filenames according to commands in a status file</td>
</tr>
<tr>
<td>ftpd</td>
<td>DARPA Internet File Transfer Protocol server</td>
</tr>
<tr>
<td>ftpser</td>
<td>PUP File Transfer Protocol Service</td>
</tr>
<tr>
<td>gatewayinfo</td>
<td>PUP GatewayInfo routing table server</td>
</tr>
<tr>
<td>gettable</td>
<td>get NIC format host tables from a host</td>
</tr>
<tr>
<td>getty</td>
<td>set terminal mode</td>
</tr>
<tr>
<td>gsa</td>
<td>group system accounting</td>
</tr>
<tr>
<td>halt</td>
<td>stop the processor</td>
</tr>
<tr>
<td>htable</td>
<td>convert NIC standard format host tables</td>
</tr>
<tr>
<td>icheck</td>
<td>file system storage consistency check</td>
</tr>
<tr>
<td>ifconfig</td>
<td>configure network interface parameters</td>
</tr>
</tbody>
</table>

August 1983 - xxviii - Fourth Berkeley Distribution
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>implog</td>
<td>IMP log interpreter</td>
</tr>
<tr>
<td>implogd</td>
<td>IMP logger process</td>
</tr>
<tr>
<td>inetd</td>
<td>DARPA little protocol server</td>
</tr>
<tr>
<td>init</td>
<td>process control initialization</td>
</tr>
<tr>
<td>insecure</td>
<td>user security monitor</td>
</tr>
<tr>
<td>kgmon</td>
<td>generate a dump of the operating system’s profile buffers</td>
</tr>
<tr>
<td>leaf</td>
<td>PUP Leaf Remote File Access Protocol Server</td>
</tr>
<tr>
<td>lpc</td>
<td>line printer control program</td>
</tr>
<tr>
<td>lpd</td>
<td>line printer daemon</td>
</tr>
<tr>
<td>mailer</td>
<td>Mailing list, forwarding, and alias manager</td>
</tr>
<tr>
<td>makedev</td>
<td>make system special files</td>
</tr>
<tr>
<td>makekey</td>
<td>generate encryption key</td>
</tr>
<tr>
<td>miscserver</td>
<td>MiscServices server for Pup</td>
</tr>
<tr>
<td>mkfss</td>
<td>construct a file system</td>
</tr>
<tr>
<td>mklost+found</td>
<td>make a lost+found directory for fsck</td>
</tr>
<tr>
<td>mknod</td>
<td>build special file</td>
</tr>
<tr>
<td>mkproto</td>
<td>construct a prototype directory</td>
</tr>
<tr>
<td>mount</td>
<td>mount and dismount file system</td>
</tr>
<tr>
<td>ncheck</td>
<td>generate names from i-numbers</td>
</tr>
<tr>
<td>netdirprint</td>
<td>print text version of Pup Network Directory</td>
</tr>
<tr>
<td>newfs</td>
<td>construct a new file system</td>
</tr>
<tr>
<td>nu</td>
<td>manage user login accounts (create, modify, destroy Unix accounts)</td>
</tr>
<tr>
<td>pac</td>
<td>printer/ploter accounting information</td>
</tr>
<tr>
<td>patchroute</td>
<td>kludge to support Stanford Pup-based subnet routing</td>
</tr>
<tr>
<td>pstat</td>
<td>print system facts</td>
</tr>
<tr>
<td>pup-mailer</td>
<td>deliver mail over the .SM PUP network</td>
</tr>
<tr>
<td>pup10arpser</td>
<td>Pup GatewayInfo routing table server</td>
</tr>
<tr>
<td>pupgateway</td>
<td>a Pup gateway program</td>
</tr>
<tr>
<td>quot</td>
<td>summarize file system ownership</td>
</tr>
<tr>
<td>quotacheck</td>
<td>file system quota consistency checker</td>
</tr>
<tr>
<td>quotaon</td>
<td>turn file system quotas on and off</td>
</tr>
<tr>
<td>rc</td>
<td>command script for auto-reboot and daemons</td>
</tr>
<tr>
<td>rdump</td>
<td>file system dump across the network</td>
</tr>
<tr>
<td>reboot</td>
<td>UNIX bootstrapping procedures</td>
</tr>
<tr>
<td>recnews</td>
<td>receive unprocessed articles via mail</td>
</tr>
<tr>
<td>renice</td>
<td>alter priority of running processes</td>
</tr>
<tr>
<td>repquota</td>
<td>summarize quotas for a file system</td>
</tr>
<tr>
<td>restore</td>
<td>incremental file system restore</td>
</tr>
<tr>
<td>rexed</td>
<td>remote execution server</td>
</tr>
<tr>
<td>rlogin</td>
<td>remote login server</td>
</tr>
<tr>
<td>rmt</td>
<td>remote magtape protocol module</td>
</tr>
<tr>
<td>route</td>
<td>manually manipulate the routing tables</td>
</tr>
<tr>
<td>routed</td>
<td>network routing daemon</td>
</tr>
<tr>
<td>restore</td>
<td>restore a file system dump across the network</td>
</tr>
<tr>
<td>rshd</td>
<td>remote shell server</td>
</tr>
<tr>
<td>rwhod</td>
<td>system status server</td>
</tr>
<tr>
<td>rxformat</td>
<td>format floppy disks</td>
</tr>
<tr>
<td>sa</td>
<td>system accounting</td>
</tr>
<tr>
<td>savecore</td>
<td>save a core dump of the operating system</td>
</tr>
<tr>
<td>sendmail</td>
<td>send mail over the internet</td>
</tr>
<tr>
<td>sendnews</td>
<td>send news articles via mail</td>
</tr>
<tr>
<td>shutdown</td>
<td>close down the system at a given time</td>
</tr>
<tr>
<td>sticky</td>
<td>executable files with persistent text</td>
</tr>
<tr>
<td>swapon</td>
<td>specify additional device for paging and swapping</td>
</tr>
<tr>
<td>sync</td>
<td>update the super block</td>
</tr>
<tr>
<td>syslog</td>
<td>log systems messages</td>
</tr>
</tbody>
</table>
9. Pup library

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>atoo</td>
<td>convert ASCII to octal numbers</td>
</tr>
<tr>
<td>bmove</td>
<td>block move a buffer</td>
</tr>
<tr>
<td>bytorder</td>
<td>discussion of byte-ordering and the Pup package</td>
</tr>
<tr>
<td>checksum</td>
<td>compute Pup-style checksum</td>
</tr>
<tr>
<td>eftp</td>
<td>Pup EFTP package</td>
</tr>
<tr>
<td>enarp</td>
<td>Address Resolution Protocol (ARP) routines</td>
</tr>
<tr>
<td>enetfilter</td>
<td>build ethernet filters</td>
</tr>
<tr>
<td>cnflush</td>
<td>flush an ethernet input file</td>
</tr>
<tr>
<td>engethost</td>
<td>determine Pup host number of ethernet interface</td>
</tr>
<tr>
<td>enopen</td>
<td>open an ethernet file</td>
</tr>
<tr>
<td>enread</td>
<td>read a packet from an ethernet file</td>
</tr>
<tr>
<td>ensetbacklog</td>
<td>set ethernet input queue backlog</td>
</tr>
<tr>
<td>ensignal</td>
<td>enable or disable signal on ethernet packet arrival</td>
</tr>
<tr>
<td>enwrite</td>
<td>write a packet to the ethernet</td>
</tr>
<tr>
<td>maddtoname</td>
<td>translate a Pup Port address to a name</td>
</tr>
<tr>
<td>mattributes</td>
<td>get Pup Network Directory entry attributes for an address</td>
</tr>
<tr>
<td>mbootdir</td>
<td>get a boot file directory from a boot server</td>
</tr>
<tr>
<td>mbootrequest</td>
<td>request a boot-load</td>
</tr>
<tr>
<td>mkissofdeath</td>
<td>send a KissOfDeath Pup</td>
</tr>
<tr>
<td>mlookup</td>
<td>translate a name to a Pup address</td>
</tr>
<tr>
<td>mmailcheck</td>
<td>find out if a user has new mail at a Pup host</td>
</tr>
<tr>
<td>mcsrvreq</td>
<td>make a MiscServices request</td>
</tr>
<tr>
<td>msendumsg</td>
<td>send a message to one or all users at a Pup host</td>
</tr>
<tr>
<td>msunbootreq</td>
<td>request a boot-load</td>
</tr>
<tr>
<td>mt imecheck</td>
<td>get time from a Pup server</td>
</tr>
<tr>
<td>overview</td>
<td>overview of Pup library routines</td>
</tr>
<tr>
<td>pupchan</td>
<td>data structure describing a Pup channel</td>
</tr>
<tr>
<td>pupclose</td>
<td>close a pup channel</td>
</tr>
<tr>
<td>pupdescip</td>
<td>access mapping between PupChans and fids</td>
</tr>
<tr>
<td>puperrmsg</td>
<td>human-readable error message from Pup package routines</td>
</tr>
<tr>
<td>pupgethost</td>
<td>get host, net numbers of local end of pup channel</td>
</tr>
<tr>
<td>pupgetport</td>
<td>get address of local, remote ends of pup channel</td>
</tr>
<tr>
<td>pupint</td>
<td>enable or disable interrupts for received Pup Packets</td>
</tr>
<tr>
<td>puplisten</td>
<td>open Pup channels on all connected networks</td>
</tr>
<tr>
<td>pumisc</td>
<td>miscellaneous Pup routines</td>
</tr>
<tr>
<td>pupnettab</td>
<td>pup configuration table</td>
</tr>
<tr>
<td>pupopen</td>
<td>open a pup channel</td>
</tr>
<tr>
<td>pupport</td>
<td>a data structure used in the Pup package, with support routines</td>
</tr>
<tr>
<td>pupprint</td>
<td>printing routines for Pup</td>
</tr>
<tr>
<td>pupread</td>
<td>read a pup packet from a pup channel</td>
</tr>
<tr>
<td>pupropopen</td>
<td>change the destination for a Pup channel</td>
</tr>
<tr>
<td>puproute</td>
<td>find a route for a Pup Internet packet</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>puprouting</td>
<td>Pup Internet routing table maintenance routines</td>
</tr>
<tr>
<td>pupsetbacklog</td>
<td>set input queue backlog for Pup channel</td>
</tr>
<tr>
<td>pupsetdfilt</td>
<td>set a default packet filter for a Pup channel</td>
</tr>
<tr>
<td>pupsetfilter</td>
<td>set the packet filter for a Pup channel</td>
</tr>
<tr>
<td>pupsetmode</td>
<td>set and read the mode for a Pup channel</td>
</tr>
<tr>
<td>pupsettimeout</td>
<td>set timeout for Pup channel</td>
</tr>
<tr>
<td>pupstring</td>
<td>manipulate strings</td>
</tr>
<tr>
<td>pupwrite</td>
<td>write a packet to a Pup channel</td>
</tr>
<tr>
<td>ringbuf</td>
<td>ring buffer package</td>
</tr>
<tr>
<td>uatimecvt</td>
<td>conversions from Unix to Alto time and vice versa</td>
</tr>
<tr>
<td>uniquesocket</td>
<td>create unique socket number</td>
</tr>
</tbody>
</table>
Permuted Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>Absolute value</td>
</tr>
<tr>
<td>abort</td>
<td>Terminate abruptly with memory image</td>
</tr>
<tr>
<td>acct</td>
<td>Login accounting</td>
</tr>
<tr>
<td>acc</td>
<td>File access</td>
</tr>
<tr>
<td>access</td>
<td>Accessibility</td>
</tr>
<tr>
<td>accept</td>
<td>Accept a connection on a socket</td>
</tr>
<tr>
<td>ac</td>
<td>Login</td>
</tr>
<tr>
<td>acct</td>
<td>Turn accounting on or off</td>
</tr>
<tr>
<td>add</td>
<td>Add a swap device for interleaved paging/swapping</td>
</tr>
<tr>
<td>adddir</td>
<td>Add directory entry attributes for an mlookup</td>
</tr>
<tr>
<td>addfnt</td>
<td>Add Pup Network Directory entry attributes for an mlookup</td>
</tr>
<tr>
<td>addrs</td>
<td>Add default route</td>
</tr>
<tr>
<td>adm</td>
<td>Administer</td>
</tr>
<tr>
<td>adv</td>
<td>Advanced</td>
</tr>
<tr>
<td>adduser</td>
<td>Create or extend a user database</td>
</tr>
<tr>
<td>admbug</td>
<td>Debug a swap device</td>
</tr>
<tr>
<td>adskip</td>
<td>Skip an address block</td>
</tr>
<tr>
<td>admin</td>
<td>Administer an account</td>
</tr>
<tr>
<td>address</td>
<td>Manage an account</td>
</tr>
<tr>
<td>adduser</td>
<td>Add a user</td>
</tr>
<tr>
<td>address</td>
<td>Address an account</td>
</tr>
<tr>
<td>adduser</td>
<td>Add a user</td>
</tr>
<tr>
<td>addrprint</td>
<td>Address Format Request User program</td>
</tr>
<tr>
<td>addrn</td>
<td>Add a network</td>
</tr>
<tr>
<td>addrn</td>
<td>Add a network</td>
</tr>
<tr>
<td>adduser</td>
<td>Add a user</td>
</tr>
<tr>
<td>addr</td>
<td>Address an account</td>
</tr>
<tr>
<td>_accounts</td>
<td>Manage user accounts</td>
</tr>
</tbody>
</table>

Fourth Berkeley Distribution - xxix - August 1983
Permuted Index

inet_makeaddr, inet_lnaof, inet_netof: Internet address manipulation routines /inet_nhoa: inet(3n)
loc: return the address of an object.
loc(3F)
(pupgetport) pupgetsreport, pupgetdsport: get addresses of local, remote ends of pup channel.
pupgetport(9)
arp: address resolution display and control.
arp(8C)
arp: Address Resolution Protocol.
arp(4P)
enarp (en10mbpaparp, ennap): Address Resolution Protocol (ARP) routines.
enarp(9)
addname: translate a Pup Port host: print IP host names and addresses.
addname(9)
net: print IP net names and addresses.
net(1)
mailaddr: mail program.
mailadd(7)
addprint: IP/ICMP Address Format Request user
addprint(8)
adduser: procedure for adding new users.
adduser(8)
time: adjust the access or modification time of a file.
time(8)
advfreq: an exploration game.
advfreq(6)
flock: apply or remove an advisory lock on an open file.
flock(2)
yes: be repetitive affirmative.
yes(1)
basename: strip filename aliases
basename(1)
learn: computer aided instruction about UNIX
learn(1)
alarm: execute a subroutine after a specified time.
alarm(3F)
alarm: schedule signal after specified time.
alarm(3C)
mailer: Mailing list, forwarding, and alias manager.
mailer(8)
unalias: remove aliases.
unalias(1)
which: locate a program file including aliases.
which(1)
aliases: aliases file for sendmail.
aliases(5)
aliased: aliases file only.
aliased(1)
newaliases: rebuild the data base for the mail aliases.
newaliases(1)
malloc, free, realloc, calloc, malloc, free, realloc, catloc,
allocators: memory allocator.
malloc(3)
malloc(3)
valloc: aligned memory allocator.
valloc(3)
valloc(3)
eye: modified vace
eye(1)
error: analyze and dispense compiler error messages.
error(1)
style: analyze surface characteristics of a document.
style(1)
analyze: Virtual UNIX postmortem crash analyzer.
analyze(8)
sigstack: set and/or get signal stack context.
sigstack(2)
alternative commands.
sig(1)
fsckblks: print alternate super block numbers for file.
fsckblks(8)
e: conversions from Unix to file format.
e(1)
Utime: load files from an Alto FTP "dump" format file.
Utime(9)
Alto time and vice versa.
Utime(9)
break-of-life server for booting 3mb lex: generator of lexical
text: convert to antique media.
text(1)
apt: apt interpreter.
apt(1)
apply: apply a command to a set of arguments.
apply(1)
apply: apply a command to a set of arguments.
apply(1)
flock: apply or remove an advisory lock on an open file.
flock(2)
propose: locate commands by keyword lookup.
propose(1)
ar: archive and library maintainer.
ar(1)
arL archive (library) file format.
ar(5)
arcv: convert archives to new format.
arcv(8)
numuser: convert Arabic numerals to English.
numuser(6)
bc: arbitrary-precision arithmetic language.
b(1)
graphics: plot: openpl, erase, label, line, circle, arc, move, cont, point, linemod, space, closepl:
plot(3X)
bp: manipulate tape
archive.
archive(1)
tar: archive file format.
tar(5)
tar: archive (library) file format.
tar(5)
tar: convert archives.
tar(1)
artifact, floppy: archiver and copy for floppy.
artifact(8)
shar: produce shell-script Archives.
shar(1)
acrc: convert archives to new format.
acrc(8)
raclile: convert archives to random libraries.
raclile(1)
artifact: archiver and copy for floppy.
artifact(8)
glob: filename expand argument list.
glob(1)
shift: manipulate argument list.
shift(1)
ined Index

rexe: return stream to a remote command.
system: issue a shell command.
system: execute a UNIX command.
test: condition.
time: time a command.
nice, nohup: run a command at low priority (th only).
switch: multi-way command branch.
remote: Remote command execution.
rehash: recompute command hash table.
unhash: discard command hash table.
hashtable: print command hashing statistics.
nohup: run command immune to hangups.
csh: a shell (command interpreter) with C-like syntax.
whatis: describe what a command is.

readonly, set, shift, times, trap, umask, wait: command language. /exec, exit, export, login, read, sh
getarg, large, return command line arguments.
repeat: execute command on terminal screen.
rc: command script for auto-reboot and daemons.
onintr: process interrupts in command scripts.
apply: apply a command to a set of arguments.
goto: command transfer.
echo: echo commands.
else: alternative commands.
intro: introduction to commands.
intro: introduction to system maintenance and operation.
rcintro - introduction to RCS commands.
at: execute commands at a later time.
 apropos: locate commands by keyword lookup.
while: repeat commands conditionally.
lastcomm: show last commands executed in reverse order.
source: read commands from file.

r68: print relocation
fstat: filter filenames according to comments in C code.
context: check for nested common to two sorted files.
socket: create an endpoint for communication.
pipe: create an interprocess communication channel.
blk: line discipline for machine-machine communications device.
dmci: DEC DMC-11/DMR-11 point-to-point communications multiplexer.
dh: DH-1/DM-11 communications multiplexer.
dz: DZ-11
users: compact list of users who are on the system.

files, and cat them.
diff: differential file and directory comparison.
resdiff: compare RCS revisions.
cmp: compare two files.
diff(3): 3-way differential file comparison operations.
strmcomp: strmcompfold: case-folded string comparison operations.
intro: introduction to compatibility library functions.
list: list:
cc: C compiler.
j77: Fortran 77 compiler.
mcl: Modula-2 compiler.
pc: Pascal compiler.
error: analyze and disperse compiler error messages.
cc68: C compiler for the MC68000.
pe68: Pascal compiler for the MC68000.
yacc: yet another parser generator.
fp: Functional Programming language compiler/interpreter.
wait: wait for background processes to complete.
as68: a68 -> .b assembler component of cc68.
ecom68: c -> .s translator component of cc68.
df68: b.out -> dll downloder component of cc68.
od68: s -> .s optimizer component of cc68.
compact, uncompress, cat: compress and uncompress files, and cat them.
checksum: compute Pup-style checksum.
learn: computer aided instruction about UNIX.
hangman: Computer version of the game hangman.
puppetnet: connect your terminal to a remote computer via Pup network.

test: condition command.
endif: terminate conditional.
if: conditional statement.
while: repeat commands conditionally.
Press format and print them on the Dover...

crc: create an Emacs window.

date: print current time.

data: load file data.

dbminit: fetch, store, delete, current working directory.

dbac: debugger.

dl: dump database.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.

dbminit: fetch, store, delete.

date: print current time.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddb:</td>
<td>debugger.</td>
</tr>
<tr>
<td>dbx:</td>
<td>debugger.</td>
</tr>
<tr>
<td>ddbx:</td>
<td>debugger.</td>
</tr>
<tr>
<td>pxx:</td>
<td>pascal.</td>
</tr>
<tr>
<td>ddt68, fddt68:</td>
<td>symbolic debugger for 68000.</td>
</tr>
<tr>
<td>p:</td>
<td>DEC CSS PCL-11 B Network Interface.</td>
</tr>
<tr>
<td>p:</td>
<td>DEC CSS PCL-11 B Network Interface.</td>
</tr>
<tr>
<td>d:</td>
<td>DEC DUNA 10 Mbs Ethernet interface.</td>
</tr>
<tr>
<td>device, dmc:</td>
<td>DEC DMC-11/DMR-11 point-to-point communications</td>
</tr>
<tr>
<td>dce:</td>
<td>DEC IMP-11/11 IMP interface.</td>
</tr>
<tr>
<td>rx:</td>
<td>DEC INUX floppy disk interface.</td>
</tr>
<tr>
<td>bad144:</td>
<td>read/write dce standard 144 bad sector information.</td>
</tr>
<tr>
<td>od:</td>
<td>dce standard 144 bad sector information.</td>
</tr>
<tr>
<td>tp:</td>
<td>dce standard 144 bad sector information.</td>
</tr>
<tr>
<td>chdir:</td>
<td>default: catchall clause in switch.</td>
</tr>
<tr>
<td>diskpart:</td>
<td>default directory.</td>
</tr>
<tr>
<td>chsh:</td>
<td>default login shell.</td>
</tr>
<tr>
<td>pupsetdfilt:</td>
<td>set a default packet filter for a Pup channel.</td>
</tr>
<tr>
<td>vgrndefs:</td>
<td>vgrind's language definition data base.</td>
</tr>
<tr>
<td>eqnchar:</td>
<td>special character definitions for eqn.</td>
</tr>
<tr>
<td>sty, sty:</td>
<td>set and get terminal state (default).</td>
</tr>
<tr>
<td>close:</td>
<td>delete a descriptor.</td>
</tr>
<tr>
<td>dbminit, fetch, store, \n</td>
<td>delete, firstkey, nextkey: data base subroutines.</td>
</tr>
<tr>
<td>pup-mailer:</td>
<td>deliver mail over the SM PUP network.</td>
</tr>
<tr>
<td>tail:</td>
<td>deliver the last part of a file.</td>
</tr>
<tr>
<td>mesg:</td>
<td>deny messages.</td>
</tr>
<tr>
<td>makedep:</td>
<td>deny messages.</td>
</tr>
<tr>
<td>makedep:</td>
<td>dependency lines for makefiles.</td>
</tr>
<tr>
<td>tset:</td>
<td>dependent initialization.</td>
</tr>
<tr>
<td>constructs</td>
<td>description by its name.</td>
</tr>
<tr>
<td>crypt, setkey, encrypt:</td>
<td>DESCRIPTION encryption.</td>
</tr>
<tr>
<td>whatis:</td>
<td>describe what a command is.</td>
</tr>
<tr>
<td>pupchan:</td>
<td>data structure describing a Pup channel.</td>
</tr>
<tr>
<td>mailaddr: mail addressing description.</td>
<td></td>
</tr>
<tr>
<td>mailaddr: mail addressing description.</td>
<td></td>
</tr>
<tr>
<td>getdiskbyname:</td>
<td>get disk description by its name.</td>
</tr>
<tr>
<td>diskbat: disk description file.</td>
<td></td>
</tr>
<tr>
<td>remote:</td>
<td>remote host description file.</td>
</tr>
<tr>
<td>close:</td>
<td>delete a descriptor.</td>
</tr>
<tr>
<td>dup, dup2:</td>
<td>duplicate a descriptor.</td>
</tr>
<tr>
<td>getfsftype, getfsent, endfsent: get file system descriptor file entry. /getfspec, getfsfile.</td>
<td></td>
</tr>
<tr>
<td>getfsizes:</td>
<td>get descriptor table size.</td>
</tr>
<tr>
<td>getfsizes:</td>
<td>get descriptor table size.</td>
</tr>
<tr>
<td>de:</td>
<td>device.</td>
</tr>
<tr>
<td>diode:</td>
<td>device.</td>
</tr>
<tr>
<td>fold:</td>
<td>fold long lines for finite width output.</td>
</tr>
<tr>
<td>ic:</td>
<td>icore control.</td>
</tr>
<tr>
<td>spawn:</td>
<td>device for interleaved paging/swapping.</td>
</tr>
<tr>
<td>swapno:</td>
<td>device for paging and swapping.</td>
</tr>
<tr>
<td>ddb:</td>
<td>device interface.</td>
</tr>
<tr>
<td>dmc:</td>
<td>device interface.</td>
</tr>
<tr>
<td>dmc:</td>
<td>device interface.</td>
</tr>
<tr>
<td>ps:</td>
<td>Evans and Sutherland Picture System 2 graphics device interface.</td>
</tr>
<tr>
<td>ethernet:</td>
<td>show status of ethernet minor devices.</td>
</tr>
<tr>
<td>dh:</td>
<td>Diff: device interface.</td>
</tr>
<tr>
<td>dms:</td>
<td>diagnostic messages to form error log.</td>
</tr>
<tr>
<td>autoconf:</td>
<td>diagnostics from the autoconfiguration code.</td>
</tr>
<tr>
<td>print wordy sentences; thesaurus for diction</td>
<td>print wordy sentences; thesaurus for diction.</td>
</tr>
<tr>
<td>print wordy sentences; thesaurus for diction</td>
<td>print wordy sentences; thesaurus for diction.</td>
</tr>
<tr>
<td>print wordy sentences; thesaurus for diction</td>
<td>print wordy sentences; thesaurus for diction.</td>
</tr>
<tr>
<td>diction, explain:</td>
<td>diction, explain:</td>
</tr>
<tr>
<td>explain:</td>
<td>diction, explain:</td>
</tr>
<tr>
<td>diff:</td>
<td>differential file and directory comparator.</td>
</tr>
<tr>
<td>diff3:</td>
<td>3-way differential file comparison.</td>
</tr>
<tr>
<td>diff:</td>
<td>differential file and directory comparator.</td>
</tr>
<tr>
<td>diff3:</td>
<td>3-way differential file comparison.</td>
</tr>
<tr>
<td>dir:</td>
<td>format of directories.</td>
</tr>
</tbody>
</table>

**ust 1983** - Fourth Berkeley Distribution
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dumpfonts</td>
<td>Show what fonts are available.</td>
</tr>
<tr>
<td>lookbib</td>
<td>Build inverted index for a bibliography.</td>
</tr>
<tr>
<td>spellbib</td>
<td>Spell out the spelling errors.</td>
</tr>
<tr>
<td>binary, .filename, .strings</td>
<td>Find the printable strings in an object, or other strings.</td>
</tr>
<tr>
<td>findex</td>
<td>Network finger server.</td>
</tr>
<tr>
<td>fing</td>
<td>User information lookup program.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>ddbm, fetch, store, delete</td>
<td>Find references in a bibliography.</td>
</tr>
<tr>
<td>nice, nohup</td>
<td>Run a command at low priority.</td>
</tr>
<tr>
<td>nice</td>
<td>Console floppy interface.</td>
</tr>
<tr>
<td>flopy</td>
<td>Floppy archiver and copier for floppy.</td>
</tr>
<tr>
<td>rx</td>
<td>Floppy disk interface.</td>
</tr>
<tr>
<td>rxformat</td>
<td>Floppy disks.</td>
</tr>
<tr>
<td>fclose</td>
<td>Close or flush an open logical unit.</td>
</tr>
<tr>
<td>exit</td>
<td>Terminate a process after flushing any pending output.</td>
</tr>
<tr>
<td>inspect</td>
<td>Print out information about UNIX.</td>
</tr>
<tr>
<td>dumpfonts</td>
<td>Show what Press fonts are available.</td>
</tr>
<tr>
<td>vwidth</td>
<td>Make troff width table for a font.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>file</td>
<td>Functions.</td>
</tr>
<tr>
<td>functions</td>
<td>Functions.</td>
</tr>
<tr>
<td>arff</td>
<td>Floppy archiver and copier for floppy.</td>
</tr>
<tr>
<td>findex</td>
<td>Network finger server.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>trap</td>
<td>Trap and repair floating point faults.</td>
</tr>
<tr>
<td>trapv</td>
<td>Trap and repair floating point overflow.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>bugfier</td>
<td>File bug reports in folders automatically.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>exit</td>
<td>Terminete a process after flushing any pending output.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>bugfier</td>
<td>File bug reports in folders automatically.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>bugfier</td>
<td>File bug reports in folders automatically.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>bugfier</td>
<td>File bug reports in folders automatically.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
<tr>
<td>bugfier</td>
<td>File bug reports in folders automatically.</td>
</tr>
<tr>
<td>fed</td>
<td>Font editor.</td>
</tr>
<tr>
<td>vformat</td>
<td>Font formats for the Berson-Varian or Versatec.</td>
</tr>
<tr>
<td>fonts</td>
<td>Fonts are available in fonts.widths.</td>
</tr>
<tr>
<td>dflmin, dflmax</td>
<td>Return extreme values.</td>
</tr>
<tr>
<td>fold</td>
<td>Fold long lines for finite width output device.</td>
</tr>
</tbody>
</table>
index

getable: get NIC format host tables from a host. getable(8C)

vtnn, or trofl. vip: format: how to format disk packs.

ansi: read and write ANSI format magnetic tapes.

uencode: format of an encoded uencode file.

dir: format of directories.

fs, inode: format of file system volume.

core: format of memory image file.

resfile: format of RCS file.

addrrm: IP/ICMP Address: format Request user program.

 tbl: format tables for nroff or troff.

tbl: DEC/mag tape formats.

vfont: font formats for the Benson-Varian Version.

scan, fscanf, scanf: formatted input conversion.

printf, fprintf, sprintf: formatted output conversion.

fmt: simple text formatter.

nroff: text formatting.

tex, latex, inter, vindex: text formatting and typesetting.

troff, nroff: text formatting and typesetting.

ms: text formatting macros.

me: macros for formatting papers.

fr77: Fortran 77 compiler.

rator: rational Fortran dialect.

frp: print Fortran file.

fsplit: split a multi-routine Fortran file into individual files.

e11: Extended Fortran Language.

intro: introduction to FORTRAN library functions.

pue, putc: write a character to a fortran logical unit.

struct: structure Fortran programs.

adage: fortune: print a random, hopefully interesting,

mailer: Mailing list.

login/: sh, for, case, if, while,:

exit, export/: sh, for, case, if, while,:

compiler/interpreter.

trp, fepe: Fortran trap and repair floating point faults.

frp: print Fortran file.

printf, sprintf: formatted output conversion.

putc, putw: put character or word on a stream.

putc, putc: write a character to a fortran logical unit.

puts: put a string on a stream.

ik: iconas frame buffer, graphics device interface.

list: compile a Fortran Lisp program.

frexp, ldexp, modf: split into mantissa and

exponent. from: who is my mail?

fing: front end for finger.

fs, inode: format of file system volume.

scan, fscanf, scanf: formatted input conversion.

mklost + found: make a lost + found directory for

fseek: file system consistency check and interactive

repair. fseek: print alternate super block numbers

for nroff' or troff.

fseek: print alternate super block numbers for

data: static information about the filesystems.

stat, bstat: get file status.

stat, bstat: get file status.

on disk: fsys: synchronize a file's in-core state with that

ftime: get date and time.

altload: load files from an Allo

ftp: file transfer program.

shutdown: shut down part of a

gamma: log gamma

compiler/interpreter. frp

bit: and, or, xor, not, shift, shift bitwise

fabs, floor, ceil: absolute value, floor, ceiling

intro: introduction to library

just 1983

Fourth Berkeley Distribution
host and network byte order.
and network byte order, htonl,
routines. PupErrMsg:
wump: the game of
sinh, cosh, tanh:
hy: Network Systems
getarg,
getpid: get process
setuid: set real and effective group
getgid, getgroup: get group
getargv: return command line arguments
setreuid: set reat and effective user
wump: the game of
indent: indent and
iphostid: set or print
printer.
pressimp: convert
abort: terminate abruptly with memory
acc: ACC 11/1/DH
implog: IMP log interpreter.
unifdcf: remove ifdefd lines.
eval, exec, exit, export, login, su, for, case.
gethostid, sethostid: get/set unique identifier of current host.
sin: return command or print
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
iptable: IP table.
ssh: log interpreter.
setsockopt: set options.
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
import: import shared strings.
dump: incremental
export: export file system.
dump: incremental file system dump.
dump: dump.
dump: dump.
dump: dump.
indent: indent and
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
setreuid: set reat and effective user
indent: indent and
inetnum, tgenent.
list: list aliases.
Permuted Index

ht: TM-03/TE-16,TU-45,TU-77 MASSBUS magtape
hy: Network Systems Hyperchannel
ik: likonas frame buffer, graphics device
ii: Interlan 10 Mb/s Ethernet
imp: IMP raw socket
loc: software loopback network
mt: TM78/TU-78 MASSBUS magtape
mtio: UNIBUS tape
pcj: DEC CSS FC-110 Network Plot
plot: graphics
rx: DEC RX02 floppy disk
tm: TM-11/TI-10 magtape
tsc: TS-11 magtape
tty: general terminal
VAX-111730 and
VAX-111750
/ineCntoa. IncCmakeaddr.
and
Sutherland Picture System

Index

atomically release blocked signals and wait for cont.
tread. twrite, trewin.
mt: UNIBUS TU45
TU58
DEClape
VAX-111750 TU58
I)upint. pupnoint: enable or disable
TM7SrrU-78 MASSBUS
pel: linemod.
linenroutc: find a route for a
DEC

10: software
ncheck: generate names from
sendmail: send mail over the
uda: UNIBUS TU45 tri-density tape drive
uu: TUSB/DECtape II UNIBUS cassette
va: Benson-Vanian
vp: Versatec

ifconfig: configure network
telnet: user
swapon: add a swap device for
sendmail: send mail over the
/inet_ntoa, inet_makeaddr, inet_ifaof, inet_netof:
flpd: DARPA
whois: ask the ARPA
puproute: find a route for a Pup
ip: Internet Protocol
inet: Internet Protocol family
iphostid: set or print
ipbroadcast: broadcasting
puprouting: Pup
intcp: Internet Control Protocol
udp: Internet User Datagram Protocol
spline: interpolate smooth curve.
ap: apl
implog: IMP log
lisp: lisp
pix: Pascal
pix: Pascal
interpreter AND executing.

atomiically release blocked signals and wait for
pupint, puppoint: enable or disable

oninit: process
intro: introduction to commands
intro: introduction to compatibility library functions.
text: introduction to FORTRAN library functions.
text: introduction to C-like library functions.
text: introduction to special files and hardware support.
text: introduction to system calls and error numbers.
text: introduction to system maintenance and operation

commands: intro
ncheck: generate names from
in a bibliography. indexbib, lookibib: build
tread, wrtie, trewin, tskip, tisate: f77 tape
tcinit: change f77

numbers

1/O, topen, telose.
1/O initialization.
1/O multiplexing.
1/O printf programmer.
1/O statistics.
1/O to/from a process.
ioctl: control device.
tcinit: change f77 1/O initialization.

imput(4)
ht(4)
hy(4)
ix(4)
im(4)
imp(4)
lx(4)
mt(4)
mtn(4)
pl(4)
plot(5)
p(4)
p(4)
rx(4)
tm(4)
tc(4)
tty(4)
tu(4)
ux(4)
va(4)
vp(4)

inetconf(8C)
telnet(1C)
impt(4)
swapon(2)
sendmail(8)
inetconf(3n)
flpd(8C)
whois(1C)
puproute(9)
inetpt(4P)
inetfpt(4F)
iphostid()
iphbcast(4P)
puproute(9)
inetrp(4P)
spline(1G)
ax(1)
implog(8C)
lisp(1)
xp(1)
xp(1)
ch(1)
pipe(2)

twic

Distribution

Fourth Berkeley Distribution

August 1983
Permutated Index
Fourth Berkeley Distribution

August 1983
Permuted Index

continue: cycle in loop.
end: terminate loop.
foreach: loop over list of names.
loc: software library.
library.
object library.

mklost + found: make a lost + found directory for file.
lower: lower the case of a filename.

p: print line printer.

ht: uurec: receive processed news articles via TM-03/TE-16, TU-45, TU-77 MASSBUS recnews: receive unprocessed articles via newatmases: rebuild the data base for the mailcheck: find out if a user has mail.
binmail: send or receive mail.
biff: be notified if mail arrives and who it is from.

mailcheck: find out if a user has mail.

mail: send and receive mail.
mailaddr: mail addressing description.
mail alias file.
mail aliases.
mail among users.
messages: system messages and junk.
mail: handle remote mail.

host: Mailing list, forwarding, and alias manager.
mail: Mailer.
mem: memory.
make: maintain program groups.
make: make.
make: archiver and library intro: introduction to system puprouting: Pup Internet routing table backup: make a backup version copy of a file.
mkdir: make a directory.

Isprint, iscntrl.
ht: uurec: receive processed news articles via TM-03/TE-16, TU-45, TU-77 MASSBUS recnews: receive unprocessed articles via newatmases: rebuild the data base for the mailcheck: find out if a user has mail at a host.
make.
man: Mailing list, forwarding, and alias manager.
mem: main memory.
make: maintain program groups.
make: archiver and library intro: introduction to system puprouting: Pup Internet routing table backup: make a backup version copy of a file.
mkdir: make a directory.

ANSI

machine: machine-machine communication (obsolete).
machines.
messages

lh(1)
ll(1)
m4(1)
m5(1)
mklost(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mnew(1)
mkdir: make a directory file.
link: make a hard link to a file.
bibtext: make a LaTeX bibliography.
link: make a link to an existing file.
many: remove a link.
find: make a lost+found directory for links.
serveto: make a MimeServices request.
mktemp: make a unique file name.
lnd: make links.
symlink: make symbolic link to a file.
makedev: make system special files.
vwidth: make troff width table for a font.
script: make typescript of terminal session.
makedep: construct dependency lines for makefiles.
create: makefile; buildmake:
makekey: generate encryption key.
makelong, getHiWord, getLoWord: miscellaneous routines.
putmisc: makekey.
route: manually manipulate the routing tables.
mailer: Mailing list, forwarding, and alias manager.
shift: manipulate argument list.
quotas: manipulate disk quotas.
PutBCPLstring, GetBCPLstring: manipulate strings. (pupstring)
route(8C): manipulate the routing tables.
unlun: manage user login accounts (create, modify, destroy)
masl: manage user login accounts.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
makedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
machedev: make system special files.
makedev: buildmake:
mkst: memory allocator.
mktemp: memory allocator.
mktemp: make system special files.
makedep: construct dependency lines for makefiles.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vfork()</td>
<td>spawn new process in a virtual memory efficient way.</td>
</tr>
<tr>
<td>abort()</td>
<td>terminate abruptly with memory image.</td>
</tr>
<tr>
<td>core()</td>
<td>format of memory image file.</td>
</tr>
<tr>
<td>vmtask()</td>
<td>report virtual memory statistics.</td>
</tr>
<tr>
<td>merge()</td>
<td>three-way file merge.</td>
</tr>
<tr>
<td>sort()</td>
<td>sort files.</td>
</tr>
<tr>
<td>resmerge()</td>
<td>merge RCS revisions.</td>
</tr>
<tr>
<td>pmerge()</td>
<td>three-way file merge.</td>
</tr>
<tr>
<td>mkstr()</td>
<td>create an error message file by massaging C source.</td>
</tr>
<tr>
<td>recv(), recvfrom(), recvmsg()</td>
<td>message from a socket.</td>
</tr>
<tr>
<td>PupilMsg()</td>
<td>human-readable error messages from Pupil package routines.</td>
</tr>
<tr>
<td>sendmsg()</td>
<td>message to one or all users at a Pupil host.</td>
</tr>
<tr>
<td>netsend()</td>
<td>send a short message to one or more users on the Ethernet.</td>
</tr>
<tr>
<td>error()</td>
<td>analyze and handle system error messages.</td>
</tr>
<tr>
<td>perror()</td>
<td>gerror. errno: error code.</td>
</tr>
<tr>
<td>syslog()</td>
<td>log system messages.</td>
</tr>
<tr>
<td>netsend()</td>
<td>send a short message to one or more users on the Ethernet.</td>
</tr>
<tr>
<td>miscesrvreq()</td>
<td>make a special request.</td>
</tr>
<tr>
<td>misk()</td>
<td>create an error message file by massaging C source.</td>
</tr>
<tr>
<td>mkfs()</td>
<td>construct a file system.</td>
</tr>
<tr>
<td>mkisod()</td>
<td>make a directory file.</td>
</tr>
<tr>
<td>mknod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>make a special file.</td>
</tr>
<tr>
<td>mkproto()</td>
<td>construct a prototype file system.</td>
</tr>
<tr>
<td>source()</td>
<td>miscellaneous Pup routines.</td>
</tr>
<tr>
<td>miscsvr()</td>
<td>Miscellaneous Services server for Pup.</td>
</tr>
<tr>
<td>filetime()</td>
<td>tell system error messages.</td>
</tr>
<tr>
<td>intro()</td>
<td>introduction to miscellaneous library functions.</td>
</tr>
<tr>
<td>.miscellaneous()</td>
<td>miscellaneous useful information.</td>
</tr>
<tr>
<td>msg()</td>
<td>system error messages.</td>
</tr>
<tr>
<td>msg()</td>
<td>send system error messages.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkisod()</td>
<td>make a directory file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>make a special file.</td>
</tr>
<tr>
<td>mkproto()</td>
<td>construct a prototype file system.</td>
</tr>
<tr>
<td>source()</td>
<td>miscellaneous Pup routines.</td>
</tr>
<tr>
<td>miscesrv()</td>
<td>Miscellaneous Services server for Pup.</td>
</tr>
<tr>
<td>filetime()</td>
<td>tell system error messages.</td>
</tr>
<tr>
<td>intro()</td>
<td>introduction to miscellaneous library functions.</td>
</tr>
<tr>
<td>miscellaneous()</td>
<td>miscellaneous useful information.</td>
</tr>
<tr>
<td>msg()</td>
<td>send system error messages.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkisod()</td>
<td>make a directory file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>make a special file.</td>
</tr>
<tr>
<td>mkproto()</td>
<td>construct a prototype file system.</td>
</tr>
<tr>
<td>source()</td>
<td>miscellaneous Pup routines.</td>
</tr>
<tr>
<td>miscesrv()</td>
<td>Miscellaneous Services server for Pup.</td>
</tr>
<tr>
<td>filetime()</td>
<td>tell system error messages.</td>
</tr>
<tr>
<td>intro()</td>
<td>introduction to miscellaneous library functions.</td>
</tr>
<tr>
<td>miscellaneous()</td>
<td>miscellaneous useful information.</td>
</tr>
<tr>
<td>msg()</td>
<td>send system error messages.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkisod()</td>
<td>make a directory file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>make a special file.</td>
</tr>
<tr>
<td>mkproto()</td>
<td>construct a prototype file system.</td>
</tr>
<tr>
<td>source()</td>
<td>miscellaneous Pup routines.</td>
</tr>
<tr>
<td>miscesrv()</td>
<td>Miscellaneous Services server for Pup.</td>
</tr>
<tr>
<td>filetime()</td>
<td>tell system error messages.</td>
</tr>
<tr>
<td>intro()</td>
<td>introduction to miscellaneous library functions.</td>
</tr>
<tr>
<td>miscellaneous()</td>
<td>miscellaneous useful information.</td>
</tr>
<tr>
<td>msg()</td>
<td>send system error messages.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkisod()</td>
<td>make a directory file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>build special file.</td>
</tr>
<tr>
<td>mkmod()</td>
<td>make a special file.</td>
</tr>
<tr>
<td>mkproto()</td>
<td>construct a prototype file system.</td>
</tr>
<tr>
<td>source()</td>
<td>miscellaneous Pup routines.</td>
</tr>
<tr>
<td>miscesrv()</td>
<td>Miscellaneous Services server for Pup.</td>
</tr>
</tbody>
</table>
random, random, initstate, setstate: better random

random, random, initstate, setstate: better random
host. iphostid: set or print Internet Protocol (IP) identifier of current...
host: print IP host names and addresses. host(1)
net: print IP net names and addresses. net(1)
banner: print large banner on printer. banner(6)
calen: print large-format calendar. calen(1)
linelen: print line lengths for a text file. linelen(1)
allusers: print list of all authorized users. allusers(1)
files, rlog: print log messages and other information about RCS... rlog(1)
nm: print name list. nm(1)
nm68: print name list of MC68000 object files. nm68(1)
hostname: set or print name of current host system. hostname(1)
catboisc: convert C/A/T files to DVI format and... catboisc(1)
vdetect: inspect and vdetect(1)
rmall: remove all jobs from the line. rmall(1)
pmail: print out information about UNIX... pmail(1)
printenv: print out mail in the post office. printenv(1)
man: find manual information by keywords;... man(1)
puproute: print Pup network routing table information. puproute(1)
68000, r68: print relocation commands in a .b file for the... r68(1)
pstal: print system facts. pstal(8)
pagesize: print system page size. pagesize(1)
imprint - cz(earina): convert files to press format and... cz(1)
convert dvi (Ti'X output) files to press format and... convert(1)
presssimp: convert press files to ImPress format and... presssimp(1)
diction: explain - explain, diction - explain, diction -
file: strings; find the

vlp: Format Lisp programs to be

banner: print large banner on
broff: troff to the ImPress
broff: troff to the ImPress
lp: line
to ImPress format and print them on the ImPress
print: pr to the line
yapp: yet another pretty
printcap: printer capability data base.
lp: line
lpd: line
dpq: prints the Dover
dprr: remove a file from the Dover
dpr: doper
lpm: remove jobs from the line
boisc: send send to the HP2680a
dviboisc: send DVI files to the HP2680a

vpr, vprm, vn, vprint: raster
Pup. (pupprint) PupPrint, PrintErrorPUP, PupPrint, size68:
(pupprint) PupPrint, PrintErrorPUP, PupPrint, size68:
dpq: prints the Dover printer queue.

adduser: reboot: UNIX bootstrapping
nice: run low priority
stop: halt a job or
_exit: terminate a
fork: create a new
fork: create a copy of this
implogd: IMP logger
kill: send signal to a
kill: send a signal to a
popen, pclose: inita/0 to/from a
wait: await completion of
exit: terminate a
init: rebuild all users

Reboot: UNIX bootstrapping

Adduser: rebuild all users

Stop: halt a job or

Exit: terminate a

Fork: create a new

Kill: send signal to a

Popen, pclose: initiate I/O to/from a

Wait: await completion of

Init: rebuild all users

Adduser: reboot: UNIX bootstrapping

Stop: halt a job or

Exit: terminate a

Fork: create a new

Kill: send signal to a

Popen, pclose: initiate I/O to/from a

Wait: await completion of

Init: rebuild all users

Adduser: reboot: UNIX bootstrapping

Stop: halt a job or

Exit: terminate a

Fork: create a new

Kill: send signal to a

Popen, pclose: initiate I/O to/from a

Wait: await completion of

Init: rebuild all users

Adduser: reboot: UNIX bootstrapping

Stop: halt a job or

Exit: terminate a

Fork: create a new

Kill: send signal to a

Popen, pclose: initiate I/O to/from a

Wait: await completion of

Init: rebuild all users

Adduser: reboot: UNIX bootstrapping

Stop: halt a job or

Exit: terminate a
permuted index

August 1983
1xx
Fourth Berkeley Distribution
address. attributes: get
locate; location and owner of
Pup Chans and fids. (pupdescrip) pupin descrip, pupout descrip.
Pup Packets. pupint.
Pup Port address to a name.
pup: raw PUP socket interface.
Pup routines /getlong, makelong, getshort.
Pup server.
PUP socket interface.
PUP Telnet Protocol Service.
pup: Xerox PUP-I protocol family.
pup10rptser: Pup Gateway Info routing table server.
Pup-based subnet routing.
pupchan: data structure describing a Pup channel.
PupChans and fids. (pupdescrip) pupin descrip, pupout descrip.
pupispec: close a pup channel.
pupdescrip: access mapping between PupChans and/ server.
eftpnot: receive-only
eftpnot: send-only
package routines.
pllpdcscrip: access mapping between pup channel.
local end of pup channel.
channel. pup channel. pup channel routing.
calling one of connected networks. puplisten.
address of local, remote ends of pup channel.
(channel) pup: Pup Xerox 1-protocol family.
maplec between PupChans and fids. (pupdescrip) received Pup Packets.
connected networks. puplisten.
endpupnettab: pup configuration table.
channel.
Pup Packets. pupint.
PupChans and fids. (pupdescrip) routines for Pup. (pupprint)
printing routines for Pup.
channel.
checksum: compute
computer via Pup network.
(pupprint) PupPrint, PrintErrorPUP.
drb: DR11-B/DR11-W general
ugentc: push character back into input stream.
push: purpose user device interface.
putc, putchar, put, putw:
put a string on a stream.
putc, putchar, put, putw: put character or word on a stream.
PutBCPlstring, GetBCPlstring. GetMesstring.
putc, putchar, put, putw: write a character to a fortran logical
unit.
on a stream.
Gettype.
putc, putchar, put, putw: put character or word on a stream.
putc, putchar, put, putw: put character or word on a stream.
putc, putchar, put, putw: put character or word on a stream.
putc, putchar, put, putw: put character or word on a stream.
putc, putchar, put, putw: put character or word on a stream.
pwd: working directory name.
PutBCPlstring, GetBCPlstring. GetMesstring.
<table>
<thead>
<tr>
<th>Command/Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd</td>
<td>change directory</td>
</tr>
<tr>
<td>eval</td>
<td>execute a command</td>
</tr>
<tr>
<td>exec</td>
<td>execute a command</td>
</tr>
<tr>
<td>exit</td>
<td>exit the shell</td>
</tr>
<tr>
<td>export</td>
<td>export variables</td>
</tr>
<tr>
<td>login</td>
<td>login to system</td>
</tr>
<tr>
<td>read</td>
<td>read input</td>
</tr>
<tr>
<td>bad144</td>
<td>read/write dce standard 144 bad sector information</td>
</tr>
<tr>
<td>bseek</td>
<td>move pointer</td>
</tr>
<tr>
<td>setgid</td>
<td>set real and effective group ID</td>
</tr>
<tr>
<td>seteuid</td>
<td>set real and effective user ID's</td>
</tr>
<tr>
<td>malloc</td>
<td>memory allocator</td>
</tr>
<tr>
<td>rcadonly</td>
<td>rearrange name list</td>
</tr>
<tr>
<td>set</td>
<td>set command</td>
</tr>
<tr>
<td>shift</td>
<td>shift command</td>
</tr>
<tr>
<td>times</td>
<td>time of day</td>
</tr>
<tr>
<td>trap</td>
<td>catch signals</td>
</tr>
<tr>
<td>umask</td>
<td>set file access permissions</td>
</tr>
<tr>
<td>wait</td>
<td>wait for a process to complete</td>
</tr>
<tr>
<td>eyacc</td>
<td>modified yacc allowing much improved error recovery</td>
</tr>
<tr>
<td>pupgetsrcport</td>
<td>network protocol for returning a stream between a server and client</td>
</tr>
<tr>
<td>pupgetdstport</td>
<td>network protocol for returning a stream between a server and client</td>
</tr>
<tr>
<td>build inv</td>
<td>build inverted index for a bibliography</td>
</tr>
<tr>
<td>refer</td>
<td>find and insert literature references in documents</td>
</tr>
<tr>
<td>strip</td>
<td>remove symbols and relocation commands in a file</td>
</tr>
<tr>
<td>readlink</td>
<td>read value of a symbolic link</td>
</tr>
<tr>
<td>recv</td>
<td>receive a message from a socket</td>
</tr>
<tr>
<td>recvfrom</td>
<td>receive a message from a socket</td>
</tr>
<tr>
<td>recvmsg</td>
<td>receive a message from a socket</td>
</tr>
<tr>
<td>re-evaluate</td>
<td>re-evaluate shell data</td>
</tr>
<tr>
<td>re-exec</td>
<td>regular expression handler</td>
</tr>
<tr>
<td>documents</td>
<td>reference C source files</td>
</tr>
<tr>
<td>cref</td>
<td>reference program</td>
</tr>
<tr>
<td>lexref</td>
<td>reference program</td>
</tr>
<tr>
<td>realpath</td>
<td>absolute path name of file</td>
</tr>
<tr>
<td>leave</td>
<td>remind you when you have to leave</td>
</tr>
<tr>
<td>calendar</td>
<td>reminder service</td>
</tr>
<tr>
<td>ruserok</td>
<td>routines for returning a stream to a remote machine</td>
</tr>
<tr>
<td>puptelnet</td>
<td>connect your terminal to a remote</td>
</tr>
</tbody>
</table>
rcmd, rresport, ruserok: routines for returning a stream to a remote command.

rev: reverse lines of a file.
rev68: reverse byte order in 68000 .b and .68.
reverse byte order in 68000 .b and .68 (b.out).
reverse line feeds in text.
reverse lines of a file.

lastcomm: show last commands executed in reverse order.

ci: check in RCS
cc: check out RCS
readdir: compare RCS revisions.
remerge: merge RCS revisions.

seek, flcbl:
ringbuf: ring buffer package.

hk: RK6-11/RK06 and RK07 moving head disk.
hk: RK6-11/RK06 and RK07 moving head disk.

about RCS files.

rm, rmdir: remove (unlink) directories or files.
rm, rmdir: remove (unlink) directories or files.
rm, rmdir: remove (unlink) directories or files.
rm, rmdir: remove (unlink) directories or files.
rm, rmdir: remove (unlink) directories or files.
rm, rmdir: remove (unlink) directories or files.
rmdir: remove (unlink) directories or files.
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).

rdir: remote tape manipulation programs.

roffbibs: run off bibliographic database.

roguics: Exploring The Dungeons of Doom.

roftbin: handling remote mail received via uucp.

remote command. rcmd, rresport, ruserok: routines for returning a stream to a remote command. rcmd, rresport, ruserok: routines for returning a stream to a remote command.

Permutated Index

rcmd(3X)
rev(1)
rev68(1)
reverse byte order in 68000 .b and .68.
reverse byte order in 68000 .b and .68 (b.out).
reverse line feeds in text.
reverse lines of a file.

lastcomm(1)
ci(1)
cc(1)
readdir(1)
remerge(1)
ringbuf(9)

rm(1)
rmdir(1)
rmdir(1)
rm(1)
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).
rmdir: remove (directory file).

route: network routing daemon.
routing: manually manipulate the routing tables.
routc: network routing daemon.

patchroute: patchroute(8)
patchroute(8)

patchroute(8)
routc(8C)

rsh(1C)

rshd(8C)
rsh(8C)

remote command. rcmd, rresport, ruserok: routines for returning a stream to a remote command.
### Permuted Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rwho</td>
<td>who's logged in on local machines</td>
</tr>
<tr>
<td>rwho(1C)</td>
<td></td>
</tr>
<tr>
<td>rwho(8C)</td>
<td></td>
</tr>
<tr>
<td>rx: DEC</td>
<td>DEC RX02 floppy disk interface</td>
</tr>
<tr>
<td>rx(4)</td>
<td></td>
</tr>
<tr>
<td>rxf:</td>
<td>format floppy disks</td>
</tr>
<tr>
<td>rxf(4)</td>
<td></td>
</tr>
<tr>
<td>o68: s *</td>
<td>optimizer component of o68</td>
</tr>
<tr>
<td>o68(1)</td>
<td></td>
</tr>
<tr>
<td>o68: s -</td>
<td>optimizer component of o68</td>
</tr>
<tr>
<td>o68(1)</td>
<td></td>
</tr>
<tr>
<td>cm68: s *</td>
<td>translator component of o68</td>
</tr>
<tr>
<td>cm68(1)</td>
<td></td>
</tr>
<tr>
<td>mbootdir:</td>
<td></td>
</tr>
<tr>
<td>mbootdir(8)</td>
<td></td>
</tr>
<tr>
<td>flpd: DARPA Internet File Transfer Protocol</td>
<td></td>
</tr>
<tr>
<td>gatewayinfo: Pup GatewayInfo routing table</td>
<td></td>
</tr>
<tr>
<td>inetd: DARPA little protocol server.</td>
<td></td>
</tr>
<tr>
<td>inetd(8C)</td>
<td></td>
</tr>
<tr>
<td>leaf: PUP Leaf Remote File Access Protocol</td>
<td></td>
</tr>
<tr>
<td>leaf(8)</td>
<td></td>
</tr>
<tr>
<td>mbootdir:</td>
<td></td>
</tr>
<tr>
<td>mbootdir(9)</td>
<td></td>
</tr>
<tr>
<td>mtimcheck:</td>
<td></td>
</tr>
<tr>
<td>mtimcheck(9)</td>
<td></td>
</tr>
<tr>
<td>pup1arpser:</td>
<td></td>
</tr>
<tr>
<td>pup1arpser(8)</td>
<td></td>
</tr>
<tr>
<td>pupecho:</td>
<td></td>
</tr>
<tr>
<td>pupecho(1)</td>
<td></td>
</tr>
<tr>
<td>pup echo protocol user and</td>
<td></td>
</tr>
</tbody>
</table>

### Additional Commands

- **getpriority, setpriority:** get/set program scheduling priority.
- **clear:** clear terminal repeatedly display output of command on terminal.
- **emacs:** a screen editor.
- **curses:** screen functions with "optimal" cursor motion.
- **ex:** vi oriented (visual) display editor based on terminal screen.
- **unssrbe:** remove script for auto-reboot and daemons.
- **rc:** command schedule process interrupts in command.
- **grep:** search for and print header (include) files.
- **include:** include search for and print header (include) files.
- **xsend, xget:** enroll mail.
- **badsect:** create files.
- **netinfo:** routing information.
- **rwho:** who's logged in on local machines.
- **sendmail:** send mail over the internet.
- **sendmsg:** send a message from a socket.
- **killpg:** send signal to a process group.
- **alias:** aliases file for sending mail.
- **send:** send mail.
- **sendmail:** send mail to a remote host.
- **sendto:** send a message from a socket.
- **sendmail:** send mail over the internet.
- **rwhod:** who's logged in on local machines.
- **send:** send mail.
- **sendmail:** send mail over the internet.
- **rwhod:** system status server.
- **mtimcheck:** get time from a Pup.
- **mbootdir:** get a boot file directory from a boot.
- **mtimcheck:** get time from a Pup.
- **inetd:** DARPA little protocol server.
- **inetd(8C)***
- **kill:** send signal to a process.
- **mess:** send a message to one or all users at a Pup host.
- **mbootdir:** get a boot file directory from a boot.
- **kill:** send signal to a process.
- **inetd:** DARPA little protocol server.
- **inetd(8C)***
- **alias:** aliases file for sending mail.
- **send:** send mail.
- **sendmail:** send mail over the internet.
continue, cd, eval, exec, exit, export, login;/sh, for case, if, while, break, shar: produce shell-script archives.

xstr: extract strings from C programs to implement shared, produce shell-script archives.

csh: change default login shell.

eval: re-evaluate shell data.

exit: leave shell.

exec: overlay shell directory stack.

rsh: remote shell.

set: change value of shell variables.

system: issue a shell command.

chsh: change default login shell.

echo: atomically release blocked signals and sigpause: atomically release blocked signals and wait for interrupt.

alias: shell macros.

suspend: suspend a shell, resuming its superior.

popd: pop shell directory stack.

sigpause: atomically release blocked signals.

set: change value of shell variables.

unset: discard shell variables.

exec: overlay shell directory stack.

exec: overlay shell with specified command.

exec: overlay shell with specified command.

rsh: remote shell.

rsh: remote shell.

rsh: remote shell.

exec: overlay shell with specified command.

rsh: remote shell.

rsh: remote shell.

sigvec: software signal facilities.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.

signal: change the action for a signal.
bad144: read/write dec
stdio: standard buffered I/O package
patchroute: print command routing
reset: reset the tty's state and get terminal
fstat: file status of a file given its name
hashstat: display system status on
exit: terminate process with status.
netstat: show network
fmove: move a file
ps: process list
stat, lstat, fstat, get file
fstat: filter file names according to command in a
ferror, feof, clearerr, fileno: stream
sysline: display system status on
etherport: show status of ethernet minor devices.
ungetc: put character back into
getline, getline, getw: get character or word from a
putchar, putc, putw: put character or word on a
putc, fputs: put a string on a
setbuf, setlinebuf: assign buffering to a
putc, fputc, putc: put character or word on a
putc: return the teletype bits to a sensible
fdate: return date and time in an ASCII string.
rindex: string operations. strcat, strncat, strcmp, strncmp, strcpy, strncpy, strlen, index,
ungetc: push character back into input
hardware: convert NIC to UNIBUS
stdin, index, rindex: string operations.
scanf: scan a formatted string
put: halt a job or process
halt: halt the processor.

- lxxx -

Fourth Berkeley Distribution
patchroute: kludge to support Stanford Pup-based routing.

alarm: execute a subroutine after a specified time.

fetch, store, delete, firstkey, nexthkey: data base routines.

lib2648: subroutines for the HP 2648 graphics terminal.

su: substitute user id temporarily.

sum: sum and count blocks in a file.

du: summarize disk usage.

quot: summarize file system ownership.

quota: summarize quotas for a file system.

SUppOIt: • ••••••

Sanford Pup-based routines. port:

Sutherland Picture System 2 graphics device.

interface. ps: Evans and Sutherland Picture System 2 graphics device.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.

swap: swap device for interleaved paging/swapping.

swab: swap bytes.

swab: swap bytes.
FOU1th
August 1983
Permuted
Screen: repeatedly display output of command on
Index
vhangup:
sysline: display system
lib2648: subroutines for the
tgetent, tgetnum, tgetnag.
tclose, tread, twrite, trewin, tskipf, tstatc:
topen, tgoto, tputs:
tty: pseudo
termcap: terminal capability data base.
termcap: terminal capability data base.
termcap: terminal configuration data base.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
termcap: terminal dependent initialization.
Fourth Berkeley Distribution

- lxxxiii -

August 1983
permuted Index

- lxxxiv -

Fourth Berkeley Distribution
Permutted Index

sh: shell; perform the command language.

unbf: unbuffered.

unid(2): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

uninode(1): unique file node.

unique(1): unique identifier.

uninode(1): unique file node.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.

unix(2): unique file system.

unic(1): unique file identifier.

unilines: unique line identifier.
mail: handle remote mail received via uuucp.
usnnap: show snapshot of the system.
connection via mail. uuencode, transmission via mail.
uuencode: format of an encoded uuencode file.
transmission via mail.
uucp, uuilog: unix to unix copy.
unx: receive processed news articles via mail.
udump: send a file to a remote host.
usnnap: show snapshot of the UUCP system.
uux: unix to unix command execution.
va: Benson-Varian interface.
valloc: aligned memory allocator.
@: arithmetic on shell variables.
abs: integer absolute value.
fabs, floor, ceil: absolute value.
getenv: variable argument list.
getenv: set variable in environment.
@: arithmetic on shell variables.
unset: discard shell variables.
unsetenv: remove environment variables.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
getenv: get value of environment variable.
getenv: get value of environment variable.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
setenv: set variable environment name.
getenv: get value of environment variable.
setenv: set value of shell variable.
Permuted Index

vhangup: virtually "hangup" the current control terminal.

vi: screen oriented consumption.

vtnit: control maximum system resource

vtnit: (visual) display editor based on ex.

vspr: Format Lisp programs to be printed with nrotT,

vtnit: Format Lisp programs to be printed with nrotT,

vmlit: control maximum system resource

vmlit: Format Lisp programs to be printed with nrotT,

vmstat: report virtual memory statistics.

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,

vmlit: Format Lisp programs to be printed with nrotT,
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>putc</td>
<td>write a character to a Fortran logical unit.</td>
<td>putc(3F)</td>
</tr>
<tr>
<td>pupwrite</td>
<td>write a packet to a PUP channel.</td>
<td>pupwrite(9)</td>
</tr>
<tr>
<td>enwrite</td>
<td>write a packet to the ethernet.</td>
<td>enwrite(9)</td>
</tr>
<tr>
<td>ansi</td>
<td>read and write ANSI format magnetic tapes.</td>
<td>ansi(1)</td>
</tr>
<tr>
<td>write, writev</td>
<td>write on a file.</td>
<td>write(2)</td>
</tr>
<tr>
<td>wall</td>
<td>write to all users.</td>
<td>wall(1)</td>
</tr>
<tr>
<td>write</td>
<td>write to another user.</td>
<td>write(1)</td>
</tr>
<tr>
<td>write, writev</td>
<td>write to another user.</td>
<td>write(1)</td>
</tr>
<tr>
<td>writev</td>
<td>write on a file.</td>
<td>write(2)</td>
</tr>
<tr>
<td>open</td>
<td>open a file for reading or writing, or create a new file.</td>
<td>open(2)</td>
</tr>
<tr>
<td>utmp</td>
<td>wtmp: login records.</td>
<td>utmp(5)</td>
</tr>
<tr>
<td>en</td>
<td>Xerox 3 Mbit/s Ethernet interface.</td>
<td>en(4)</td>
</tr>
<tr>
<td>pup</td>
<td>Xerox PUP-1 protocol family.</td>
<td>pup(4F)</td>
</tr>
<tr>
<td>xsend</td>
<td>xget, xenroll: secret mail.</td>
<td>xsend(1)</td>
</tr>
<tr>
<td>bit</td>
<td>xor, not, nshift, lshift bitwise functions.</td>
<td>bit(3F)</td>
</tr>
<tr>
<td>shared</td>
<td>xor, extract strings from C programs to implement</td>
<td>xstr(1)</td>
</tr>
<tr>
<td>strings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>j0, j1, jn</td>
<td>y0, y1, yn: Bessel functions.</td>
<td>j0(3M)</td>
</tr>
<tr>
<td>eyacc</td>
<td>yacc: allowing much improved error recovery.</td>
<td>eyacc(1)</td>
</tr>
<tr>
<td>modified</td>
<td>yacc: yet another compiler-compiler.</td>
<td>yacc(1)</td>
</tr>
<tr>
<td>yapp</td>
<td>yapp: yet another pretty printer.</td>
<td>yapp(1)</td>
</tr>
<tr>
<td>yes</td>
<td>yess: be repetitively affirmative.</td>
<td>yes(1)</td>
</tr>
<tr>
<td>j0, j1, jn, y0, y1</td>
<td>yn: Bessel functions.</td>
<td>j0(3M)</td>
</tr>
<tr>
<td>zork</td>
<td>zork: the game of dungeon.</td>
<td>zork(6)</td>
</tr>
</tbody>
</table>