

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

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

Copyright 1979, Bell Telephone Laboratories, Incorporated.
Holders of a UNIXTM/32V software license are permitted to
copy this document, or any portion of it, as necessary for
licensed use of the software, provided this copyright notice and
statement of permission are included.

PREFACE

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 advice 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 4.1 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 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 Kridle, 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. Babaoğlu
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 Kridle, 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. Babaoğlu

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 *bin*ary 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 *ty(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(1)* 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 '#').

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/lem/filex* refers to the file *filex* in the directory *lem*; *lem* is itself a subdirectory of *usr*; *usr* 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 *ls(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 *f77(1)*, the Pascal compiler *pc(1)*, and interpreter *pi(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 in response 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.

Sty and *gty*. These system calls have been extensively altered, see *ioctl*(2) and *ty*(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 don't 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 *ty(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 *<sys/dir.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

intro	introduction to commands
adb	debugger
addbib	create or extend bibliographic database
allusers	print list of all authorized users
altoload	load files from an Alto FTP
ansi	read and write ANSI format magnetic tapes
apl	apl interpreter
apply	apply a command to a set of arguments
apropos	locate commands by keyword lookup
ar	archive and library maintainer
arptab	show contents of kernel ARP table
as	VAX-11 assembler
as68	.a68 -> .b assembler component of cc68
at	execute commands at a later time
awk	pattern scanning and processing language
backup	make a backup version copy of a file
basename	strip filename affixes
bboard	bulletin board reading program
bc	arbitrary-precision arithmetic language
bibtex	make a LaTeX bibliography
biff	be notified if mail arrives and who it is from
binmail	send or receive mail among users
boise	send files to the HP2680a printer using TCP
btroff	troff to the ImPrint printer
buildmake	preprocessor to provide extended syntax for makefiles
cal	print calendar
calen	print large-format calendar
calendar	reminder service
cat	catenate and print
catboise	convert C/A/T files to DVI format and print on Boise
catdvi	convert C/A/T files to DVI format
cb	C program beautifier
cc	C compiler
cc68	C compiler for the MC68000
ccom68	.c -> .s translator component of cc68
cd	change working directory
checknr	check nroff/troff files
chfn	change finger entry
chgrp	change group
chmod	change mode
chsh	change default login shell
ci	check in RCS revisions
clear	clear terminal screen
cmp	compare two files
cnest	check for nested comments in C code
co	check out RCS revisions
col	filter reverse line feeds
colert	filter nroff output for CRT previewing
colrm	remove columns from a file
comm	select or reject lines common to two sorted files
compact	compress and uncompress files, and cat them
congraph	plot connectivity of a graph
cp	copy

cparen	cparen - add parentheses to C expressions
cref	cross reference program
crypt	encode/decode
csh	a shell (command interpreter) with C-like syntax
ctags	create a tags file
cxref	cross reference C source files
cz	convert files to press format and print them on the Dover.
dataio	load the data i/o prom programmer
date	print and set the date
s t -	list contents of an Emacs data base .br d b p r i n t - print an entry from an Emacs database
dbx	debugger
dc	desk calculator
dcat	convert troff phototypesetter output files to press format and print them on the Dover.
dd	convert and copy a file
ddt68	symbolic debugger for 68000
deroff	remove nroff, troff, tbl and eqn constructs
detex	remove TeX constructs
df	disk free
diction	print wordy sentences; thesaurus for diction
diff	differential file and directory comparator
diff3	3-way differential file comparison
dl68	b.out -> .dl downloader component of cc68
dlx	download with error correction - 68000 Sun1 monitor
dpq	prints the Dover printer queue
dpr	dover printer spooler
dprm	remove a file from the Dover printer queue
dtree	print directory tree structures
dtroff	troff to the Dover
du	summarize disk usage
dumpfonts	show what Press fonts are available in fonts.widths
dvi Boise	send DVI files to the HP2680a printer using TCP
dviimp	convert DVI files to impress format
dvip	convert a dvi (TeX output) file to press format.
dvipress	convert dvi (TeX output) files to press format and print them on the Dover.
echo	echo arguments
ed	text editor
efl	Extended Fortran Language
eftprec	receive-only PUP/EI*IP file transfer program with routing
eftpsend	send-only PUP/EI*IP file transfer program with routing
emacs	a screen editor
eqn	typeset mathematics
error	analyze and disperse compiler error messages
etherport	show status of ethernet minor devices
ex	text editor
exlog	extract data from system load log file
expand	expand tabs to spaces, and vice versa
explain	explain, diction - print wordy sentences; thesaurus for diction
expr	evaluate arguments as an expression
eyacc	modified yacc allowing much improved error recovery
f77	Fortran 77 compiler
false	provide truth values
fed	font editor
file	determine file type
find	find files
fing	front end for finger
finger	user information lookup program

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
ld	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
linclen	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

lxfref	lisp cross reference program
m4	macro processor
mail	send and receive mail
mailcheck	find out if a user has mail at a PUP host
make	maintain program groups
makedep	construct dependency lines for makefiles
man	find manual information by keywords; print out the manual
merge	three-way file merge
msg	permit or deny messages
mkdir	make a directory
mkstr	create an error message file by massaging C source
mod	Modula-2 compiler
more	file perusal filter for crt viewing
msgs	system messages and junk mail program
mt	magnetic tape manipulating program
mv	move or rename files
net	print IP net names and addresses
netalias	keeping track of remote user names and passwords
netscnd	send a short message to one or more users on the Ethernet
netstat	show network status
netupd	update a directory from one on another system
newaliases	rebuild the data base for the mail aliases file
nice	run a command at low priority (<i>sh</i> only)
nm	print name list
nm68	print name list of MC68000 object files
nroff	text formatting
o68	.s -> .s optimizer component of cc68
od	octal, decimal, hex, ascii dump
olddb	debugger
pagesize	print system page size
passwd	change login password
pc	Pascal compiler
pc68	Pascal compiler for the MC68000
pdx	pascal debugger
pi	Pascal interpreter code translator
ping	IP/ICMP echo user program
pix	Pascal interpreter and executor
plot	graphics filters
pmerge	pascal file merger
pr	print file
pr68	print extended statistics on .b file
pressimp	convert press files to ImPress format and print them on the ImPrint printer.
print	pr to the line printer
printenv	print out the environment
prmail	print out mail in the post office
prof	display profile data
ps	process status
pti	phototypesetter interpreter
ptx	permuted index
pupecho	Pup Echo protocol user and server
pupftp	Pup File Transfer Program
puproute	print Pup network routing table information
puptelnet	connect your terminal to a remote computer via Pup network
pwd	working directory name
px	Pascal interpreter
pxp	Pascal execution profiler

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
res	change RCS file attributes
rcsdiff	compare RCS revisions
rcsintro	rcsintro - 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
ruptime	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
soelim	eliminate .so's from nroff 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
stty	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
tangle	convert web file into pascal file, tex file
tar	tape archiver

le of Contents

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

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

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
execve	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
getgid	get group identity
getgroups	get group access list
gethostid	get/set unique identifier of current host
gethostname	get/set name of current host
getitimer	get/set value of interval timer
getpagesize	get system page size
getpeername	get name of connected peer
getpgrp	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
mknod	make a special file
mount	mount or remove file system
open	open a file for reading or writing, or create a new file

Table of Contents

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

3. C Library Subroutines

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

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
strcmpfold	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
etime	return elapsed execution time
exit	terminate process with status
fdate	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
hostname	get name of current host
idate	return date or time in numerical form
index	tell about character objects

Table of Contents

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

M. Math Library

intro	introduction to mathematical library functions
exp	exponential, logarithm, power, square root
floor	absolute value, floor, ceiling functions
gamma	log gamma function
hypot	Euclidean distance
j0	bessel functions
sin	trigonometric functions
sinh	hyperbolic functions

N. Internet Network Library

intro	introduction to network library functions
byteorder	convert values between host and network byte order
gethostent	get network host entry
getnetent	get network entry
getprotoent	get protocol entry
getservent	get service entry
inet	Internet address manipulation routines

O. C Standard I/O Library Subroutines

intro	standard buffered input/output package
fclose	close or flush a stream
ferror	stream status inquiries
fopen	open a stream
fread	buffered binary input/output
fseek	reposition a stream
getc	get character or word from stream
gets	get a string from a stream
printf	formatted output conversion
putc	put character or word on a stream
puts	put a string on a stream
scanf	formatted input conversion

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
getfsent	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
getpw	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	get process times
utime	set file 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 LH/DH 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 LH/DH IMP interface
ct	phototypesetter interface
de	DEC DEUNA 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

gmr	Grinnell Systems display
hk	RK6-11/RK06 and RK07 moving head disk
hp	MASSBUS disk interface
ht	TM-03/TE-16,TU-45,TU-77 MASSBUS magtape interface
hy	Network Systems Hyperchannel interface
ik	Ikonas frame buffer, graphics device interface
il	Interlan 10 Mb/s Ethernet interface
imp	1822 network interface
imp	IMP raw socket interface
inet	Internet protocol family
ip	Internet Protocol
ipbroadcast	broadcasting Internet Protocol packets
kg	KI-11/DL-11W line clock
lo	software loopback network interface
lp	line printer
mem	main memory
mt	TM78/TU-78 MASSBUS magtape interface
mtio	UNIX magtape interface
null	data sink
pcl	DEC CSS PCL-11 B Network Interface
ps	Evans and Sutherland Picture System 2 graphics device interface
pty	pseudo terminal driver
pup	Xerox PUP-I protocol family
pup	raw PUP socket interface
rx	DEC RX02 floppy disk interface
tcp	Internet Transmission Control Protocol
tm	TM-11/TE-10 magtape interface
ts	TS-11 magtape interface
tty	general terminal interface
tu	VAX-11/730 and VAX-11/750 TU58 console cassette interface
uda	UDA-50 disk controller interface
udp	Internet User Datagram Protocol
un	Ungermann-Bass interface
up	unibus storage module controller/drives
ut	UNIBUS TU45 tri-density tape drive interface
uu	TU58/DECtape II UNIBUS cassette interface
va	Benson-Varian interface
vp	Versatec interface
vv	Protcon proNET 10 Megabit ring

5. File Formats

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

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

6. Games

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

7. Miscellaneous

intro	miscellaneous useful information pages
ascii	map of ASCII character set
environ	user environment
eqnchar	special character definitions for eqn

Table of Contents

hier	file system hierarchy
mailaddr	mail addressing description
man	macros to typeset manual
me	macros for formatting papers
ms	text formatting macros
term	conventional names for terminals

8. System Maintenance

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

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

Table of Contents

telnetd	DARPA TELNET protocol server
telsr	PUP Telnet Protocol Service
tftpd	DARPA Trivial File Transfer Protocol server
timeck	poll the localnet for the current time
trpt	transliterate protocol trace
tunefs	tune up an existing file system
update	periodically update the super block
utime	adjust the access or modification time of a file
uuclean	uucp spool directory clean-up
uurec	receive processed news articles via mail
uusnap	show snapshot of the UUCP system
vipw	edit the password file

9. Pup library

atoo	convert ASCII to octal numbers
bmove	block move a buffer
byteorder	discussion of byte-ordering and the Pup package
checksum	compute Pup-style checksum
eftp	Pup EFTP package
enarp	Address Resolution Protocol (ARP) routines
enctfilter	build ethernet filters
enflush	flush an ethernet input file
engethost	determine Pup host number of ethernet interface
enopen	open an ethernet file
enread	read a packet from an ethernet file
ensetbacklog	set ethernet input queue backlog
ensignal	enable or disable signal on ethernet packet arrival
enwrite	write a packet to the ethernet
maddtoname	translate a Pup Port address to a name
mattributes	get Pup Network Directory entry attributes for an address
mbootdir	get a boot file directory from a boot server
mbootrequest	request a boot-load
mkissofdeath	send a KissOfDeath Pup
mlookup	translate a name to a Pup address
mmailcheck	find out if a user has new mail at a Pup host
msscrvreq	make a MiscServices request
msendumsg	send a message to one or all users at a Pup host
msunbootreq	request a boot-load
mtimecheck	get time from a Pup server
overview	overview of Pup library routines
pupchan	data structure describing a Pup channel
pupclose	close a pup channel
pupdescrip	access mapping between PupChans and fids
puperrmsg	human-readable error message from Pup package routines
pupgethost	get host, net numbers of local end of pup channel
pupgetport	get address of local, remote ends of pup channel
pupint	enable or disable interrupts for received Pup Packets
puplisten	open Pup channels on all connected networks
pupmisc	miscellaneous Pup routines
pupnettab	pup configuration table
pupopen	open a pup channel
pupport	a data structure used in the Pup package, with support routines
pupprint	printing routines for Pup
pupread	read a pup packet from a pup channel
pupreopen	change the destination for a Pup channel
puproute	find a route for a Pup Internet packet

puprouting	Pup Internet routing table maintenance routines
pupsetbacklog	set input queue backlog for pup channel
pupsetdfilt	set a default packet filter for a Pup channel
pupsetfilter	set the packet filter for a Pup channel
pupsetmode	set and read the mode for a Pup channel
pupsettimeout	set timeout for pup channel
pupstring	manipulate strings
pupwrite	write a packet to a pup channel
ringbuf	ring buffer package
uatimecvt	conversions from Unix to Alto time and vice versa
uniquesocket	create unique socket number

PERMUTED INDEX

	@: arithmetic on shell variables.	csH(1)
as68: .a68	-> .b assembler component of cc68.	as68(1)
ld68: .b	-> b.out linker for the MC68000.	ld68(1)
dl68: b.out	-> .dl downloader component of cc68.	dl68(1)
unpcnt: remove lines beginning with	% from a file.	unpcnt(1)
o68: .s	-> .s optimizer component of cc68.	o68(1)
ccom68: .c	-> .s translator component of cc68.	ccom68(1)
imp:	1822 network interface.	imp(4)
lib2648: subroutines for the IIP	2648 graphics terminal.	lib2648(3X)
ec:	3Com 10 Mb/s Ethernet interface.	ec(4)
breathlife: breath-of-life server for bootloading	3mb Altos.	breathlife(8)
diff3:	3-way differential file comparison.	diff3(1)
sendbug: mail a system bug report to	4bsd-bugs.	sendbug(1)
rev68: reverse byte order in 68000 .b and .	68 (b.out) files.	rev68(1)
size68: prints sizes of segments in a .b or .	68 file.	size68(1)
ddt68, fddt68: symbolic debugger for	68000.	ddt68(1)
print relocation commands in a .b file for the	68000. r168:	r168(1)
rev68: reverse byte order in	68000 .b and .68 (b.out) files.	rev68(1)
dlx: download with error correction -	68000 Sun1 monitor.	dlx(1)
	750rom: details of Vax-11/750 boot ROMs.	750rom(8)
a68: .	a68 -> .b assembler component of cc68.	as68(1)
	aardvark: yet another exploration game.	aardvark(6)
	abort: generate a fault.	abort(3)
	abort: terminate abruptly with memory image.	abort(3F)
abort: terminate	abruptly with memory image.	abort(3F)
	abs: integer absolute value.	abs(3)
abs: integer	absolute value.	abs(3)
fabs, floor, ceil:	absolute value, floor, ceiling functions.	floor(3M)
	ac: login accounting.	ac(8)
	acc: ACC LH/DH IMP interface.	acc(4)
acc:	ACC LH/DH IMP interface.	acc(4)
accept:	accept a connection on a socket.	accept(2)
	accept: accept a connection on a socket.	accept(2)
	access: determine accessibility of a file.	access(3F)
	access: determine accessibility of file.	access(2)
getgroups: get group	access list.	getgroups(2)
initgroups: initialize group	access list.	initgroups(3X)
setgroups: set group	access list.	setgroups(2)
/pupindescrip, pupoutdescrip, pupfidtochan:	access mapping between PupChans and fids.	pupdescrip(9)
filetime: tell minutes since file	(access, modification) time.	filetime(8)
utime: adjust the	access or modification time of a file.	utime(8)
leaf: PUP Leaf Remote File	Access Protocol Server.	leaf(8)
access: determine	accessability of a file.	access(3F)
access: determine	accessability of file.	access(2)
fstat: filter filenames	according to commands in a status file.	fstat(8)
ac: login	accounting.	ac(8)
ddacct: Dump Dover	Accounting.	ddacct(8)
gsa: group system	accounting.	gsa(8)
sa: accton: system	accounting.	sa(8)
acct: execution	accounting file.	acct(5)
pac: printer/ploter	accounting information.	pac(8)
acct: turn	accounting on or off.	acct(2)
user login accounts (create, modify, destroy Unix	accounts). nu: manage	nu(8)
nu: manage user login	accounts (create, modify, destroy Unix accounts).	nu(8)
	acct: execution accounting file.	acct(5)
	acct: turn accounting on or off.	acct(2)
	accton: system accounting.	sa(8)
sa,	acos, atan, atan2: trigonometric functions.	sin(3M)
sin, cos, tan, asin,	action for a signal.	signal(3F)
signal: change the	A/D converter.	ad(4)
ad: Data Translation	ad: Data Translation A/D converter.	ad(4)
fortune: print a random, hopefully interesting,	adage.	fortune(6)
	adb: debugger.	adb(1)
swapon:	add a swap device for interleaved paging/swapping.	swapon(2)
- create an Emacs data base .br/ d b a d d :	add entry to an Emacs data base .br d b e r e a t e	dbadd(1)
cparen -	add parentheses to C expressions.	cparen(1)
	addbib: create or extend bibliographic database.	addbib(1)
adduser: procedure for	adding new users.	adduser(8)
swapon: specify	additional device for paging and swapping.	swapon(8)
get Pup Network Directory entry attributes for an	address, mattributes:	mattributes(9)
mlookup: translate a name to a Pup	address.	mlookup(9)
addrfmt: IP/ICMP	Address Format Request user program.	addrfmt(8)

Permuted Index

inet_makcaddr, inet_lnaof, inet_nctof:	Internet address manipulation routines. /inet_ntoa,	inet(3n)
loc:	return the address of an object.	loc(3F)
(puppetport) pupgetsrcport, pupgetdstport:	get address of local, remote ends of pup channel.	pupgetport(9)
arp:	address resolution display and control.	arp(8C)
arp:	Address Resolution Protocol.	arp(4P)
(enarp) en10mbpuparp, ennoarp:	Address Resolution Protocol (ARP) routines.	enarp(9)
maddtoname:	translate a Pup Port address to a name.	maddtoname(9)
host:	print IP host names and addresses.	host(1)
net:	print IP net names and addresses.	nct(1)
mailaddr:	mail addressing description.	mailaddr(7)
program.	addrfmt: IP/ICMP Address Format Request user	addrfmt(8)
utime:	adduser: procedure for adding new users.	adduser(8)
adjust the access or modification time of a file.	utime(8)	
adventure: an exploration game.	adventure(6)	
flock:	apply or remove an advisory lock on an open file.	flock(2)
yes:	be repetitively affirmative.	yes(1)
basename:	strip filename allixes.	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.	csh(1)
which:	locate a program file including aliases file for sendmail.	csh(1)
newaliases:	rebuild the data base for the mail aliases and paths (csh only).	aliases(5)
aliases:	aliases file.	which(1)
valloc:	aligned memory allocator.	newaliases(1)
malloc, free, realloc, calloc, alloca:	memory allocator.	aliases(5)
valloc:	aligned memory allocator.	valloc(3)
cyacc:	modified yacc allowing much improved error recovery.	malloc(3)
limit:	allusers: print list of all authorized users.	malloc(3)
renice:	alter per-process resource limitations.	valloc(3)
fsckblks:	print alternate super block numbers for fsck -b.	valloc(3)
else:	alternative commands.	cyacc(1)
altoload:	load files from an Alto FTP "dump" format file.	allusers(1)
Uatimecvt:	conversions from Unix to Alto time and vice versa.	csh(1)
format file.	altoload: load files from an Alto FTP "dump"	renice(8)
breath-of-life server for bootloading 3mb	Altos. breathlife:	fsckblks(8)
lex:	generator of lexical analysis programs.	csh(1)
error:	analyze and disperse compiler error messages.	altoload(1)
style:	analyze surface characteristics of a document.	uatimecvt(9)
analyze:	Virtual UNIX postmortem crash analyzer.	altoload(1)
sigstack:	set and/or get signal stack context.	breathlife(8)
worms:	animate worms on a display terminal.	lex(1)
rain:	animated raindrops display.	error(1)
ansi:	read and write ANSI format magnetic tapes.	style(1)
ansi:	read and write ANSI format magnetic tapes.	analyze(8)
bcd:	convert to antique media.	analyzer(8)
apl:	apl interpreter.	sigstack(2)
apl:	apl interpreter.	worms(6)
apply:	apply a command to a set of arguments.	rain(6)
apply:	apply a command to a set of arguments.	ansi(1)
flock:	apply or remove an advisory lock on an open file.	ansi(1)
apropos:	locate commands by keyword lookup.	bcd(6)
ar:	archive and library maintainer.	apl(1)
ar:	archive (library) file format.	apl(1)
number:	convert Arabic numerals to English.	apply(1)
bc:	arbitrary-precision arithmetic language.	apply(1)
graphics/ plot:	openpl, erase, label, line, circle, arc, move, cont, point, linemod, space, closepl:	flock(2)
tp:	manipulate tape archive.	apropos(1)
ar:	archive and library maintainer.	ar(1)
tar:	tape archive file format.	ar(5)
ar:	archive (library) file format.	number(6)
tar:	tape archiver.	bc(1)
arfl, fcopy:	archiver and copier for floppy.	plot(3X)
shar:	produce shell-script archives.	tp(1)
arev:	convert archives to new format.	ar(1)
ranlib:	convert archives to random libraries.	tar(5)
glob:	filename expand argument list.	ar(5)
shift:	manipulate argument list.	tar(1)
		arfl(8V)
		shar(1)
		arev(8)
		ranlib(1)
		arev(8)
		arfl(8V)
		csh(1)
		csh(1)

varargs: variable	argument list.	varargs(3)
apply: apply a command to a set of	arguments.	apply(1)
echo: echo	arguments.	csh(1)
echo: echo	arguments.	echo(1)
getarg, iargc: return command line	arguments.	getarg(3F)
expr: evaluate	arguments as an expression.	expr(1)
traper: trap	arithmetic errors.	traper(3F)
bc: arbitrary-precision	arithmetic language.	bc(1)
@:	arithmetic on shell variables.	csh(1)
	arithmetic: provide drill in number facts.	arithmetic(6)
cn10mbpuparp, ennoarp: Address Resolution Protocol	arp: address resolution display and control.	arp(8C)
arptab: show contents of kernel	arp: Address Resolution Protocol.	arp(4P)
whois: ask the	(ARP) routines. (enarp)	enarp(9)
	ARP table.	arptab(1)
	ARPA Internet NIC about a user.	whois(1C)
enable or disable signal on ethernet packet	arptab: show contents of kernel ARP table.	arptab(1)
biff: be notified if mail	arrival. ensignal:	ensignal(9)
expire: remove outdated news	arrives and who it is from.	biff(1)
vnews: read news	articles.	expire(8)
recnews: receive unprocessed	articles via mail.	vnews(1)
sendnews: send news	articles via mail.	recnews(8)
uurec: receive processed news	articles via mail.	sendnews(8)
expr: evaluate arguments	as an expression.	uurec(8)
	as: VAX-11 assembler.	expr(1)
gmtime, asctime, timezone: convert date and time to	as68: .a68 -> .b assembler component of cc68.	as(1)
ascii: map of	ASCII. ctime, localtime,	as68(1)
od: octal, decimal, hex,	ASCII character set.	ctime(3)
	ascii dump.	ascii(7)
	ascii: map of ASCII character set.	od(1)
fdate: return date and time in an	ASCII string.	ascii(7)
atof, atoi, atol: convert	ASCII to numbers.	fdate(3F)
atoo: convert	ASCII to octal numbers.	atof(3)
ctime, localtime, gmtime,	asctime, timezone: convert date and time to ASCII.	atoo(9)
sin, cos, tan,	asin, acos, atan, atan2: trigonometric functions.	ctime(3)
whois:	ask the ARPA Internet NIC about a user.	sin(3M)
as: VAX-11	assembler.	whois(1C)
a.out:	assembler and link editor output.	as(1)
as68: .a68 -> .b	assembler component of cc68.	a.out(5)
	assert: program verification.	as68(1)
setbuf, setbuffer, setlinebuf:	assign buffering to a stream.	assert(3X)
shutdown: close down the system	at a given time.	setbuf(3S)
at: execute commands	at a later time.	shutdown(8)
mailcheck: find out if a user has mail	at a PUP host.	at(1)
mmailcheck: find out if a user has new mail	at a Pup host.	mailcheck(1)
msendumsg: send a message to one or all users	at a Pup host.	mmailcheck(9)
	at: execute commands at a later time.	msendumsg(9)
	at low priority (<i>sh</i> only).	at(1)
nice, nohup: run a command	atan, atan2: trigonometric functions.	nice(1)
sin, cos, tan, asin, acos,	atan2: trigonometric functions.	sin(3M)
sin, cos, tan, asin, acos, atan,	atof, atoi, atol: convert ASCII to numbers.	sin(3M)
	atoi, atol: convert ASCII to numbers.	atof(3)
atof,	atol: convert ASCII to numbers.	atof(3)
atof, atoi,	atomically release blocked signals and wait for	atof(3)
interrupt. sigpause:	atoo: convert ASCII to octal numbers.	sigpause(2)
	attributes.	atoo(9)
rcs: change RCS file	attributes for an address.	rcs(1)
mattributes: get Pup Network Directory entry	authorized users.	mattributes(9)
allusers: print list of all	autocall unit interface.	allusers(1)
dn: DN-11	autoconf: diagnostics from the autoconfiguration	dn(4)
code.	autoconfiguration code.	autoconf(4)
autoconf: diagnostics from the	automatically.	autoconf(4)
bugfiler: file bug reports in folders	auto-reboot and daemons.	bugfiler(8)
rc: command script for	available in fonts.widths.	rc(8)
dumpfonts: show what Press fonts are	average. loadlog:	dumpfonts(1)
log the current time, number of users, and load	average load log data on a weekly basis.	loadlog(1)
loadavg:	await completion of process.	loadavg(1)
wait:	awk: pattern scanning and processing language.	wait(1)
	-b. fsckblks:	awk(1)
print alternate super block numbers for fsck	backgammon: the game.	fsckblks(8)
	background.	backgammon(6)
bg: place job in	background processes to complete.	csh(1)
wait: wait for	backlog.	csh(1)
enstbacklog: set ethernet input queue	backlog for pup channel.	enstbacklog(9)
pupsetbacklog: set input queue	backup: make a backup version copy of a file.	pupsetbacklog(9)
	backup version copy of a file.	backup(1)
backup: make a		backup(1)

bad144: read/write dec standard 144	bad sector information.	bad144(8)
badsect: create files to contain	bad sectors.	badsect(8)
symlchk: check for	bad symbolic links.	symlchk(1)
information.	bad144: read/write dec standard 144 bad sector	bad144(8)
banner: print large	badsect: create files to contain bad sectors.	badsect(8)
getbanner: get system login	banner on printer.	banner(6)
gettytab: terminal configuration data	banner: print large banner on printer.	banner(6)
hosts: host name data	banner string.	getbanner(3)
networks: network name data	base.	gettytab(5)
phones: remote host phone number data	base.	hosts(5)
printcap: printer capability data	base.	networks(5)
protocols: protocol name data	base.	phones(5)
services: service name data	base.	printcap(5)
termcap: terminal capability data	base.	protocols(5)
vgrindfs: vgrind's language definition data	base.	services(5)
base .br/ d b a d d : add entry to an Emacs data	base .br d b e r e a t e - create an Emacs data	termcap(5)
/base .br d b e r e a t e - create an Emacs data	base .br d b l i s t - list contents of an Emacs/	vgrindfs(5)
/.br d b l i s t - list contents of an Emacs data	base .br d b p r i n t - print an entry from a	dbadd(1)
newaliases: rebuild the data	base for the mail aliases file.	dbadd(1)
ttytype: data	base of terminal types by port.	dbadd(1)
fetch, store, delete, firstkey, nextkey: data	base subroutines. dbminit,	newaliases(1)
vi: screen oriented (visual) display editor	based on ex.	ttytype(5)
loadavg: average load log data on a weekly	basename: strip filename affixes.	dbm(3X)
	basis.	vi(1)
	bboard: bulletin board reading program.	basename(1)
	bc: arbitrary-precision arithmetic language.	loadavg(1)
	bcd: convert to antique media.	bboard(1)
	bcmp, bzero, fls: bit and byte string operations.	bc(1)
bcopy, operations.	bcopy, bcmp, bzero, fls: bit and byte string	bcd(6)
cb: C program	beautifier.	bstring(3)
unpcnt: remove lines	beginning with % from a file.	bstring(3)
va:	Benson-Varian interface.	cb(1)
vfont: font formats for the	Benson-Varian or Versatec.	unpcnt(1)
j0, j1, jn, y0, y1, yn:	bessel functions.	va(4)
changing/ random, srandom, initstate, setstate:	bessel functions: of two kinds for integer orders.	vfont(5)
	better random number generator; routines for	j0(3M)
	bg: place job in background.	bessel(3I)
	bibliographic database.	random(3)
addbib: create or extend	bibliographic database.	csh(1)
roffbib: run off	bibliographic database.	addbib(1)
sortbib: sort	bibliography.	roffbib(1)
bibtex: make a LaTeX	bibliography. indxbib, lookbib: build inverted	sortbib(1)
index for a bibliography, find references in a	bibliography, find references in a bibliography.	bibtex(1)
indxbib, lookbib: build inverted index for a	bibtex: make a LaTeX bibliography.	lookbib(1)
from.	bill: be notified if mail arrives and who it is	lookbib(1)
comsat:	bill server.	biff(1)
install: install	binaries.	comsat(8C)
whereis: locate source,	binary, and or manual for program.	install(1)
find the printable strings in a object, or other	binary, file. strings:	whereis(1)
uencode.udecode: encode/decode a	binary file for transmission via mail.	strings(1)
fread, fwrite: buffered	binary input/output.	uencode(1C)
buildnetdir: build	binary-format Pup Network Directory.	fread(3S)
bind:	bind a name to a socket.	buildnetdir(8)
	bind: bind a name to a socket.	bind(2)
	binmail: send or receive mail among users.	bind(2)
	bit and byte string operations.	binmail(1)
bcopy, bcmp, bzero, fls:	bit: and, or, xor, not, rshift, lshift bitwise	bstring(3)
functions.	bitwise functions.	bit(3I)
bit: and, or, xor, not, rshift, lshift	bk: line discipline for machine-machine	bit(3I)
communication (obsolete).	block.	bk(4)
sync: update the super	block.	sync(8)
update: periodically update the super	block move a buffer.	update(8)
bmove:	block numbers for fsck -b.	bmove(9)
fsckblks: print alternate super	block signals.	fsckblks(8)
sigblock:	blocked signals and wait for interrupt.	sigblock(2)
sigpause: atomically release	blocks in a file.	sigpause(2)
sum: sum and count	bmove: block move a buffer.	sum(1)
	board reading program.	bmove(9)
	boggle.	bboard(1)
bboard: bulletin	boggle: play the game of boggle.	boggle(6)
boggle: play the game of	Boise. catboise:	boggle(6)
convert C/A/T files to DVI format and print on	boise: send files to the 11P2680a printer using TCP.	catboise(1)
	book of changes and other cookies.	boise(1)
	boot file directory from a boot server.	ching(6)
ching: the		mbootdir(9)
mbootdir: get a		

750rom: details of Vax-11/750	boot ROMs.	750rom(8)
mbootdir: get a boot file directory from a	boot server.	mbootdir(9)
mbootrequest: request a	boot-load.	mbootrequest(9)
msunbootreq: request a	boot-load.	msunbootreq(9)
breathlife: breath-of-life server for	bootloading 3mb Altos.	breathlife(8)
reboot: UNIX	bootstrapping procedures.	reboot(8)
mille: play Mille	Bournes.	mille(6)
rev68: reverse byte order in 68000 .b and .68	(b.out) files.	rev68(1)
d b a d d : add entry to an Emacs data base .	br d b e r e a t e - create an Emacs data base .br/	dbadd(1)
/ .br d b c r e a t e - create an Emacs data base .	br d b l i s t - list contents of an Emacs data/	dbadd(1)
/-- list contents of an Emacs data base .	br d b p r i n t - print an entry from a	dbadd(1)
switch: multi-way command	branch.	csh(1)
login,/ sh, for, case, if, while, :, . .	break, continue, cd, eval, exec, exit, export,	sh(1)
	break: exit while/foreach loop.	csh(1)
	breaksw: exit from switch.	csh(1)
	breathlife: breath-of-life server for bootloading	breathlife(8)
3mb Altos.	breathlife: breath-of-life server for bootloading 3mb Altos.	breathlife(8)
breathlife:	bring job into foreground.	csh(1)
fg:	brk, sbrk: change data segment size.	brk(2)
	broadcasting Internet Protocol packets.	ipbroadcast(4P)
ipbroadcast :	btroff: troff to the ImPrint printer.	btroff(1)
bmove: block move a	buffer.	bmove(9)
ik: Ikonas frame	buffer, graphics device interface.	ik(4)
ringbuf: ring	buffer package.	ringbuf(9)
fread, fwrite:	buffered binary input/output.	fread(3S)
stdio: standard	buffered input/output package.	intro(3S)
setbuf, setbuffer, setlinebuf: assign	buffering to a stream.	setbuf(3S)
generate a dump of the operating system's profile	buffers. kgmon:	kgmon(8)
gripe: mail a local system	bug report.	gripe(1)
sendbug: mail a system	bug report to 4bsd-bugs.	sendbug(1)
bugfiler: file	bug reports in folders automatically.	bugfiler(8)
automatically.	bugfiler: file bug reports in folders	bugfiler(8)
buildnetdir:	build binary-format Pup Network Directory.	buildnetdir(8)
efinit, efwdinsert, efginsert, efchinsert, efAND:	build ethernet filters. (enclfilter) ensetfilt,	enclfilter(9)
references in a bibliography. indxib , lookbib :	build inverted index for a bibliography, find	lookbib(1)
mknod :	build special file.	mknod(8)
config :	build system configuration files.	config(8)
for makefiles.	buildmake: preprocessor to provide extended syntax	buildmake(1)
Directory.	buildnetdir: build binary-format Pup Network	buildnetdir(8)
bboard :	bulletin board reading program.	bboard(1)
ntohs : convert values between host and network	byte order. htonl , htons , ntohl ,	byteorder(3n)
rev68: reverse	byte order in 68000 .b and .68 (b.out) files.	rev68(1)
bcopy , bcmp , bzero , fls : bit and	byte string operations.	bstring(3)
package.	byteorder: discussion of byte-ordering and the Pup	byteorder(9)
byteorder : discussion of	byte-ordering and the Pup package.	byteorder(9)
swab : swap	bytes.	swab(3)
bcopy , bcmp ,	bzero , fls : bit and byte string operations.	bstring(3)
ccom68 :	c -> .s translator component of cc68.	ccom68(1)
cnest : check for nested comments in	C code.	cnest(1)
cc :	C compiler.	cc(1)
cc68 :	C compiler for the MC68000.	cc68(1)
(except) raise , raise_sys):	C exception handling.	except(3)
cparen - add parentheses to	C expressions.	cparen(1)
cb :	C program beautifier.	cb(1)
indent : indent and format	C program source.	indent(1)
lint : a	C program verifier.	lint(1)
xstr : extract strings from	C programs to implement shared strings.	xstr(1)
mkstr : create an error message file by massaging	C source.	mkstr(1)
cxref : cross reference	C source files.	cxref(1)
hypot ,	cabs: Euclidean distance.	hypot(3M)
	cal: print calendar.	cal(1)
	calculate default disk partition sizes.	diskpart(8)
diskpart :	calculator.	dc(1)
dc : desk	calen: print large-format calendar.	calen(1)
	calendar.	cal(1)
cal : print	calendar.	calen(1)
calen : print large-format	calendar: reminder service.	calendar(1)
	call.	syscall(2)
syscall : indirect system	call graph profile data.	gprof(1)
gprof : display	caller.	getuid(3F)
getuid , getgid : get user or group ID of the	calloc, alloca : memory allocator.	malloc(3)
malloc , free , realloc ,	calls and error numbers.	intro(2)
intro : introduction to system	canfield.	canfield(6)
canfield , cfscores : the solitaire card game	canfield, cfscores : the solitaire card game	canfield(6)
canfield .	capability data base.	printcap(5)
printcap : printer	capability data base.	termcap(5)
termcap : terminal		

canfield, cfscores: the solitaire cribbage: the	card game canfield.	canfield(6)
cd, eval, exec, exit, export, login./ sh, for, lower: lower the	card game cribbage.	cribbage(6)
strempfold, strncmpfold:	case, if, while, :, ., ., break, continue,	sh(1)
tu: VAX-11/730 and VAX-11/750 TU58 console	case of a filename.	lower(1)
uu: TU58/DE:tape II UNIBUS	case: selector in switch.	csh(1)
catman: create the	case-folded string comparison operations.	strcmpfold(3)
catdvi: convert	cassette interface.	tu(4)
catboise: convert	cassette interface.	uu(4)
uncompact, ccat: compress and uncompress files, and print on Boise.	cat: catenate and print.	cat(1)
default:	cat files for the manual.	catman(8)
cat:	C/A/T files to DVI format.	catdvi(1)
	C/A/T files to DVI format and print on Boise.	catboise(1)
	cat them. compact,	compact(1)
	catboise: convert C/A/T files to DVI format and	catboise(1)
	catchall clause in switch.	csh(1)
	catdvi: convert C/A/T files to DVI format.	catdvi(1)
	catenate and print.	cat(1)
	catman: create the cat files for the manual.	catman(8)
	cb: C program beautifier.	cb(1)
	cc: C compiler.	cc(1)
as68: .a68 -> .b assembler component of	cc68.	as68(1)
ccom68: .c -> .s translator component of	cc68.	ccom68(1)
dl68: b.out -> .dl downloader component of	cc68.	dl68(1)
o68: .s -> .s optimizer component of	cc68.	o68(1)
compact, uncompact,	cc68: C compiler for the MC68000.	cc68(1)
	ccat: compress and uncompress files, and cat them.	compact(1)
	ccom68: .c -> .s translator component of cc68.	ccom68(1)
	cd: change directory.	csh(1)
	cd: change working directory.	cd(1)
case, if, while, :, ., ., break, continue,	cd, eval, exec, exit, export, login, read./ /for,	sh(1)
fabs, floor,	ceil: absolute value, floor, ceiling functions.	floor(3M)
fabs, floor, ceil: absolute value, floor,	ceiling functions.	floor(3M)
canfield,	cfscores: the solitaire card game canfield.	canfield(6)
chdir:	change current working directory.	chdir(2)
brk, sbrk:	change data segment size.	brk(2)
chdir:	change default directory.	chdir(3F)
chsh:	change default login shell.	chsh(1)
cd:	change directory.	csh(1)
chdir:	change directory.	csh(1)
ioinit:	change I77 I/O initialization.	ioinit(3F)
chfn:	change finger entry.	chfn(1)
chgrp:	change group.	chgrp(1)
passwd:	change login password.	passwd(1)
chmod:	change mode.	chmod(1)
chmod:	change mode of a file.	chmod(3F)
chmod:	change mode of file.	chmod(2)
umask:	change or display file creation mask.	csh(1)
chown:	change owner.	chown(8)
chown:	change owner and group of a file.	chown(2)
rcs:	change RCS file attributes.	rcs(1)
chroot:	change root directory.	chroot(2)
signal:	change the action for a signal.	signal(3F)
pupreopen:	change the destination for a Pup channel.	pupreopen(9)
rename:	change the name of a file.	rename(2)
set:	change value of shell variable.	csh(1)
cd:	change working directory.	cd(1)
ching: the book of	changes and other cookies.	ching(6)
better random number generator: routines for	changing generators. /srandom, initstate, setstate:	random(3)
verch: version	changing program for Pascal sources.	verch(1)
pipe: create an interprocess communication	channel.	pipe(2)
pupchan: data structure describing a Pup	channel.	pupchan(9)
pupclose: close a pup	channel.	pupclose(9)
get host, net numbers of local end of pup	channel. pupgethost, pupgetnet:	pupgethost(9)
get address of local, remote ends of pup	channel. /pupgetsreport, pupgetdstport:	pupgetport(9)
pupopen: open a pup	channel.	pupopen(9)
pupread: read a pup packet from a pup	channel.	pupread(9)
pupreopen: change the destination for a Pup	channel.	pupreopen(9)
pupsetbacklog: set input queue backlog for pup	channel.	pupsetbacklog(9)
pupsetdfilt: set a default packet filter for a Pup	channel.	pupsetdfilt(9)
pupsetfilter: set the packet filter for a Pup	channel.	pupsetfilter(9)
pupsetmode: set and read the mode for a Pup	channel. pupsetmode,	pupsetmode(9)
pupsettimeout: set timeout for pup	channel.	pupsettimeou(9)
pupwrite: write a packet to a pup	channel.	pupwrite(9)
puplisten, puplistenall: open Pup	channels on all connected networks.	puplisten(9)
ungetc: push	character back into input stream.	ungetc(3S)
isspace, ispunct, isprint, iscntrl, isascii:	character classification macros. /isdigit, isalnum,	ctypc(3)

eqnchar: special character definitions for eqn.	eqnchar(7)
getc, fgetc: get a character from a logical unit.	getc(3F)
index, rindex, lnblnk, len: tell about character objects.	index(3F)
getc, getchar, fgetc, getw: get character or word from stream.	getc(3S)
putc, putchar, fputc, putw: put character or word on a stream.	putc(3S)
ascii: map of ASCII character set.	ascii(7)
putc, fputc: write a character to a fortran logical unit.	putc(3F)
style: analyze surface characteristics of a document.	style(1)
tr: translate characters.	tr(1)
snake, snscore: display chase game.	snake(6)
chdir: change current working directory.	chdir(2)
chdir: change default directory.	chdir(3F)
chdir: change directory.	csh(1)
check. file system directory consistency	dcheck(8)
ichk: file system storage consistency	ichk(8)
fsck: file system consistency	fsck(8)
symbchk: check for bad symbolic links.	symbchk(1)
cnest: check for nested comments in C code.	cnest(1)
ci: check in RCS revisions.	ci(1)
checknr: check nroff/troff files.	checknr(1)
co: check out RCS revisions.	co(1)
eqn, neqn, checkeq: typeset mathematics.	eqn(1)
quotacheck: file system quota consistency	quotacheck(8)
fastboot, fasthalt: reboot/halt the system without checking the disks.	fastboot(8)
newsr: information file for readnews(1) and checknews(1).	newsr(5)
timecheck: checknr: check nroff/troff files.	checknr(1)
checksum: compute Pup-style checks and sets Pup network time.	timecheck(1)
checksum: compute Pup-style checksum.	checksum(9)
checksum: compute Pup-style checksum.	checksum(9)
chess: the game of chess.	chess(6)
chess: the game of chess.	chess(6)
chfn: change finger entry.	chfn(1)
chgrp: change group.	chgrp(1)
ching: the book of changes and other cookies.	ching(6)
chmod: change mode.	chmod(1)
chmod: change mode of a file.	chmod(3F)
chmod: change mode of file.	chmod(2)
chown: change owner.	chown(8)
chown: change owner and group of a file.	chown(2)
chroot: change root directory.	chroot(2)
chsh: change default login shell.	chsh(1)
ci: check in RCS revisions.	ci(1)
closepl:/ plot: openpl, erase, label, line.	plot(3X)
ispunct, isprint, iscntrl, isascii: character classification macros. /isdigit, isalnum, isspace,	ctype(3)
default: catchall	csh(1)
uuclean: uucp spool directory	uuclean(8C)
clear: clear terminal screen.	clear(1)
clri: clear i-node.	clri(8)
clear: clear terminal screen.	clear(1)
ferror, feof, clearerr, fileno: stream status inquiries.	ferror(3S)
csh: a shell (command interpreter) with C-like syntax.	csh(1)
kg: KL-11/D1-11W line	kg(4)
cron: clock daemon.	cron(8)
pupclose: close a pup channel.	pupclose(9)
close: delete a descriptor.	close(2)
shutdown: close down the system at a given time.	shutdown(8)
fclose, fflush: close or flush a stream.	fclose(3S)
opendir, readdir, telldir, seekdir, rewinddir, closedir: directory operations.	directory(3)
syslog, openlog, closelog: control system log.	syslog(3)
circle, arc, move, cont, point, linemod, space, closepl: graphics interface. /erase, label, line.	plot(3X)
clri: clear i-node.	clri(8)
cmp: compare two files.	cmp(1)
cnest: check for nested comments in C code.	cnest(1)
co: check out RCS revisions.	co(1)
code. autoconf: diagnostics from the autoconfiguration	autoconf(4)
code. cnest: check for nested comments in C	cnest(1)
code translator. pi: Pascal interpreter	pi(1)
col: filter reverse line feeds.	col(1)
colrt: filter nroff output for CRT previewing.	colrt(1)
collect system diagnostic messages to form error	dmsg(8)
colrm: remove columns from a file.	colrm(1)
columns from a file.	colrm(1)
comm: select or reject lines common to two sorted	comm(1)
command. csh: overlay shell with specified	csh(1)
command. time: time	csh(1)
command. rcmd, rresvport, ruscrok: routines for returning a stream to a remote	rcmd(3X)

rexec: return stream to a remote system: issue a shell system: execute a UNIX test: condition time: time a nice, nohup: run a switch: multi-way remote: Remote uux: unix to unix rehash: recompute unhash: discard hashstat: print nohup: run csh: a shell whatis: describe what a	command. command. command. command. command at low priority (<i>sh</i> only). command branch. command execution. command execution. command hash table. command hash table. command hashing statistics. command immune to hangups. (command interpreter) with C-like syntax. command is. command language. /exec, exit, export, login, read, command line arguments. command on terminal screen. command repeatedly. command script for auto-reboot and daemons. command scripts. command to a set of arguments. command transfer. commands. commands. commands. intro: commands. commands at a later time. commands by keyword lookup. commands conditionally. commands executed in reverse order. commands from file. commands in a .b file for the 68000. commands in a status file. comments in C code. common to two sorted files. communication. communication channel. communication (obsolete). communications device. communications multiplexer. communications multiplexer. compact list of users who are on the system. compact, uncompact, ccat: compress and uncompress comparator. compare RCS revisions. compare two files. comparison. comparison operations. compatibility library functions. compile a Franz Lisp program. compiler. compiler. compiler. compiler. compiler error messages. compiler for the MC68000. compiler for the MC68000. compiler-compiler. compiler/interpreter. complete. completion of process. component of cc68. component of cc68. component of cc68. compress and uncompress files, and cat them. compute Pup-style checksum. computer aided instruction about UNIX. Computer version of the game hangman. computer via Pup network. comsat: biff server. condition command. conditional. conditional statement. conditionally.	rexec(3X) system(3) system(3F) test(1) time(1) nice(1) csh(1) remote(1) uux(1C) csh(1) csh(1) csh(1) csh(1) csh(1) whatis(1) sh(1) getarg(3F) screen(1) csh(1) rc(8) csh(1) apply(1) csh(1) csh(1) intro(1) intro(8) resintro(1) at(1) apropos(1) csh(1) lastcomm(1) csh(1) rl68(1) fstat(8) cnest(1) comm(1) socket(2) pipe(2) bk(4) dmc(4) dh(4) dz(4) users(1) compact(1) diff(1) rcdiff(1) cmp(1) diff3(1) strempfold(3) intro(3C) lizt(1) cc(1) f77(1) mod(1) pc(1) error(1) cc68(1) pc68(1) yacc(1) fp(1) csh(1) wait(1) as68(1) ecom68(1) dl68(1) o68(1) compact(1) checksum(9) learn(1) hangman(6) puptelnet(1) comsat(8C) test(1) csh(1) csh(1) csh(1)
readonly, set, shift, times, trap, umask, wait; getarg, iargc: return screen: repeatedly display output of repeat: execute rc: onintr: process interrupts in apply: apply a goto: else: alternative intro: introduction to resintro - introduction to RCS at: execute apropos: locate while: repeat lastcomm: show last source: read rl68: print relocation fstat: filter filenames according to cnest: check for nested comm: select or reject lines socket: create an endpoint for pipe: create an interprocess bk: line discipline for machine-machine dmc: DEC DMC-11/DMR-11 point-to-point dh: D11-11/DM-11 dz: DZ-11 users: files, and cat them. diff: differential file and directory rcdiff: cmp: diff3: 3-way differential file strempfold, strempfold: case-folded string intro: introduction to lizt: cc: C f77: Fortran 77 mod: Modula-2 pc: Pascal error: analyze and disperse cc68: C pc68: Pascal yacc: yet another fp: Functional Programming language wait: wait for background processes to wait: await as68: .a68 -> .b assembler ecom68: .c -> .s translator dl68: b.out -> .dl downloader o68: .s -> .s optimizer compact, uncompact, ccat: checksum: learn: hangman: puptelnet: connect your terminal to a remote test: endif: terminate if: while: repeat commands		

gettytab: terminal	config: build system configuration files.	config(8)
config: build system	configuration data base.	gettytab(5)
setpupnettab, getpupnettab, endpupnettab: pup	configuration files.	config(8)
ifconfig:	configuration table. (pupnettab)	pupnettab(9)
	configure network interface parameters.	ifconfig(8C)
	congraph: plot connectivity of a graph.	congraph(1)
	connect: initiate a connection on a socket.	connect(2)
	connect to a remote system.	tip(1C)
tip, cu:	connect your terminal to a remote computer via Pup	puptelnet(1)
network. puptelnet:	connected networks.	puplisten(9)
puplisten, puplistenall: open Pup channels on all	connected peer.	getpeername(2)
getpeername: get name of	connected sockets.	socketpair(2)
socketpair: create a pair of	connection.	shutdown(2)
shutdown: shut down part of a full-duplex	connection on a socket.	accept(2)
accept: accept a	connection on a socket.	connect(2)
connect: initiate a	connections on a socket.	listen(2)
listen: listen for	connectivity of a graph.	congraph(1)
congraph: plot	cons: VAX-11 console interface.	cons(4)
	consistency check.	dcheck(8)
dcheck: file system directory	consistency check.	icheck(8)
icheck: file system storage	consistency check and interactive repair.	fsock(8)
fsock: file system	consistency checker.	quotacheck(8)
quotacheck: file system quota	console cassette interface.	tu(4)
tu: VAX-11/730 and VAX-11/750 TUS8	console floppy interface.	fl(4)
fl:	console interface.	cons(4)
cons: VAX-11	construct a file. what:	what(1)
show what versions of object modules were used to	construct a file system.	mkfs(8)
mkfs:	construct a new file system.	newfs(8)
newfs:	construct a prototype file system.	mkproto(8)
mkproto:	construct dependency lines for makefiles.	makedep(1)
makedep:	constructs.	deroff(1)
deroff: remove nroff, troff, tbl and eqn	constructs.	detex(1)
detex: remove TeX	constructs.	unsubscribe(1)
unsubscribe: remove Scribe	consumption. getrlimit,	getrlimit(2)
setrlimit: control maximum system resource	consumption.	vlimit(3C)
vlimit: control maximum system resource	cont, point, linemod, space, closepl: graphics/	plot(3X)
/openpl, erase, label, line, circle, arc, move,	contain bad sectors.	badsect(8)
badsect: create files to	contents of an Emacs data base .br d b l i s t - list	dbadd(1)
/create an Emacs data base .br d b l i s t - list	ls: list	ls(1)
ls: list	contents of directory.	arptab(1)
arptab: show	contents of kernel ARP table.	sigstack(2)
sigstack: set and/or get signal stack	context.	sh(1)
sh, for, case, if, while, : . . . break.	continue, cd, eval, exec, exit, export, login,/	csh(1)
	continue: cycle in loop.	arp(8C)
arp: address resolution display and	control.	fcntl(2)
fcntl: file	control.	ioctl(2)
ioctl:	control device.	init(8)
init: process	control initialization.	getrlimit(2)
getrlimit, setrlimit:	control maximum system resource consumption.	vlimit(3C)
vlimit:	control maximum system resource consumption.	lpc(8)
lpc: line printer	control program.	tcp(4P)
tcp: Internet Transmission	Control Protocol.	syslog(3)
syslog, openlog, closelog:	control system log.	vhangup(2)
vhangup: virtually "hangup" the current	control terminal.	uda(4)
uda: UDA-50 disk	controller interface.	up(4)
up: unibus storage module	controller/drives.	term(7)
term:	conventional names for terminals.	ccvt(3)
ccvt, fevt, gcv: output	conversion.	long(3I)
long, short: integer object	conversion.	printf(3S)
printf, sprintf, sprintf: formatted output	conversion.	scanf(3S)
scanf, fscanf, sscanf: formatted input	conversion.	units(1)
units:	conversion program.	uftimecv(9)
Uftimecv:	conversions from Unix to Alto time and vice versa.	undump(1)
undump:	convert a core dump to an executable a.out file.	dvip(1)
dvip, dvid:	convert a dvi (TeX output) file to press format.	dd(1)
dd:	convert and copy a file.	number(6)
number:	convert Arabic numerals to English.	arcv(8)
arcv:	convert archives to new format.	ranlib(1)
ranlib:	convert archives to random libraries.	atof(3)
atof, atoi, atol:	convert ASCII to numbers.	atoc(9)
atoc:	convert ASCII to octal numbers.	catdvi(1)
catdvi:	convert C/A/T files to DVI format.	catboise(1)
Boise. catboise:	convert C/A/T files to DVI format and print on	ctime(3)
ctime, localtime, gmtime, asctime, timezone:	convert date and time to ASCII.	dviimp(1)
dviimp:	convert DVI files to impress format.	dvipress(1)
print them on the Dover.. dvipress:	convert dvi (TEX output) files to press format and	cz(1)
Dover.. cz (czarina):	convert files to press format and print them on the	

htable:	convert NIC standard format host tables.	htable(8)
them on the ImPrint printer..	pressimp: convert press files to Impress format and print	pressimp(1)
iprint -	convert text files to DVI format.	iprint(1)
bed:	convert to antique media.	bed(6)
format and print them on the Dover..	deat: convert troff phototypesetter output files to press	dcat(1)
htonl, htons, ntohl, ntohs:	convert values between host and network byte order.	byteorder(3n)
tangle, weave:	convert web file into pascal file, tex file.	tangle(1)
ad: Data Translation A/D	converter.	ad(4)
ching: the book of changes and other	cookies.	ching(6)
arff, fcopy: archiver and	copier for floppy.	arff(8V)
cp:	copy.	cp(1)
rcp: remote file	copy.	rcp(1C)
uucp, uulog: unix to unix	copy.	uucp(1C)
dd: convert and	copy a file.	dd(1)
backup: make a backup version	copy of a file.	backup(1)
fork: create a	copy of this process.	fork(3F)
savecore: save a	core dump of the operating system.	savecore(8)
undump: convert a	core dump to an executable a.out file.	undump(1)
	core: format of memory image file.	core(5)
gcore: get	core images of running processes.	gcore(1)
dlx: download with error	correction - 68000 Sun1 monitor.	dlx(1)
functions. sin,	cos, tan, asin, acos, atan, atan2: trigonometric	sin(3M)
sinh,	cosh, tanh: hyperbolic functions.	sinh(3M)
wc: word	count.	wc(1)
sum: sum and	count blocks in a file.	sum(1)
	cp: copy.	cp(1)
	cparen - add parentheses to C expressions.	cparen(1)
top: display and update information about the top	cpu processes.	top(1)
analyze: Virtual UNIX postmortem	crash analyzer.	analyze(8)
	crash: what happens when the system crashes.	crash(8V)
crash: what happens when the system	crashes.	crash(8V)
	creat: create a new file.	creat(2)
	fork: create a copy of this process.	fork(3F)
	creat: create a new file.	creat(2)
open: open a file for reading or writing, or	create a new file.	open(2)
	fork: create a new process.	fork(2)
	socketpair: create a pair of connected sockets.	socketpair(2)
	ctags: create a tags file.	ctags(1)
/entry to an Emacs data base .br d b c r e a t e --	create an Emacs data base .br d b l i s t - list/	dbadd(1)
	socket: create an endpoint for communication.	socket(2)
	mkstr: create an error message file by massaging C source.	mkstr(1)
	pipe: create an interprocess communication channel.	pipe(2)
	badsect: create files to contain bad sectors.	badsect(8)
nu: manage user login accounts	(create, modify, destroy Unix accounts).	nu(8)
	addbib: create or extend bibliographic database.	addbib(1)
	catman: create the cat files for the manual.	catman(8)
UniqueSocket:	create unique socket number.	uniquesocket(9)
umask: change or display file	creation mask.	umask(2)
umask: set file	creation mode mask.	umask(2)
	cref: cross reference program.	cref(1)
cribbage: the card game	cribbage.	cribbage(6)
	cribbage: the card game cribbage.	cribbage(6)
	cron: clock daemon.	cron(8)
	cxref: cross reference C source files.	cxref(1)
	cref: cross reference program.	cref(1)
	lxref: lisp cross reference program.	lxref(1)
	pxref: Pascal cross-reference program.	pxref(1)
colert: filter nroff output for	CRT previewing.	colert(1)
more, page: file perusal filter for	ert viewing.	more(1)
	crypt: encode/decode.	crypt(1)
	crypt, setkey, encrypt: DES encryption.	crypt(3)
	esh: a shell (command interpreter) with C-like	esh(1)
	css: DEC IMP-11A LH/DH IMP interface.	css(4)
	pcl: DEC CSS PCL-11 B Network Interface.	pcl(4)
	ct: phototypesetter interface.	ct(4)
	ctags: create a tags file.	ctags(1)
convert date and time to ASCII.	ctime, localtime, gmtime, asctime, timezone:	ctime(3)
time,	ctime, ltime, gmtime: return system time.	time(3F)
tip,	cu: connect to a remote system.	tip(1C)
vhangup: virtually "hangup" the	current control terminal.	vhangup(2)
gethostid, sethostid: get/set unique identifier of	current host.	gethostid(2)
gethostname, sethostname: get/set name of	current host.	gethostname(2)
hostname: get name of	current host.	hostname(3F)
set or print Internet Protocol (IP) identifier of	current host. iphostid:	iphostid(1)
hostid: set or print identifier of	current host system.	hostid(1)
hostname: set or print name of	current host system.	hostname(1)

jobs: print	current job list.	esh(1)
sigsetmask: set	current signal mask.	sigsetmask(2)
timeck: poll the localnet for the	current time.	timeck(8C)
loadlog: log the	current time, number of users, and load average.	loadlog(1)
whoami: print effective	current user id.	whoami(1)
chdir: change	current working directory.	chdir(2)
getcwd: get pathname of	current working directory.	getcwd(3F)
getwd: get	current working directory pathname.	getwd(3)
motion.	cursor: screen functions with "optimal" cursor	cursor(3X)
courses: screen functions with "optimal"	cursor motion.	cursor(3X)
spline: interpolate smooth	curve.	spline(1G)
	cxref: cross reference C source files.	cxref(1)
continue:	cycle in loop.	esh(1)
print them on the Dover..	cz (czarina): convert files to press format and	cz(1)
them on the Dover.. cz	(czarina): convert files to press format and print	cz(1)
cron: clock	daemon.	cron(8)
lpd: line printer	daemon.	lpd(8)
routed: network routing	daemon.	routed(8C)
re: command script for auto-reboot and	daemons.	rc(8)
flpd: DARPA Internet File Transfer Protocol server.	DARPA Internet File Transfer Protocol server.	flpd(8C)
inctd: DARPA little protocol server.	DARPA little protocol server.	inctd(8C)
telnetd: DARPA TELNET protocol server.	DARPA TELNET protocol server.	telnetd(8C)
tfpd: DARPA Trivial File Transfer Protocol server.	DARPA Trivial File Transfer Protocol server.	tfpd(8C)
eval: re-evaluate shell	data.	esh(1)
gprof: display call graph profile	data.	gprof(1)
prof: display profile	data.	prof(1)
ttys: terminal initialization	data.	ttys(5)
gettytab: terminal configuration	data base.	gettytab(5)
hosts: host name	data base.	hosts(5)
networks: network name	data base.	networks(5)
phones: remote host phone number	data base.	phones(5)
printcap: printer capability	data base.	printcap(5)
protocols: protocol name	data base.	protocols(5)
services: service name	data base.	services(5)
termcap: terminal capability	data base.	termcap(5)
vgrindfs: vgrind's language definition	data base.	vgrindfs(5)
data base .br/ d b a d d : add entry to an Emacs	data base .br d b e r e a t e - create an Emacs	dbadd(1)
/data base .br d b c r e a t e - create an Emacs	data base .br d b l i s t - list contents of an/	dbadd(1)
/base .br d b l i s t -- list contents of an Emacs	data base .br d b p r i n t - print an entry from/	dbadd(1)
newaliases: rebuild the	data base for the mail aliases file.	newaliases(1)
ttytype:	data base of terminal types by port.	ttytype(5)
dbmunit, fetch, store, delete, firstkey, nextkey:	data base subroutines.	dbm(3X)
exlog: extract	data from system load log file.	exlog(1)
dataio: load the	data i/o prom programmer.	dataio(1)
loadavg: average load log	data on a weekly basis.	loadavg(1)
brk, sbrk: change	data segment size.	brk(2)
null:	data sink.	null(4)
pupchan:	data structure describing a Pup channel.	pupchan(9)
support routines. port: a	data structure used in the Pup package, with	pupport(9)
ad:	Data Translation A/D converter.	ad(4)
types: primitive system	data types.	types(5)
addbib: create or extend bibliographic	database.	addbib(1)
rofbib: run off bibliographic	database.	rofbib(1)
sortbib: sort bibliographic	database.	sortbib(1)
join: relational	database operator.	join(1)
udp: Internet User	Datagram Protocol.	udp(4P)
	dataio: load the data i/o prom programmer.	dataio(1)
date: print and set the	date.	date(1)
gettimeofday, settimeofday: get/set	date and time.	gettimeofday(2)
time, ftime: get	date and time.	time(3C)
ftime: return	date and time in an ASCII string.	ftime(3F)
localtime, gmtime, asctime, timezone: convert	date and time to ASCII. ctime.	ctime(3)
touch: update	date last modified of a file.	touch(1)
idate, itime: return	date or time in numerical form.	idate(3F)
	date: print and set the date.	date(1)
data base subroutines.	dbmunit, fetch, store, delete, firstkey, nextkey:	dbm(3X)
	dbx: debugger.	dbx(1)
	dbx: debugger.	olddb(1)
	dc: desk calculator.	dc(1)
press format and print them on the Dover..	dcat: convert troff phototypesetter output files to	dcat(1)
	dcheck: file system directory consistency check.	dcheck(8)
	dd: convert and copy a file.	dd(1)
	ddacct: Dump Dover Accounting.	ddacct(8)
	ddt68, fddt68: symbolic debugger for 68000.	ddt68(1)
	dc: DI:C DEUNA 10 Mb/s Ethernet interface.	dc(4)
adb:	debugger.	adb(1)

dbx:	debugger.	dbx(1)
dbx:	debugger.	olddb(1)
pdx: pascal	debugger.	pdx(1)
ddt68, fddt68: symbolic	debugger for 68000.	ddt68(1)
pel:	DEC CSS PCL-11 B Network Interface.	pcl(4)
de:	DEC DEUNA 10 Mb/s Ethernet interface.	de(4)
device. dmc:	DEC DMC-11/DMR-11 point-to-point communications	dmc(4)
css:	DEC IMP-11A I11/D11 IMP interface.	css(4)
rx:	DEC RX02 floppy disk interface.	rx(4)
bad144: read/write	dec standard 144 bad sector information.	bad144(8)
od: octal.	decimal, hex, ascii dump.	od(1)
tp:	DEC/mag tape formats.	tp(5)
	default: catchall clause in switch.	esh(1)
chdir: change	default directory.	chdir(3F)
diskpart: calculate	default disk partition sizes.	diskpart(8)
chsh: change	default login shell.	chsh(1)
pupscdfilt: set a	default packet filter for a Pup channel.	pupscdfilt(9)
vgrindcfs: vgrind's language	definition data base.	vgrindcfs(5)
eqnchar: special character	definitions for eqn.	eqnchar(7)
stty, gtty: set and get terminal state	(defunct).	stty(3C)
close:	delete a descriptor.	close(2)
dbminit, fetch, store,	delete, firstkey, nextkey: data base subroutines.	dbm(3X)
pup-mailer:	deliver mail over the .SM PUP network.	pup-mailer(8)
tail:	deliver the last part of a file.	tail(1)
msg: permit or	deny messages.	msg(1)
makedep: construct	dependency lines for makefiles.	makedep(1)
tset: terminal	dependent initialization.	tset(1)
constructs.	deroff: remove nroff, troff, tbl and eqn	deroff(1)
crypt, setkey, encrypt:	DES encryption.	crypt(3)
whatis:	describe what a command is.	whatis(1)
pupchan: data structure	describing a Pup channel.	pupchan(9)
mailaddr: mail addressing	description.	mailaddr(7)
getdiskbyname: get disk	description by its name.	getdisk(3X)
disktab: disk	description file.	disktab(5)
remote: remote host	description file.	remote(5)
close: delete a	descriptor.	close(2)
dup, dup2: duplicate a	descriptor.	dup(2)
getfstype, setfsent, endfsent: get file system	descriptor file entry. /getfsspec, getfsfile.	getfsent(3X)
getdtablesize: get	descriptor table size.	getdtablesize(2)
dc:	desk calculator.	dc(1)
pupreopen: change the	destination for a Pup channel.	pupreopen(9)
nu: manage user login accounts (create, modify,	destroy Unix accounts).	nu(8)
750rom:	details of Vax-11/750 boot ROMs.	750rom(8)
access:	determine accessibility of a file.	access(3F)
access:	determine accessibility of file.	access(2)
file:	determine file type.	file(1)
engethost:	determine Pup host number of ethernet interface.	engethost(9)
de: DEC	detex: remove TeX constructs.	detex(1)
DEC DMC-11/DMR-11 point-to-point communications	DEUNA 10 Mb/s Ethernet interface.	de(4)
drum: paging	device. dmc:	dmc(4)
fold: fold long lines for finite width output	device.	drum(4)
ioctl: control	device.	fold(1)
swapon: add a swap	device.	ioctl(2)
swapon: specify additional	device for interleaved paging/swapping.	swapon(2)
drb: DR11-B/DR11-W general purpose user	device for paging and swapping.	swapon(8)
ik: Ikonas frame buffer, graphics	device interface.	drb(4)
ps: Evans and Sutherland Picture System 2 graphics	device interface.	ik(4)
etherport: show status of ethernet minor	device interface.	ps(4)
	devices.	etherport(1)
fmin, fmax, frac, dflmin, dflmax,	df: disk free.	df(1)
fmin, fmax, frac, dflmin,	dfrac, inmax: return extreme values.	fmin(3F)
values. fmin, fmax, frac,	dflmax, dfrac, inmax: return extreme values.	fmin(3F)
	dflmin, dflmax, dfrac, inmax: return extreme	fmin(3F)
dh:	dh: D11-11/DM-11 communications multiplexer.	dh(4)
dmsg: collect system	D11-11/DM-11 communications multiplexer.	dh(4)
autoconf:	diagnostic messages to form error log.	dmsg(8)
print wordy sentences; thesaurus for	diagnostics from the autoconfiguration code.	autoconf(4)
diction— print wordy sentences; thesaurus for	diction, diction, explain:	diction(1)
diction. explain,	diction, explain.	explain(1)
for diction.	diction— print wordy sentences; thesaurus for	explain(1)
	diction, explain: print wordy sentences; thesaurus	diction(1)
diff: differential file and directory comparator.	diff: differential file and directory comparator.	diff(1)
diff3: 3-way differential file comparison.	diff3: 3-way differential file comparison.	diff3(1)
diff:	differential file and directory comparator.	diff(1)
diff3: 3-way	differential file comparison.	diff3(1)
dir: format of directories.	dir: format of directories.	dir(5)

dir: format of	directories.	dir(5)
rm, rmdir: remove (unlink) files or	directories.	rm(1)
mmdir, m: remove (unlink)	directories or files.	rmdir(1)
buildnetdir: build binary-format Pup Network	Directory.	buildnetdir(8)
cd: change working	directory.	cd(1)
chdir: change current working	directory.	chdir(2)
chdir: change default	directory.	chdir(3F)
chroot: change root	directory.	chroot(2)
cd: change	directory.	csh(1)
chdir: change	directory.	csh(1)
getcwd: get pathname of current working	directory.	getcwd(3F)
ls: list contents of	directory.	ls(1)
mkdir: make a	directory.	mkdir(1)
netdirprint: print text version of Pup Network	Directory.	netdirprint(8)
scandir: scan a	directory.	scandir(3)
uuclean: uucp spool	directory clean-up.	uuclean(8C)
diff: differential file and	directory comparator.	diff(1)
dcheck: file system	directory consistency check.	dcheck(8)
unlink: remove	directory entry.	unlink(2)
unlink: remove a	directory entry.	unlink(3F)
mattributes: get Pup Network	Directory entry attributes for an address.	mattributes(9)
mkdir: make a	directory file.	mkdir(2)
rmdir: remove a	directory file.	rmdir(2)
mklost + found: make a lost + found	directory for fsck.	mklost + found(8)
mbootdir: get a boot file	directory from a boot server.	mbootdir(9)
netupd: update a	directory from one on another system.	netupd(1)
pwd: working	directory name.	pwd(1)
readdir, telldir, seekdir, rewinddir, closedir:	directory operations. opendir,	directory(3)
getwd: get current working	directory pathname.	getwd(3)
popd: pop shell	directory stack.	csh(1)
pushd: push shell	directory stack.	csh(1)
dtree: print	directory tree structures.	dtree(1)
pupint, pupnoint: enable or	disable interrupts for received Pup Packets.	pupint(9)
ensignal: enable or	disable signal on ethernet packet arrival.	ensignal(9)
quota: display	disc usage and limits.	quota(1)
unhash:	discard command hash table.	csh(1)
unset:	discard shell variables.	csh(1)
(obsolete). bk: line	discipline for machine-machine communication	bk(4)
byteorder:	discussion of byte-ordering and the Pup package.	byteorder(9)
synchronize a file's in-core state with that on	disk. fsync:	fsync(2)
hk: RK6-11/RK06 and RK07 moving head	disk.	hk(4)
uda: UDA-50	disk controller interface.	uda(4)
getdiskbyname: get	disk description by its name.	getdisk(3X)
disktab:	disk description file.	disktab(5)
df:	disk free.	df(1)
hp: MASSBUS	disk interface.	hp(4)
rx: DEC RX02 floppy	disk interface.	rx(4)
format: how to format	disk packs.	format(8V)
diskpart: calculate default	disk partition sizes.	diskpart(8)
quota: manipulate	disk quotas.	quota(2)
drtest: standalone	disk test program.	drtest(8)
du: summarize	disk usage.	du(1)
diskpart: calculate default disk partition sizes.	diskpart: calculate default disk partition sizes.	diskpart(8)
reboot/halt the system without checking the	disks. fastboot, fasthalt:	fastboot(8)
rxformat: format floppy	disks.	rxformat(8V)
mount, umount: mount and	disktab: disk description file.	disktab(5)
error: analyze and	dismount file system.	mount(8)
gmr: Grinnell Systems	disperse compiler error messages.	error(1)
rain: animated raindrops	display.	gmr(4)
arp: address resolution	display.	rain(6)
processes. top:	display and control.	arp(8C)
gprof:	display and update information about the top cpu	top(1)
snake, snscore:	display call graph profile data.	gprof(1)
quota:	display chase game.	snake(6)
vi: screen oriented (visual)	display disc usage and limits.	quota(1)
umask: change or	display editor based on ex.	vi(1)
screen: repeatedly	display file creation mask.	csh(1)
prof:	display output of command on terminal screen.	screen(1)
sysline:	display profile data.	prof(1)
worms: animate worms on a	display system status on status line of a terminal.	sysline(1)
hypot, cabs: Euclidean	display terminal.	worms(6)
rdist: remote file	distance.	hypot(3M)
dl68: b.out -> .	distribution program.	rdist(1)
dl68: b.out -> .dl downloader component of cc68.	dl downloader component of cc68.	dl68(1)
dl68: b.out -> .dl downloader component of cc68.	dl68: b.out -> .dl downloader component of cc68.	dl68(1)
dlx: download with error correction - 68000 Sun1	dlx: download with error correction - 68000 Sun1	dlx(1)
monitor.		

communications device.	dmc: DEC DMC-11/DMR-11 point-to-point	dmc(4)
dmc: DEC	DMC-11/DMR-11 point-to-point communications device.	dmc(4)
error log.	dmesg: collect system diagnostic messages to form	dmesg(8)
	dmf: DMF-32, terminal multiplexor.	dmf(4)
	dmf: DMF-32, terminal multiplexor.	dmf(4)
	dn: DN-11 autocal unit interface.	dn(4)
	DN-11 autocal unit interface.	dn(4)
	doctor: interact with a psychoanalyst.	doctor(6)
style: analyze surface characteristics of a	document.	style(1)
refer: find and insert literature references in	documents.	refer(1)
w: who is on and what they are	doing.	w(1)
rogue: Exploring The Dungeons of	Doom.	rogue(6)
convert files to press format and print them on the	Dover.. cz (czarina):	cz(1)
output files to press format and print them on the	Dover.. dcat: convert troff phototypesetter	dcat(1)
dtroff: troff to the	Dover.	dtroff(1)
output) files to press format and print them on the	Dover.. dvi: convert dvi (TEX	dvi(1)
ddacct: Dump	Dover Accounting.	ddacct(8)
dpq: prints the	Dover printer queue.	dpq(1)
dprm: remove a file from the	Dover printer queue.	dprm(1)
dpr: dover printer spooler.	dpr: dover printer spooler.	dpr(1)
shutdown: shut	down part of a full-duplex connection.	shutdown(2)
shutdown: close	down the system at a given time.	shutdown(8)
monitor. dlx:	download with error correction - 68000 Sun1	dlx(1)
dl68: b.out -> .dl	downloader component of cc68.	dl68(1)
	dpq: prints the Dover printer queue.	dpq(1)
	dpr: dover printer spooler.	dpr(1)
	dprm: remove a file from the Dover printer queue.	dprm(1)
	dr: DR11-B/DR11-W interface.	dr(4)
interface. drb:	DR11-B/DR11-W general purpose user device	drb(4)
dr:	DR11-B/DR11-W interface.	dr(4)
rand,	drand, irand: return random values.	rand(3F)
graph:	draw a graph.	graph(1G)
interface.	drb: DR11-B/DR11-W general purpose user device	drb(4)
arithmetic: provide	drill in number facts.	arithmetic(6)
ut: UNIBUS TU45 tri-density tape	drive interface.	ut(4)
pty: pseudo terminal	driver.	pty(4)
	drtest: standalone disk test program.	drtest(8)
	drum: paging device.	drum(4)
	dtime: return elapsed execution time.	dtime(3I)
	dtrec: print directory tree structures.	dtrec(1)
	dtroff: troff to the Dover.	dtroff(1)
	du: summarize disk usage.	du(1)
dump: incremental file system	dump.	dump(8)
od: octal, decimal, hex, ascii	dump.	od(1)
rdump: file system	dump across the network.	rdump(8C)
rrestore: restore a file system	dump across the network.	rrestore(8C)
ddacct:	Dump Dover Accounting.	ddacct(8)
	dump, dumpdates: incremental dump format.	dump(5)
dumpfs:	dump file system information.	dumpfs(8)
dump, dumpdates: incremental	dump format.	dump(5)
altoload: load files from an Alto LIP	"dump" format file.	altoload(1)
	dump: incremental file system dump.	dump(8)
savecore: save a core	dump of the operating system.	savecore(8)
kgmon: generate a	dump of the operating system's profile buffers.	kgmon(8)
undump: convert a core	dump to an executable a.out file.	undump(1)
dump,	dumpdates: incremental dump format.	dump(5)
fonts.widths.	dumpfs: show what Press fonts are available in	dumpfs(1)
	dumpfs: dump file system information.	dumpfs(8)
zork: the game of	dungeon.	zork(6)
rogue: Exploring The	Dungeons of Doom.	rogue(6)
	dup, dup2: duplicate a descriptor.	dup(2)
dup,	dup2: duplicate a descriptor.	dup(2)
dup, dup2:	duplicate a descriptor.	dup(2)
dviimp: convert	DVI files to impress format.	dviimp(1)
dviboise: send	DVI files to the HP2680a printer using TCP.	dviboise(1)
catdvi: convert C/A/T files to	DVI format.	catdvi(1)
iprint - convert text files to	DVI format.	iprint(1)
catboise: convert C/A/T files to	DVI format and print on Boise.	catboise(1)
dvip, dvid: convert a	dvi (TeX output) file to press format.	dvip(1)
them on the Dover.. dvi: convert	dvi (TeX output) files to press format and print	dvi(1)
using TCP.	dviboise: send DVI files to the HP2680a printer	dviboise(1)
format.. dvi,	dvid: convert a dvi (TeX output) file to press	dvip(1)
	dviimp: convert DVI files to impress format.	dviimp(1)
press format..	dvip, dvid: convert a dvi (TeX output) file to	dvip(1)
format and print them on the Dover..	dvi: convert a dvi (TeX output) file to press	dvi(1)
	dvipress: convert dvi (TeX output) files to press	dvipress(1)
	dz: DZ-11 communications multiplexer.	dz(4)

dz:	DZ-11 communications multiplexer.	dz(4)
	ec: 3Com 10 Mb/s Ethernet interface.	ec(4)
echo:	echo arguments.	csh(1)
echo:	echo arguments.	echo(1)
	echo: echo arguments.	csh(1)
	echo: echo arguments.	echo(1)
pupecho, echoserve: Pup	Echo protocol user and server.	pupecho(1)
ping: IP/ICMP	echo user program.	ping(1)
pupecho.	echoserve: Pup Echo protocol user and server.	pupecho(1)
	ecvt, fevt, gvvt: output conversion.	ecvt(3)
	ed: text editor.	ed(1)
end, etext,	edata: last locations in program.	end(3)
ex,	edit: text editor.	ex(1)
vipw:	edit the password file.	vipw(8)
edquota:	edit user quotas.	edquota(8)
ed: text	editor.	ed(1)
emacs: a screen	editor.	emacs(1)
ex, edit: text	editor.	ex(1)
fed: font	editor.	fed(1)
ld: link	editor.	ld(1)
sed: stream	editor.	sed(1)
vi: screen oriented (visual) display	editor based on ex.	vi(1)
a.out: assembler and link	editor output.	a.out(5)
	edquota: edit user quotas.	edquota(8)
efinit, efwdinsert, efginsert, cfchinsert,	efAND: build ethernet filters. /ensetfilt.	enetfilter(9)
/ensetfilt, cfinit, cfwdinsert, efginsert,	efchinsert, cfAND: build ethernet filters.	enetfilter(9)
whoami: print	effective current user id.	whoami(1)
setregid: set real and	effective group ID.	setregid(2)
setreuid: set real and	effective user ID's.	setreuid(2)
vfork: spawn new process in a virtual memory	efficient way.	vfork(2)
build ethernet filters. (enetfilter) ensetfilt,	efinit, efwdinsert, efginsert, cfchinsert, cfAND:	enetfilter(9)
	efl: Extended Fortran Language.	efl(1)
(enetfilter) ensetfilt, efinit, efwdinsert,	efginsert, cfchinsert, cfAND: build ethernet/	enetfilter(9)
eftp: Pup	EI-TTP package.	eftp(9)
	eftp: Pup EI-TTP package.	eftp(9)
program with routing.	eftprec: receive-only PUP/EI-TTP file transfer	eftprec(1)
with routing.	eftpsend: send-only PUP/EI-TTP file transfer program	eftpsend(1)
ethernet filters. (enetfilter) ensetfilt, cfinit,	efwdinsert, efginsert, cfchinsert, cfAND: build	enetfilter(9)
grep,	efrep, fgrep: search a file for a pattern.	grep(1)
etime, dtime: return	elapsed execution time.	etime(3I)
insque, remque: insert/remove	element from a queue.	insque(3)
soelim:	eliminate .so's from nroff input.	soelim(1)
	else: alternative commands.	csh(1)
	emacs: a screen editor.	emacs(1)
Emacs data base .br/ d b a d d : add entry to an	Emacs data base .br d b c r e a t e - create an	dbadd(1)
/an Emacs data base .br d b e r e a t e -- create an	Emacs data base .br d b l i s t - list contents/	dbadd(1)
/data base .br d b l i s t - list contents of an	Emacs data base .br d b p r i n t - print an entry/	dbadd(1)
	en: Xerox 3 Mb/s Ethernet interface.	en(4)
(ARP) routines. (enarp)	en10mbpuparp, ennoarp: Address Resolution Protocol	enarp(9)
Packets. pupint, pupoint:	enable or disable interrupts for received Pup	pupint(9)
arrival. ensignal:	enable or disable signal on ethernet packet	ensignal(9)
setquota:	enable/disable quotas on a file system.	setquota(2)
Protocol (ARP) routines.	(enarp) en10mbpuparp, ennoarp: Address Resolution	enarp(9)
uencode: format of an	encoded uencode file.	uencode(5)
crypt:	encode/decode.	crypt(1)
mail. uencode,uencode:	encode/decode a binary file for transmission via	uencode(1C)
crypt, setkey,	encrypt: DES encryption.	crypt(3)
crypt, setkey, encrypt: DES	encryption.	crypt(3)
makekey: generate	encryption key.	makekey(8)
	end, etext, edata: last locations in program.	end(3)
finger: front	end for finger.	finger(1)
pupgetnet: get host, net numbers of local	end of pup channel. pupgethost.	pupgethost(9)
logout:	end session.	csh(1)
	end: terminate loop.	csh(1)
/getfsspec, getfsfile, getfstype, setfsent,	endfsent: get file system descriptor file entry.	getfscnt(3X)
getgrent, getgrgid, getgrnam, setgrent,	endgrent: get group file entry.	getgrent(3)
gethostbyaddr, gethostbyname, sethostent,	endhostent: get network host entry. gethostent,	gethostent(3n)
	endif: terminate conditional.	csh(1)
getnetent, getnetbyaddr, getnetbyname, setnetent,	endnetent: get network entry.	getnetent(3n)
socket: create an	endpoint for communication.	socket(2)
getprotobynumber, getprotobyname, setprotoent,	endprotoent: get protocol entry. getprotoent,	getprotoent(3n)
(pupnettab) setpupnettab, getpupnettab,	endpupnettab: pup configuration table.	pupnettab(9)
getpwent, getpwuid, getpwnam, setpwent,	endpwent: get password file entry.	getpwent(3)
pupgetdtpport: get address of local, remote	ends of pup channel. (pupgetport) pupgetreport,	pupgetport(9)
getservbyport, getservbyname, setservent,	endservent: get service entry. getservent,	getservent(3n)
	endsw: terminate switch.	csh(1)

enstat: print	enet: ethernet packet filter.	enet(4)
eflginsert, efchinsert, cfAND: build ethernet/	enet (packet filter) information.	enstat(8)
interface.	(enetfilter) ensetfilt, cfinit, cfwdinsert,	enetfilter(9)
number: convert Arabic numerals to	enflush: flush an ethernet input file.	enflush(9)
routines. (enarp) enl0mbpuparp.	engethost: determine Pup host number of ethernet	engethost(9)
	English.	number(6)
	ennoarp: Address Resolution Protocol (ARP)	enarp(9)
	enopen: open an ethernet file.	enopen(9)
	enread: read a packet from an ethernet file.	enread(9)
	enroll: secret mail.	xsend(1)
	ensetbacklog: set ethernet input queue backlog.	ensetbacklog(9)
efchinsert, cfAND: build ethernet/ (enetfilter)	ensetfilt, cfinit, cfwdinsert, eflginsert,	enetfilter(9)
packet arrival.	ensignal: enable or disable signal on ethernet	ensignal(9)
	enstat: print enet (packet filter) information.	enstat(8)
	entries from name list.	nlist(3)
	enry.	chfn(1)
	entry. getfsent, getfsspec, getfsfile, getfstype,	getfsent(3X)
setfsent, endfsent: get file system descriptor file	entry. getgrent, getgrgid,	getgrent(3)
getgrnam, setgrent, endgrent: get group file	entry. gethostent, gethostbyaddr, gethostbyname,	gethostent(3n)
sethostent, endhostent: get network host	entry. getnetent, getnetbyaddr,	getnetent(3n)
getnetbyname, setnetent, endnetent: get network	entry. /getprotobyname, getprotobyname,	getprotoent(3n)
setprotoent, endprotoent: get protocol	entry. getpwent, getpwuid,	getpwent(3)
getpwnam, setpwent, endpwent: get password file	entry. getservent, getservbyport,	getservent(3n)
getservbyname, setservent, endservent: get service	entry.	unlink(2)
unlink: remove directory	entry.	unlink(3F)
unlink: remove a directory	entry attributes for an address.	mattributes(9)
mattributes: get Pup Network Directory	entry from a /br d b l i s t - list contents	dbadd(1)
of an Emacs data base .br d b p r i n t - print an	entry to an Emacs data base .br d b c r e a t e -	dbadd(1)
create an Emacs data base .br/ d b a d d : add	environ: execute a file. execd,	execd(3)
execv, execl, execlp, execlvp, exec, exece, execl,	environ: user environment.	environ(7)
	environment.	cs(1)
	environment.	environ(7)
setenv: set variable in	environment.	printenv(1)
environ: user	environment name.	getenv(3)
printenv: print out the	environment variables.	cs(1)
getenv: value for	environment variables.	getenv(3F)
unsetenv: remove	enwrite: write a packet to the ethernet.	enwrite(9)
getenv: get value of	enq.	enqchar(7)
	enq constructs.	deroll(1)
enqchar: special character definitions for	enq, neqn, checkcq: typeset mathematics.	enq(1)
deroll: remove nroll, troff, tbl and	enqchar: special character definitions for enq.	enqchar(7)
	erase. label, line, circle, arc, move, cont, point,	plot(3X)
linemod, space, closepl: graphics/ plot: openpl,	error: analyze and disperse compiler error	error(1)
messages.	error correction - 68000 Sun1 monitor.	dlx(1)
dlx: download with	error log.	dmsg(8)
dmsg: collect system diagnostic messages to form	error message file by massaging C source.	mkstr(1)
mkstr: create an	error message from Pup package routines.	puperrmsg(9)
PupErrMsg: human-readable	error messages.	error(1)
error: analyze and disperse compiler	error messages.	perror(3)
perror, sys_errlist, sys_err: system	error messages.	perror(3F)
perror, gerror, icrno: get system	error numbers.	intro(2)
intro: introduction to system calls and	error recovery.	cyacc(1)
cyacc: modified yacc allowing much improved	errors.	spell(1)
spell, spellin, spellout: find spelling	errors.	traper(3F)
traper: trap arithmetic	etext, edata: last locations in program.	end(3)
end,	ethernet.	enwrite(9)
enwrite: write a packet to the	Ethernet, netsend:	netsend(1)
send a short message to one or more users on the	ethernet file.	enopen(9)
enopen: open an	ethernet file.	enread(9)
enread: read a packet from an	ethernet filters. (enetfilter) ensetfilt, cfinit,	enetfilter(9)
cfwdinsert, eflginsert, efchinsert, cfAND: build	ethernet input file.	enflush(9)
enflush: flush an	ethernet input queue backlog.	ensetbacklog(9)
ensetbacklog: set	Ethernet interface.	de(4)
de: DEC DEUNA 10 Mb/s	Ethernet interface.	cc(4)
cc: 3Com 10 Mb/s	Ethernet interface.	en(4)
en: Xerox 3 Mb/s	ethernet interface.	engethost(9)
engethost: determine Pup host number of	ethernet interface.	il(4)
il: Interlan 10 Mb/s	ethernet minor devices.	etherport(1)
etherport: show status of	ethernet packet arrival.	ensignal(9)
ensignal: enable or disable signal on	ethernet packet filter.	enet(4)
enet:	etherport: show status of ethernet minor devices.	etherport(1)
	etime, dtime: return elapsed execution time.	etime(3F)
	Euclidean distance.	hypot(3M)
hypot, cabs:	eval, exec, exit, export, login, read, readonly, /	sh(1)
/if, while, : , , break, continue, cd,	eval: re-evaluate shell data.	cs(1)
	expr:	expr(1)
expr:	evaluate arguments as an expression.	expr(1)

device interface. ps:	Evans and Sutherland Picture System 2 graphics	ps(4)
history: print history	event list.	cs(1)
screen oriented (visual) display editor based on	ex. vi:	vi(1)
	ex, edit: text editor.	ex(1)
lpq: spool queue	examination program.	lpq(1)
(except) raise, raise_sys(): C	(except) raise, raise_sys(): C exception handling.	except(3)
execl, execv, execl, execlp, execvp,	exception handling.	except(3)
/while, :, ., ., break, continue, cd, eval,	exec, exece, exect, environ: execute a file.	execl(3)
execl, execv, execl, execlp, execvp, exec,	exec, exit, export, login, read, readonly, set,/	sh(1)
execl, execv, execl, execlp, execvp, exec,	exec: overlay shell with specified command.	cs(1)
execl, execv, execl, execlp, execvp, exec,	exece, exect, environ: execute a file.	execl(3)
execl, execv, execl, execlp, execvp, exec,	execl, execv, execl, execlp, execvp, exece,	execl(3)
execl, execv, execl, execlp, execvp, exec,	execl, execlp, execvp, exec, exece, exect,	execl(3)
execl, execv, execl, execlp, execvp, exec,	execlp, execvp, exec, exece, exect, environ:	execl(3)
execl, execv, execl, execlp, execvp, exec,	exect, environ: execute a file.	execl(3)
execl, execv, execl, execlp, execvp, exec,	executable a.out file.	undump(1)
execl, execv, execl, execlp, execvp, exec,	executable files with persistent text.	sticky(8)
execl, execv, execl, execlp, execvp, exec,	execute a file. execl, execv, execl,	execl(3)
execl, execv, execl, execlp, execvp, exec,	execute a file.	execvc(2)
execl, execv, execl, execlp, execvp, exec,	execute a subroutine after a specified time.	alarm(3F)
execl, execv, execl, execlp, execvp, exec,	execute a UNIX command.	system(3F)
execl, execv, execl, execlp, execvp, exec,	execute command repeatedly.	cs(1)
execl, execv, execl, execlp, execvp, exec,	execute commands at a later time.	at(1)
execl, execv, execl, execlp, execvp, exec,	executed in reverse order.	lastcomm(1)
execl, execv, execl, execlp, execvp, exec,	execution.	remote(1)
execl, execv, execl, execlp, execvp, exec,	execution.	uux(1C)
execl, execv, execl, execlp, execvp, exec,	execution accounting file.	acct(5)
execl, execv, execl, execlp, execvp, exec,	execution for an interval.	sleep(1)
execl, execv, execl, execlp, execvp, exec,	execution for an interval.	sleep(3F)
execl, execv, execl, execlp, execvp, exec,	execution for an interval.	sleep(3)
execl, execv, execl, execlp, execvp, exec,	execution profile.	monitor(3)
execl, execv, execl, execlp, execvp, exec,	execution profiler.	pxp(1)
execl, execv, execl, execlp, execvp, exec,	execution server.	rexc(8C)
execl, execv, execl, execlp, execvp, exec,	execution time.	etime(3F)
execl, execv, execl, execlp, execvp, exec,	execution time profile.	profil(2)
execl, execv, execl, execlp, execvp, exec,	executor.	pix(1)
execl, execv, execl, execlp, execvp, exec,	execv, execl, execlp, execvp, exec, exece, exect,	execl(3)
execl, execv, execl, execlp, execvp, exec,	execvc: execute a file.	execvc(2)
execl, execv, execl, execlp, execvp, exec,	execvp, exec, exece, exect, environ: execute a	execl(3)
execl, execv, execl, execlp, execvp, exec,	existing file.	link(3F)
execl, execv, execl, execlp, execvp, exec,	existing file system.	tunefs(8)
execl, execv, execl, execlp, execvp, exec,	exit, export, login, read, readonly, set, shift,/	sh(1)
execl, execv, execl, execlp, execvp, exec,	exit from switch.	cs(1)
execl, execv, execl, execlp, execvp, exec,	exit: leave shell.	cs(1)
execl, execv, execl, execlp, execvp, exec,	_exit: terminate a process.	exit(2)
execl, execv, execl, execlp, execvp, exec,	exit: terminate a process after flushing any	exit(3)
execl, execv, execl, execlp, execvp, exec,	exit: terminate process with status.	exit(3F)
execl, execv, execl, execlp, execvp, exec,	exit while/foreach loop.	cs(1)
execl, execv, execl, execlp, execvp, exec,	exlog: extract data from system load log file.	exlog(1)
execl, execv, execl, execlp, execvp, exec,	exp, log, log10, pow, sqrt: exponential, logarithm,	exp(3M)
execl, execv, execl, execlp, execvp, exec,	expand argument list.	cs(1)
execl, execv, execl, execlp, execvp, exec,	expand tabs to spaces, and vice versa.	expand(1)
execl, execv, execl, execlp, execvp, exec,	expand, unexpand: expand tabs to spaces, and vice	expand(1)
execl, execv, execl, execlp, execvp, exec,	expire: remove outdated news articles.	expire(8)
execl, execv, execl, execlp, execvp, exec,	explain, diction - print wordy sentences; thesaurus	explain(1)
execl, execv, execl, execlp, execvp, exec,	explain: print wordy sentences; thesaurus for	diction(1)
execl, execv, execl, execlp, execvp, exec,	exploration game.	aardvark(6)
execl, execv, execl, execlp, execvp, exec,	exploration game.	adventure(6)
execl, execv, execl, execlp, execvp, exec,	Exploring The Dungeons of Doom.	rogue(6)
execl, execv, execl, execlp, execvp, exec,	exponent.	frexp(3)
execl, execv, execl, execlp, execvp, exec,	exponential, logarithm, power, square root.	exp(3M)
execl, execv, execl, execlp, execvp, exec,	export, login, read, readonly, set, shift, times,/	sh(1)
execl, execv, execl, execlp, execvp, exec,	expr: evaluate arguments as an expression.	expr(1)
execl, execv, execl, execlp, execvp, exec,	expression.	expr(1)
execl, execv, execl, execlp, execvp, exec,	expression handler.	regex(3)
execl, execv, execl, execlp, execvp, exec,	expressions.	cparen(1)
execl, execv, execl, execlp, execvp, exec,	extend bibliographic database.	addbib(1)
execl, execv, execl, execlp, execvp, exec,	Extended Fortran Language.	efl(1)
execl, execv, execl, execlp, execvp, exec,	extended statistics on .b file.	pr68(1)
execl, execv, execl, execlp, execvp, exec,	extended syntax for makefiles.	buildmake(1)
execl, execv, execl, execlp, execvp, exec,	extract data from system load log file.	exlog(1)
execl, execv, execl, execlp, execvp, exec,	extract strings from C programs to implement shared	xstr(1)
execl, execv, execl, execlp, execvp, exec,	eyacc: modified yacc allowing much improved error	eyacc(1)
execl, execv, execl, execlp, execvp, exec,	f77: Fortran 77 compiler.	f77(1)
execl, execv, execl, execlp, execvp, exec,	f77 I/O initialization.	ioinit(3F)
execl, execv, execl, execlp, execvp, exec,	f77 tape I/O. topen,	topen(3F)

remote: remote host description	file.	remotc(5)
rename: change the name of a	file.	rename(2)
rename: rename a	file.	rename(3F)
rev: reverse lines of a	file.	rev(1)
rmdir: remove a directory	file.	rmdir(2)
size: size of an object	file.	size(1)
size68: prints sizes of segments in a .b or .68	file.	size68(1)
the printable strings in a object, or other binary	file. strings: find	strings(1)
sum: sum and count blocks in a	file.	sum(1)
symlink: make symbolic link to a	file.	symlink(2)
tail: deliver the last part of a	file.	tail(1)
weave: convert web file into pascal file, tex	file. tangle,	tangle(1)
touch: update date last modified of a	file.	touch(1)
undump: convert a core dump to an executable a.out	file.	undump(1)
uniq: report repeated lines in a	file.	uniq(1)
unpct: remove lines beginning with % from a	file.	unpct(1)
utime: adjust the access or modification time of a	file.	utime(8)
uucode: format of an encoded uucode	file.	uucode(5)
vipw: edit the password	file.	vipw(8)
versions of object modules were used to construct a	file. what: show what	what(1)
write, writev: write on a	file.	write(2)
filetime: tell minutes since	file (access, modification) time.	filetime(8)
leaf: PUP Leaf Remote	File Access Protocol Server.	leaf(8)
diff: differential	file and directory comparator.	diff(1)
rcs: change RCS	file attributes.	rcs(1)
bugfiler:	file bug reports in folders automatically.	bugfiler(8)
mkstr: create an error message	file by massaging C source.	mkstr(1)
diff3: 3-way differential	file comparison.	diff3(1)
fcntl:	file control.	fcntl(2)
rcp: remote	file copy.	rcp(1C)
umask: change or display	file creation mask.	umask(2)
umask: set	file creation mode mask.	umask(2)
	file: determine file type.	file(1)
mbootdir: get a boot	file directory from a boot server.	mbootdir(9)
rdist: remote	file distribution program.	rdist(1)
setfsent, endfsent: get file system descriptor	file entry. /getfsspec, getfsfile, getfstype,	getfsent(3X)
getgrgid, getgrnam, setgrent, endgrent: get group	file entry. getgrent,	getgrent(3)
getpwnam, setpwent, endpwent: get password	file entry. getpwent, getpwuid,	getpwent(3)
grep, egrep, fgrep: search a	file for a pattern.	grep(1)
open: open a	file for reading or writing, or create a new file.	open(2)
newsr: information	file for readnews(1) and checknews(1).	newsr(5)
aliases: aliases	file for sendmail.	aliases(5)
rl68: print relocation commands in a .b	file for the 68000.	rl68(1)
uucode, uucode: encode/decode a binary	file for transmission via mail.	uucode(1C)
ar: archive (library)	file format.	ar(5)
tar: tape archive	file format.	tar(5)
dprm: remove a	file from the Dover printer queue.	dprm(1)
which: locate a program	file including aliases and paths (csh only).	which(1)
fsplit: split a multi-routine Fortran	file into individual files.	fsplit(1)
tangle, weave: convert web	file into pascal file, tex file.	tangle(1)
split: split a	file into pieces.	split(1)
merge: three-way	file merge.	merge(1)
pmerge: pascal	file merger.	pmerge(1)
mktemp: make a unique	file name.	mktemp(3)
fseek, ftell: reposition a	file on a logical unit.	fseek(3F)
more, page:	file perusal filter for crt viewing.	more(1)
stat, lstat, fstat: get	file status.	stat(2)
stat, lstat, fstat: get	file status.	stat(3F)
mkfs: construct a	file system.	mkfs(8)
mkproto: construct a prototype	file system.	mkproto(8)
mount, umount: mount or remove	file system.	mount(2)
mount, umount: mount and dismount	file system.	mount(8)
newfs: construct a new	file system.	newfs(8)
repquota: summarize quotas for a	file system.	repquota(8)
setquota: enable/disable quotas on a	file system.	setquota(2)
tunefs: tune up an existing	file system.	tunefs(8)
repair. fsck:	file system consistency check and interactive	fsck(8)
getfsfile, getfstype, setfsent, endfsent: get	file system descriptor file entry. /getfsspec,	getfsent(3X)
dcheck:	file system directory consistency check.	dcheck(8)
dump: incremental	file system dump.	dump(8)
rdump:	file system dump across the network.	rdump(8C)
rrestore: restore a	file system dump across the network.	rrestore(8C)
hier:	file system hierarchy.	hier(7)
dumpfs: dump	file system information.	dumpfs(8)
quot: summarize	file system ownership.	quot(8)
quotacheck:	file system quota consistency checker.	quotacheck(8)

quotaon, quotaoff: turn	file system quotas on and off.	quotaon(8)
restore: incremental	file system restore.	restore(8)
ichk: check	file system storage consistency check.	ichk(8)
mtab: mounted	file system table.	mtab(5)
fs, inode: format of	file system volume.	fs(5)
tangle, weave: convert web file into pascal	file, tex file.	tangle(1)
utime: set	file times.	utime(3C)
utimes: set	file times.	utimes(2)
uuser: send a	file to a remote host.	uuser(1C)
truncate: truncate a	file to a specified length.	truncate(2)
dvip, dvid: convert a dvi (TeX output)	file to press format.	dvip(1)
ftpser: PUP	File Transfer Protocol Service.	ftpser(8)
ftp:	file transfer program.	ftp(1C)
pupftp: Pup	File Transfer Program.	pupftp(1)
tftp: trivial	file transfer program.	tftp(1C)
cftpcc: receive-only PUP/EFTP	file transfer program with routing.	cftpcc(1)
cftpssend: send-only PUP/EFTP	file transfer program with routing.	cftpssend(1)
ftpd: DARPA Internet	File Transfer Protocol server.	ftpd(8C)
tftpd: DARPA Trivial	File Transfer Protocol server.	tftpd(8C)
file: determine	file type.	file(1)
lower: lower the case of a	filename.	lower(1)
basename: strip	filename affixes.	basename(1)
glob:	filename expand argument list.	glob(1)
fstat: filter	filenames according to commands in a status file.	fstat(8)
ferror, feof, clearerr,	fileno: stream status inquiries.	ferror(3S)
checknr: check nroff/troff	files.	checknr(1)
cmp: compare two	files.	cmp(1)
comm: select or reject lines common to two sorted	files.	comm(1)
config: build system configuration	files.	config(8)
xref: cross reference C source	files.	xref(1)
find: find	files.	find(1)
split a multi-routine Fortran file into individual	files. fsplit:	split(1)
ident: identify	files.	ident(1)
include: search for and print header (include)	files.	include(1)
makedev: make system special	files.	makedev(8)
mv: move or rename	files.	mv(1)
nm68: print name list of MC68000 object	files.	nm68(1)
reverse byte order in 68000 .b and .68 (b.out)	files. rev68:	rev68(1)
print log messages and other information about RCS	files. rlog:	rlog(1)
mdir, rm: remove (unlink) directories or	files.	mdir(1)
sort: sort or merge	files.	sort(1)
compact, uncompact, ccatt: compress and uncompress	files, and cat them.	compact(1)
intro: introduction to special	files and hardware support.	intro(4)
catman: create the cat	files for the manual.	catman(8)
altoload: load	files from an Alto EFTP "dump" format file.	altoload(1)
fsync: synchronize a	file's in-core state with that on disk.	fsync(2)
imprint - print text	files on Imprint-10.	imprint(1)
m, rmdir: remove (unlink)	files or directories.	m(1)
badsect: create	files to contain bad sectors.	badsect(8)
catdvi: convert C/A/T	files to DVI format.	catdvi(1)
iprint - convert text	files to DVI format.	iprint(1)
catboise: convert C/A/T	files to DVI format and print on Boise.	catboise(1)
dviimp: convert DVI	files to impress format.	dviimp(1)
ImPrint printer.. pressimp: convert press	files to ImPress format and print them on the	pressimp(1)
cz (czarina): convert	files to press format and print them on the Dover..	cz(1)
decat: convert troff phototypesetter output	files to press format and print them on the Dover..	decat(1)
dviipress: convert dvi (TeX output)	files to press format and print them on the Dover..	dviipress(1)
boise: send	files to the HP2680a printer using TCP.	boise(1)
dvi Boise: send DVI	files to the HP2680a printer using TCP.	dvi Boise(1)
sticky: executable	files with persistent text.	sticky(8)
fstab: static information about the	filesystems.	fstab(5)
modification) time.	filetime: tell minutes since file (access,	filetime(8)
enet: ethernet packet	filter.	enet(4)
file. fstat:	filter filenames according to commands in a status	fstat(8)
pupsetdfilt: set a default packet	filter for a Pup channel.	pupsetdfilt(9)
pupsetfilter: set the packet	filter for a Pup channel.	pupsetfilter(9)
more, page: file perusal	filter for crt viewing.	more(1)
enstat: print enet (packet	filter) information.	enstat(8)
colcrt:	filter nroff output for CRT previewing.	colcrt(1)
col:	filter reverse line feeds.	col(1)
efginsert, cfchinsert, cfANID: build ethernet	filters. /enstatf, cfinit, cfwdinsert,	efginsert(9)
plot: graphics	filters.	plot(1G)
puproute:	find a route for a Pup Internet packet.	puproute(9)
refer:	find and insert literature references in documents.	refer(1)
find:	find files.	find(1)
find: find files.	find: find files.	find(1)

look:	find lines in a sorted list.	look(1)
manual. man:	find manual information by keywords; print out the	man(1)
ttyname, isatty, ttyslot:	find name of a terminal.	ttynam(3)
ttynam, isatty:	find name of a terminal port.	ttynam(3F)
library. lorder68:	find ordering relation for an MC68000 object	lorder68(1)
lorder:	find ordering relation for an object library.	lorder(1)
mailcheck:	find out if a user has mail at a PUP host.	mailcheck(1)
mmailcheck:	find out if a user has new mail at a Pup host.	mmailcheck(9)
lookbib: build inverted index for a bibliography,	find references in a bibliography. indxib,	lookbib(1)
spell, spellin, spellout:	find spelling errors.	spell(1)
binary, file. strings:	find the printable strings in a object, or other	strings(1)
fing: front end for finger.	fing: front end for finger.	fing(1)
fingd: network	fingd: network finger server.	fingd(8)
finger.	finger.	fing(1)
chfn: change	finger entry.	chfn(1)
fingd: network	finger server.	fingd(8)
finger: user information lookup program.	finger: user information lookup program.	finger(1)
fold: fold long lines for	finite width output device.	fold(1)
head: give	first few lines.	head(1)
dbminit, fetch, store, delete,	firstkey, nextkey: data base subroutines.	dbm(3X)
fish: play "Go	Fish".	fish(6)
fish: play "Go Fish".	fish: play "Go Fish".	fish(6)
(sh only).	(sh only).	nice(1)
nice, nohup: run a command at low priority	fi: console floppy interface.	fi(4)
arff,	floppy: archiver and copier for floppy.	arff(8V)
extreme values. flmin,	fmax, ffrac, dflmin, dflmax, dffrac, inmax: return	flmin(3F)
return extreme values.	flmin, flmax, ffrac, dflmin, dflmax, dffrac, inmax:	flmin(3F)
trpspe, fpercent: trap and repair	floating point faults.	trpspe(3F)
trapov: trap and repair	floating point overflow.	trapov(3F)
file.	lock: apply or remove an advisory lock on an open	lock(2)
functions. fabs,	floor, ceil: absolute value, floor, ceiling	floor(3M)
fabs, floor, ceil: absolute value,	floor, ceiling functions.	floor(3M)
arff, fscopy: archiver and copier for	floppy.	arff(8V)
rx: DI:C RX02	floppy disk interface.	rx(4)
rxformat: format	floppy disks.	rxformat(8V)
fi: console	floppy interface.	fi(4)
fclose, fflush: close or	flush a stream.	fclose(3S)
enflush:	flush an ethernet input file.	enflush(9)
flush:	flush: flush output to a logical unit.	flush(3F)
flush: flush output to a logical unit.	flush: flush output to a logical unit.	flush(3F)
exit: terminate a process after	flushing any pending output.	exit(3)
device.	fmt: simple text formatter.	fmt(1)
fold:	fold: fold long lines for finite width output	fold(1)
bugfiler: file bug reports in	fold long lines for finite width output device.	fold(1)
vwidth: make troff width table for a	folders automatically.	bugfiler(8)
fed:	font.	vwidth(1)
vfont:	font editor.	fed(1)
inspect and print out information about UNIX	font formats for the Benson-Varian or Versatec.	vfont(5)
dumpfonts: show what Press	fonts. vfontinfo:	vfontinfo(1)
dumpfonts: show what Press fonts are available in	fonts are available in fonts.widths.	dumpfonts(1)
dumpfonts: show what Press fonts are available in	fonts.widths.	dumpfonts(1)
fopen, freopen, fdopen: open a stream.	foreach: loop over list of names.	fopen(3S)
foreach: loop over list of names.	foreground.	csh(1)
foreground.	fork: create a copy of this process.	csh(1)
fork: create a copy of this process.	fork: create a new process.	fork(3F)
fork: create a new process.	form.	fork(2)
form.	form error log.	idate(3F)
form error log.	format.	dmesg(8)
format.	format.	ar(5)
format.	format.	arev(8)
format.	format.	catdvi(1)
format.	format.	dump(5)
format.	format.	dviimp(1)
format. dvip,	format. dvip,	dvip(1)
format.	format.	iprint(1)
format.	format.	tar(5)
catboise: convert C/A/T files to DVI	format and print on Boise.	catboise(1)
cz (czarina): convert files to press	format and print them on the Dover..	cz(1)
convert troff phototypesetter output files to press	format and print them on the Dover.. dcat:	dcat(1)
dvipress: convert dvi (TEX output) files to press	format and print them on the Dover..	dvipress(1)
pressimp: convert press files to lmpress	format and print them on the lmpress printer..	pressimp(1)
indent: indent and	format C program source.	indent(1)
format: how to	format disk packs.	format(8V)
altoload: load files from an Alto I:TP "dump"	format file.	altoload(1)
rxformat:	format floppy disks.	rxformat(8V)
htable: convert NIC standard	format host tables.	htable(8)

gettable: get NIC	format: host tables from a host.	gettable(8C)
	format: how to format disk packs.	format(8V)
vtroff, or troff. vlp:	Format Lisp programs to be printed with nroff,	vlp(1)
ansi: read and write ANSI	format magnetic tapes.	ansi(1)
uuencode:	format of an encoded uuencode file.	uuencode(5)
dir:	format of directories.	dir(5)
fs, inode:	format of file system volume.	fs(5)
core:	format of memory image file.	core(5)
rcsfile:	format of RCS file.	rcsfile(5)
addrfmt: IP/ICMP Address	Format Request user program.	addrfmt(8)
tbl:	format tables for nroff or troff.	tbl(1)
tp: DEC/mag tape	formats.	tp(5)
vfont: font	formats for the Benson-Varian or Versatec.	vfont(5)
scanf, fscanf, sscanf:	formatted input conversion.	scanf(3S)
printf, sprintf, sprintf:	formatted output conversion.	printf(3S)
fmt: simple text	formatter.	fmt(1)
nroff: text	formatting.	nroff(1)
tex, latex, initex, virtex: text	formatting and typesetting.	tex(1)
troff, nroff: text	formatting and typesetting.	troff(1)
ms: text	formatting macros.	ms(7)
me: macros for	formatting papers.	me(7)
f77:	Fortran 77 compiler.	f77(1)
ratfor: rational	Fortran dialect.	ratfor(1)
fpr: print	Fortran file.	fpr(1)
fsplit: split a multi-routine	Fortran file into individual files.	fsplit(1)
cl: Extended	Fortran Language.	cl(1)
intro: introduction to	FORTRAN library functions.	intro(3F)
putc, fputc: write a character to a	fortran logical unit.	putc(3F)
struct: structure	Fortran programs.	struct(1)
adage.	fortune: print a random, hopefully interesting,	fortune(6)
mailer: Mailing list,	forwarding, and alias manager.	mailer(8)
login./ sh, for, case, if, while, :, .	, break, continue, cd, eval, exec, exit, export,	sh(1)
exit, export./ sh, for, case, if, while, :	, ., break, continue, cd, eval, exec,	sh(1)
compiler/interpreter.	fp: Functional Programming language	fp(1)
trpspe,	specht: trap and repair floating point faults.	trpspe(3F)
	fpr: print Fortran file.	fpr(1)
printf,	sprintf, sprintf: formatted output conversion.	printf(3S)
putc, putchar,	fputc, putw: put character or word on a stream.	putc(3S)
putc,	fputc: write a character to a fortran logical unit.	putc(3F)
puts,	fputs: put a string on a stream.	puts(3S)
ik: Ikonas	frame buffer, graphics device interface.	ik(4)
lisz: compile a	Franz Lisp program.	lisz(1)
	fread, fwrite: buffered binary input/output.	fread(3S)
df: disk	free.	df(1)
malloc,	free, realloc, calloc, alloca: memory allocator.	malloc(3)
fopen,	freopen, fdopen: open a stream.	fopen(3S)
exponent.	frex, ldexp, modf: split into mantissa and	frex(3)
from: who is my mail	from?.	from(1)
ring:	front end for finger.	ring(1)
scanf,	fs, inode: format of file system volume.	fs(5)
	fscanf, sscanf: formatted input conversion.	scanf(3S)
mklost + found: make a lost + found directory for	fseek.	mklost + found(8)
fsckblks: print alternate super block numbers for	fsck -b.	fsckblks(8)
repair.	fsck: file system consistency check and interactive	fsck(8)
fsck -b.	fsckblks: print alternate super block numbers for	fsckblks(8)
	fseek, fstell: reposition a file on a logical unit.	fseek(3F)
	fseek, fstell, rewind: reposition a stream.	fseek(3S)
individual files.	fsplit: split a multi-routine Fortran file into	fsplit(1)
	fstab: static information about the filesystems.	fstab(5)
status file.	fstat: filter filenames according to commands in a	fstat(8)
stat, lstat,	fstat: get file status.	stat(2)
stat, lstat,	fstat: get file status.	stat(3F)
on disk.	fsync: synchronize a file's in-core state with that	fsync(2)
fseek,	fstell: reposition a file on a logical unit.	fseek(3F)
fseek,	fstell, rewind: reposition a stream.	fseek(3S)
time,	ftime: get date and time.	time(3C)
altoload: load files from an Alto	FTP "dump" format file.	altoload(1)
	ftp: file transfer program.	ftp(1C)
	ftpd: DARPA Internet File Transfer Protocol server.	ftpd(8C)
	ftpsr: PUP File Transfer Protocol Service.	ftpsr(8)
shutdown: shut down part of a	full-duplex connection.	shutdown(2)
gamma: log gamma	function.	gamma(3M)
compiler/interpreter. fp:	Functional Programming language	fp(1)
bit: and, or, xor, not, lshift, lshift bitwise	functions.	bit(3F)
fabs, floor, ceil: absolute value, floor, ceiling	functions.	floor(3M)
intro: introduction to library	functions.	intro(3)

intro: introduction to compatibility library functions.	intro(3C)
intro: introduction to FORTRAN library functions.	intro(3F)
intro: introduction to mathematical library functions.	intro(3M)
intro: introduction to network library functions.	intro(3n)
intro: introduction to miscellaneous library functions.	intro(3X)
j0, j1, jn, y0, y1, yn: bessell functions.	j0(3M)
cos, tan, asin, acos, atan, atan2: trigonometric functions.	sin(3M)
sinh, cosh, tanh: hyperbolic functions.	sinh(3M)
bessel functions: of two kinds for integer orders.	bessel(3F)
cursor: screen functions with "optimal" cursor motion.	cursor(3X)
fread, fwrite: buffered binary input/output.	fread(3S)
aardvark: yet another exploration game.	aardvark(6)
adventure: an exploration game.	adventure(6)
backgammon: the game.	backgammon(6)
monop: Monopoly game.	monop(6)
snake, sncore: display chase game.	snake(6)
trek: trekkie game.	trek(6)
worm: Play the growing worm game.	worm(6)
canfield, cfscores: the solitaire card game canfield.	canfield(6)
cribbage: the card game cribbage.	cribbage(6)
hangman: Computer version of the game hangman.	hangman(6)
boggle: play the game of boggle.	boggle(6)
chess: the game of chess.	chess(6)
zork: the game of zork.	zork(6)
wump: the game of hunt-the-wumpus.	wump(6)
gamma: log gamma function.	gamma(3M)
gamma: log gamma function.	gamma(3M)
gateway: a Pup gateway program.	pupgateway(8)
gateway: a Pup gateway program.	pupgateway(8)
gatewayinfo: Pup GatewayInfo routing table server.	gatewayinfo(8)
GatewayInfo routing table server.	gatewayinfo(8)
pup10arpser: Pup GatewayInfo routing table server.	pup10arpser(8)
gcov: get core images of running processes.	gcov(1)
gcvt: output conversion.	gcvt(3)
generate a dump of the operating system's profile.	kgmon(8)
generate a fault.	abort(3)
makekey: generate encryption key.	makekey(8)
ncheck: generate names from i-numbers.	ncheck(8)
rand, srand: random number generator.	rand(3C)
lex: generator of lexical analysis programs.	lex(1)
/srandom, initstate, setstate: better random number generator; routines for changing generators.	random(3)
random number generator; routines for changing generators.	random(3)
perror, perror: get system error messages.	perror(3F)
getarg, larg: return command line arguments.	getarg(3F)
getbanner: get system login banner string.	getbanner(3)
GetBCPI.string, PutBCPI.string, GetMesastring, pupstring(9)	pupstring(9)
getc, fgetc: get a character from a logical unit.	getc(3F)
getc, getchar, fgetc, getw: get character or word.	getc(3S)
getchar, fgetc, getw: get character or word from stream.	getc(3S)
getcwd: get pathname of current working directory.	getcwd(3F)
getdiskbyname: get disk description by its name.	getdisk(3X)
getdtablesize: get descriptor table size.	getdtablesize(2)
getgid, getgid: get group identity.	getgid(2)
getenv: get value of environment variables.	getenv(3F)
getenv: value for environment name.	getenv(3)
geteuid: get user identity.	getuid(2)
getfsent, getfsspec, getfsfile, getfstype, getfsent, endfsent: get file system descriptor/ system descriptor file entry.	getfsent(3X)
getfsent, getfsspec, endfsent: get file system descriptor/ descriptor file/ getfsent, getfsspec, getfsfile, getuid.	getfsent(3X)
getfsent, getfsspec, getfsfile, getuid.	getfsent(3X)
getgid: get user or group ID of the caller.	getuid(3F)
getgid, getegid: get group identity.	getgid(2)
getgrent, getgrgid, getgrnam, setgrent, endgrent: get group file entry.	getgrent(3)
getgrgid, getgrnam, setgrent, endgrent: get group file entry.	getgrent(3)
getgrnam, setgrent, endgrent: get group file entry.	getgrent(3)
getgroups: get group access list.	getgroups(2)
(pupmisc) getlong, makelong, getshort, makeshort, getHiWord, getLoWord: miscellaneous Pup routines.	pupmisc(9)
endhostent: get network host entry.	gethostent(3n)
gethostent, gethostent: get network host entry.	gethostent(3n)
sethostent, endhostent: get network host entry.	gethostent(3n)
current host.	gethostid(2)
host.	gethostid(2)
timer.	gethostname(2)
getitimer, setitimer: get/set value of interval.	getitimer(2)
getlog: get user's login name.	getlog(3F)
getlogin: get login name.	getlogin(3)
getlong, makelong, getshort, makeshort, getHiWord,	pupmisc(9)

getlong, makelong, getshort, makeshort, getHiWord, (pupstring) GetBCPLstring, PutBCPLstring, get network entry. getnetent, entry. getnetent, getnetbyaddr, endnetent: get network entry.

getpid, scheduling priority.

protocol entry. getprotoent, getprotobynumber, endprotoent: get protocol entry. getprotoent, setprotoent, endprotoent: get protocol entry. table. (pupnettab) setpupnettab,

get password file entry. getpwent, getpwuid, password file entry. getpwent, resource consumption, utilization.

entry, getservent, getservbyport, endservent: get service entry. getservent, setservent, endservent: get service entry. gettimeofday, settimeofday: gethostname, sethostname: getpriority, setpriority: gethostid, sethostid: getitimer, setitimer: miscellaneous Pup/ (pupmisc) getlong, makelong,

sockets.

getc, getchar, fgetc,

head: shutdown: close down the system at a given time.

ASCII. ctime, localtime, time, ctime, ltime, fish: play setjmp, longjmp: non-local

congraph: plot connectivity of a graph: draw a graph: draw a graph profile data.

gprof: display call ik: Ikonas frame buffer, ps: Evans and Sutherland Picture System 2 plot: arc, move, cont, point, linemod, space, closepl: plot: lib2648: subroutines for the IIP 2648

vgrind: gmrl: chgrp: change getpgrp: get process ingroup: show membership in a specified killpg: send signal to a process setpgrp: set process getgroups: get initgroups: initialize setgroups: set group:

getLoWord: miscellaneous Pup routines. (pupmisc) GetMesastring, PutMesastring: manipulate strings. getnetbyaddr, getnetbyname, setnetent, endnetent: getnetbyname, setnetent, endnetent: get network getnetent, getnetbyaddr, getnetbyname, setnetent, getpagesize: get system page size. getpass: read a password. getpeername: get name of connected peer. getpgrp: get process group. getpid: get process id. getpid, getppid: get process identification. getppid: get process identification. getpriority, setpriority: get/set program getprotobyname, setprotoent, endprotoent: get getprotobynumber, getprotobyname, setprotoent, getprotoent, getprotobynumber, getprotobyname, getpupnettab, endpupnettab: pup configuration getpw: get name from uid. getpwent, getpwuid, getpwnam, setpwent, endpwent: getpwnam, setpwent, endpwent: get password file getpwnam, setpwnam, setpwent, endpwent: get getrlimit, setrlimit: control maximum system getusage: get information about resource gets, fgets: get a string from a stream. getservbyname, setservent, endservent: get service getservbyport, getservbyname, setservent, getservent, getservbyport, getservbyname, get/set date and time. get/set name of current host. get/set program scheduling priority. get/set unique identifier of current host. get/set value of interval timer. getshort, makeshort, getHiWord, getLoWord: getsockname: get socket name. getsockopt, setsockopt: get and set options on gettable: get NIC format host tables from a host. gettimeofday, settimeofday: get/set date and time. getty: set terminal mode. gettytab: terminal configuration data base. getuid, geteuid: get user identity. getuid, getgid: get user or group ID of the caller. getw: get character or word from stream. getwd: get current working directory pathname. give first few lines. glob: filename expand argument list. gmrl: Grinnell Systems display. gmtime, asctime, timezone: convert date and time to gmtime: return system time. "Go fish". goto. goto: command transfer. gprof: display call graph profile data. graph. graph: draw a graph. graph profile data. graphics device interface. graphics device interface. graphics filters. graphics interface. /erase, label, line, circle, graphics interface. graphics terminal. grep, egrep, fgrep: search a file for a pattern. grind nice listings of programs. Grinnell Systems display. gripe: mail a local system bug report. group. group. group. group. group. group access list. group access list. group access list. group file.

pupmisc(9)
pupstring(9)
getnetent(3n)
getnetent(3n)
getnetent(3n)
getpagesize(2)
getpass(3)
getpeername(2)
getpgrp(2)
getpid(3F)
getpid(2)
getpid(2)
getpriority(2)
getprotoent(3n)
getprotoent(3n)
getprotoent(3n)
pupnettab(9)
getpw(3C)
getpwent(3)
getpwent(3)
getpwnam(3)
getrlimit(2)
getusage(2)
gets(3S)
getservent(3n)
getservent(3n)
getservent(3n)
gettimeofday(2)
gethostname(2)
getpriority(2)
gethostid(2)
getitimer(2)
pupmisc(9)
getsockname(2)
getsockopt(2)
gettable(8C)
gettimeofday(2)
getty(8)
gettytab(5)
getuid(2)
getuid(3F)
getc(3S)
getwd(3)
head(1)
shutdown(8)
csh(1)
gmrl(4)
ctime(3)
time(3F)
fish(6)
setjmp(3)
csh(1)
gprof(1)
congraph(1)
graph(1G)
graph(1G)
gprof(1)
ik(4)
ps(4)
plot(1G)
plot(3X)
plot(5)
lib2648(3X)
grep(1)
vgrind(1)
gmrl(4)
gripe(1)
chgrp(1)
getpgrp(2)
ingroup(1)
killpg(2)
setpgrp(2)
getgroups(2)
initgroups(3X)
setgroups(2)
group(5)

getgrgid, getgrnam, setgrent, endgrent: get	group file entry. getgrent.	getgrent(3)
setregid: set real and effective	group: group file.	group(5)
setruuid, setgid, setegid, setrgid: set user and	group ID.	setregid(2)
getuid, getgid: get user or	group ID. setuid, seteuid.	setuid(3)
getgid, getegid: get	group ID of the caller.	getuid(3F)
groups: show	group identity.	getgid(2)
chown: change owner and	group memberships.	groups(1)
gsa: group system accounting.	group of a file.	chown(2)
make: maintain program	group system accounting.	gsa(8)
	groups.	make(1)
worm: Play the	groups: show group memberships.	groups(1)
	growing worm game.	worm(6)
stty.	gsa: group system accounting.	gsa(8)
stop:	gtty: set and get terminal state (defunct).	stty(3C)
reboot: reboot system or	halt a job or process.	csch(1)
	halt processor.	reboot(2)
rmail:	halt: stop the processor.	halt(8)
re_comp, re_exec: regular expression	handle remote mail received via uuip.	rmail(1)
(except) raise, raise_sys(): C exception	handler.	regex(3)
hangman: Computer version of the game	handling.	except(3)
	hangman.	hangman(6)
	hangman: Computer version of the game hangman.	hangman(6)
vhangup: virtually	"hangup" the current control terminal.	vhangup(2)
nohup: run command immune to	hangups.	csch(1)
crash: what	happens when the system crashes.	crash(8V)
link: make a	hard link to a file.	link(2)
intro: introduction to special files and	hardware support.	intro(4)
rehash: recompute command	hash table.	csch(1)
unhash: discard command	hash table.	csch(1)
hashstat: print command	hashing statistics.	csch(1)
	hashstat: print command hashing statistics.	csch(1)
leave: remind you when you	have to leave.	leave(1)
include: search for and print	header (include) files.	include(1)
od: octal, decimal,	hex, ascii dump.	od(1)
	hier: file system hierarchy.	hier(7)
hier: file system	hierarchy.	hier(7)
history: print	history event list.	csch(1)
	history: print history event list.	csch(1)
fortune: print a random,	hk: RK6-11/RK06 and RK07 moving head disk.	hk(4)
sethostid: get/set unique identifier of current	hopefully interesting, adage.	fortune(6)
gethostname, sethostname: get/set name of current	host. gethostid.	gethostid(2)
gettable: get NIC format host tables from a	host.	gethostname(2)
hostname: get name of current	host.	gettable(8C)
print Internet Protocol (IP) identifier of current	host. iphostid: set or	hostname(3F)
mailcheck: find out if a user has mail at a PUP	host.	iphostid(1)
find out if a user has new mail at a Pup	host. mailcheck:	mailcheck(1)
send a message to one or all users at a Pup	host. mmailcheck:	mmailcheck(9)
uucsend: send a file to a remote	host. msendumsg:	msendumsg(9)
htonl, htons, ntohl, ntohs: convert values between	host.	uucsend(1C)
remote: remote	host and network byte order.	byteorder(3n)
gethostbyname, sethostent, endhostent: get network	host description file.	remote(5)
hosts:	host entry. gethostent, gethostbyaddr.	gethostent(3n)
host: print IP	host name data base.	hosts(5)
pupgethost, pupgetnet: get	host names and addresses.	host(1)
engethost: determine Pup	host, net numbers of local end of pup channel.	pupgethost(9)
phones: remote	host number of ethernet interface.	engethost(9)
	host phone number data base.	phones(5)
runtime: show	host: print IP host names and addresses.	host(1)
hostid: set or print identifier of current	host status of local machines.	runtime(1C)
hostname: set or print name of current	host system.	hostid(1)
htable: convert NIC standard format	host system.	hostname(1)
gettable: get NIC format	host tables.	htable(8)
system.	host tables from a host.	gettable(8C)
	hostid: set or print identifier of current host	hostid(1)
locate: location and owner of Pup network	hostname: set or print name of current host system.	hostname(1)
	hostname: get name of current host.	hostname(3F)
	hosts.	locate(1)
uptime: show	hosts: host name data base.	hosts(5)
format:	how long system has been up.	uptime(1)
lib2648: subroutines for the	how to format disk packs.	format(8V)
	IIP 2648 graphics terminal.	lib2648(3X)
boise: send files to the	hp: MASSBUS disk interface.	hp(4)
dviboise: send DVI files to the	IIP2680a printer using TCP.	boise(1)
interface.	IIP2680a printer using TCP.	dviboise(1)
	ht: TM-03/TF-16,TU-45,TU-77 MASSBUS magtape	ht(4)
	htable: convert NIC standard format host tables.	htable(8)

host and network byte order. and network byte order. htonl, routines. PupErrMsg: wump: the game of	htonl, htons, ntohl, ntohs: convert values between htonl, htons, ntohl, ntohs: convert values between host human-readable error message from Pup package hy: Network Systems Hyperchannel interface. hy: Network Systems hyperbolic functions. Hyperchannel interface. hypot, cabs: Euclidean distance. iargc: return command line arguments. icheck: file system storage consistency check. id. ID. ID. setuid, seteuid, setruid, id. ID of the caller. id temporarily. idate, itime: return date or time in numerical ident: identify files. identification. identifier of current host. identifier of current host. identifier of current host system. identify files. identity. identity. identity. ID's. ierrno: get system error messages. if a user has mail at a PUP host. if a user has new mail at a Pup host. if: conditional statement. if mail arrives and who it is from. if, while, : . . . break, continue, cd, ifconfig: configure network interface parameters. ifdef'd lines. II UNIBUS cassette interface. ik: Ikonas frame buffer, graphics device interface. Ikonas frame buffer, graphics device interface. il: Interlan 10 Mb/s Ethernet interface. image. image file. images of running processes. immediate notification. immune to hangups. imp: 1822 network interface. imp: IMP raw socket interface. IMP interface. IMP interface. IMP log interpreter. IMP logger process. imp: IMP raw socket interface. IMP-11A I.II/DII IMP interface. implement shared strings. implog: IMP log interpreter. implogd: IMP logger process. imp: css: DEC xstr: extract strings from C programs to	byteorder(3n) byteorder(3n) puperrmsg(9) wump(6) hy(4) sinh(3M) hy(4) hypot(3M) getarg(3F) icheck(8) getpid(3F) setregid(2) setuid(3) whoami(1) getuid(3F) su(1) idate(3F) ident(1) getpid(2) gethostid(2) iphostid(1) hostid(1) ident(1) getgid(2) getuid(2) setreuid(2) perror(3F) mailcheck(1) mmailcheck(9) csh(1) bill(1) sh(1) ifconfig(8C) unifdel(1) uu(4) ik(4) ik(4) il(4) abort(3F) core(5) gcore(1) csh(1) csh(1) imp(4) imp(4P) acc(4) css(4) implog(8C) implogd(8C) imp(4P) css(4) xstr(1) implog(8C) implogd(8C) dviimp(1) pressimp(1) imprint(1) pressimp(1) btroff(1) itroff(1) imprint(1) eyacc(1) include(1) include(1) which(1) sync(2) dump(5) dump(8) restore(8) indent(1) indent(1) independent operation routines. tgetent, index. index for a bibliography, find references in a index, rindex, lnbk, len: tell about character index, rindex: string operations. strcat,
-------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

last:	indicate last logins of users and teletypes.	last(1)
syscall:	indirect system call.	syscall(2)
fsplit: split a multi-routine Fortran file into bibliography, find references in a bibliography.	individual files.	fsplit(1)
	indxbib, lookbib: build inverted index for a	lookbib(1)
	inet: Internet protocol family.	inet(4F)
inet_lnaof, inet_netof: Internet address/	inet_addr, inet_network, inet_ntoa, inet_makeaddr,	inet(3n)
	inetd: DARPA little protocol server.	inetd(8C)
inet_addr, inet_network, inet_ntoa, inet_makeaddr,	inet_lnaof, inet_netof: Internet address/	inet(3n)
address/ inet_addr, inet_network, inet_ntoa,	inet_makeaddr, inet_lnaof, inet_netof: Internet	inet(3n)
/inet_network, inet_ntoa, inet_makeaddr, inet_lnaof,	inet_netof: Internet address manipulation routines.	inet(3n)
inet_netof: Internet address/ inet_addr,	inet_network, inet_ntoa, inet_makeaddr, inet_lnaof,	inet(3n)
Internet address/ inet_addr, inet_network,	inet_ntoa, inet_makeaddr, inet_lnaof, inet_netof:	inet(3n)
bad144: read/write dec standard 144 bad sector	information.	bad144(8)
dumpfs: dump file system	information.	dumpfs(8)
enstat: print enet (packet filter)	information.	enstat(8)
pac: printer/ploter accounting	information.	pac(8)
puproute: print Pup network routing table	information.	puproute(1)
rlog: print log messages and other	information about RCS files.	rlog(1)
getrusage: get	information about resource utilization.	getrusage(2)
vtimes: get	information about resource utilization.	vtimes(3C)
fstab: static	information about the filesystems.	fstab(5)
top: display and update	information about the top cpu processes.	top(1)
vfontinfo: inspect and print out	information about UNIX fonts.	vfontinfo(1)
man: find manual	information by keywords; print out the manual.	man(1)
newsrc:	information file for readnews(1) and checknews(1).	newsrc(5)
finger: user	information lookup program.	finger(1)
miscellaneous: miscellaneous useful	information pages.	intro(7)
	ingroup: show membership in a specified group.	ingroup(1)
	init: process control initialization.	init(8)
tex, latex,	initex, vrtex: text formatting and typesetting.	tex(1)
	initgroups: initialize group access list.	initgroups(3X)
init: process control	initialization.	init(8)
ioinit: change f77 I/O	initialization.	ioinit(3F)
tset: terminal dependent	initialization.	tset(1)
ttys: terminal	initialization data.	ttys(5)
initgroups:	initialize group access list.	initgroups(3X)
connect:	initiate a connection on a socket.	connect(2)
popen, pclose:	initiate I/O to/from a process.	popen(3)
generator; routines for changing/ random, srandom,	initstate, setstate: better random number	random(3)
flmin, flmax, flfrac, dflmin, dflmax, dflfrac,	inmax: return extreme values.	flmin(3F)
clri: clear	i-node.	clri(8)
fs,	inode: format of file system volume.	fs(5)
read, readv: read	input.	read(2)
soclim: eliminate .so's from nroff	input.	soclim(1)
scanf, fscanf, sscanf: formatted	input conversion.	scanf(3S)
enflush: flush an ethernet	input file.	enflush(9)
ensetbacklog: set ethernet	input queue backlog.	ensetbacklog(9)
pupsetbacklog: set	input queue backlog for pup channel.	pupsetbacklog(9)
ungetc: push character back into	input stream.	ungetc(3S)
fread, fwrite: buffered binary	input/output.	fread(3S)
stdio: standard buffered	input/output package.	intro(3S)
fferror, feof, clearerr, fileno: stream status	inquiries.	fferror(3S)
	insecure: user security monitor.	insecure(8)
refer: find and	insert literature references in documents.	refer(1)
insque, remque:	insert/remove element from a queue.	insque(3)
vfontinfo:	inspect and print out information about UNIX fonts.	vfontinfo(1)
	insque, remque: insert/remove element from a queue.	insque(3)
install:	install binaries.	install(1)
	install: install binaries.	install(1)
learn: computer aided	instruction about UNIX.	learn(1)
doctor:	interact with a psychoanalyst.	doctor(6)
fsck: file system consistency check and	interactive repair.	fsck(8)
fortune: print a random, hopefully	interesting, adage.	fortune(6)
acc: ACC L11/D11 IMP	interface.	acc(4)
cons: VAX-11 console	interface.	cons(4)
css: DEC IMP-11A L11/D11 IMP	interface.	css(4)
ct: phototypesetter	interface.	ct(4)
de: DEC DEUNA 10 Mb/s Ethernet	interface.	de(4)
dn: DN-11 autocall unit	interface.	dn(4)
dr: DR11-B/DR11-W	interface.	dr(4)
drb: DR11-B/DR11-W general purpose user device	interface.	drb(4)
cc: 3Com 10 Mb/s Ethernet	interface.	cc(4)
en: Xerox 3 Mb/s Ethernet	interface.	en(4)
engethost: determine Pup host number of ethernet	interface.	engethost(9)
fl: console floppy	interface.	fl(4)
hp: MASSBUS disk	interface.	hp(4)

ht: TM-03/TE-16,TU-45,TU-77 MASSBUS magtape	interface.	ht(4)
hy: Network Systems Hyperchannel	interface.	hy(4)
ik: Ikonas frame buffer, graphics device	interface.	ik(4)
il: Interlan 10 Mb/s Ethernet	interface.	il(4)
imp: 1822 network	interface.	imp(4)
imp: IMP raw socket	interface.	imp(4P)
lo: software loopback network	interface.	lo(4)
mt: TM78/TU-78 MASSBUS magtape	interface.	mt(4)
mtio: UNIX magtape	interface.	mtio(4)
pcl: DEC CSS PCI-11 B Network	Interface.	pcl(4)
cont, point, linemod, space, closepl: graphics	interface.	/erase, label, line, circle, arc, move,
plot: graphics	interface.	plot(3X)
and Sutherland Picture System 2 graphics device	interface.	plot(5)
pup: raw PUP socket	interface.	ps(4)
rx: DEC RX02 floppy disk	interface.	pup(4P)
tm: TM-11/TE-10 magtape	interface.	rx(4)
ts: TS-11 magtape	interface.	tm(4)
tty: general terminal	interface.	ts(4)
tu: VAX-11/730 and VAX-11/750 TU58 console cassette	interface.	tty(4)
uda: UDA-50 disk controller	interface.	tu(4)
un: Ungermann-Bass	interface.	uda(4)
ut: UNIBUS TU45 tri-density tape drive	interface.	un(4)
uu: TU58/DECtape II UNIBUS cassette	interface.	ut(4)
va: Benson-Varian	interface.	uu(4)
vp: Versatec	interface.	va(4)
ifconfig: configure network	interface parameters.	vp(4)
telnet: user	interface to the TELNET protocol.	ifconfig(8C)
il:	Interlan 10 Mb/s Ethernet interface.	telnet(1C)
swapon: add a swap device for	interleaved paging/swapping.	il(4)
sendmail: send mail over the	internet.	swapon(2)
/inet_ntoa, inet_makeaddr, inet_lnaof, inet_netof:	Internet address manipulation routines.	sendmail(8)
ftpd: DARPA	Internet File Transfer Protocol server.	inet(3n)
whois: ask the ARPA	Internet NIC about a user.	ftpd(8C)
puproute: find a route for a Pup	Internet packet.	whois(1C)
ip:	Internet Protocol.	puproute(9)
inct:	Internet protocol family.	ip(4P)
iphostid: set or print	Internet Protocol (IP) identifier of current host.	inct(4F)
ipbroadcast: broadcasting	Internet Protocol packets.	iphostid(1)
puprouting: Pup	Internet routing table maintenance routines.	ipbroadcast(4P)
tcp:	Internet Transmission Control Protocol.	puprouting(9)
udp:	Internet User Datagram Protocol.	tcp(4P)
spline:	interpolate smooth curve.	udp(4P)
apl: apl	interpreter.	spline(1G)
implog: IMP log	interpreter.	apl(1)
lisp: lisp	interpreter.	implog(8C)
pti: phototypesetter	interpreter.	lisp(1)
px: Pascal	interpreter and executor.	pti(1)
pix: Pascal	interpreter code translator.	px(1)
pi: Pascal	interpreter with C-like syntax.	pix(1)
csh: a shell (command	interprocess communication channel.	pi(1)
pipe: create an	interrupt, sigpause:	csh(1)
atomically release blocked signals and wait for	interrupts for received Pup Packets.	pipe(2)
pupint, pupoint: enable or disable	interrupts in command scripts.	sigpause(2)
onintr: process	introduction to commands.	pupint(9)
intro:	introduction to compatibility library functions.	csh(1)
intro:	introduction to FORTRAN library functions.	intro(1)
intro:	introduction to library functions.	intro(3C)
intro:	introduction to mathematical library functions.	intro(3F)
intro:	introduction to miscellaneous library functions.	intro(3)
intro:	introduction to network library functions.	intro(3M)
networking:	introduction to networking facilities.	intro(3X)
resintro:	introduction to RCS commands.	intro(3n)
intro:	introduction to special files and hardware support.	intro(4N)
intro:	introduction to system calls and error numbers.	resintro(1)
commands. intro:	introduction to system maintenance and operation	intro(4)
ncheck: generate names from	i-numbers.	intro(2)
in a bibliography. indxbib, lookbib: build	inverted index for a bibliography, find references	intro(8)
tread, twrite, trewin, tskipf, tstate: f77 tape	I/O. open, telose,	ncheck(8)
ioinit: change f77	I/O initialization.	lookbib(1)
select: synchronous	i/o multiplexing.	open(3F)
dataio: load the data	i/o prom programmer.	ioinit(3F)
iostat: report	I/O statistics.	select(2)
popen, pclose: initiate	I/O to/from a process.	dataio(1)
	ioctl: control device.	iostat(1)
	ioinit: change f77 I/O initialization.	popen(3)
		ioctl(2)
		ioinit(3F)

	iostat: report I/O statistics.	iostat(1)
host: print	IP host names and addresses.	host(1)
iphostid: set or print Internet Protocol	(IP) identifier of current host.	iphostid(1)
	ip: Internet Protocol.	ip(4P)
net: print	IP net names and addresses.	net(1)
packets.	ipbroadcast: broadcasting Internet Protocol	ipbroadcast(4P)
identifier of current host.	iphostid: set or print Internet Protocol (IP)	iphostid(1)
addrfmt:	IP/ICMP Address Format Request user program.	addrfmt(8)
ping:	IP/ICMP echo user program.	ping(1)
	iprint - convert text files to DVI format.	iprint(1)
rand, drand,	irand: return random values.	rand(3F)
isascii:/ isalpha, isupper, islower, isdigit,	isalnum, isspace, ispunct, isprint, iscntrl,	ctype(3)
isspace, ispunct, isprint, iscntrl, isascii:/	isalpha, isupper, islower, isdigit, isalnum,	ctype(3)
isalnum, isspace, ispunct, isprint, iscntrl,	isascii: character classification macros. /isdigit,	ctype(3)
ttynam,	isatty: find name of a terminal port.	ttynam(3F)
	isatty, ttyslot: find name of a terminal.	ttynam(3)
/isdigit, isalnum, isspace, ispunct, isprint,	iscntrl, isascii: character classification macros.	ctype(3)
iscntrl, isascii:/ isalpha, isupper, islower,	isdigit, isalnum, isspace, ispunct, isprint,	ctype(3)
isprint, iscntrl, isascii:/ isalpha, isupper,	islower, isdigit, isalnum, isspace, ispunct,	ctype(3)
/islower, isdigit, isalnum, isspace, ispunct,	isprint, iscntrl, isascii: character classification/	ctype(3)
/isupper, islower, isdigit, isalnum, isspace,	ispunct, isprint, iscntrl, isascii: character/	ctype(3)
isalpha, isupper, islower, isdigit, isalnum,	isspace, ispunct, isprint, iscntrl, isascii:/	ctype(3)
system:	issue a shell command.	system(3)
ispunct, isprint, iscntrl, isascii:/	isupper, islower, isdigit, isalnum, isspace,	ctype(3)
isalpha,	itime: return date or time in numerical form.	idate(3F)
idate,	itroff: troff to the lmpint printer.	itroff(1)
	j0, j1, jn, y0, y1, yn: bessel functions.	j0(3M)
	j1, jn, y0, y1, yn: bessel functions.	j0(3M)
	jn, y0, y1, yn: bessel functions.	j0(3M)
	bg: place	job in background.
	fg: bring	job into foreground.
jobs: print current	job list.	job list.
stop: halt a	job or process.	job or process.
kill: kill	jobs and processes.	jobs and processes.
lprm: remove	jobs from the line printer spooling queue.	jobs from the line printer spooling queue.
	jobs: print current job list.	jobs: print current job list.
	join: relational database operator.	join(1)
msgs: system messages and	junk mail program.	msgs(1)
netalias:	keeping track of remote user names and passwords.	netalias(1)
arptab: show contents of	kernel ARP table.	arptab(1)
makekey: generate encryption	key.	makekey(8)
apropos: locate commands by	keyword lookup.	apropos(1)
man: find manual information by	keywords: print out the manual.	man(1)
	kg: KL-11/DL-11W line clock.	kg(4)
profile buffers.	kgmon: generate a dump of the operating system's	kgmon(8)
kill:	kill jobs and processes.	kill(1)
	kill: kill jobs and processes.	kill(1)
	kill: send a signal to a process.	kill(3F)
	kill: send signal to a process.	kill(2)
	kill: terminate a process with extreme prejudice.	kill(1)
	killpg: send signal to a process group.	killpg(2)
bessel functions: of two	kinds for integer orders.	bessel(3F)
mkisssofdeath: send a	KissOfDeath Pup.	mkisssofdeath(9)
kg:	KL-11/DL-11W line clock.	kg(4)
routing. patchroute:	kludge to support Stanford Pup-based subnet	patchroute(8)
ment,	kmem: main memory.	mem(4)
linemod, space, closepl:/ plot: openpl, erase,	label, line, circle, arc, move, cont, point,	plot(3X)
awk: pattern scanning and processing	language.	awk(1)
bc: arbitrary-precision arithmetic	language.	bc(1)
efl: Extended Fortran	language.	efl(1)
set, shift, times, trap, umask, wait: command	language. /exit, export, login, read, readonly,	sh(1)
fp: Functional Programming	language compiler/interpreter.	fp(1)
vgrindfs: vgrind's	language definition data base.	vgrindfs(5)
calen: print	large-format calendar.	calen(1)
order.	lastcomm: show last commands executed in reverse	lastcomm(1)
bibtex: make a	LaTeX bibtexography.	bibtex(1)
typesetting. tex,	latex, initex, virtex: text formatting and	tex(1)
	latex: TeX with a macro package preloaded.	latex(1)
	ld: link editor.	ld(1)
	ld68: .b -> b.out linker for the MC68000.	ld68(1)
frexp,	ldexp, modf: split into mantissa and exponent.	frexp(3)
leaf: PUP	leaf: PUP Leaf Remote File Access Protocol Server.	leaf(8)
	Leaf Remote File Access Protocol Server.	leaf(8)
	learn: computer aided instruction about UNIX.	learn(1)
leave: remind you when you have to	leave.	leave(1)
	leave: remind you when you have to leave.	leave(1)

exit:	leave shell.	csh(1)
index, rindex, lnlnk,	len: tell about character objects.	index(3F)
truncate: truncate a file to a specified	length.	truncate(2)
linelen: print line	lengths for a text file.	linelen(1)
lex: generator of	lex: generator of lexical analysis programs.	lex(1)
acc: ACC	lexical analysis programs.	lex(1)
css: DEC IMP-11A	I.II/DII IMP interface.	acc(4)
terminal.	I.II/DII IMP interface.	css(4)
ranlib: convert archives to random	lib2648: subroutines for the IIP 2648 graphics	lib2648(3X)
lorder: find ordering relation for an object	libraries.	ranlib(1)
find ordering relation for an MC68000 object	library.	lorder(1)
ar: archive	library. lorder68:	lorder68(1)
intro: introduction to	(library) file format.	ar(5)
intro: introduction to compatibility	library functions.	intro(3)
intro: introduction to FORTRAN	library functions.	intro(3C)
intro: introduction to mathematical	library functions.	intro(3F)
intro: introduction to network	library functions.	intro(3M)
intro: introduction to miscellaneous	library functions.	intro(3n)
ar: archive and	library functions.	intro(3X)
overview: overview of Pup	library maintainer.	ar(1)
limit: alter per-process resource	library routines.	overview(9)
unlimit: remove resource	limit: alter per-process resource limitations.	csh(1)
quota: display disc usage and	limitations.	csh(1)
getarg, iargc: return command	limitations.	csh(1)
space, closepl:/ plot: openpl, erase, label,	limits.	quota(1)
kg: KI-11/DI-11W	line arguments.	getarg(3F)
(obsolete). bk:	line, circle, arc, move, cont, point, linemod,	plot(3X)
col: filter reverse	line clock.	kg(4)
linelen: print	line discipline for machine-machine communication	bk(4)
sysline: display system status on status	line feeds.	col(1)
lpr: off	line lengths for a text file.	linelen(1)
lp:	line of a terminal.	sysline(1)
print: pr to the	line print.	lpr(1)
lpc:	line printer.	lp(4)
lpd:	line printer.	print(1)
lprm: remove jobs from the	line printer control program.	lpc(8)
/erase, label, line, circle, arc, move, cont, point,	line printer daemon.	lpd(8)
head: give first few	line printer spooling queue.	lprm(1)
unifdef: remove ifdef'd	linelen: print line lengths for a text file.	linelen(1)
unpnt: remove	linemod, space, closepl: graphics interface.	plot(3X)
comm: select or reject	lines.	head(1)
fold: fold long	lines.	unifdef(1)
makedep: construct dependency	lines beginning with % from a file.	unpnt(1)
uniq: report repeated	lines common to two sorted files.	comm(1)
look: find	lines for finite width output device.	fold(1)
rev: reverse	lines for makefiles.	makedep(1)
readlink: read value of a symbolic	lines in a file.	uniq(1)
ld:	lines in a sorted list.	look(1)
a.out: assembler and	lines of a file.	rev(1)
link: make a hard	link.	readlink(2)
symlink: make symbolic	link editor.	ld(1)
link: make a	link editor output.	a.out(5)
ld68: .b -> b.out	link: make a hard link to a file.	link(2)
ln: make	link: make a link to an existing file.	link(3F)
symchck: check for bad symbolic	link to a file.	link(2)
lxref:	link to an existing file.	symlink(2)
lisp:	linker for the MC68000.	link(3F)
liszt: compile a Franz	links.	ld68(1)
troff, vlp: Format	links.	ln(1)
glob: filename expand argument	lint: a C program verifier.	symchck(1)
history: print history event	lisp cross reference program.	lint(1)
jobs: print current job	lisp interpreter.	lxref(1)
shift: manipulate argument	lisp: lisp interpreter.	lisp(1)
getgroups: get group access	lisp: lisp interpreter.	lisp(1)
initgroups: initialize group access	Lisp program.	liszt(1)
look: find lines in a sorted	Lisp programs to be printed with nroff, vtroff, or	vlp(1)
nlist: get entries from name	list.	csh(1)
nm: print name	list.	csh(1)
setgroups: set group access	list.	csh(1)
	list.	csh(1)
	list.	getgroups(2)
	list.	initgroups(3X)
	list.	look(1)
	list.	nlist(3)
	list.	nm(1)
	list.	setgroups(2)

symorder: rearrange name	list.	symorder(1)
varargs: variable argument	list.	varargs(3)
/- create an Emacs data base .br d b l i s t -	list contents of an Emacs data base .br/	dbadd(1)
ls:	list contents of directory.	ls(1)
mailer: Mailing	list, forwarding, and alias manager.	mailer(8)
allusers: print	list of all authorized users.	allusers(1)
nm68: print name	list of MC68000 object files.	nm68(1)
foreach: loop over	list of names.	csh(1)
users: compact	list of users who are on the system.	users(1)
listen:	listen for connections on a socket.	listen(2)
	listen: listen for connections on a socket.	listen(2)
vgrind: grind nice	listings of programs.	vgrind(1)
	lisz: compile a Franz Lisp program.	lisz(1)
refer: find and insert	literature references in documents.	refer(1)
inetd: DARPA	little protocol server.	inetd(8C)
	ln: make links.	ln(1)
index, rindex,	lnblnk, len: tell about character objects.	index(3F)
	lo: software loopback network interface.	lo(4)
loadlog: log the current time, number of users, and	load average.	loadlog(1)
altload:	load files from an Alto FTP "dump" format file.	altload(1)
loadavg: average	load log data on a weekly basis.	loadavg(1)
exlog: extract data from system	load log file.	exlog(1)
dataio:	load the data i/o prom programmer.	dataio(1)
	loadavg: average load log data on a weekly basis.	loadavg(1)
load average.	loadlog: log the current time, number of users, and	loadlog(1)
	loc: return the address of an object.	loc(3F)
timeck: poll the	localnet for the current time.	timeck(8C)
and time to ASCII. ctime,	localtime, gmtime, asctime, timezone: convert date	ctime(3)
(csh only). which:	locate a program file including aliases and paths	which(1)
apropos:	locate commands by keyword lookup.	apropos(1)
	locate: location and owner of Pup network hosts.	locate(1)
whereis:	locate source, binary, and or manual for program.	whereis(1)
locate:	location and owner of Pup network hosts.	locate(1)
end, etext, edata: last	locations in program.	end(3)
flock: apply or remove an advisory	lock on an open file.	flock(2)
	lock: reserve a terminal.	lock(1)
collect system diagnostic messages to form error	log. dmsg:	dmsg(8)
syslog, openlog, closlog: control system	log.	syslog(3)
loadavg: average load	log data on a weekly basis.	loadavg(1)
exlog: extract data from system load	log file.	exlog(1)
gamma:	log gamma function.	gamma(3M)
implog: IMP	log interpreter.	implog(8C)
power, square root. exp,	log, log10, pow, sqrt: exponential, logarithm,	exp(3M)
rlog: print	log messages and other information about RCS files.	rlog(1)
syslog:	log systems messages.	syslog(8)
average. loadlog:	log the current time, number of users, and load	loadlog(1)
square root. exp, log,	log10, pow, sqrt: exponential, logarithm, power,	exp(3M)
exp, log, log10, pow, sqrt: exponential,	logarithm, power, square root.	exp(3M)
rwho: who's	logged in on local machines.	rwho(1C)
implogd: IMP	logger process.	implogd(8C)
flush: flush output to a	logical unit.	flush(3F)
fseek, fcell: reposition a file on a	logical unit.	fseek(3F)
getc, fgetc: get a character from a	logical unit.	getc(3F)
putc, fputc: write a character to a fortran	logical unit.	putc(3F)
rlogin: remote	login.	rlogin(1C)
ac:	login accounting.	ac(8)
accounts). nu: manage user	login accounts (create, modify, destroy Unix	nu(8)
getbanner: get system	login banner string.	getbanner(3)
	login: login new user.	csh(1)
getlog: get user's	login name.	getlog(3F)
getlogin: get	login name.	getlogin(3)
login:	login new user.	csh(1)
passwd: change	login password.	passwd(1)
/break, continue, cd, eval, exec, exit, export,	login, read, readonly, set, shift, times, trap,/	sh(1)
utmp, wtmp:	login records.	utmp(5)
rlogind: remote	login server.	rlogind(8C)
chsh: change default	login shell.	chsh(1)
	login: sign on.	login(1)
last: indicate last	logins of users and teletypes.	last(1)
	logout: end session.	csh(1)
setjmp,	longjmp: non-local goto.	setjmp(3)
	look: find lines in a sorted list.	look(1)
find references in a bibliography. indxbib,	lookbib: build inverted index for a bibliography,	lookbib(1)
apropos: locate commands by keyword	lookup.	apropos(1)
finger: user information	lookup program.	finger(1)
break: exit while/foreach	loop.	csh(1)

continue: cycle in	loop.	csh(1)
end: terminate	loop.	csh(1)
foreach:	loop over list of names.	csh(1)
lo: software	loopback network interface.	lo(4)
library.	lorder: find ordering relation for an object	lorder(1)
object library.	lorder68: find ordering relation for an MC68000	lorder68(1)
mklost + found: make a	lost + found directory for fsck.	mklost + found(8)
	lower: lower the case of a filename.	lower(1)
	lower:	lower(1)
	lp: line printer.	lp(4)
	lpc: line printer control program.	lpc(8)
	lpd: line printer daemon.	lpd(8)
	lpq: spool queue examination program.	lpq(1)
	lpr: off line print.	lpr(1)
queue.	lprm: remove jobs from the line printer spooling	lprm(1)
	ls: list contents of directory.	ls(1)
	lseek: move read/write pointer.	lseek(2)
bit: and, or, xor, not, rshift,	lshift bitwise functions.	bit(3F)
stat,	lstat, fstat: get file status.	stat(2)
stat,	lstat, fstat: get file status.	stat(3F)
time, ctime,	ltime, grmtime: return system time.	time(3F)
	lxref: lisp cross reference program.	lxref(1)
	m4: macro processor.	m4(1)
bk: line discipline for	machine-machine communication (obsolete).	bk(4)
ruptime: show host status of local	machines.	ruptime(1C)
rwho: who's logged in on local	machines.	rwho(1C)
latex: TeX with a	macro package preloaded.	latex(1)
m4:	macro processor.	m4(1)
alias: shell	macros.	csh(1)
isprint, iscntrl, isascii: character classification	macros. /isdigit, isalnum, isspace, ispunct,	ctype(3)
ms: text formatting	macros.	ms(7)
translate version 6 manual macros to version 7	macros. tman:	tman(1)
me:	macros for formatting papers.	me(7)
man:	macros to typeset manual.	man(7)
tman: translate version 6 manual	macros to version 7 macros.	tman(1)
	maddtoname: translate a Pup Port address to a name.	maddtoname(9)
	mt: magnetic tape manipulating program.	mt(1)
ansi: read and write ANSI format	magnetic tapes.	ansi(1)
ht: TM-03/TI-16,TU-45,TU-77 MASSBUS	magtape interface.	ht(4)
mt: TM78/TU-78 MASSBUS	magtape interface.	mt(4)
mtio: UNIX	magtape interface.	mtio(4)
tm: TM-11/TI-10	magtape interface.	tm(4)
ts: TS-11	magtape interface.	ts(4)
rmt: remote	magtape protocol module.	rmt(8C)
mail: send and receive	mail.	mail(1)
recnews: receive unprocessed articles via	mail.	recnews(8)
sendnews: send news articles via	mail.	sendnews(8)
encode/decode a binary file for transmission via	mail. uuencode,uudecode:	uuencode(1C)
uurec: receive processed news articles via	mail.	uurec(8)
xsend, xget, enroll: secret	mail.	xsend(1)
gripe:	mail a local system bug report.	gripe(1)
sendbug:	mail a system bug report to 4bsd-bugs.	sendbug(1)
mailaddr:	mail addressing description.	mailaddr(7)
newaliases: rebuild the data base for the	mail aliases file.	newaliases(1)
binmail: send or receive	mail among users.	binmail(1)
bill: be notified if	mail arrives and who it is from.	bill(1)
mailcheck: find out if a user has	mail at a PUP host.	mailcheck(1)
mmailcheck: find out if a user has new	mail at a PUP host.	mmailcheck(9)
from: who is my	mail from?.	from(1)
pmail: print out	mail in the post office.	pmail(1)
sendmail: send	mail over the internet.	sendmail(8)
pup-mailer: deliver	mail over the .SM PUP network.	pup-mailer(8)
msgs: system messages and junk	mail program.	msgs(1)
mail: handle remote	mail received via uucp.	rmail(1)
	mail: send and receive mail.	mail(1)
	mailaddr: mail addressing description.	mailaddr(7)
host.	mailcheck: find out if a user has mail at a PUP	mailcheck(1)
manager.	mailer: Mailing list, forwarding, and alias	mailer(8)
mailer:	Mailing list, forwarding, and alias manager.	mailer(8)
mcm, kcm:	main memory.	mem(4)
make:	maintain program groups.	make(1)
ar: archive and library	maintainer.	ar(1)
intro: introduction to system	maintenance and operation commands.	intro(8)
puprouting: Pup Internet routing table	maintenance routines.	puprouting(9)
backup:	make a backup version copy of a file.	backup(1)
mkdir:	make a directory.	mkdir(1)

mkdir:	make a directory file.	mkdir(2)
link:	make a hard link to a file.	link(2)
bibtex:	make a LaTeX bibliography.	bibtex(1)
link:	make a link to an existing file.	link(3F)
mklost+found:	make a lost+found directory for fsck.	mklost+found(8)
mcsrvreq:	make a MiscServices request.	mcsrvreq(9)
mknod:	make a special file.	mknod(2)
mktemp:	make a unique file name.	mktemp(3)
ln:	make links.	ln(1)
	make: maintain program groups.	make(1)
symlink:	make symbolic link to a file.	symlink(2)
makedev:	make system special files.	makedev(8)
vwidth:	make troff width table for a font.	vwidth(1)
script:	make typescript of terminal session.	script(1)
	makedep: construct dependency lines for makefiles.	makedep(1)
	makedev: make system special files.	makedev(8)
preprocessor to provide extended syntax for makedep: construct dependency lines for	makefiles. buildmake:	buildmake(1)
	makefiles.	makedep(1)
	makekey: generate encryption key.	makekey(8)
getLoWord: miscellaneous Pup/ (pupmisc) getlong, routines. (pupmisc) getlong, makelong, getshort, allocator. the manual.	makelong, getshort, makeshort, getHiWord, makeshort, getHiWord, getLoWord: miscellaneous Pup malloc, free, realloc, calloc, alloca: memory	pupmisc(9) pupmisc(9) malloc(3)
	man: find manual information by keywords; print out man: macros to typeset manual.	man(1) man(7)
	Unix accounts). nu: manage user login accounts (create, modify, destroy manager.	nu(8) mailer(8)
mailer: Mailing list, forwarding, and alias shift:	manipulate argument list.	csh(1)
	manipulate disk quotas.	quota(2)
PutBCPLstring, GetMesastring, PutMesastring:	manipulate strings. (pupstring) GetBCPLstring,	pupstring(9)
	tp: manipulate tape archive.	tp(1)
	route: manually manipulate the routing tables.	route(8C)
mt: magnetic tape	manipulating program.	mt(1)
rtar, rdd, rmt: remote tape	manipulation programs.	rtar(1)
inet_lnaof, inet_nctof: Internet address	manipulation routines. /inet_ntoa, inet_makeaddr,	inet(3n)
frexp, ldexp, modf: split into	mantissa and exponent.	frexp(3)
catman: create the cat files for the	manual.	catman(8)
find manual information by keywords; print out the man: macros to typeset	manual. man:	man(1) man(7)
whereis: locate source, binary, and or manual. man: find	manual. manually manipulate the routing tables.	whereis(1) man(1)
trman: translate version 6	manual information by keywords; print out the manual macros to version 7 macros.	trman(1)
	manually manipulate the routing tables.	route(8C)
pupindescrip, pupoutdescrip, pupfidtochan: access	mapping between PupChans and fids. (pupdescrip)	pupdescrip(9)
umask: change or display file creation	mask.	csh(1)
sigsetmask: set current signal	mask.	sigsetmask(2)
umask: set file creation mode	mask.	umask(2)
mkstr: create an error message file by hp:	massaging C source.	mkstr(1)
ht: TM-03/TE-16,TU-45,TU-77	MASSBUS disk interface.	hp(4)
mt: TM78/TU-78	MASSBUS magtape interface.	ht(4)
intro: introduction to	MASSBUS magtape interface.	mt(4)
eqn, neqn, checkeq: typeset	mathematical library functions.	intro(3M)
attributes for an address.	mathematics.	eqn(1)
getrlimit, setrlimit: control	mattributes: get Pup Network Directory entry	mattributes(9)
vlimit: control	maximum system resource consumption.	getrlimit(2)
server.	maximum system resource consumption.	vlimit(3C)
	mbootdir: get a boot file directory from a boot	mbootdir(9)
	mbootrequest: request a boot-load.	mbootrequest(9)
dc: DEC DEUNA 10	Mb/s Ethernet interface.	dc(4)
cc: 3Com 10	Mb/s Ethernet interface.	cc(4)
en: Xerox 3	Mb/s Ethernet interface.	en(4)
il: Interlan 10	Mb/s Ethernet interface.	il(4)
cc68: C compiler for the	MC68000.	cc68(1)
ld68: .b -> b.out linker for the	MC68000.	ld68(1)
pc68: Pascal compiler for the	MC68000.	pc68(1)
nm68: print name list of	MC68000 object files.	nm68(1)
lorder68: find ordering relation for an	MC68000 object library.	lorder68(1)
	me: macros for formatting papers.	me(7)
	measure terminal output rate.	ttime(1)
ttime:	media.	bed(6)
bed: convert to antique	Megabit ring.	vv(4)
vv: Proteon proNET 10	mem, kmem: main memory.	mem(4)
	membership in a specified group.	ingroup(1)
ingroup: show	memberships.	groups(1)
groups: show group	memory.	mcn(4)
mem, kmem: main	memory allocator.	malloc(3)
malloc, free, realloc, calloc, alloca:	memory allocator.	valloc(3)
valloc: aligned		

vfork: spawn new process in a virtual	memory efficient way.	vfork(2)
abort: terminate abruptly with	memory image.	abort(3F)
core: format of	memory image file.	core(5)
vmstat: report virtual	memory statistics.	vmstat(1)
merge: three-way file	merge.	merge(1)
sort: sort or	merge files.	sort(1)
rcsmerge:	merge RCS revisions.	rcsmerge(1)
pmerge: pascal file	merge: three-way file merge.	merge(1)
mkstr: create an error	merger.	pmerge(1)
recv, recvfrom, recvmsg: receive a	msg: permit or deny messages.	msg(1)
send, sendto, sendmsg: send a	message file by massaging C source.	mkstr(1)
PupErrMsg: human-readable error	message from a socket.	recv(2)
msendumsg: send a	message from a socket.	send(2)
netsend: send a short	message from Pup package routines.	puperrmsg(9)
error: analyze and disperse compiler error	message to one or all users at a Pup host.	msendumsg(9)
msg: permit or deny	message to one or more users on the Ethernet.	netsend(1)
perror, sys_errlist, sys_nerr: system error	messages.	error(1)
perror, gerror, ierrno: get system error	messages.	msg(1)
psignal, sys_siglist: system signal	messages.	perror(3)
syslog: log systems	messages.	perror(3F)
msgs: system	messages.	psignal(3)
rlog: print log	messages and junk mail program.	syslog(8)
dmsg: collect system diagnostic	messages and other information about RCS files.	msgs(1)
mille: play	messages to form error log.	rlog(1)
etherport: show status of ethernet	Mille Bourmes.	dmsg(8)
filetime: tell	mille: play Mille Bourmes.	mille(6)
intro: introduction to	minor devices.	mille(6)
pages.	minutes since file (access, modification) time.	etherport(1)
getshort, makeshort, getHiWord, getLoWord:	miscellaneous library functions.	filetime(8)
miscellaneous:	miscellaneous: miscellaneous useful information	intro(3X)
miscsrvreq: make a	miscellaneous Pup routines. /getlong, makelong,	intro(7)
miscserver:	miscellaneous useful information pages.	pupmisc(9)
miscsrvreq: make a	miscserver: MiscServices server for Pup.	intro(7)
miscserver:	MiscServices request.	miscserver(8)
mkdir: make a directory.	MiscServices server for Pup.	miscsrvreq(9)
mkdir: make a directory file.	mkdir: make a directory.	miscserver(8)
mkfs: construct a file system.	mkdir: make a directory file.	mkdir(1)
mkissofdeath: send a KissOfDeath Pup.	mkfs: construct a file system.	mkdir(2)
mklost + found: make a lost + found directory for fsck.	mkissofdeath: send a KissOfDeath Pup.	mkfs(8)
mknod: build special file.	mklost + found: make a lost + found directory for fsck.	mkissofdeath(9)
mknod: make a special file.	mknod: build special file.	mklost + found(8)
mkproto: construct a prototype file system.	mknod: make a special file.	mknod(8)
mkstr: create an error message file by massaging C	mkproto: construct a prototype file system.	mknod(2)
mktemp: make a unique file name.	mkstr: create an error message file by massaging C	mkproto(8)
mlookup: translate a name to a Pup address.	mktemp: make a unique file name.	mkstr(1)
mmailcheck: find out if a user has new mail at a	mlookup: translate a name to a Pup address.	mktemp(3)
mod: Modula-2 compiler.	mmailcheck: find out if a user has new mail at a	mlookup(9)
mode.	mod: Modula-2 compiler.	mmailcheck(9)
mode.	mode.	mod(1)
mode for a Pup channel.	mode.	chmod(1)
mode mask.	mode for a Pup channel.	getty(8)
mode of a file.	mode mask.	pupsetmode(9)
mode of file.	mode of a file.	umask(2)
modf: split into mantissa and exponent.	mode of file.	chmod(3F)
modification time.	modf: split into mantissa and exponent.	chmod(2)
modification time of a file.	modification time.	frexp(3)
modified of a file.	modification time of a file.	filetime(8)
modified yacc allowing much improved error	modified of a file.	utime(8)
modify, destroy Unix accounts).	modified yacc allowing much improved error	touch(1)
Modula-2 compiler.	modify, destroy Unix accounts).	eyacc(1)
module.	Modula-2 compiler.	nu(8)
module controller/drives.	module.	mod(1)
modules were used to construct a file.	module controller/drives.	rmt(8C)
moncontrol: prepare execution profile.	modules were used to construct a file.	up(4)
monitor.	moncontrol: prepare execution profile.	what(1)
monitor.	monitor.	monitor(3)
monitor, monstartup, moncontrol: prepare execution	monitor, monstartup, moncontrol: prepare execution	dlx(1)
monop: Monopoly game.	monop: Monopoly game.	insecure(8)
Monopoly game.	monstartup, moncontrol: prepare execution profile.	monitor(3)
monstartup, moncontrol: prepare execution profile.	more, page: file pcrusal filter for crt viewing.	monop(6)
more, page: file pcrusal filter for crt viewing.	more users on the Ethernet.	monop(6)
more users on the Ethernet.	motion.	monop(3)
mount and dismount file system.	mount and dismount file system.	more(1)
		netsend(1)
		curses(3X)
		mount(8)

mount, umount:	mount or remove file system.	mount(2)
	mount, umount: mount and dismount file system.	mount(8)
	mount, umount: mount or remove file system.	mount(2)
mtab:	mounted file system table.	mtab(5)
bmove: block	move a buffer.	bmove(9)
plot: openpl, erase, label, line, circle, arc,	move, cont, point, linemod, space, closepl:/	plot(3X)
mv:	move or rename files.	mv(1)
lseek:	move read/write pointer.	lseek(2)
hk: RK6-11/RK06 and RK07	moving head disk.	hk(4)
	ms: text formatting macros.	ms(7)
	mscopyreq: make a MiscServices request.	mscopyreq(9)
Pup host.	msendumsg: send a message to one or all users at a	msendumsg(9)
	msgs: system messages and junk mail program.	msgs(1)
	msunbootreq: request a boot-load.	msunbootreq(9)
	mt: magnetic tape manipulating program.	mt(1)
	mt: TM78/TU-78 MASSBUS magtape interface.	mt(4)
	mtab: mounted file system table.	mtab(5)
	mtimecheck: get time from a Pup server.	mtimecheck(9)
	mtio: UNIX magtape interface.	mtio(4)
eyacc: modified yacc allowing	much improved error recovery.	eyacc(1)
dh: DH-11/DM-11 communications	multiplexer.	dh(4)
dz: DZ-11 communications	multiplexer.	dz(4)
select: synchronous i/o	multiplexing.	select(2)
dmf: DMF-32, terminal	multiplexor.	dmf(4)
fsplit: split a	multi-routine Fortran file into individual files.	fsplit(1)
switch:	multi-way command branch.	csh(1)
	mv: move or rename files.	mv(1)
from: who is	my mail from?.	from(1)
getdiskbyname: get disk description by its	name.	getdisk(3X)
getenv: value for environment	name.	getenv(3)
getlog: get user's login	name.	getlog(3F)
getlogin: get login	name.	getlogin(3)
getsockname: get socket	name.	getsockname(2)
maddtoname: translate a Pup Port address to a	name.	maddtoname(9)
mktemp: make a unique file	name.	mktemp(3)
pwd: working directory	name.	pwd(1)
tty: get terminal	name.	tty(1)
hosts: host	name data base.	hosts(5)
networks: network	name data base.	networks(5)
protocols: protocol	name data base.	protocols(5)
services: service	name data base.	services(5)
getpw: get	name from uid.	getpw(3C)
nlist: get entries from	name list.	nlist(3)
nm: print	name list.	nm(1)
symorder: rearrange	name list.	symorder(1)
nm68: print	name list of MC68000 object files.	nm68(1)
rename: change the	name of a file.	rename(2)
ttynam, isatty, ttyslot: find	name of a terminal.	ttynam(3)
ttynam, isatty: find	name of a terminal port.	ttynam(3F)
getpeername: get	name of connected peer.	getpeername(2)
gethostname, sethostname: get/set	name of current host.	gethostname(2)
hostname: get	name of current host.	hostname(3F)
hostname: set or print	name of current host system.	hostname(1)
whereami: report	name of terminal.	whereami(1)
mlookup: translate a	name to a Pup address.	mlookup(9)
bind: bind a	name to a socket.	bind(2)
foreach: loop over list of	names.	csh(1)
host: print IP host	names and addresses.	host(1)
net: print IP net	names and addresses.	net(1)
netalias: keeping track of remote user	names and passwords.	netalias(1)
term: conventional	names for terminals.	term(7)
ncheck: generate	names from i-numbers.	ncheck(8)
	ncheck: generate names from i-numbers.	ncheck(8)
eqn,	neqn, checkeq: typeset mathematics.	eqn(1)
cnest: check for	nested comments in C code.	cnest(1)
net: print IP	net names and addresses.	net(1)
pupgethost, pupgetnet: get host,	net numbers of local end of pup channel.	pupgethost(9)
	net: print IP net names and addresses.	net(1)
passwords.	netalias: keeping track of remote user names and	netalias(1)
Directory.	netdirprint: print text version of Pup Network	netdirprint(8)
on the Ethernet.	netsend: send a short message to one or more users	netsend(1)
	netstat: show network status.	netstat(1)
system.	netupd: update a directory from one on another	netupd(1)
pup-mailer: deliver mail over the .SM PUP	network.	pup-mailer(8)
connect your terminal to a remote computer via Pup	network. puptelnet:	pupclnet(1)
rdump: file system dump across the	network.	rdump(8C)

rrestore: restore a file system dump across the	network.	rrestore(8C)
nthl, ntohs: convert values between host and	network byte order. htonl, htons,	byteorder(3n)
buildnetdir: build binary-format Pup	Network Directory.	buildnetdir(8)
netdirprint: print text version of Pup	Network Directory.	netdirprint(8)
mattributes: get Pup	Network Directory entry attributes for an address.	mattributes(9)
getnetbyname, setnetent, endnetent: get	network entry. getnetent, getnetbyaddr,	getnetent(3n)
findg:	network finger server.	findg(8)
gethostbyname, sethostent, endhostent: get	network host entry. gethostent, gethostbyaddr,	gethostent(3n)
locate: location and owner of Pup	network hosts.	locate(1)
imp: 1822	network interface.	imp(4)
lo: software loopback	network interface.	lo(4)
pcl: DEC CSS PCL-11 B	Network Interface.	pcl(4)
ifconfig: configure	network interface parameters.	ifconfig(8C)
intro: introduction to	network library functions.	intro(3n)
networks:	network name data base.	networks(5)
routed:	network routing daemon.	routed(8C)
puproute: print Pup	network routing table information.	puproute(1)
netstat: show	network status.	netstat(1)
hy:	Network Systems II hyperchannel interface.	hy(4)
timecheck: checks and sets Pup	network time.	timecheck(1)
networking: introduction to	networking facilities.	intro(4N)
puplistenall: open Pup channels on all connected	networking: introduction to networking facilities.	intro(4N)
creat: create a	networks. puplisten,	puplisten(9)
open a file for reading or writing, or create a	networks: network name data base.	networks(5)
newfs: construct a	new file.	creat(2)
arcv: convert archives to	new file. open:	open(2)
mmailcheck: find out if a user has	new file system.	newfs(8)
fork: create a	new format.	arcv(8)
vfork: spawn	new mail at a Pup host.	mmailcheck(9)
login: login	new process.	fork(2)
adduser: procedure for adding	new process in a virtual memory efficient way.	vfork(2)
aliases file.	new user.	csh(1)
expire: remove outdated	new users.	adduser(8)
vnews: read	newaliases: rebuild the data base for the mail	newaliases(1)
sendnews: send	news: construct a new file system.	newfs(8)
uurec: receive processed	news articles.	expire(8)
checknews(1).	news articles via mail.	vnews(1)
dbminit, fetch, store, delete, firstkey,	news articles via mail.	sendnews(8)
whois: ask the ARPA Internet	news:rc: information file for readnews(1) and	uurec(8)
gettable: get	nextkey: data base subroutines.	newsrc(5)
htable: convert	NIC about a user.	dbm(3X)
vgrind: grind	NIC format host tables from a host.	whois(1C)
(sh only).	NIC standard format host tables.	gettable(8C)
only). nice,	nice listings of programs.	htable(8)
setjmp, longjmp:	nice, nohup: run a command at low priority	vgrind(1)
bit: and, or, xor,	nice: run low priority process.	nice(1)
notify: request immediate	nice: set program priority.	csh(1)
biff: be	nlist: get entries from name list.	nice(3C)
soelim: eliminate .so's from	nm: print name list.	nlist(3)
tbl: format tables for	nm68: print name list of MC68000 object files.	nm(1)
colert: filter	nohup: run a command at low priority (sh	nm68(1)
troff,	nohup: run command immune to hangups.	nice(1)
deroff: remove	non-local goto.	csh(1)
vlp: Format Lisp programs to be printed with	not, rshift, lshift bitwise functions.	setjmp(3)
checknr: check	notification.	bit(3F)
network byte order. htonl, htons,	notified if mail arrives and who it is from.	csh(1)
order. htonl, htons, nthl,	notify: request immediate notification.	csh(1)
destroy Unix accounts).	nroff input.	soelim(1)
UniqueSocket: create unique socket	nroff or troff.	tbl(1)
phones: remote host phone	nroff output for CRT previewing.	colert(1)
arithmetic: provide drill in	nroff: text formatting.	nroff(1)
rand, srand: random	nroff: text formatting and typesetting.	troff(1)
random, srand, initstate, setstate: better random	nroff, troff, tbl and eqn constructs.	deroff(1)
	nroff, vtroff, or troff.	vlp(1)
	nroff/troff files.	checknr(1)
	nthl, ntohs: convert values between host and	byteorder(3n)
	ntohs: convert values between host and network byte	byteorder(3n)
	nu: manage user login accounts (create, modify,	nu(8)
	null: data sink.	null(4)
	number.	uniquesocket(9)
	number: convert Arabic numerals to English.	number(6)
	number data base.	phones(5)
	number facts.	arithmetic(6)
	number generator.	rand(3C)
	number generator; routines for changing generators.	random(3)

engethost: determine Pup host
 loadlog: log the current time,
 atof, atoi, atol: convert ASCII to
 atoo: convert ASCII to octal
 intro: introduction to system calls and error
 fsckblks: print alternate super block
 pupgethost, pupgetnet: get host, net
 number: convert Arabic
 idate, itime: return date or time in

 loc: return the address of an
 long, short: integer
 size: size of an
 nm68: print name list of MC68000
 lorder: find ordering relation for an
 lorder68: find ordering relation for an MC68000
 what: show what versions of
 strings: find the printable strings in a
 index, rindex, lnblnk, len: tell about character
 line discipline for machine-machine communication
 od:
 atoo: convert ASCII to

 pmail: print out mail in the post
 netupd: update a directory from
 msendumsg: send a message to
 netsend: send a short message to

 nohup: run a command at low priority (*sh*
 program file including aliases and paths (*cs*
 file. open:
 pupopen:
 fopen, freopen, fdopen:
 enopen:
 flock: apply or remove an advisory lock on an
 a new file.
 puplisten, puplistenall:
 closedir: directory operations.
 syslog,
 cont, point, linemod, space, closepl:/ plot:
 savcore: save a core dump of the
 kgmon: generate a dump of the
 intro: introduction to system maintenance and
 tgetstr, tgoto, tputs: terminal independent
 bcopy, bcmp, bzero, fls: bit and byte string
 telldir, seekdir, rewinddir, closedir: directory
 strcmpfold: case-folded string comparison
 strepy, strnepy, strlen, index, rindex: string
 join: relational database
 curses: screen functions with
 o68: *s -> .s*
 stty: set terminal
 getsockopt, setsockopt: get and set
 ntohs: convert values between host and network byte
 lastcomm: show last commands executed in reverse
 rev68: reverse byte
 lorder68: find
 lorder: find
 bessel functions: of two kinds for integer
 vi: screen
 rev68: reverse byte order in 68000 .b and .68 (b.
 expire: remove
 a.out: assembler and link editor
 terminate a process after flushing any pending
 ecvt, fcvt, gcvt:
 printf, sprintf, sprintf: formatted
 fold: fold long lines for finite width
 dvip, dvid: convert a dvi (TeX
 Dover.. dcat: convert troff phototypesetter
 Dover.. dvipress: convert dvi (TeX
 colct: filter nroff
 screen: repeatedly display
 ttime: measure terminal
 flush: flush
 foreach: loop
 sendmail: send mail

 number of ethernet interface.
 number of users, and load average.
 numbers.
 numbers.
 numbers.
 numbers for fsck -b.
 numbers of local end of pup channel.
 numerals to English.
 numerical form.
 o68: *.s -> .s* optimizer component of cc68.
 object.
 object conversion.
 object file.
 object files.
 object library.
 object library.
 object modules were used to construct a file.
 object, or other binary, file.
 objects.
 (obsolete). bk:
 octal, decimal, hex, ascii dump.
 octal numbers.
 od: octal, decimal, hex, ascii dump.
 office.
 one on another system.
 one or all users at a Pup host.
 one or more users on the Ethernet.
 onintr: process interrupts in command scripts.
 only). nice,
 only). which: locate a
 open a file for reading or writing, or create a new
 open a pup channel.
 open a stream.
 open an ethernet file.
 open file.
 open: open a file for reading or writing, or create
 open Pup channels on all connected networks.
 opendir, readdir, telldir, seekdir, rewinddir,
 openlog, closelog: control system log.
 openpl, erase, label, line, circle, arc, move,
 operating system.
 operating system's profile buffers.
 operation commands.
 operation routines. tgetent, tgetnum, tgetflag.
 operations.
 operations. opendir, readdir,
 operations. streampfold,
 operations. strcat, strcmp, strncmp,
 operator.
 "optimal" cursor motion.
 optimizer component of cc68.
 options.
 options on sockets.
 order. htonl, htons, ntohs,
 order.
 order in 68000 .b and .68 (b.out) files.
 ordering relation for an MC68000 object library.
 ordering relation for an object library.
 orders.
 oriented (visual) display editor based on ex.
 out) files.
 outdated news articles.
 output.
 output. exit:
 output conversion.
 output conversion.
 output device.
 output) file to press format.
 output files to press format and print them on the
 output) files to press format and print them on the
 output for CRT previewing.
 output of command on terminal screen.
 output rate.
 output to a logical unit.
 over list of names.
 over the internet.

pup-mailer: deliver mail	over the .SM PUP network.	pup-mailer(8)
trapov: trap and repair floating point	overflow.	trapov(3F)
exec:	overlay shell with specified command.	csh(1)
overview:	overview of Pup library routines.	overview(9)
	overview: overview of Pup library routines.	overview(9)
chown: change	owner.	chown(8)
chown: change	owner and group of a file.	chown(2)
locate: location and	owner of Pup network hosts.	locate(1)
quot: summarize file system	ownership.	quot(8)
	pac: printer/ploter accounting information.	pac(8)
puproute: find a route for a Pup Internet	packet.	puproute(9)
ensignal: enable or disable signal on ethernet	packet arrival.	ensignal(9)
enet: ethernet	packet filter.	enet(4)
pupsetdfilt: set a default	packet filter for a Pup channel.	pupsetdfilt(9)
pupsetfilter: set the	packet filter for a Pup channel.	pupsetfilter(9)
enstat: print enet	(packet filter) information.	enstat(8)
pupread: read a pup	packet from a pup channel.	pupread(9)
enread: read a	packet from an ethernet file.	enread(9)
pupwrite: write a	packet to a pup channel.	pupwrite(9)
enwrite: write a	packet to the ethernet.	enwrite(9)
ipbroadcast: broadcasting Internet Protocol	packets.	ipbroadcast(4P)
enable or disable interrupts for received Pup	Packets. pupint, pupnoint:	pupint(9)
format: how to format disk	packs.	format(8V)
more.	page: file perusal filter for crt viewing.	more(1)
getpagesize: get system	page size.	getpagesize(2)
pagesize: print system	page size.	pagesize(1)
miscellaneous: miscellaneous useful information	pages.	intro(7)
	pagesize: print system page size.	pagesize(1)
tk:	paginator for the Tektronix 4014.	tk(1)
swapon: specify additional device for	paging and swapping.	swapon(8)
drum:	paging device.	drum(4)
swapon: add a swap device for interleaved	paging/swapping.	swapon(2)
socketpair: create a	pair of connected sockets.	socketpair(2)
me: macros for formatting	papers.	me(7)
ifconfig: configure network interface	parameters.	ifconfig(8C)
cparen - add	parentheses to C expressions.	cparen(1)
diskpart: calculate default disk	partition sizes.	diskpart(8)
pc:	Pascal compiler.	pc(1)
pc68:	Pascal compiler for the MC68000.	pc68(1)
pxref:	Pascal cross-reference program.	pxref(1)
pdx:	pascal debugger.	pdx(1)
pxp:	Pascal execution profiler.	pxp(1)
pmerge:	pascal file merger.	pmerge(1)
tangle, weave: convert web file into	pascal file, tex file.	tangle(1)
px:	Pascal interpreter.	px(1)
pix:	Pascal interpreter and executor.	pix(1)
pi:	Pascal interpreter code translator.	pi(1)
verch: version changing program for	Pascal sources.	verch(1)
	passwd: change login password.	passwd(1)
	passwd: password file.	passwd(5)
getpass: read a	password.	getpass(3)
passwd: change login	password.	passwd(1)
passwd:	password file.	passwd(5)
vipw: edit the	password file.	vipw(8)
getpwuid, getpwnam, setpwent, endpwent: get	password file entry. getpwent,	getpwent(3)
netalias: keeping track of remote user names and	passwords.	netalias(1)
subnet routing.	patchroute: kludge to support Stanford Pup-based	patchroute(8)
getwd: get current working directory	pathname.	getwd(3)
getcwd: get	pathname of current working directory.	getcwd(3F)
which: locate a program file including aliases and	paths (csh only).	which(1)
grep, egrep, fgrep: search a file for a	pattern.	grep(1)
awk:	pattern scanning and processing language.	awk(1)
	pause: stop until signal.	pause(3C)
	pc: Pascal compiler.	pc(1)
	pc68: Pascal compiler for the MC68000.	pc68(1)
	pci: DEC CSS PCL-11 B Network Interface.	pci(4)
pci: DEC CSS	PCI-11 B Network Interface.	pci(4)
popen,	pelose: initiate I/O to/from a process.	popen(3)
	pdx: pascal debugger.	pdx(1)
getpeername: get name of connected	peer.	getpeername(2)
exit: terminate a process after flushing any	pending output.	exit(3)
msg:	permit or deny messages.	msg(1)
ptx:	permuted index.	ptx(1)
limit: alter	per-process resource limitations.	csh(1)
	perror, gerror, icrmo: get system error messages.	perror(3F)
messages.	perror, sys_errlist, sys_nerr: system error	perror(3)

sticky: executable files with more, page: file	persistent text.	sticky(8)
phones: remote host	perusal filter for crt viewing.	more(1)
	phone number data base.	phones(5)
	phones: remote host phone number data base.	phones(5)
	ct: phototypesetter interface.	ct(4)
	pti: phototypesetter interpreter.	pti(1)
print them on the Dover..	phototypesetter output files to press format and	dcat(1)
	phototypesetter simulator.	tc(1)
	pi: Pascal interpreter code translator.	pi(1)
ps: Evans and Sutherland	Picture System 2 graphics device interface.	ps(4)
	ping: IP/ICMP echo user program.	ping(1)
	pipe: create an interprocess communication channel.	pipe(2)
	tee: pipe fitting.	tee(1)
	pix: Pascal interpreter and excucutor.	pix(1)
	bg: place job in background.	csh(1)
	fish: play "Go Fish".	fish(6)
	mille: play Mille Bournes.	mille(6)
	boggle: play the game of boggle.	boggle(6)
	worm: Play the growing worm game.	worm(6)
	congraph: plot connectivity of a graph.	congraph(1)
	plot: graphics filters.	plot(1G)
	plot: graphics interface.	plot(5)
move, cont, point, linemod, space, closepl:/	plot: openpl, erase, label, line, circle, arc,	plot(3X)
vtroff: troff to a raster	plotter.	vtroff(1)
	pmerge: pascal file merger.	pmerge(1)
	point faults.	trpfp(3F)
trpfp, fpcent: trap and repair floating	point, linemod, space, closepl: graphics interface.	plot(3X)
/erase, label, line, circle, arc, move, cont,	point overflow.	trapov(3F)
trapov: trap and repair floating	pointer.	lseek(2)
	lseek: move read/write	lseek(2)
dmc: DEC DMC-11/DMR-11	point-to-point communications device.	drac(4)
	poll the localnet for the current time.	timeck(8C)
	timeck:	timeck(8C)
	popd: pop shell directory stack.	csh(1)
	popd: pop shell directory stack.	csh(1)
	popen, pclose: initiate I/O to/from a process.	popen(3)
	port.	tlynam(3F)
ttynam, isatty: find name of a terminal	port.	ttytype(5)
ttytype: data base of terminal types by	port: a data structure used in the Pup package,	pupport(9)
with support routines.	Port address to a name.	maddtoname(9)
maddtoname: translate a Pup	post office.	prmail(1)
prmail: print out mail in the	postmortem crash analyzer.	analyze(8)
analyze: Virtual UNIX	pow, sqrt: exponential, logarithm, power, square	exp(3M)
root. exp, log, log10,	power, square root.	exp(3M)
exp, log, log10, pow, sqrt: exponential, logarithm,	pr: print file.	pr(1)
	pr to the line printer.	print(1)
	pr68: print extended statistics on .b file.	pr68(1)
	preloaded.	latex(1)
latex: TeX with a macro package	prepare execution profile.	monitor(3)
monitor, monstartup, moncontrol:	preprocessor to provide extended syntax for	buildmake(1)
makefiles. buildmake:	press files to lmPress format and print them on the	pressimp(1)
lmPrint printer.. pressimp: convert	Press fonts are available in fonts.widths.	dumpfonts(1)
dumpfonts: show what	press format..	dvip(1)
dvip, dvid: convert a dvi (TeX output) file to	press format and print them on the Dover..	cz(1)
cz (zarina): convert files to	press format and print them on the Dover..	dcat(1)
dcat: convert troff phototypesetter output files to	press format and print them on the Dover..	dvipress(1)
dvipress: convert dvi (TeX output) files to	pressimp: convert press files to lmPress format and	pressimp(1)
print them on the lmPrint printer..	pretty printer.	yapp(1)
yapp: yet another	previewing.	colert(1)
colert: filter nroff output for CRT	primitive system data types.	types(5)
	types:	types(5)
	cat: catenate and	cat(1)
	lpr: off line	lpr(1)
	fortune: print a random, hopefully interesting, adage.	fortune(6)
	fsckblks: print alternate super block numbers for fsck -b.	fsckblks(8)
contents of an Emacs data base .br d b p r i n t -	print an entry from a ./br d b l i s t - list	dbadd(1)
	date: print and set the date.	date(1)
	cal: print calendar.	cal(1)
	hashstat: print command hashing statistics.	csh(1)
	jobs: print current job list.	csh(1)
	dtree: print directory tree structures.	dtree(1)
	whoami: print effective current user id.	whoami(1)
	enstat: print enet (packet filter) information.	enstat(8)
	pr68: print extended statistics on .b file.	pr68(1)
	pr: print file.	pr(1)
	fpr: print Fortran file.	fpr(1)
include: search for and	print header (include) files.	include(1)
history:	print history event list.	csh(1)
hostid: set or	print identifier of current host system.	hostid(1)

host. iphostid: set or	print Internet Protocol (IP) identifier of current	iphostid(1)
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)
catboise: convert C/A/T files to DVI format and	print on Boise.	catboise(1)
vfontinfo: inspect and	print out information about UNIX fonts.	vfontinfo(1)
pmail:	print out mail in the post office.	pmail(1)
printenv:	print out the environment.	printenv(1)
man: find manual information by keywords;	print out the manual.	man(1)
	print: pr to the line printer.	print(1)
puproute:	print Pup network routing table information.	puproute(1)
68000. r168:	print relocation commands in a .b file for the	r168(1)
pstat:	print system facts.	pstat(8)
pagesize:	print system page size.	pagesize(1)
imprint -	print text files on Imprint-10.	imprint(1)
netdirprint:	print text version of Pup Network Directory.	netdirprint(8)
cz (czarina): convert files to press format and	print them on the Dover..	cz(1)
phototypesetter output files to press format and	print them on the Dover.. dcat: convert troff	dcat(1)
convert dvi (TEX output) files to press format and	print them on the Dover.. dvipress:	dvipress(1)
pressimp: convert press files to ImPress format and	print them on the ImPrint printer..	pressimp(1)
diction,explain:	print wordy sentences; thesaurus for diction.	diction(1)
explain, diction -	print wordy sentences; thesaurus for diction.	explain(1)
file. strings: find the	printable strings in a object, or other binary,	strings(1)
	printcap: printer capability data base.	printcap(5)
vlp: Format Lisp programs to be	printed with nroff, vtroff, or troff.	vlp(1)
	printenv: print out the environment.	printenv(1)
banner: print large banner on	printer.	banner(6)
btroff: troff to the ImPrint	printer.	btroff(1)
itroff: troff to the ImPrint	printer.	itroff(1)
lp: line	printer.	lp(4)
to ImPress format and print them on the ImPrint	printer.. pressimp: convert press files	pressimp(1)
print: pr to the line	printer.	print(1)
yapp: yet another pretty	printer.	yapp(1)
printcap:	printer capability data base.	printcap(5)
lpc: line	printer control program.	lpc(8)
lpd: line	printer daemon.	lpd(8)
dpq: prints the Dover	printer queue.	dpq(1)
dprm: remove a file from the Dover	printer queue.	dprm(1)
dpr: dover	printer spooler.	dpr(1)
lprm: remove jobs from the line	printer spooling queue.	lprm(1)
boise: send files to the IIP2680a	printer using TCP.	boise(1)
dviboise: send DVI files to the IIP2680a	printer using TCP.	dviboise(1)
pac:	printer/ploter accounting information.	pac(8)
vpr, vprm, vpq, vprint: raster	printer/plotter spooler.	vpr(1)
Pup. (pupprint) PupPrint,	Print!ErrorPUP, PupTypeName: printing routines for	pupprint(9)
conversion.	printf, sprintf, sprintf: formatted output	printf(3S)
(pupprint) PupPrint, Print!ErrorPUP, PupTypeName:	printing routines for Pup.	pupprint(9)
size68:	prints sizes of segments in a .b or .68 file.	size68(1)
dpq:	prints the Dover printer queue.	dpq(1)
setpriority: get/set program scheduling	priority. getpriority,	getpriority(2)
nice: set program	priority.	nice(3C)
nice, nohup: run a command at low	priority (sh only).	nice(1)
renice: alter	priority of running processes.	renice(8)
nice: run low	priority process.	cs(1)
	pmail: print out mail in the post office.	pmail(1)
adduser:	procedure for adding new users.	adduser(8)
reboot: UNIX bootstrapping	procedures.	reboot(8)
nice: run low priority	process.	cs(1)
stop: halt a job or	process.	cs(1)
_exit: terminate a	process.	exit(2)
fork: create a new	process.	fork(2)
fork: create a copy of this	process.	fork(3F)
implogd: IMP logger	process.	implogd(8C)
kill: send signal to a	process.	kill(2)
kill: send a signal to a	process.	kill(3F)
popen, pclose: initiate I/O to/from a	process.	popen(3)
wait: await completion of	process.	wait(1)
exit: terminate a	process after flushing any pending output.	exit(3)
init:	process control initialization.	init(8)

getpgrp: get	process group.	getpgrp(2)
killpg: send signal to a	process group.	killpg(2)
setpgrp: set	process group.	setpgrp(2)
getpid: get	process id.	getpid(3F)
getpid, getppid: get	process identification.	getpid(2)
vfork: spawn new	process in a virtual memory efficient way.	vfork(2)
onintr:	process interrupts in command scripts.	csh(1)
ps:	process status.	ps(1)
times: get	process times.	times(3C)
wait, wait3: wait for	process to terminate.	wait(2)
wait: wait for a	process to terminate.	wait(3F)
ptrace:	process trace.	ptrace(2)
kill: terminate a	process with extreme prejudice.	kill(1)
exit: terminate	process with status.	exit(3F)
uurec: receive	processed news articles via mail.	uurec(8)
kill: kill jobs and	processes.	csh(1)
gcore: get core images of running	processes.	gcore(1)
renice: alter priority of running	processes.	renice(8)
display and update information about the top cpu	processes. top:	top(1)
wait: wait for background	processes to complete.	csh(1)
awk: pattern scanning and	processing language.	awk(1)
halt: stop the	processor.	halt(8)
m4: macro	processor.	m4(1)
reboot: reboot system or halt	processor.	reboot(2)
shar:	produce shell-script archives.	shar(1)
	prof: display profile data.	prof(1)
	profil: execution time profile.	profil(2)
monitor, monstartup, moncontrol: prepare execution	profile.	monitor(3)
profil: execution time	profile.	profil(2)
kgmon: generate a dump of the operating system's	profile buffers.	kgmon(8)
gprof: display call graph	profile data.	gprof(1)
prof: display	profile data.	prof(1)
pxp: Pascal execution	profiler.	pxp(1)
addrfmt: IP/ICMP Address Format Request user	program.	addrfmt(8)
bboard: bulletin board reading	program.	bboard(1)
cref: cross reference	program.	cref(1)
drtest: standalone disk test	program.	drtest(8)
end, etext, edata: last locations in	program.	end(3)
finger: user information lookup	program.	finger(1)
ftp: file transfer	program.	ftp(1C)
lisz: compile a Franz Lisp	program.	lisz(1)
lpc: line printer control	program.	lpc(8)
lpq: spool queue examination	program.	lpq(1)
lxref: lisp cross reference	program.	lxref(1)
msgs: system messages and junk mail	program.	msgs(1)
mt: magnetic tape manipulating	program.	mt(1)
ping: IP/ICMP echo user	program.	ping(1)
pupftp: Pup File Transfer	Program.	pupftp(1)
gateway: a Pup gateway	program.	pupgateway(8)
pxref: Pascal cross-reference	program.	pxref(1)
rdist: remote file distribution	program.	rdist(1)
tftp: trivial file transfer	program.	tftp(1C)
units: conversion	program.	units(1)
whereis: locate source, binary, and or manual for	program.	whereis(1)
cb: C	program beautifier.	cb(1)
only). which: locate a	program file including aliases and paths (csh	which(1)
verch: version changing	program for Pascal sources.	verch(1)
make: maintain	program groups.	make(1)
nice: set	program priority.	nice(3C)
getpriority, setpriority: get/set	program scheduling priority.	getpriority(2)
indent: indent and format C	program source.	indent(1)
assert:	program verification.	assert(3X)
lint: a C	program verifier.	lint(1)
cfstprec: receive-only PUP/EFIP file transfer	program with routing.	cfstprec(1)
cfstpsend: send-only PUP/EFIP file transfer	program with routing.	cfstpsend(1)
dataio: load the data i/o prom	programmer.	dataio(1)
fp: Functional	Programming language compiler/interpreter.	fp(1)
lex: generator of lexical analysis	programs.	lex(1)
rtar, rdd, rmt: remote tape manipulation	programs.	rtar(1)
struct: structure Fortran	programs.	struct(1)
vgrind: grind nice listings of	programs.	vgrind(1)
troll. vlp: Format Lisp	programs to be printed with nroff, vtroff, or	vlp(1)
xstr: extract strings from C	programs to implement shared strings.	xstr(1)
dataio: load the data i/o	prom programmer.	dataio(1)
vv: Proteon	proNET 10 Megabit ring.	vv(4)
vv:	Proteon proNET 10 Megabit ring.	vv(4)

arp: Address Resolution Protocol.	arp(4P)
ip: Internet Protocol.	ip(4P)
tcp: Internet Transmission Control Protocol.	tcp(4P)
telnet: user interface to the TELNET protocol.	telnet(1C)
udp: Internet User Datagram Protocol.	udp(4P)
(enarp) enl0mbpuparp, ennoarp: Address Resolution Protocol (ARP) routines.	enarp(9)
getprotobyname, setprotoent, endprotoent: get protocol entry.	getprotoent(3n)
inet: Internet protocol family.	inet(4F)
pup: Xerox PUP-I protocol family.	pup(4F)
iphostid: set or print Internet Protocol (IP) identifier of current host.	iphostid(1)
rmt: remote magtape protocol module.	rmt(8C)
protocols: protocol name data base.	protocols(5)
ipbroadcast: broadcasting Internet Protocol packets.	ipbroadcast(4P)
ftpd: DARPA Internet File Transfer Protocol server.	ftpd(8C)
inetd: DARPA little protocol server.	inetd(8C)
leaf: PUP Leaf Remote File Access Protocol Server.	leaf(8)
telnetd: DARPA TELNET protocol server.	telnetd(8C)
ftspd: DARPA Trivial File Transfer Protocol server.	ftspd(8C)
ftpsr: PUP File Transfer Protocol Service.	ftpsr(8)
telser: PUP Telnet Protocol Service.	telser(8)
trpt: transliterate protocol trace.	trpt(8C)
pupecho, echoserve: PUP Echo protocol user and server.	pupecho(1)
protocols: protocol name data base.	protocols(5)
mkproto: construct a prototype file system.	mkproto(8)
arithmic: provide drill in number facts.	arithmic(6)
buildmake: preprocessor to provide extended syntax for makefiles.	buildmake(1)
false, true: provide truth values.	false(1)
true, false: provide truth values.	true(1)
device interface.	ps(4)
	ps(1)
pty:	pty(4)
	psignal(3)
	pstat(8)
doctor: interact with a psychoanalyst.	doctor(6)
	pti(1)
	ptrace(2)
	ptx(1)
	pty(4)
	miscserver(8)
	mkissofdeath(9)
	pupprint(9)
	mlookup(9)
	pupchan(9)
	pupclose(9)
	pupgethost(9)
	pupgetport(9)
	pupopen(9)
	pupread(9)
	pupreopen(9)
	pupsetbacklog(9)
	pupsetdfilt(9)
	pupsetfilter(9)
	pupsetmode(9)
	pupsettimeout(9)
	pupwrite(9)
	puplisten(9)
	pupnettab(9)
	pupecho(1)
	cftp(9)
	ftpsr(8)
	pupftsp(1)
	pupgateway(8)
	gatewayinfo(8)
	pupl0arpsr(8)
	mailcheck(1)
	mmailcheck(9)
	msendumsg(9)
	engethost(9)
	puproute(9)
	puprouting(9)
	leaf(8)
	overview(9)
	pup-mailer(8)
	pupclnet(1)
	buildnetdir(8)
	netdirprint(8)

address. mattributes: get	Pup Network Directory entry attributes for an	mattributes(9)
locate: location and owner of	Pup network hosts.	locate(1)
puproute: print	Pup network routing table information.	puproute(1)
timecheck: checks and sets	Pup network time.	timecheck(1)
byteorder: discussion of byte-ordering and the	Pup package.	byteorder(9)
PupErrMsg: human-readable error message from	Pup package routines.	puperrmsg(9)
port: a data structure used in the	Pup package, with support routines.	puppori(9)
pupread: read a	pup packet from a pup channel.	pupread(9)
puppoint: enable or disable interrupts for received	Pup Packets. pupint,	pupint(9)
maddoname: translate a	Pup Port address to a name.	maddoname(9)
	pup: raw PUP socket interface.	pup(4P)
makeshort, getHiWord, getLoWord: miscellaneous	Pup routines. /getlong, makelong, getshort,	pupmisc(9)
mtimecheck: get time from a	Pup server.	mtimecheck(9)
pup: raw	PUP socket interface.	pup(4P)
telser:	PUP Telnet Protocol Service.	telser(8)
	pup: Xerox PUP-I protocol family.	pup(4F)
	pup10arpsc: Pup GatewayInfo routing table server.	pup10arpsc(8)
patchroute: kludge to support Stanford	Pup-based subnet routing.	patchroute(8)
pupoutdescrip, pupfidtochan: access mapping between	pupchan: data structure describing a Pup channel.	pupchan(9)
	PupChans and fids. (pupdescrip) pupindescrip,	pupdescrip(9)
pupfidtochan: access mapping between PupChans and/	pupclose: close a pup channel.	pupclose(9)
server.	(pupdescrip) pupindescrip, pupoutdescrip,	pupdescrip(9)
eftprec: receive-only	pupcho, echoscrv: Pup Echo protocol user and	pupcho(1)
efpsend: send-only	PUP/EFTP file transfer program with routing.	eftprec(1)
package routines.	PUP/EFTP file transfer program with routing.	efpsend(1)
fids. (pupdescrip) pupindescrip, pupoutdescrip,	PupErrMsg: human-readable error message from Pup	puperrmsg(9)
pup channel. (pupgetport) pupgetsreport,	pupfidtochan: access mapping between PupChans and	pupdescrip(9)
local end of pup channel.	pupftp: Pup File Transfer Program.	pupftp(1)
channel. pupsetmode,	pupgetdstport: get address of local, remote ends of	pupgetport(9)
pup channel. pupgethost,	pupgethost, pupgetnet: get host, net numbers of	pupgethost(9)
address of local, remote ends of pup channel.	pupgetmode: set and read the mode for a Pup	pupsetmode(9)
remote ends of pup channel. (pupgetport)	pupgetnet: get host, net numbers of local end of	pupgethost(9)
pup: Xerox	(pupgetport) pupgetsreport, pupgetdstport: get	pupgetport(9)
mapping between PupChans and fids. (pupdescrip)	pupgetsreport, pupgetdstport: get address of local,	pupgetport(9)
received Pup Packets.	PUP-I protocol family.	pup(4F)
networks. puplisten,	pupindescrip, pupoutdescrip, pupfidtochan: access	pupdescrip(9)
	pupint, puppoint: enable or disable interrupts for	pupint(9)
getHiWord, getLoWord: miscellaneous Pup routines.	puplisten, puplistenall: open Pup channels on all	puplisten(9)
endpupnettab: Pup configuration table.	puplistenall: open Pup channels on all connected	puplisten(9)
Pup Packets. pupint,	pup-mailer: deliver mail over the .SM PUP network.	pup-mailer(8)
	(pupmisc) getlong, makelong, getshort, makeshort,	pupmisc(9)
PupChans and fids. (pupdescrip) pupindescrip,	(pupnettab) setpupnettab, getpupnettab,	pupnettab(9)
routines for Pup. (pupprint)	puppoint: enable or disable interrupts for received	pupint(9)
printing routines for Pup.	pupopen: open a pup channel.	pupopen(9)
channel.	pupoutdescrip, pupfidtochan: access mapping between	pupdescrip(9)
information.	PupPrint, PrintErrorPUP, PupTypeName: printing	pupprint(9)
routines.	(pupprint) PupPrint, PrintErrorPUP, PupTypeName:	pupprint(9)
channel.	pupread: read a pup packet from a pup channel.	pupread(9)
channel.	pupreopen: change the destination for a Pup	pupreopen(9)
Pup channel.	puproute: find a route for a Pup Internet packet.	puproute(9)
	puproute: print Pup network routing table	puproute(1)
GetMesastring, PutMesastring: manipulate strings.	puprouting: Pup Internet routing table maintenance	puprouting(9)
checksum: compute	pupsetbacklog: set input queue backlog for pup	pupsetbacklog(9)
computer via Pup network.	pupsetdfilt: set a default packet filter for a Pup	pupsetdfilt(9)
(pupprint) PupPrint, PrintErrorPUP,	pupsetfilter: set the packet filter for a Pup	pupsetfilter(9)
drb: DR11-B/DR11-W general	pupsetmode, pupgetmode: set and read the mode for a	pupsetmode(9)
ungetc:	pupsettimeout: set timeout for pup channel.	pupsettimeout(9)
pushd:	(pupstring) GetBCPLstring, PutBCPLstring,	pupstring(9)
puts, fputs:	Pup-style checksum.	checksum(9)
putc, putchar, fputc, putw:	puptelnet: connect your terminal to a remote	puptelnet(1)
manipulate strings. (pupstring) GetBCPLstring,	PupTypeName: printing routines for Pup.	pupprint(9)
unit.	pupwrite: write a packet to a pup channel.	pupwrite(9)
on a stream.	purpose user device interface.	drb(4)
stream. putc,	push character back into input stream.	ungetc(3S)
GetBCPLstring, PutBCPLstring, GetMesastring,	pushd: push shell directory stack.	csh(1)
putc, putchar, fputc,	put a string on a stream.	csh(1)
putw:	put character or word on a stream.	puts(3S)
unit.	PutBCPLstring, GetMesastring, PutMesastring:	putc(3S)
on a stream.	putc, fputc: write a character to a fortran logical	pupstring(9)
stream. putc,	putc, putchar, fputc, putw: put character or word	putc(3F)
GetBCPLstring, PutBCPLstring, GetMesastring,	putc, putchar, fputc, putw: put character or word on a	putc(3S)
putc, putchar, fputc,	PutMesastring: manipulate strings. (pupstring)	putc(3S)
putw:	puts, fputs: put a string on a stream.	pupstring(9)
unit.	putw: put character or word on a stream.	puts(3S)
on a stream.	pwd: working directory name.	putc(3S)
stream. putc,		pwd(1)

	px: Pascal interpreter.	px(1)
	pxp: Pascal execution profiler.	pxp(1)
	pxref: Pascal cross-reference program.	pxref(1)
	qsort: quick sort.	qsort(3F)
	qsort: quicker sort.	qsort(3)
	queue.	dpq(1)
dpm: remove a file from the Dover printer	queue.	dprm(1)
insque, remque: insert/remove element from a	queue.	insque(3)
lprm: remove jobs from the line printer spooling	queue.	lprm(1)
enstbacklog: set ethernet input	queue backlog.	enstbacklog(9)
pupsetbacklog: set input	queue backlog for pup channel.	pupsetbacklog(9)
lpq: spool	queue examination program.	lpq(1)
qsort:	quick sort.	qsort(3F)
qsort:	quicker sort.	qsort(3)
	quiz: test your knowledge.	quiz(6)
	quot: summarize file system ownership.	quot(8)
quotacheck: file system	quota consistency checker.	quotacheck(8)
	quota: display disc usage and limits.	quota(1)
	quota: manipulate disk quotas.	quota(2)
	quotacheck: file system quota consistency checker.	quotacheck(8)
quotaon,	quotaoff: turn file system quotas on and off.	quotaon(8)
off.	quotaon, quotaoff: turn file system quotas on and	quotaon(8)
edquota: edit user	quotas.	edquota(8)
quota: manipulate disk	quotas.	quota(2)
repquota: summarize	quotas for a file system.	repquota(8)
setquota: enable/disable	quotas on a file system.	setquota(2)
quotaon, quotaoff: turn file system	quotas on and off.	quotaon(8)
	rain: animated raindrops display.	rain(6)
rain: animated	raindrops display.	rain(6)
(except)	raise, raise_sys(): C exception handling.	except(3)
(except) raise,	raise_sys(): C exception handling.	except(3)
	rand, drand, irand: return random values.	rand(3F)
	rand, srand: random number generator.	rand(3C)
	random, hopefully interesting, adage.	fortune(6)
fortune: print a	random libraries.	ranlib(1)
ranlib: convert archives to	random number generator.	rand(3C)
rand, srand:	random number generator; routines for changing/	random(3)
random, srandom, initstate, setstate: better	random, srandom, initstate, setstate: better random	random(3)
number generator; routines for changing/	random values.	rand(3F)
rand, drand, irand: return	ranlib: convert archives to random libraries.	ranlib(1)
	raster plotter.	vtroff(1)
vtroff: troff to a	raster printer/plotter spooler.	vpr(1)
vpr, vprm, vpq, vprint:	rate.	ttime(1)
ttime: measure terminal output	ratfor: rational Fortran dialect.	ratfor(1)
	rational Fortran dialect.	ratfor(1)
ratfor:	raw PUP socket interface.	pup(4P)
pup:	raw socket interface.	imp(4P)
imp: IMP	re: command script for auto-reboot and daemons.	rc(8)
	remd, resvport, ruserok: routines for returning a	remd(3X)
stream to a remote command.	rep: remote file copy.	rep(1C)
	res: change RCS file attributes.	res(1)
	RCS commands.	resintro(1)
resintro - introduction to	RCS file.	resfile(5)
resfile: format of	RCS file attributes.	res(1)
res: change	RCS files. rlog:	rlog(1)
print log messages and other information about	RCS revisions.	ci(1)
ci: check in	RCS revisions.	co(1)
co: check out	RCS revisions.	resdiff(1)
resdiff: compare	RCS revisions.	resmerge(1)
resmerge: merge	resdiff: compare RCS revisions.	resdiff(1)
	resfile: format of RCS file.	resfile(5)
	resintro - introduction to RCS commands.	resintro(1)
	resmerge: merge RCS revisions.	resmerge(1)
rtar,	rdd, rmt: remote tape manipulation programs.	rtar(1)
	rdist: remote file distribution program.	rdist(1)
	rdump: file system dump across the network.	rdump(8C)
enread:	read a packet from an ethernet file.	enread(9)
getpass:	read a password.	getpass(3)
pupread:	read a pup packet from a pup channel.	pupread(9)
ansi:	read and write ANSI format magnetic tapes.	ansi(1)
source:	read commands from file.	csh(1)
read, readv:	read input.	read(2)
vnews:	read news articles.	vnews(1)
/continue, cd, eval, exec, exit, export, login,	read, readonly, set, shift, times, trap, umask,/	sh(1)
	read, readv: read input.	read(2)
pupsetmode, pupgetmode: set and	read the mode for a Pup channel.	pupsetmode(9)

readlink:	read value of a symbolic link.	readlink(2)
directory operations.	readdir, telldir, seekdir, rewinddir, closedir:	directory(3)
open:	open a file for	open(2)
bboard:	bulletin board	bboard(1)
newsrsc:	information file for	readlink(2)
command/ /cd, eval, excc, exit, export, login, read,	readnews(1) and checknews(1).	newsrsc(5)
read,	readonly, set, shift, times, trap, umask, wait:	sh(1)
bad144:	readv: read input.	read(2)
lseek:	read/write dec standard 144 bad sector information.	bad144(8)
setregid:	read/write pointer.	lseek(2)
setreuid:	real and effective group ID.	setregid(2)
malloc, free,	real and effective user ID's.	setreuid(2)
symorder:	realloc, calloc, alloca: memory allocator.	malloc(3)
	rearrange name list.	symorder(1)
reboot:	reboot: reboot system or halt processor.	reboot(2)
	reboot system or halt processor.	reboot(2)
fastboot, fasthalt:	reboot: UNIX bootstrapping procedures.	reboot(8)
newaliases:	reboot/halt the system without checking the disks.	fastboot(8)
recv, recvfrom, recvmsg:	rebuild the data base for the mail aliases file.	newaliases(1)
mail: send and	receive a message from a socket.	recv(2)
binmail: send or	receive mail.	mail(1)
uurec:	receive mail among users.	binmail(1)
pupint, pupoint:	receive processed news articles via mail.	uurec(8)
enable or disable interrupts for	receive unprocessed articles via mail.	recnews(8)
mail: handle remote mail	received Pup Packets.	pupint(9)
routing. cftprec:	received via uucp.	rmail(1)
	receive-only PUP/EFTP file transfer program with	cftprec(1)
	recnews: receive unprocessed articles via mail.	recnews(8)
	re_comp, re_exec: regular expression handler.	regex(3)
	recompute command hash table.	csh(1)
rehash:	records.	utmp(5)
utmp, wtmp: login	recovery.	eyacc(1)
eyacc: modified yacc allowing much improved error	recv, recvfrom, recvmsg: receive a message from a	recv(2)
socket.	recvfrom, recvmsg: receive a message from a socket.	recv(2)
recv,	recv, recvfrom, recvmsg: receive a message from a socket.	recv(2)
recv, recvfrom,	eval:	csh(1)
eval:	re-evaluate shell data.	csh(1)
re_comp,	re_exec: regular expression handler.	regex(3)
documents.	refer: find and insert literature references in	refer(1)
exref: cross	reference C source files.	exref(1)
cref: cross	reference program.	cref(1)
lxref: lisp cross	reference program.	lxref(1)
build inverted index for a bibliography, find	references in a bibliography. indxib, lookbib:	lookbib(1)
refer: find and insert literature	references in documents.	refer(1)
re_comp, re_exec:	regular expression handler.	regex(3)
	rehash: recompute command hash table.	csh(1)
comm: select or	reject lines common to two sorted files.	comm(1)
lorder68: find ordering	relation for an MC68000 object library.	lorder68(1)
lorder: find ordering	relation for an object library.	lorder(1)
join:	relational database operator.	join(1)
sigpause: atonically	release blocked signals and wait for interrupt.	sigpause(2)
strip: remove symbols and	relocation bits.	strip(1)
r168: print	relocation commands in a .b file for the 68000.	r168(1)
leave:	remind you when you have to leave.	leave(1)
calendar:	reminder service.	calendar(1)
ruserok: routines for returning a stream to a	remote command. rcmd, rresvport,	rcmd(3X)
rexec: return stream to a	remote command.	rexec(3X)
remote:	Remote command execution.	remote(1)
puptelnet: connect your terminal to a	remote computer via Pup network.	puptelnet(1)
pupgetsrport, pupgetdstport: get address of local,	remote ends of pup channel. (pupgetport)	pupgetport(9)
rexecd:	remote execution server.	rexecd(8C)
leaf: PUP Leaf	Remote File Access Protocol Server.	leaf(8)
rep:	remote file copy.	rep(1C)
rdist:	remote file distribution program.	rdist(1)
uusend: send a file to a	remote host.	uusend(1C)
remote:	remote host description file.	remote(5)
phones:	remote host phone number data base.	phones(5)
rlogin:	remote login.	rlogin(1C)
rlogind:	remote login server.	rlogind(8C)
mnt:	remote magtape protocol module.	mnt(8C)
mail: handle	remote mail received via uucp.	mail(1)
	remote: Remote command execution.	remote(1)
	remote: remote host description file.	remote(5)
rsh:	remote shell.	rsh(1C)
rshd:	remote shell server.	rshd(8C)
tip, cu: connect to a	remote system.	tip(1C)
rtar, rdd, rmt:	remote tape manipulation programs.	rtar(1)

netalias: keeping track of	remote user names and passwords.	netalias(1)
unlink:	remove a directory entry.	unlink(3F)
rmdir:	remove a directory file.	rmdir(2)
dpm:	remove a file from the Dover printer queue.	dpm(1)
unalias:	remove aliases.	csh(1)
flock: apply or	remove an advisory lock on an open file.	flock(2)
colrm:	remove columns from a file.	colrm(1)
unlink:	remove directory entry.	unlink(2)
unsetenv:	remove environment variables.	csh(1)
mount, umount: mount or	remove file system.	mount(2)
unifdef:	remove ifdefed lines.	unifdef(1)
lprm:	remove jobs from the line printer spooling queue.	lprm(1)
unpnt:	remove lines beginning with % from a file.	unpnt(1)
deroff:	remove nroff, troff, tbl and eqn constructs.	deroff(1)
expire:	remove outdated news articles.	expire(8)
unlimit:	remove resource limitations.	csh(1)
unsubscribe:	remove Scribe constructs.	unsubscribe(1)
strip:	remove symbols and relocation bits.	strip(1)
detex:	remove TeX constructs.	detex(1)
rmdir, rm:	remove (unlink) directories or files.	rmdir(1)
rm, rmdir:	remove (unlink) files or directories.	rm(1)
insque:	remque: insert/remove element from a queue.	insque(3)
rename:	rename a file.	rename(3F)
mv: move or	rename: change the name of a file.	rename(2)
mv: move or	rename files.	mv(1)
mv: move or	rename: rename a file.	renamc(3F)
mv: move or	renice: alter priority of running processes.	renice(8)
mv: move or	repair.	fsck(8)
mv: move or	repair floating point faults.	trpfpc(3F)
mv: move or	repair floating point overflow.	trapov(3F)
mv: move or	repeat commands conditionally.	csh(1)
mv: move or	repeat: execute command repeatedly.	csh(1)
mv: move or	repeated lines in a file.	uniq(1)
mv: move or	repeatedly.	csh(1)
mv: move or	repeatedly display output of command on terminal	screen(1)
mv: move or	repetitively affirmative.	yes(1)
mv: move or	report.	gripe(1)
mv: move or	report I/O statistics.	iostat(1)
mv: move or	report name of terminal.	whereami(1)
mv: move or	report repeated lines in a file.	uniq(1)
mv: move or	report to 4bsd-bugs.	sendbug(1)
mv: move or	report virtual memory statistics.	vmstat(1)
mv: move or	reports in folders automatically.	bugfiler(8)
mv: move or	reposition a file on a logical unit.	fsck(3F)
mv: move or	reposition a stream.	fsck(3S)
mv: move or	repquota: summarize quotas for a file system.	repquota(8)
mv: move or	request.	mcsrvreq(9)
mv: move or	request a boot-load.	mbootrequest(9)
mv: move or	request a boot-load.	msunbootreq(9)
mv: move or	notify: request immediate notification.	csh(1)
mv: move or	Request user program.	addrfmt(8)
mv: move or	lock: reserve a terminal.	lock(1)
mv: move or	reset: reset the teletype bits to a sensible state.	reset(1)
mv: move or	reset: reset the teletype bits to a sensible state.	reset(1)
mv: move or	arp: address resolution display and control.	arp(8C)
mv: move or	arp: Address Resolution Protocol.	arp(4P)
mv: move or	(enarp) enl0mbpuparp, ennoarp: Address Resolution Protocol (ARP) routines.	enarp(9)
mv: move or	getrlimit, setrlimit: control maximum system resource consumption.	getrlimit(2)
mv: move or	vlimit: control maximum system resource consumption.	vlimit(3C)
mv: move or	limit: alter per-process resource limitations.	csh(1)
mv: move or	unlimit: remove resource limitations.	csh(1)
mv: move or	getrusage: get information about resource utilization.	getrusage(2)
mv: move or	vtimes: get information about resource utilization.	vtimes(3C)
mv: move or	restore: incremental file system restore.	restore(8)
mv: move or	rrestore: restore a file system dump across the network.	rrestore(8C)
mv: move or	restore: incremental file system restore.	restore(8)
mv: move or	suspend: suspend a shell, resuming its superior.	csh(1)
mv: move or	getarg, iarg: return command line arguments.	getarg(3F)
mv: move or	fdate: return date and time in an ASCII string.	fdate(3F)
mv: move or	idate, itime: return date or time in numerical form.	idate(3F)
mv: move or	etime, dtime: return elapsed execution time.	etime(3F)
mv: move or	fmin, flmax, flrac, dflmin, dflmax, dflrac, inmax: return extreme values.	fmin(3F)
mv: move or	rand, drand, irand: return random values.	rand(3F)
mv: move or	rexec: return stream to a remote command.	rexec(3X)
mv: move or	time, ctime, ltime, gmtime: return system time.	time(3F)
mv: move or	loc: return the address of an object.	loc(3F)

remd, rresvport, ruserok: routines for returning a stream to a remote command.	remd(3X)
(b.out) files.	rev(1)
files. rev68: reverse byte order in 68000 .b and .68	rev68(1)
col: filter	rev68(1)
rev: reverse line feeds.	col(1)
reverse lines of a file.	rev(1)
reverse order.	lastcomm(1)
revisions.	ci(1)
revisions.	co(1)
revisions.	resdiff(1)
revisions.	rcsmerge(1)
rewind: reposition a stream.	fseek(3S)
rewinddir, closedir: directory operations.	directory(3)
rexec: return stream to a remote command.	rexcc(3X)
rexecd: remote execution server.	rexecd(8C)
rindex, lnblink, len: tell about character objects.	index(3F)
rindex: string operations. strcat, strncat,	string(3)
ring.	vv(4)
ring buffer package.	ringbuf(9)
ringbuf: ring buffer package.	ringbuf(9)
RK07 moving head disk.	hk(4)
RK6-11/RK06 and RK07 moving head disk.	hk(4)
r68: print relocation commands in a .b file for	r68(1)
rlog: print log messages and other information	rlog(1)
rlogin: remote login.	rlogin(1C)
rlogind: remote login server.	rlogind(8C)
rm: remove (unlink) directories or files.	rmdir(1)
rm, rmdir: remove (unlink) files or directories.	rm(1)
rmail: handle remote mail received via uucp.	rmail(1)
rmdir: remove a directory file.	rmdir(2)
rmdir: remove (unlink) files or directories.	rm(1)
rmdir, rm: remove (unlink) directories or files.	rmdir(1)
rmt: remote magtape protocol module.	rmt(8C)
rmt: remote tape manipulation programs.	rtar(1)
roffbib: run off bibliographic database.	roffbib(1)
rogue: Exploring The Dungeons of Doom.	rogue(6)
ROOMS.	750rom(8)
root. exp, log, log10,	exp(3M)
root directory.	chroot(2)
route for a Pup Internet packet.	puproute(9)
route: manually manipulate the routing tables.	route(8C)
routed: network routing daemon.	routed(8C)
routines. (enarp) en10mbpuparp,	enarp(9)
routines. /inet_ntoa, inet_makeaddr, inet_lnaof,	inet(3n)
routines.	overview(9)
routines. PupErrMsg:	supermsg(9)
routines. (pupmisc) getlong, makelong, getshort,	pupmisc(9)
routines. port: a data	pupport(9)
routines.	puprouting(9)
routines. tgetent, tgetnum, tgetflag, tgetstr,	termcap(3X)
routines for changing generators. /initstate,	random(3)
routines for Pup. (pupprint)	pupprint(9)
routines for returning a stream to a remote	remd(3X)
routing. elfprec:	elfprec(1)
routing. elfpsend:	elfpsend(1)
routing. patchroute:	patchroute(8)
routing daemon.	routed(8C)
routing table information.	puproute(1)
routing table maintenance routines.	puprouting(9)
routing table server.	gatewayinfo(8)
routing table server.	pup10arpser(8)
routing tables.	route(8C)
rrestore: restore a file system dump across the	rrestore(8C)
rresvport, ruserok: routines for returning a stream	remd(3X)
rsh: remote shell.	rsh(1C)
rshd: remote shell server.	rshd(8C)
rshift, lshift bitwise functions.	bit(3F)
rtar, rdd, rmt: remote tape manipulation programs.	rtar(1)
run a command at low priority (sh only).	nicc(1)
run command immune to hangups.	csh(1)
run low priority process.	csh(1)
run off bibliographic database.	roffbib(1)
running processes.	gcore(1)
running processes.	renice(8)
ruptime: show host status of local machines.	ruptime(1C)
ruserok: routines for returning a stream to a	remd(3X)
lastcomm: show last commands executed in	
ci: check in RCS	
co: check out RCS	
resdiff: compare RCS	
rcsmerge: merge RCS	
fseek, fiell,	
opendir, readdir, telldir, seekdir,	
index,	
strcmp, strncmp, strepy, strncpy, strlen, index,	
vv: Proteon proNET 10 Megabit	
ringbuf:	
hk: RK6-11/RK06 and	
hk:	
the 68000,	
about RCS files.	
rmdir,	
rm,	
rtar, rdd,	
750rom: details of Vax-11/750 boot	
pow, sqrt: exponential, logarithm, power, square	
chroot: change	
puproute: find a	
ennoarp: Address Resolution Protocol (ARP)	
inet_netof: Internet address manipulation	
overview; overview of Pup library	
human-readable error message from Pup package	
makeshort, getLiWord, getLoWord: miscellaneous Pup	
structure used in the Pup package, with support	
puprouting: Pup Internet routing table maintenance	
tgoto, tputs: terminal independent operation	
setstate: better random number generator;	
PupPrint, Print!errorPUP, PupTypeName: printing	
command. remd, rresvport, ruserok:	
receive-only PUP/EFTP file transfer program with	
send-only PUP/EFTP file transfer program with	
kludge to support Stanford Pup-based subnet	
routed: network	
puproute: print Pup network	
puprouting: Pup Internet	
gatewayinfo: Pup GatewayInfo	
pup10arpser: Pup GatewayInfo	
route: manually manipulate the	
network.	
to a remote command. remd,	
bit: and, or, xor, not,	
nice, nohup:	
nohup:	
nicc:	
roffbib:	
gcore: get core images of	
renice: alter priority of	
remote command. remd, rresvport,	

rexecd: remote execution	server.	rexecd(8C)
rlogind: remote login	server.	rlogind(8C)
rshd: remote shell	server.	rshd(8C)
rwhod: system status	server.	rwhod(8C)
telnetd: DARPA TELNET protocol	server.	telnetd(8C)
tftpd: DARPA Trivial File Transfer Protocol	server.	tftpd(8C)
breathlife: breath-of-life	server for bootloading 3mb Altos.	breathlife(8)
miscserver: MiscServices	server for Pup.	miscserver(8)
	services: service name data base.	services(5)
	session.	cs(1)
logout: end	session.	script(1)
script: make typescript of terminal	set.	ascii(7)
ascii: map of ASCII character	set a default packet filter for a Pup channel.	pupsetdfilt(9)
pupsetdfilt:	sty, gty:	sty(3C)
sty, gty:	set and get terminal state (defunct).	
pupsetmode, pupgetmode:	set and read the mode for a Pup channel.	pupsetmode(9)
sigstack:	set and/or get signal stack context.	sigstack(2)
	set: change value of shell variable.	cs(1)
sigsetmask:	set current signal mask.	sigsetmask(2)
ensetbacklog:	set ethernet input queue backlog.	ensetbacklog(9)
umask:	set file creation mode mask.	umask(2)
utime:	set file times.	utime(3C)
utimes:	set file times.	utimes(2)
setgroups:	set group access list.	setgroups(2)
pupsetbacklog:	set input queue backlog for pup channel.	pupsetbacklog(9)
apply: apply a command to a	set of arguments.	apply(1)
getsockopt, setsockopt: get and	set options on sockets.	getsockopt(2)
hostid:	set or print identifier of current host system.	hostid(1)
current host. iphostid:	set or print Internet Protocol (IP) identifier of	iphosid(1)
hostname:	set or print name of current host system.	hostname(1)
setpgpr:	set process group.	setpgpr(2)
nice:	set program priority.	nice(3C)
setregid:	set real and effective group ID.	setregid(2)
setreuid:	set real and effective user ID's.	setreuid(2)
eval, exec, exit, export, login, read, readonly,	set, shift, times, trap, umask, wait: command/ /cd,	sh(1)
getty:	set terminal mode.	getty(8)
stty:	set terminal options.	stty(1)
tabs:	set terminal tabs.	tabs(1)
date: print and	set the date.	date(1)
pupsetfilter:	set the packet filter for a Pup channel.	pupsetfilter(9)
pupsettimeout:	set timeout for pup channel.	pupsettimeout(9)
setuid, seteuid, setruid, setgid, setegid, setrgid:	set user and group ID.	setuid(3)
setenv:	set variable in environment.	cs(1)
a stream.	setbuf, setbuffer, setlinebuf: assign buffering to	setbuf(3S)
stream. setbuf,	setbuffer, setlinebuf: assign buffering to a	setbuf(3S)
setuid, seteuid, setruid, setgid,	setegid, setrgid: set user and group ID.	setuid(3)
	setenv: set variable in environment.	cs(1)
user and group ID. setuid,	seteuid, setruid, setgid, setegid, setrgid: set	setuid(3)
entry. getfsent, getfsspec, getfsfile, getfstype,	setfsent, endfsent: get file system descriptor file	getfsent(3X)
setuid, seteuid, setruid,	setgid, setegid, setrgid: set user and group ID.	setuid(3)
getgrent, getgrgid, getgrnam,	setgrent, endgrent: get group file entry.	getgrent(3)
	setgroups: set group access list.	setgroups(2)
gethostent, gethostbyaddr, gethostbyname,	sethostent, endhostent: get network host entry.	gethostent(3n)
host. gethostid,	sethostid: get/set unique identifier of current	gethostid(2)
gethostname,	sethostname: get/set name of current host.	gethostname(2)
getitimer,	setitimer: get/set value of interval timer.	getitimer(2)
	setjmp, longjmp: non-local goto.	setjmp(3)
crypt.	setkey, encrypt: DES encryption.	crypt(3)
setbuf, setbuffer,	setlinebuf: assign buffering to a stream.	setbuf(3S)
getnetent, getnetbyaddr, getnetbyname,	setnetent, endnetent: get network entry.	getnetent(3n)
	setpgpr: set process group.	setpgpr(2)
getpriority,	setpriority: get/set program scheduling priority.	getpriority(2)
getprotoent, getprotobyname, getprotobynumber, getprotobynumber,	setprotoent, endprotoent: get protocol entry.	getprotoent(3n)
configuration table. (pupnettab)	setpupnettab, getpupnettab, endpupnettab: pup	pupnettab(9)
getpwent, getpwuid, getpwnam,	setpwent, endpwent: get password file entry.	getpwent(3)
	setquota: enable/disable quotas on a file system.	setquota(2)
setuid, seteuid, setruid, setgid, setegid,	setregid: set real and effective group ID.	setregid(2)
consumption. getrlimit,	setreuid: set real and effective user ID's.	setreuid(2)
group ID. setuid, seteuid,	setrgid: set user and group ID.	setuid(3)
timecheck: checks and	setrlimit: control maximum system resource	getrlimit(2)
getservent, getservbyport, getservbyname,	setruid, setgid, setegid, setrgid: set user and	setuid(3)
getsockopt,	sets Pup network time.	timecheck(1)
for changing/ random, srandom, initstate,	setservent, endservent: get service entry.	getservent(3n)
gettimeofday,	setsockopt: get and set options on sockets.	getsockopt(2)
set user and group ID.	setstate: better random number generator; routines	random(3)
	settimeofday: get/set date and time.	gettimeofday(2)
	setuid, seteuid, setruid, setgid, setegid, setrgid:	setuid(3)

Permuted Index

continue, cd, eval, exec, exit, export, login/	sh, for, case, if, while, :, . . . , break,	sh(1)
xstr: extract strings from C programs to implement	shar: produce shell-script archives.	shar(1)
chsh: change default login	shared strings.	xstr(1)
exit: leave	shell.	chsh(1)
rsh: remote	shell.	csh(1)
system: issue a	shell command.	rsh(1C)
csh: a	shell (command interpreter) with C-like syntax.	system(3)
eval: re-evaluate	shell data.	csh(1)
popd: pop	shell directory stack.	csh(1)
pushd: push	shell directory stack.	csh(1)
alias:	shell macros.	csh(1)
suspend: suspend a	shell, resuming its superior.	csh(1)
rshd: remote	shell server.	rshd(8C)
set: change value of	shell variable.	csh(1)
@: arithmetic on	shell variables.	csh(1)
unset: discard	shell variables.	csh(1)
exec: overlay	shell with specified command.	csh(1)
shar: produce	shell-script archives.	shar(1)
/exec, exit, export, login, read, readonly, set,	shift: manipulate argument list.	csh(1)
long,	shift, times, trap, umask, wait: command language.	sh(1)
netsend: send a	short: integer object conversion.	long(3F)
arptab:	short message to one or more users on the Ethernet.	netsend(1)
groups:	show contents of kernel ARP table.	arptab(1)
ruptime:	show group memberships.	groups(1)
uptime:	show host status of local machines.	ruptime(1C)
lastcomm:	show how long system has been up.	uptime(1)
ingroup:	show last commands executed in reverse order.	lastcomm(1)
netstat:	show membership in a specified group.	ingroup(1)
uusnap:	show network status.	netstat(1)
etherport:	show snapshot of the UUCP system.	uusnap(8C)
fonts.widths, dumpfonts:	show status of ethernet minor devices.	etherport(1)
construct a file. what:	show what Press fonts are available in	dumpfonts(1)
shutdown:	show what versions of object modules were used to	what(1)
connection.	shut down part of a full-duplex connection.	shutdown(2)
login:	shutdown: close down the system at a given time.	shutdown(8)
pause: stop until	shutdown: shut down part of a full-duplex	shutdown(2)
signal: change the action for a	sigblock: block signals.	sigblock(2)
alarm: schedule	sign on.	login(1)
signal: simplified software	signal.	pause(3C)
sigvec: software	signal.	signal(3F)
sigsetmask: set current	signal after specified time.	alarm(3C)
psignal, sys_siglist: system	signal: change the action for a signal.	signal(3F)
ensignal: enable or disable	signal facilities.	signal(3C)
sigstack: set and/or get	signal facilities.	sigvec(2)
kill: send	signal mask.	sigsetmask(2)
kill: send a	signal messages.	psignal(3)
killpg: send	signal on ethernet packet arrival.	ensignal(9)
sigblock: block	signal: simplified software signal facilities.	signal(3C)
sigpause: atomically release blocked	signal stack context.	sigstack(2)
wait for interrupt.	signal to a process.	kill(2)
signal:	signal to a process.	kill(3F)
tc: phototypesetter	signal to a process group.	killpg(2)
trigonometric functions.	signals.	sigblock(2)
filetime: tell minutes	signals and wait for interrupt.	sigpause(2)
null: data	sigpause: atomically release blocked signals and	sigpause(2)
brk, sbrk: change data segment	sigsetmask: set current signal mask.	sigsetmask(2)
getdtablesize: get descriptor table	sigstack: set and/or get signal stack context.	sigstack(2)
getpagesize: get system page	sigvec: software signal facilities.	sigvec(2)
pagesize: print system page	simplified software signal facilities.	signal(3C)
size:	simulator.	tc(1)
file.	sin, cos, tan, asin, acos, atan, atan2:	sin(3M)
diskpart: calculate default disk partition	since file (access, modification) time.	filetime(8)
size68: prints	sinh, cosh, tanh: hyperbolic functions.	sinh(3M)
size:	sink.	null(4)
size of an object file.	size.	brk(2)
size: size of an object file.	size.	getdtablesize(2)
size68: prints sizes of segments in a .b or .68	size.	getpagesize(2)
sizes.	size.	pagesize(1)
sizes of segments in a .b or .68 file.	size of an object file.	size(1)
sleep: suspend execution for an interval.	size: size of an object file.	size(1)
sleep: suspend execution for an interval.	size68: prints sizes of segments in a .b or .68	size68(1)
	sizes.	diskpart(8)
	sleep: suspend execution for an interval.	size68(1)
	sleep: suspend execution for an interval.	sleep(1)
		sleep(3F)

pup-mailer: deliver mail over the .	sleep: suspend execution for interval.	sleep(3)
spline: interpolate	SM PUP network.	pup-mailer(8)
	smooth curve.	spline(1G)
	snake, ssnake: display chase game.	snake(6)
uusnap: show	snapshot of the UUCP system.	uusnap(8C)
snake,	snscore: display chase game.	snake(6)
accept: accept a connection on a	socket.	accept(2)
bind: bind a name to a	socket.	bind(2)
connect: initiate a connection on a	socket.	connect(2)
listen: listen for connections on a	socket.	listen(2)
recv, recvfrom, recvmsg: receive a message from a	socket.	recv(2)
send, sendto, sendmsg: send a message from a	socket.	send(2)
	socket: create an endpoint for communication.	socket(2)
imp: IMP raw	socket interface.	imp(4P)
pup: raw PUP	socket interface.	pup(4P)
getsockname: get	socket name.	getsockname(2)
UniqueSocket: create unique	socket number.	uniqusocket(9)
	socketpair: create a pair of connected sockets.	socketpair(2)
getsockopt, setsockopt: get and set options on	sockets.	getsockopt(2)
socketpair: create a pair of connected	sockets.	socketpair(2)
	soelim: eliminate .so's from nroff input.	soelim(1)
lo:	software loopback network interface.	lo(4)
signal: simplified	software signal facilities.	signal(3C)
sigvec:	software signal facilities.	sigvec(2)
canfield, cfscores: the	solitaire card game canfield.	canfield(6)
qsort: quicker	sort.	qsort(3)
qsort: quick	sort.	qsort(3F)
tsort: topological	sort.	tsort(1)
sortbib:	sort bibliographic database.	sortbib(1)
sort:	sort or merge files.	sort(1)
	sort: sort or merge files.	sort(1)
	sortbib: sort bibliographic database.	sortbib(1)
comm: select or reject lines common to two	sorted files.	comm(1)
look: find lines in a	sorted list.	look(1)
soelim: eliminate	.so's from nroff input.	soelim(1)
soelim: eliminate .	so's from nroff input.	soelim(1)
indent: indent and format C program	source.	indent(1)
mkstr: create an error message file by massaging C	source.	mkstr(1)
whereis: locate	source, binary, and or manual for program.	whereis(1)
xref: cross reference C	source files.	xref(1)
	source: read commands from file.	csh(1)
verch: version changing program for Pascal	sources.	verch(1)
line, circle, arc, move, cont, point, linemod,	space, closepl: graphics interface. /erase, label,	plot(3X)
expand, unexpand: expand tabs to	spaces, and vice versa.	expand(1)
way, vfork:	spawn new process in a virtual memory efficient	vfork(2)
exec: overlay shell with	specified command.	csh(1)
ingroup: show membership in a	specified group.	ingroup(1)
truncate: truncate a file to a	specified length.	truncate(2)
alarm: schedule signal after	specified time.	alarm(3C)
alarm: execute a subroutine after a	specified time.	alarm(3F)
swapon:	specify additional device for paging and swapping.	swapon(8)
	spell, spellin, spellout: find spelling errors.	spell(1)
spell,	spellin, spellout: find spelling errors.	spell(1)
spell, spellin, spellout: find	spelling errors.	spell(1)
spell, spellin,	spellout: find spelling errors.	spell(1)
	spline: interpolate smooth curve.	spline(1G)
split:	split a file into pieces.	split(1)
files, fsplit:	split a multi-routine Fortran file into individual	fsplit(1)
frexp, ldexp, modf:	split into mantissa and exponent.	frexp(3)
	split: split a file into pieces.	split(1)
uuclean: uucp	spool directory clean-up.	uuclean(8C)
lpq:	spool queue examination program.	lpq(1)
dpr: dover printer	spooler.	dpr(1)
vpr, vprm, vprq, vprint: raster printer/plotter	spooler.	vpr(1)
lprm: remove jobs from the line printer	spooling queue.	lprm(1)
printf, fprintf,	sprintf: formatted output conversion.	printf(3S)
exp, log, log10, pow,	sqrt: exponential, logarithm, power, square root.	exp(3M)
log10, pow, sqrt: exponential, logarithm, power,	square root. exp, log,	exp(3M)
rand,	srand: random number generator.	rand(3C)
generator: routines for changing/ random,	random, initState, setState: better random number	random(3)
scanf, fscanf,	scanf: formatted input conversion.	scanf(3S)
	stab: symbol table types.	stab(5)
popd: pop shell directory	stack.	csh(1)
pushd: push shell directory	stack.	csh(1)
sigstack: set and/or get signal	stack context.	sigstack(2)
drtest:	standalone disk test program.	drtest(8)

bad144: read/write dec	standard 144 bad sector information.	bad144(8)
stdio:	standard buffered input/output package.	intro(3S)
htable: convert NIC	standard format host tables.	htable(8)
patchroute: kludge to support	Stanford Pup-based subnet routing.	patchroute(8)
	stat, lstat, fstat: get file status.	stat(2)
	stat, lstat, fstat: get file status.	stat(3F)
reset: reset the teletype bits to a sensible	state.	reset(1)
stty, gtty: set and get terminal	state (defunct).	stty(3C)
fsync: synchronize a file's in-core	state with that on disk.	fsync(2)
if: conditional	statement.	csh(1)
fstab:	static information about the filesystems.	fstab(5)
hashstat: print command hashing	statistics.	csh(1)
iostat: report I/O	statistics.	iostat(1)
vmstat: report virtual memory	statistics.	vmstat(1)
pr68: print extended	statistics on .b file.	pr68(1)
exit: terminate process with	status.	exit(3F)
netstat: show network	status.	netstat(1)
ps: process	status.	ps(1)
stat, lstat, fstat: get file	status.	stat(2)
stat, lstat, fstat: get file	status.	stat(3F)
fstat: filter filenames according to commands in a	status file.	fstab(8)
ferror, feof, clearerr, fileno: stream	status inquiries.	ferror(3S)
sysline: display system status on	status line of a terminal.	sysline(1)
etherport: show	status of ethernet minor devices.	etherport(1)
ruptime: show host	status of local machines.	ruptime(1C)
sysline: display system	status on status line of a terminal.	sysline(1)
rwhod: system	status server.	rwhod(8C)
	stdio: standard buffered input/output package.	intro(3S)
	sticky: executable files with persistent text.	sticky(8)
	stop: halt a job or process.	csh(1)
halt:	stop the processor.	halt(8)
pause:	stop until signal.	pause(3C)
icheck: file system	storage consistency check.	icheck(8)
up: unibus	storage module controller/drives.	up(4)
subroutines. dbminit, fetch,	store, delete, firstkey, nextkey: data base	dbm(3X)
strlen, index, rindex: string operations.	streat, strncat, stremp, strncmp, strepy, strncpy,	string(3)
rindex: string operations. streat, strncat,	stremp, strncmp, strepy, strncpy, strlen, index,	string(3)
comparison operations.	strempfold, strncmpfold: case-folded string	strempfold(3)
operations. streat, strncat, stremp, strncmp,	strepy, strncpy, strlen, index, rindex: string	string(3)
fclose, fflush: close or flush a	stream.	fclose(3S)
fopen, freopen, fdopen: open a	stream.	fopen(3S)
fseek, ftell, rewind: reposition a	stream.	fseek(3S)
getchar, fgctc, getw: get character or word from	stream. getc,	getc(3S)
gets, fgets: get a string from a	stream.	gets(3S)
putchar, fputc, putw: put character or word on a	stream. putc,	putc(3S)
puts, fputs: put a string on a	stream.	puts(3S)
setbuffer, setlinebuf: assign buffering to a	stream. setbuf,	setbuf(3S)
ungetc: push character back into input	stream.	ungetc(3S)
sed:	stream editor.	sed(1)
ferror, feof, clearerr, fileno:	stream status inquiries.	ferror(3S)
rcmd, rresvport, ruserok: routines for returning a	stream to a remote command.	rcmd(3X)
rexec: return	stream to a remote command.	rexec(3X)
fdate: return date and time in an ASCII	string.	fdate(3F)
getbanner: get system login banner	string.	getbanner(3)
strempfold, strncmpfold: case-folded	string comparison operations.	strempfold(3)
gets, fgets: get a	string from a stream.	gets(3S)
puts, fputs: put a	string on a stream.	puts(3S)
bcopy, bcmp, bzero, bts: bit and byte	string operations.	bstring(3)
strncmp, strepy, strncpy, strlen, index, rindex:	string operations. streat, strncat, stremp,	string(3)
GetMesastring, PutMesastring: manipulate	strings. (pupstring) GetBCPI.string, PutBCPI.string,	pupstring(9)
extract strings from C programs to implement shared	strings. xstr:	xstr(1)
other binary, file.	strings: find the printable strings in a object, or	strings(1)
strings. xstr: extract	strings from C programs to implement shared	xstr(1)
strings: find the printable	strings in a object, or other binary, file.	strings(1)
basename:	strip filename affixes.	basename(1)
strip: remove symbols and relocation bits.	strip.	strip(1)
strlen, index, rindex: string operations.	strcat, stremp, strncmp, strepy, strncpy, strlen,	string(3)
strncat, stremp, strncmp, strepy, strncpy, strlen,	strncmp, strepy, strncpy, strlen, index, rindex:	string(3)
string operations. streat, strncat, stremp,	strncmpfold: case-folded string comparison	strempfold(3)
operations. strempfold,	strepy, strlen, index, rindex: string operations.	string(3)
streat, strncat, stremp, strncmp, strepy,	struct: structure Fortran programs.	struct(1)
pupchan: data	structure describing a Pup channel.	pupchan(9)
struct:	structure Fortran programs.	struct(1)
routines. port: a data	structure used in the Pup package, with support	pupport(9)
dtrec: print directory tree	structures.	dtrec(1)

	stty, gtty: set and get terminal state (defunct).	stty(3C)
	stty: set terminal options.	stty(1)
document.	style: analyze surface characteristics of a	style(1)
	su: substitute user id temporarily.	su(1)
patchroute: kludge to support Stanford Pup-based	subnet routing.	patchroute(8)
alarm: execute a	subroutine after a specified time.	alarm(3F)
fetch, store, delete, firstkey, nextkey: data base	subroutines. dbminit,	dbm(3X)
lib2648:	subroutines for the HP 2648 graphics terminal.	lib2648(3X)
su:	substitute user id temporarily.	su(1)
sum:	sum and count blocks in a file.	sum(1)
	sum: sum and count blocks in a file.	sum(1)
du:	summarize disk usage.	du(1)
quot:	summarize file system ownership.	quot(8)
repquota:	summarize quotas for a file system.	repquota(8)
dix: download with error correction - 68000	Sun1 monitor.	dx(1)
sync: update the	super block.	sync(8)
update: periodically update the	super block.	update(8)
fscbblks: print alternate	super block numbers for fsck -b.	fscbblks(8)
sync: update	super-block.	sync(2)
suspend: suspend a shell, resuming its	superior.	csh(1)
intro: introduction to special files and hardware	support.	intro(4)
a data structure used in the Pup package, with	support routines. port:	puppet(9)
patchroute: kludge to	support Stanford Pup-based subnet routing.	patchroute(8)
style: analyze	surface characteristics of a document.	style(1)
suspend:	suspend a shell, resuming its superior.	csh(1)
sleep:	suspend execution for an interval.	sleep(1)
sleep:	suspend execution for an interval.	sleep(3F)
sleep:	suspend execution for interval.	sleep(3)
	suspend: suspend a shell, resuming its superior.	csh(1)
interface. ps: Evans and	Sutherland Picture System 2 graphics device	ps(4)
	swab: swap bytes.	swab(3)
swab:	swap bytes.	swab(3)
swapon: add a	swap device for interleaved paging/swapping.	swapon(2)
paging/swapping.	swapon: add a swap device for interleaved	swapon(2)
swapping.	swapon: specify additional device for paging and	swapon(8)
swapon: specify additional device for paging and	swapping.	swapon(8)
breaksw: exit from	switch.	csh(1)
case: selector in	switch.	csh(1)
default: catchall clause in	switch.	csh(1)
endsw: terminate	switch.	csh(1)
	switch: multi-way command branch.	csh(1)
stab:	symbol table types.	stab(5)
ddt68, fddt68:	symbolic debugger for 68000.	ddt68(1)
readlink: read value of a	symbolic link.	readlink(2)
symlink: make	symbolic link to a file.	symlink(2)
symlchk: check for bad	symbolic links.	symlchk(1)
strip: remove	symbols and relocation bits.	strip(1)
	symlchk: check for bad symbolic links.	symlchk(1)
	symlink: make symbolic link to a file.	symlink(2)
	symorder: rearrange name list.	symorder(1)
	sync: update super-block.	sync(2)
	sync: update the super block.	sync(8)
disk. fsync:	synchronize a file's in-core state with that on	fsync(2)
select:	synchronous i/o multiplexing.	select(2)
csh: a shell (command interpreter) with C-like	syntax.	csh(1)
buildmake: preprocessor to provide extended	syntax for makefiles.	buildmake(1)
	syscall: indirect system call.	syscall(2)
perror,	sys_errlist, sys_nerr: system error messages.	perror(3)
terminal.	sysline: display system status on status line of a	sysline(1)
	syslog: log systems messages.	syslog(8)
	syslog, openlog, closelog: control system log.	syslog(3)
perror, sys_errlist,	sys_nerr: system error messages.	perror(3)
psignal,	sys_siglist: system signal messages.	psignal(3)
gmr: Grinnell	Systems display.	gmr(4)
hy: Network	Systems Hyperchannel interface.	hy(4)
syslog: log	systems messages.	syslog(8)
kgmon: generate a dump of the operating	system's profile buffers.	kgmon(8)
arptab: show contents of kernel ARP	table.	arptab(1)
rehash: recompute command hash	table.	csh(1)
unhash: discard command hash	table.	csh(1)
mtab: mounted file system	table.	mtab(5)
getpupnettab, endpupnettab: pup configuration	table. (pupnettab) setpupnettab,	pupnettab(9)
vwidth: make troff width	table for a font.	vwidth(1)
puproute: print Pup network routing	table information.	puproute(1)
puprouting: Pup Internet routing	table maintenance routines.	puprouting(9)
gatewayinfo: Pup GatewayInfo routing	table server.	gatewayinfo(8)

pup10arpser: Pup GatewayInfo routing	table server.	pup10arpser(8)
getdtablesize: get descriptor	table size.	getdtablesize(2)
stab: symbol	table types.	stab(5)
htable: convert NIC standard format host	tables.	htable(8)
route: manually manipulate the routing	tables.	route(8C)
tbl: format	tables for nroff or troff.	tbl(1)
gettable: get NIC format host	tables from a host.	gettable(8C)
tabs: set terminal	tabs.	tabs(1)
	tabs: set terminal tabs.	tabs(1)
expand, unexpand: expand	tabs to spaces, and vice versa.	expand(1)
ctags: create a	tags file.	ctags(1)
	tail: deliver the last part of a file.	tail(1)
	talk: talk to another user.	talk(1)
talk:	talk to another user.	talk(1)
functions. sin, cos,	tan, asin, acos, atan, atan2: trigonometric	sin(3M)
tex file.	tangle, weave: convert web file into pascal file,	tangle(1)
sinh, cosh,	tanh: hyperbolic functions.	sinh(3M)
tp: manipulate	tape archive.	tp(1)
tar:	tape archive file format.	tar(5)
tar:	tape archiver.	tar(1)
ut: UNIBUS TU45 tri-density	tape drive interface.	ut(4)
tp: DEC/mag	tape formats.	tp(5)
tclose, tread, twrite, trewin, tskipf, tstate: f77	tape I/O. topen,	topen(3F)
mt: magnetic	tape manipulating program.	mt(1)
rtar, rdd, rmt: remote	tape manipulation programs.	rtar(1)
ansi: read and write ANSI format magnetic	tapes.	ansi(1)
	tar: tape archive file format.	tar(5)
	tar: tape archiver.	tar(1)
deroff: remove nroff, troff,	tbl and eqn constructs.	deroff(1)
	tbl: format tables for nroff or troff.	tbl(1)
	tc: phototypesetter simulator.	tc(1)
tape I/O. topen,	tclose, tread, twrite, trewin, tskipf, tstate: f77	topen(3F)
boise: send files to the HP2680a printer using	TCP.	boise(1)
send DVI files to the HP2680a printer using	TCP. dviboise:	dviboise(1)
	tcp: Internet Transmission Control Protocol.	tcp(4P)
	tec: pipe fitting.	tec(1)
tk: paginator for the	Tektronix 4014.	tk(1)
reset: reset the	teletype bits to a sensible state.	reset(1)
last: indicate last logins of users and	teletypes.	last(1)
index, rindex, lnbkln, len:	tell about character objects.	index(3F)
time. filetime:	tell minutes since file (access, modification)	filetime(8)
operations. opendir, readdir,	tellmdir, seekdir, rewinddir, closedir: directory	directory(3)
telnet: user interface to the	TELNET protocol.	telnet(1C)
telnetd: DARPA	TELNET protocol server.	telnetd(8C)
telser: PUP	Telnet Protocol Service.	telser(8)
	telnet: user interface to the TELNET protocol.	telnet(1C)
	telnetd: DARPA TELNET protocol server.	telnetd(8C)
	telser: PUP Telnet Protocol Service.	telser(8)
su: substitute user id	temporarily.	su(1)
	term: conventional names for terminals.	term(7)
lib2648: subroutines for the HP 2648 graphics	termcap: terminal capability data base.	termcap(5)
lock: reserve a	terminal.	lib2648(3X)
sysline: display system status on status line of a	terminal.	lock(1)
ttyname, isatty, ttyslot: find name of a	terminal.	sysline(1)
vhangup: virtually "hangup" the current control	terminal.	ttynam(3)
whereami: report name of	terminal.	vhangup(2)
worms: animate worms on a display	terminal.	whereami(1)
	terminal.	worms(6)
termcap:	terminal capability data base.	termcap(5)
gettytab:	terminal configuration data base.	gettytab(5)
tset:	terminal dependent initialization.	tset(1)
pty: pseudo	terminal driver.	pty(4)
tgetent, tgetnum, tgetflag, tgetstr, tgoto, tputs:	terminal independent operation routines.	termcap(3X)
ttys:	terminal initialization data.	ttys(5)
tty: general	terminal interface.	tty(4)
getty: set	terminal mode.	getty(8)
dmf: DMI-32,	terminal multiplexor.	dmf(4)
tty: get	terminal name.	tty(1)
stty: set	terminal options.	stty(1)
ttime: measure	terminal output rate.	ttime(1)
ttynam, isatty: find name of a	terminal port.	ttynam(3F)
clear: clear	terminal screen.	clear(1)
screen: repeatedly display output of command on	terminal screen.	screen(1)
script: make typescript of	terminal session.	script(1)
stty, gtty: set and get	terminal state (defunct).	stty(3C)
tabs: set	terminal tabs.	tabs(1)

puptelnet: connect your	terminal to a remote computer via Pup network.	puptelnet(1)
ttytype: data base of	terminal types by port.	ttytype(5)
term: conventional names for	terminals.	term(7)
wait, wait3: wait for process to	terminate.	wait(2)
wait: wait for a process to	terminate.	wait(3F)
_exit:	terminate a process.	exit(2)
output. exit:	terminate a process after flushing any pending	exit(3)
kill:	terminate a process with extreme prejudice.	kill(1)
abort:	terminate abruptly with memory image.	abort(3F)
endif:	terminate conditional.	csh(1)
end:	terminate loop.	csh(1)
exit:	terminate process with status.	exit(3F)
endsw:	terminate switch.	csh(1)
	test: condition command.	test(1)
drtest: standalone disk	test program.	drtest(8)
quiz:	test your knowledge.	quiz(6)
detex: remove	TeX constructs.	detex(1)
tangle, weave: convert web file into pascal file,	tex file.	tangle(1)
typesetting.	tex, latex, initex, virtex: text formatting and	tex(1)
dvip. dvid: convert a dvi	(TeX output) file to press format.	dvip(1)
on the Dover.. dvipress: convert dvi	(TeX output) files to press format and print them	dvipress(1)
latex:	TeX with a macro package preloaded.	latex(1)
sticky: executable files with persistent	text.	sticky(8)
ed:	text editor.	ed(1)
ex, edit:	text editor.	ex(1)
linelen: print line lengths for a	text file.	linelen(1)
imprint - print	text files on Imprint-10.	imprint(1)
iprint - convert	text files to DVI format.	iprint(1)
fmt: simple	text formatter.	fmt(1)
nroff:	text formatting.	nroff(1)
tex, latex, initex, virtex:	text formatting and typesetting.	tex(1)
troff, nroff:	text formatting and typesetting.	troff(1)
ms:	text formatting macros.	ms(7)
ncdirprint: print	text version of Pup Network Directory.	ncdirprint(8)
	tftp: trivial file transfer program.	tftp(1C)
	tftpd: DARPA Trivial File Transfer Protocol server.	tftpd(8C)
terminal independent operation routines.	tgetent, tgetnum, tgetflag, tgetstr, tgoto, tputs:	termcap(3X)
independent operation routines. tgetent, tgetnum,	tgetflag, tgetstr, tgoto, tputs: terminal	termcap(3X)
independent operation routines. tgetent,	tgetnum, tgetflag, tgetstr, tgoto, tputs: terminal	termcap(3X)
operation routines. tgetent, tgetnum, tgetflag,	tgetstr, tgoto, tputs: terminal independent	termcap(3X)
routines. tgetent, tgetnum, tgetflag, tgetstr,	tgoto, tputs: terminal independent operation	termcap(3X)
diction.explain: print wordy sentences;	thesaurus for diction.	diction(1)
explain, diction- print wordy sentences;	thesaurus for diction.	explain(1)
merge:	three-way file merge.	merge(1)
alarm: schedule signal after specified	time.	alarm(3C)
alarm: execute a subroutine after a specified	time.	alarm(3F)
at: execute commands at a later	time.	at(1)
ctime, dtime: return elapsed execution	time.	ctime(3F)
tell minutes since file (access, modification)	time. filetype:	filetime(8)
gettimeofday, settimeofday: get/set date and	time.	gettimeofday(2)
shutdown: close down the system at a given	time.	shutdown(8)
time, ftime: get date and	time.	time(3C)
time, ctime, ltime, gmtime: return system	time.	time(3F)
timecheck: checks and sets Pup network	time.	timecheck(1)
timeck: poll the localnet for the current	time.	timeck(8C)
time:	time a command.	time(1)
UAtimecvt: conversions from Unix to Alto	time and vice versa.	uatimecvt(9)
time:	time command.	csh(1)
	time, ctime, ltime, gmtime: return system time.	time(3F)
mtimecheck: get	time from a Pup server.	mtimecheck(9)
	time, ftime: get date and time.	time(3C)
fdate: return date and	time in an ASCII string.	fdate(3F)
idate, itime: return date or	time in numerical form.	idate(3F)
loadlog: log the current	time, number of users, and load average.	loadlog(1)
utime: adjust the access or modification	time of a file.	utime(8)
profil: execution	time profile.	profil(2)
	time: time a command.	time(1)
	time: time command.	csh(1)
gmtime, asctime, timezone: convert date and	time to ASCII. ctime, localtime,	ctime(3)
	timecheck: checks and sets Pup network time.	timecheck(1)
	timeck: poll the localnet for the current time.	timeck(8C)
	timeout for pup channel.	pupsettimeout(9)
pupsettimeout: set	timer.	getitimer(2)
getitimer, sctitimer: get/set value of interval	times.	times(3C)
times: get process	times.	utime(3C)
utime: set file	times.	utime(3C)
utimes: set file	times.	utimes(2)

exit, export, login, read, readonly, set, shift, ctime, localtime, gmtime, asctime,	times: get process times.	times(3C)
	times, trap, umask, wait: command language. /exec,	sh(1)
	timezone: convert date and time to ASCII.	ctime(3)
	tip, cu: connect to a remote system.	tip(1C)
	tk: paginator for the Tektronix 4014.	tk(1)
	tm: TM-11/TE-10 magtape interface.	tm(4)
ht: TM-03/TE-16,TU-45,TU-77 MASSBUS magtape interface.		ht(4)
tm: TM-11/TE-10 magtape interface.		tm(4)
mt: TM78/TU-78 MASSBUS magtape interface.		mt(4)
popen, pclose: initiate I/O	to/from a process.	popen(3)
top: display and update information about the cpu processes.	top cpu processes.	top(1)
tstate: f77 tape I/O.	top: display and update information about the top	top(1)
tsort:	topen, tclose, tread, twrite, trewin, tskipf,	topen(3F)
	topological sort.	tsort(1)
	touch: update date last modified of a file.	touch(1)
	tp: DEC/mag tape formats.	tp(5)
	tp: manipulate tape archive.	tp(1)
tgetent, tgetnum, tgetflag, tgetstr, tgoto,	tputs: terminal independent operation routines.	termcap(3X)
	tr: translate characters.	tr(1)
ptrace: process	trace.	ptrace(2)
trpt: transliterate protocol	trace.	trpt(8C)
netalias: keeping	track of remote user names and passwords.	netalias(1)
ftpser: PUP File	Transer Protocol Service.	ftpser(8)
goto: command	transfer.	csh(1)
ftp: file	transfer program.	ftp(1C)
pupftp: Pup File	Transfer Program.	pupftp(1)
tftp: trivial file	transfer program.	tftp(1C)
eftprec: receive-only PUP/EFTP file	transfer program with routing.	eftprec(1)
eftpsend: send-only PUP/EFTP file	transfer program with routing.	eftpsend(1)
ftpd: DARPA Internet File	Transfer Protocol server.	ftpd(8C)
tftpd: DARPA Trivial File	Transfer Protocol server.	tftpd(8C)
mlookup:	translate a name to a Pup address.	mlookup(9)
maddtoname:	translate a Pup Port address to a name.	maddtoname(9)
tr:	translate characters.	tr(1)
macros. tman:	translate version 6 manual macros to version 7	tman(1)
ad: Data	Translation A/D converter.	ad(4)
pi: Pascal interpreter code	translator.	pi(1)
ccom68: .c -> .s	translator component of cc68.	ccom68(1)
trpt:	transliterate protocol trace.	trpt(8C)
tcp: Internet	Transmission Control Protocol.	tcp(4P)
uuencode, uudecode: encode/decode a binary file for	transmission via mail.	uuencode(1C)
trpfp, spcent:	trap and repair floating point faults.	trpfp(3F)
trapov:	trap and repair floating point overflow.	trapov(3F)
traper:	trap arithmetic errors.	traper(3F)
export, login, read, readonly, set, shift, times,	trap, umask, wait: command language. /exec, exit,	sh(1)
	traper: trap arithmetic errors.	traper(3F)
	trapov: trap and repair floating point overflow.	trapov(3F)
I/O. topen, tclose,	tread, twrite, trewin, tskipf, tstate: f77 tape	topen(3F)
dtree: print directory	tree structures.	dtree(1)
	trek: trekkie game.	trek(6)
	trekkie game.	trek(6)
topen, tclose, tread, twrite,	trewin, tskipf, tstate: f77 tape I/O.	topen(3F)
ut: UNIBUS TU45	tri-density tape drive interface.	ut(4)
sin, cos, tan, asin, acos, atan, atan2:	trigonometric functions.	sin(3M)
tftp:	trivial file transfer program.	tftp(1C)
tftpd: DARPA	Trivial File Transfer Protocol server.	tftpd(8C)
7 macros.	tman: translate version 6 manual macros to version	tman(1)
tbl: format tables for nroff or	troff.	tbl(1)
Lisp programs to be printed with nroff, vtroff, or	troff. vlp: Format	vlp(1)
	troff, nroff: text formatting and typesetting.	troff(1)
and print them on the Dover. deat: convert	troff phototypesetter output files to press format	deat(1)
deroff: remove nroff,	troff, tbl and eqn constructs.	deroff(1)
vtroff:	troff to a raster plotter.	vtroff(1)
dtroff:	troff to the Dover.	dtroff(1)
btroff:	troff to the ImPrint printer.	btroff(1)
itroff:	troff to the ImPrint printer.	itroff(1)
vwidth: make	troff width table for a font.	vwidth(1)
faults.	trpfp, spcent: trap and repair floating point	trpfp(3F)
	trpt: transliterate protocol trace.	trpt(8C)
	true, false: provide truth values.	true(1)
false,	true: provide truth values.	false(1)
truncate:	truncate a file to a specified length.	truncate(2)
	truncate: truncate a file to a specified length.	truncate(2)
false, true: provide	truth values.	false(1)
true, false: provide	truth values.	true(1)
	ts: TS-11 magtape interface.	ts(4)

ts:	TS-11 magtape interface.	ts(4)
topen, tclose, tread, twrite, trewin,	tset: terminal dependent initialization.	tset(1)
topen, tclose, tread, twrite, trewin, tskipf,	tskipf, tstate: /77 tape I/O.	topen(3F)
	tsort: topological sort.	tsort(1)
	tstate: /77 tape I/O.	topen(3F)
	ttime: measure terminal output rate.	ttime(1)
	tty: general terminal interface.	tty(4)
	tty: get terminal name.	tty(1)
	ttynam, isatty: find name of a terminal port.	ttynam(3F)
	ttynam, isatty, ttyslot: find name of a terminal.	ttynam(3)
	ttys: terminal initialization data.	ttys(5)
ttynam, isatty,	ttyslot: find name of a terminal.	ttynam(3)
	ttysize: data base of terminal types by port.	ttysize(5)
interface,	tu: VAX-11/730 and VAX-11/750 TU58 console cassette	tu(4)
ut: UNIBUS	TU45 tri-density tape drive interface.	ui(4)
ht: TM-03/TE-16,	TU-45, TU-77 MASSBUS magtape interface.	hi(4)
tu: VAX-11/730 and VAX-11/750	TU58 console cassette interface.	tu(4)
	uu: TU58/DI:Ctape II UNIBUS cassette interface.	uu(4)
ht: TM-03/TE-16, TU-45,	TU-77 MASSBUS magtape interface.	hi(4)
tunefs:	tune up an existing file system.	tunefs(8)
	tunefs: tune up an existing file system.	tunefs(8)
topen, tclose, tread,	twrite, trewin, tskipf, tstate: /77 tape I/O.	topen(3F)
file: determine file	type.	file(1)
stab: symbol table	types.	stab(5)
types: primitive system data	types.	types(5)
ttytype: data base of terminal	types by port.	ttytype(5)
	types: primitive system data types.	types(5)
script: make	typescript of terminal session.	script(1)
man: macros to	typeset manual.	man(7)
eqn, neqn, checkeq:	typeset mathematics.	eqn(1)
tex, latex, initex, virtex: text formatting and	typesetting.	tex(1)
troff, nroff: text formatting and	typesetting.	troff(1)
vice versa.	UAtimecvt: conversions from Unix to Alto time and	uatimecvt(9)
	uda: UDA-50 disk controller interface.	uda(4)
uda:	UDA-50 disk controller interface.	uda(4)
	udp: Internet User Datagram Protocol.	udp(4P)
getpw: get name from	uid.	getpw(3C)
	ul: do underlining.	ul(1)
	umask: change or display file creation mask.	cs(1)
	umask: set file creation mode mask.	umask(2)
login, read, readonly, set, shift, times, trap,	umask, wait: command language. /exec, exit, export,	sh(1)
mount,	umount: mount and dismount file system.	mount(8)
mount,	umount: mount or remove file system.	mount(2)
	un: Ungermann-Bass interface.	un(4)
	unalias: remove aliases.	cs(1)
cat them, compact,	uncompact, ccat: compress and uncompress files, and	compact(1)
compact, uncompact, ccat: compress and	uncompress files, and cat them.	compact(1)
ul: do	underlining.	ul(1)
file.	undump: convert a core dump to an executable a.out	undump(1)
expand,	unexpand: expand tabs to spaces, and vice versa.	expand(1)
un:	Ungermann-Bass interface.	un(4)
	ungetc: push character back into input stream.	ungetc(3S)
	unhash: discard command hash table.	cs(1)
uu: TU58/DI:Ctape II	UNIBUS cassette interface.	uu(4)
up:	unibus storage module controller/drives.	up(4)
ut:	UNIBUS TU45 tri-density tape drive interface.	ul(4)
	unifdef: remove ifdef'ed lines.	unifdef(1)
	uniq: report repeated lines in a file.	uniq(1)
mktemp: make a	unique file name.	mktemp(3)
gethostid, sethostid: get/set	unique identifier of current host.	gethostid(2)
UniqueSocket: create	unique socket number.	uniquesocket(9)
	UniqueSocket: create unique socket number.	uniquesocket(9)
flush: flush output to a logical	unit.	flush(3I)
fseek, ftell: reposition a file on a logical	unit.	fseek(3I)
getc, fgetc: get a character from a logical	unit.	getc(3F)
putc, fputc: write a character to a fortran logical	unit.	putc(3F)
dn: DN-11 autocall	unit interface.	dn(4)
	units: conversion program.	units(1)
learn: computer aided instruction about	UNIX.	learn(1)
manage user login accounts (create, modify, destroy	Unix accounts), nu:	nu(8)
reboot:	UNIX bootstrapping procedures.	reboot(8)
system: execute a	UNIX command.	system(3F)
uux: unix to	unix command execution.	uux(1C)
uucp, uulog: unix to	unix copy.	uucp(1C)
vfontinfo: inspect and print out information about	UNIX fonts.	vfontinfo(1)
mtio:	UNIX magtape interface.	mtio(4)

analyze: Virtual	UNIX postmortem crash analyzer.	analyze(8)
UAtimecvt: conversions from	Unix to Alto time and vice versa.	uaticcvt(9)
uux:	unix to unix command execution.	uux(1C)
uucp, uulog:	unix to unix copy.	uucp(1C)
rmdir, rm: remove	unlimit: remove resource limitations.	cs(1)
rm, rmdir: remove	(unlink) directories or files.	rmdir(1)
	(unlink) files or directories.	rm(1)
	unlink: remove a directory entry.	unlink(3F)
	unlink: remove directory entry.	unlink(2)
	unpent: remove lines beginning with % from a file.	unpent(1)
recnews: receive	unprocessed articles via mail.	recnews(8)
	unsubscribe: remove Scribe constructs.	unsubscribe(1)
	unset: discard shell variables.	cs(1)
	unsetenv: remove environment variables.	cs(1)
uptime: show how long system has been	up.	uptime(1)
tunefs: tune	up an existing file system.	tunefs(8)
	up: unibus storage module controller/drives.	up(4)
netupd:	update a directory from one on another system.	netupd(1)
touch:	update date last modified of a file.	touch(1)
top: display and	update information about the top cpu processes.	top(1)
	update: periodically update the super block.	update(8)
sync:	update super-block.	sync(2)
sync:	update the super block.	sync(8)
update: periodically	update the super block.	update(8)
	uptime: show how long system has been up.	uptime(1)
du: summarize disk	usage.	du(1)
quota: display disc	usage and limits.	quota(1)
port: a data structure	used in the Pup package, with support routines.	pupport(9)
what: show what versions of object modules were	used to construct a file.	what(1)
miscellaneous: miscellaneous	useful information pages.	intro(7)
login: login new	user.	cs(1)
talk: talk to another	user.	talk(1)
whois: ask the ARPA Internet NIC about a	user.	whois(1C)
write: write to another	user.	write(1)
seteuid, setruid, setgid, setegid, setrgid: set	user and group ID. setuid,	setuid(3)
pupecho, echoserve: Pup Echo protocol	user and server.	pupecho(1)
udp: Internet	User Datagram Protocol.	udp(4P)
drb: DR11-B/DR11-W general purpose	user device interface.	drb(4)
environ:	user environment.	environ(7)
mailcheck: find out if a	user has mail at a PUP host.	mailcheck(1)
mmailcheck: find out if a	user has new mail at a Pup host.	mmailcheck(9)
whoami: print effective current	user id.	whoami(1)
su: substitute	user id temporarily.	su(1)
getuid, geteuid: get	user identity.	getuid(2)
setreuid: set real and effective	user ID's.	setreuid(2)
finger:	user information lookup program.	finger(1)
telnet:	user interface to the TELNET protocol.	telnet(1C)
accounts). nu: manage	user login accounts (create, modify, destroy Unix	nu(8)
netalias: keeping track of remote	user names and passwords.	netalias(1)
getuid, getgid: get	user or group ID of the caller.	getuid(3F)
addrfmt: IP/ICMP Address Format Request	user program.	addrfmt(8)
ping: IP/ICMP echo	user program.	ping(1)
edquota: edit	user quotas.	edquota(8)
insecure:	user security monitor.	insecure(8)
adduser: procedure for adding new	users.	adduser(8)
allusers: print list of all authorized	users.	allusers(1)
binmail: send or receive mail among	users.	binmail(1)
wall: write to all	users.	wall(1)
loadlog: log the current time, number of	users, and load average.	loadlog(1)
last: indicate last logins of	users and teletypes.	last(1)
msendmsg: send a message to one or all	users at a Pup host.	msendmsg(9)
	users: compact list of users who are on the system.	users(1)
getlog: get	user's login name.	getlog(3F)
netsend: send a short message to one or more	users on the Ethernet.	netsend(1)
users: compact list of	users who are on the system.	users(1)
boise: send files to the HP2680a printer	using TCP.	boise(1)
dvi Boise: send DVI files to the HP2680a printer	using TCP.	dvi Boise(1)
ut: UNIBUS TU45 tri-density tape drive interface.	utilization.	ut(4)
getrusage: get information about resource	utilization.	getrusage(2)
vtimes: get information about resource	utilization.	vtimes(3C)
file.	utime: adjust the access or modification time of a	utime(8)
	utime: set file times.	utime(3C)
	utimes: set file times.	utimes(2)
	utmp, wtmp: login records.	utmp(5)
	uu: TU58/DICtape II UNIBUS cassette interface.	uu(4)
	uuclean: uucp spool directory clean-up.	uuclean(8C)

mail: handle remote mail received via	uucp.	rmail(1)
uuclean:	uucp spool directory clean-up.	uuclean(8C)
uusnap: show snapshot of the	UUCP system.	uusnap(8C)
transmission via mail.	uucp, uulog: unix to unix copy.	uucp(1C)
uencode:	uudecode: encode/decode a binary file for	uencode(1C)
uencode: format of an encoded	uencode file.	uencode(5)
transmission via mail.	uencode: format of an encoded uencode file.	uencode(5)
uucp,	uencode,uudecode: encode/decode a binary file for	uencode(1C)
	uulog: unix to unix copy.	uucp(1C)
	uurec: receive processed news articles via mail.	uurec(8)
	uusend: send a file to a remote host.	uusend(1C)
	uusnap: show snapshot of the UUCP system.	uusnap(8C)
	uux: unix to unix command execution.	uux(1C)
	va: Benson-Varian interface.	va(4)
	valloc: aligned memory allocator.	valloc(3)
	value.	abs(3)
abs: integer absolute	value, floor, ceil functions.	floor(3M)
fabs, floor, ceil: absolute	value for environment name.	getenv(3)
getenv:	value of a symbolic link.	readlink(2)
readlink: read	value of environment variables.	getenv(3F)
getenv: get	value of interval timer.	getitimer(2)
getitimer, setitimer: get/set	value of shell variable.	csh(1)
set: change	values.	false(1)
false, true: provide truth	values. flmin, flmax, ffrac,	flmin(3F)
dfmin, dfmax, dfrac, inmax: return extreme	values.	rand(3F)
rand, drand, irand: return random	values.	truc(1)
true, false: provide truth	values between host and network byte order.	byteorder(3n)
htonl, htons, ntohl, ntohs: convert	varargs: variable argument list.	varargs(3)
set: change value of shell	variable.	csh(1)
varargs:	variable argument list.	varargs(3)
setenv: set	variable in environment.	csh(1)
@: arithmetic on shell	variables.	csh(1)
unset: discard shell	variables.	csh(1)
unsetenv: remove environment	variables.	csh(1)
getenv: get value of environment	variables.	getenv(3F)
as:	VAX-11 assembler.	as(1)
cons:	VAX-11 console interface.	cons(4)
interface. tu:	VAX-11/730 and VAX-11/750 TU58 console cassette	tu(4)
750rom: details of	Vax-11/750 boot ROMs.	750rom(8)
tu: VAX-11/730 and	VAX-11/750 TU58 console cassette interface.	tu(4)
assert: program	verch: version changing program for Pascal sources.	verch(1)
lint: a C program	verification.	assert(3X)
expand, unexpand: expand tabs to spaces, and vice	verifier.	lint(1)
conversions from Unix to Alto time and vice	versa.	expand(1)
vfont: font formats for the Benson-Varian or	versa. Uatimecvt:	uatimecvt(9)
vp:	Versatec.	vfont(5)
trman: translate	Versatec interface.	vp(4)
trman: translate version 6 manual macros to	version 6 manual macros to version 7 macros.	trman(1)
verch:	version 7 macros.	trman(1)
backup: make a backup	version changing program for Pascal sources.	verch(1)
netdirprint: print text	version copy of a file.	backup(1)
hangman: Computer	version of Pup Network Directory.	netdirprint(8)
file. what: show what	version of the game hangman.	hangman(6)
Versatec.	versions of object modules were used to construct a	what(1)
UNIX fonts.	vfont: font formats for the Benson-Varian or	vfont(5)
efficient way.	vfontinfo: inspect and print out information about	vfontinfo(1)
	vfork: spawn new process in a virtual memory	vfork(2)
	vgrind: grind nice listings of programs.	vgrind(1)
	vgrindefs: vgrind's language definition data base.	vgrindefs(5)
	vgrind's language definition data base.	vgrindefs(5)
	vhangup: virtually "hangup" the current control	vhangup(2)
	vi: screen oriented (visual) display editor based	vi(1)
recnews: receive unprocessed articles	via mail.	recnews(8)
sendnews: send news articles	via mail.	sendnews(8)
encode/decode a binary file for transmission	via mail. uencode,uudecode:	uencode(1C)
uurec: receive processed news articles	via mail.	uurec(8)
connect your terminal to a remote computer	via Pup network. puptelnet:	puptelnet(1)
mail: handle remote mail received	via uucp.	mail(1)
expand, unexpand: expand tabs to spaces, and	vice versa.	expand(1)
Uatimecvt: conversions from Unix to Alto time and	vice versa.	uatimecvt(9)
more, page: file perusal filter for crt	viewing.	more(1)
	vipw: edit the password file.	vipw(8)
	virtex: text formatting and typesetting.	tex(1)
tex, latex, initex.	virtual memory efficient way.	vfork(2)
vfork: spawn new process in a	virtual memory statistics.	vmstat(1)
vmstat: report	Virtual UNIX postmortem crash analyzer.	analyze(8)
analyze:		

vhangup:	virtually "hangup" the current control terminal.	vhangup(2)
vi: screen oriented consumption.	(visual) display editor based on ex.	vi(1)
vtroff, or troff.	vlimit: control maximum system resource	vlimit(3C)
	vlp: Format Lisp programs to be printed with nroff.	vlp(1)
	vmstat: report virtual memory statistics.	vmstat(1)
	vnews: read news articles.	vnews(1)
fs, inode: format of file system	volume.	fs(5)
	vp: Versatec interface.	vp(4)
vpr, vprm, spooler.	vpq, vprint: raster printer/plotter spooler.	vpr(1)
vpr, vprm, vpq, vpr,	vpr, vprm, vpq, vprint: raster printer/plotter	vpr(1)
	vprint: raster printer/plotter spooler.	vpr(1)
	vprm, vpq, vprint: raster printer/plotter spooler.	vpr(1)
vlp: Format Lisp programs to be printed with nroff,	vtimes: get information about resource utilization.	vtimes(3C)
	vtroff, or troff.	vlp(1)
	vtroff: troff to a raster plotter.	vtroff(1)
	vv: Proteon proNET 10 Mcgabit ring.	vv(4)
	vwidth: make troff width table for a font.	vwidth(1)
	w: who is on and what they are doing.	w(1)
	wait: await completion of process.	wait(1)
read, readonly, set, shift, times, trap, umask,	wait: command language. /exec, exit, export, login,	sh(1)
wait:	wait for a process to terminate.	wait(3F)
wait:	wait for background processes to complete.	csh(1)
sigpause: atomically release blocked signals and	wait for interrupt.	sigpause(2)
wait, wait3:	wait for process to terminate.	wait(2)
	wait: wait for a process to terminate.	wait(3F)
	wait: wait for background processes to complete.	csh(1)
	wait, wait3: wait for process to terminate.	wait(2)
	wait3: wait for process to terminate.	wait(2)
	wall: write to all users.	wall(1)
	wc: word count.	wc(1)
	weave: convert web file into pascal file, tex file.	tangle(1)
tangle, weave: convert	web file into pascal file, tex file.	tangle(1)
loadavg: average load log data on a	weekly basis.	loadavg(1)
what: show what versions of object modules	were used to construct a file.	what(1)
whatis: describe	what a command is.	whatis(1)
crash:	what happens when the system crashes.	crash(8V)
dumpfonts: show	what Press fonts are available in fonts.widths.	dumpfonts(1)
used to construct a file.	what: show what versions of object modules were	what(1)
w: who is on and	what they are doing.	w(1)
construct a file. what: show	what versions of object modules were used to	what(1)
	whatis: describe what a command is.	whatis(1)
	when the system crashes.	crash(8V)
crash: what happens	when you have to leave.	leave(1)
leave: remind you	whereami: report name of terminal.	whereami(1)
	whereis: locate source, binary, and or manual for	whereis(1)
program.	which: locate a program file including aliases and	which(1)
paths (csh only).	while, : . . . , break, continue, cd, eval,	sh(1)
exec, exit, export, login./ sh, for, case, if,	while: repeat commands conditionally.	csh(1)
	while/foreach loop.	csh(1)
break: exit	who are on the system.	users(1)
users: compact list of users	who is my mail from?	from(1)
from:	who is on and what they are doing.	w(1)
w:	who is on the system.	who(1)
who:	who it is from.	biff(1)
biff: be notified if mail arrives and	who: who is on the system.	who(1)
	whoami: print effective current user id.	whoami(1)
	whois: ask the ARPA Internet NIC about a user.	whois(1C)
	who's logged in on local machines.	rwho(1C)
rwho:	width output device.	fold(1)
fold: fold long lines for finite	width table for a font.	vwidth(1)
vwidth: make troff	widths. dumpfonts:	dumpfonts(1)
show what Press fonts are available in fonts.	without checking the disks.	fastboot(8)
fastboot, fasthalt: reboot/halt the system	word count.	wc(1)
wc:	word from stream.	getc(3S)
getc, getchar, fgetc, getw: get character or	word on a stream.	putc(3S)
putc, putchar, fputc, putw: put character or	wordy sentences; thesaurus for diction.	diction(1)
diction, explain: print	wordy sentences; thesaurus for diction.	explain(1)
explain, diction - print	working directory.	cd(1)
cd: change	working directory.	chdir(2)
chdir: change current	working directory.	getcwd(3F)
getcwd: get pathname of current	working directory name.	pwd(1)
pwd:	working directory pathname.	getwd(3)
getwd: get current	worm game.	worm(6)
worm: Play the growing	worm: Play the growing worm game.	worm(6)
	worms: animate worms on a display terminal.	worms(6)
worms: animate	worms on a display terminal.	worms(6)

putc, fputc:	write a character to a fortran logical unit.	putc(3F)
pupwrite:	write a packet to a pup channel.	pupwrite(9)
enwrite:	write a packet to the ethernet.	enwrite(9)
ansi: read and	write ANSI format magnetic tapes.	ansi(1)
write, writev:	write on a file.	write(2)
wall:	write to all users.	wall(1)
write:	write to another user.	write(1)
	write: write to another user.	write(1)
	write, writev: write on a file.	write(2)
write,	writev: write on a file.	write(2)
open: open a file for reading or	writing, or create a new file.	open(2)
utmp,	wtmp: login records.	utmp(5)
	wump: the game of hunt-the-wumpus.	wump(6)
en:	Xerox 3 Mb/s Ethernet interface.	en(4)
pup:	Xerox PUP-1 protocol family.	pup(4F)
xsend,	xget, enroll: secret mail.	xsend(1)
bit: and, or,	xor, not, rshift, lshift bitwise functions.	bit(3F)
	xsend, xget, enroll: secret mail.	xsend(1)
shared strings.	xstr: extract strings from C programs to implement	xstr(1)
j0, j1, jn,	y0, y1, yn: bessell functions.	j0(3M)
j0, j1, jn, y0,	y1, yn: bessell functions.	j0(3M)
eyacc: modified	yacc allowing much improved error recovery.	eyacc(1)
	yacc: yet another compiler-compiler.	yacc(1)
	yapp: yet another pretty printer.	yapp(1)
	yes: be repetitively affirmative.	yes(1)
j0, j1, jn, y0, y1,	yn: bessell functions.	j0(3M)
	zork: the game of dungeon.	zork(6)

