CB-UNIX

Programmer's Manual

Edition 2.3

J. D. Doan
Editor

May 1981

This document contains proprietary information of Bell Telephone Laboratories and is not to be disclosed, reproduced or published without Bell Laboratories approval.

Department 59473
Bell Telephone Laboratories, Incorporated
Columbus, OH 43213

UNIX is a trademark of Bell Laboratories.

This manual was set on a Graphic Systems, Inc. phototypesetter driven by the TROFF formatting program operating under the CB-UNIX system. The text of the manual was prepared using the ED text editor.

ACKNOWLEDGEMENTS

The form of this manual follows that of the *UNIX Programmer's Manual—Seventh Edition, Volume 1* developed by M. D. McIlroy. A large part of the present manual's contents is descended from the *UNIX Programmer's Manual—Sixth Edition* by K. Thompson and D. M. Ritchie (Bell Telephone Laboratories, May 1975) and the *PWB/UNIX User's Manual* by T. A. Dolotta, R. C. Haight, and E. M. Piskorik, eds. (Bell Telephone Laboratories, May 1977). A special credit should also be given to the UNIX support effort of department 3624; their support of UNIX helped make this manual possible. The number of our colleagues who have contributed to UNIX and CB-UNIX software and documentation is, by now, too large to list here, but the usefulness and acceptance of UNIX and of CB-UNIX is a true measure of their collective success.

Columbus, Ohio J.D.D.

INTRODUCTION

This manual describes the features of CB-UNIX. It provides neither a general overview of UNIX (for that, see "The UNIX Time-Sharing System," *BSTJ*, Vol. 57, No. 6, Part 2, pp. 1905-29, by D. M. Ritchie and K. Thompson), nor details of the implementation of the system (see "UNIX Implementation," *BSTJ*, same issue, pp. 1931-46). The manual is organized as follows:

1. Title Page
2. Acknowledgements
3. Introduction
4. How To Get Started
5. Table of Contents (including a permuted index)
6. Manual pages organized into 8 sections

The table of contents is organized by section and alphabetized within each section. The permuted index is derived from the table of contents. In the permuted index (and in general throughout the manual), each manual page name is followed by the manual section to which it refers (e.g. *ed*(1)); this convention is necessary because of the duplication of names between the various sections.

The manual pages are divided into eight sections; each section is further sub-divided into 2 sub-sections. Each section starts off with an introduction to the kinds of things which are documented in the section. Following this introduction, in the sub-section labeled "basic" is the documentation for those things which are considered to be basic to the section and are in the "prime" support class of the CB-UNIX support group. The second sub-section in each section is used to document routines added by the local application group, i.e. not supported by the CB-UNIX support group. The sections of the manual are:

1. Commands and Application Programs:
    1. General-Purpose Commands.
    1C. Communications Commands.
    1G. Graphics Commands.
    1M. System Maintenance Commands.
    1S. SCCS Commands.
    1X. Games.
1. Local Commands and Application Programs.
2. System Calls.
2. Local System Calls.
3. Subroutines:
    3C. C and Assembler Library Routines.
    3M. Mathematical Library Routines.
    3S. Standard I/O Library Routines.
    3X. Miscellaneous Routines.
3. Local Subroutines.
4. Device Interfaces and Special Files.
4. Local Device Interfaces and Special Files.
5. File Formats, Tables and Macros.
5. Local File Formats, Tables and Macros.
6. UNIX System Explantions.
6. Local UNIX System Explantions.
7. Kinks and Conventions.
7. Local Kinks and Conventions.
8. Stand-Alone Utilities.
8. Local Stand-Alone Utilities.

Section 1 (*Commands and Application Programs*) describes programs intended to be invoked directly by the user or by command language procedures, in contradistinction to subroutines, which are intended to be called by the user's programs. Commands generally reside in the directory **/bin** (for **bin**ary programs). Some programs also reside in **/usr/bin**, to save space in **/bin**. These directories are searched automatically by the command interpreter called the *shell*. Sub-class 1C contains communication programs such as *fget*, *dpr*, *cu*, etc. These entries may differ from system to system. Sub-class 1G contains graphics commands that involve graphics output on some device. Some examples of 1G commands are *gex*, *plot*, *graph*, etc. which mostly reside in the directory **/usr/bin**. Sub-class 1M contains system maintenance programs such as *fsck*, *mkfs*, etc., which generally reside in the directory **/etc**. These commands are not intended for use by the ordinary user due to their privileged nature. Sub-class 1S contains Source Code Control System commands such as *admin*, *get*, *delta*, etc. These entries mostly reside in the directory **/usr/bin**. Sub-class 1X are Games such as *ttt*, *wump*, *startrek*, etc., which reside in the directory **/usr/games**.

Section 2 (*System Calls*) describes the entries into the UNIX supervisor, including the Assembler and C language interfaces. In the Assembler, these system calls are invoked by the *sys* operation code, which is a synonym for the *trap* instruction.

Section 3 (*Subroutines*) describes the available subroutines. Their binary versions reside in various system libraries in directories **/lib** and **/usr/lib**. Sub-class 3C contains C and Assembler Library routines, which reside in **/lib/libc.a**. Sub-class 3M contains mathematical routines (*pow*, *log*, *sin*, etc.), which reside in **/lib/libm.a**. Sub-class 3S contains Standard I/O Library routines, also found in **/lib/libc.a**.

Section 4 (*Device Interfaces and Special Files*) discusses the characteristics of each special "file" that actually refers to an input/output device. The names in **Section 4** refer to the name of the device (i.e. RP04, RP03, TU16) rather than the names of the special file itself. Pseudo devices, like multiplexors and shared memory are also documented here.

Section 5 (*File Formats, Tables and Macros*) documents the structure of particular kinds of files; for example, the form of the output of the Assembler and the Loader is given. Excluded are files used by only one command, for example, the assembler's intermediate files. Some of the macro packages available, particularly for the text formatting programs *nroff*(1) and *troff*(1), are documented in **Section 5**.

Section 6 (*UNIX System Explanations*) includes boot procedures, system error messages, changable system parameters, etc.,.

Section 7 (*Kinks and Conventions*) documents proscribed conventions for file naming, C programming, etc.,.

Section 8 (*Stand-alone Utilities*) tells the story of programs which stand apart from UNIX in the sense that they run on the "bare" machine. For instance, *sacopy*, a device to device copy program is documented here.

Each section consists of a number of independent entries of a page or so each. The name of the entry appears in the upper corners of its pages. Entries within each section are alphabetized, with the exception of the introductory entry that begins each section. The page numbers of each entry start at 1. Some entries may describe several routines, commands, etc. In such cases, the entry appears only once, alphabetized under its "major" name.

All entries are based on a common format, not all of whose parts always appear:

The **NAME** part repeats the name of the entry and states (very briefly) its purpose.

The **SYNOPSIS** part summarizes the use of the program being described. A few conventions are used, particularly in Section 1 (*Commands*):

Boldface strings are literals and are to be typed just as they appear.

Italic strings usually represent substitutable argument prototypes and program names found elsewhere in the manual (they are underlined in the typed version of the entries).

Square brackets [ ] around an argument prototype indicate that the argument is optional. When an argument prototype is given as "name" or "file", 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 **–**, plus **+**, or equal sign **=** is often taken to be some sort of flag 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 −, +, or =.

The **DESCRIPTION** part discusses the subject at hand.

The **EXAMPLE(S)** part gives example(s) of usage, where appropriate.

The **FILES** part gives the file names that are built into the program.

The **SEE ALSO** part gives pointers to related information.

The **DIAGNOSTICS** part discusses the diagnostic indications that may be produced. Messages that are intended to be self-explanatory are not listed.

The **BUGS** part gives known bugs and sometimes deficiencies. Occasionally, the suggested fix is also described.

The **ASSEMBLER** part summarizes the procedure for invoking system calls from PDP-11 Assembler source programs. This part appears only in Section 2 and in the Introduction to Section 3.

All entries are available on-line via the *man*(1) command.

HOW TO GET STARTED

This discussion provides the basic information you need to get started on UNIX (we will use "UNIX" here to mean both "UNIX" and "CB-UNIX", unless the distinction matters): how to log in and log out, how to communicate through your terminal, and how to run a program. (See *UNIX for Beginners* by B. W. Kernighan for a more complete introduction to the system.)

Logging in. You must dial up UNIX from an appropriate terminal. UNIX supports full-duplex ASCII terminals. You must also have a valid user name, which may be obtained, together with the telephone number of the UNIX system, from the administrator of your system. Usually, the same telephone number serves terminals operating at speeds of 110, 150, and 300 baud. A different number may be used for 1200-baud service. After a data connection is established, the *login* procedure depends on the kind of terminal you are using.

300-baud terminals: These terminals generally have a speed switch that should be set to **300** (or **30**, for 30 characters per second) and a half-/full-duplex switch that should be set to full-duplex. When a connection is established, the system types **login:** and you then type your user name followed by the "return" key. If you have a password (and you should!), the system asks for it, but does not print ("echo") it on the terminal. The system may prompt you for a dialup password which is established by the system administrator (you need to know it to use any dial port into the system). After you have logged in, the "return", "new-line", and "line-feed" keys will give exactly the same result.

Higher-speed terminals: Terminals designed to run at higher data rates than 300 baud (i.e., 1200 baud) can be utilized in full-duplex mode provided *input* remains character-by-character, typing speed.

TELETYPE® Model 37 (and other terminals less than 300 baud): 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 150-baud terminal is in use. The system then will type "LOGIN:", this time at 150 baud (another "break" at this point will get you down to 110 baud); you respond with your user name. At this point the system will prompt you for any necessary passwords (see *300-baud terminals* above). From the TELETYPE Model 37, and any other terminal that has the "new-line" function (combined "carriage-return" and "line-feed" pair), terminate each line you type with the "new-line" key (*not* the "return" key).

Non-dial Terminals: In this case, the terminal should be prompting with the message "LOGIN:", if it is not, typing a "return" will usually cause it to do so. If it types garbage back at you, it is probably at the wrong speed; typing "break" will cause UNIX to cycle the terminal to another speed. If you try a few "breaks" and "returns" with no luck, see your local system guru or administrator - maybe the terminal is out of order.

It is important that you type your login name in lower case if possible; if you type upper-case letters, UNIX will assume that your terminal cannot generate lower-case letters and that you mean all subsequent upper-case input to be treated as lower case. When you have logged in successfully, the shell will type a $ to you. (The shell is described below under *How to run a program.*)

For more information, consult *login*(1) and *getty*(1M), which discuss the login sequence in more detail, *tty*(4), which discusses terminal input/output, and *stty*(1), which tells you how to describe the characteristics of your terminal to the system (*profile*(5) explains how to accomplish this last task automatically every time you log in).

Logging out. There are four ways to log out:

You can simply hang up the phone.

You can log out by typing an end-of-file indication (ASCII EOT character, usually typed as "control d") to the Shell. The Shell will terminate and the "LOGIN:" message will appear again (on some dial-up lines the line will be hung-up without the "LOGIN:" message appearing).

You can also log in directly as another user by giving a *login* command.

You can sit around for a while. After a specified interval has elapsed with no activity on your part you will be automatically logged out. The default wait interval is specified by the system administrator when the system is built.

How to communicate through your terminal. When you type to UNIX, a gnome deep in the system is gathering your characters and saving them. These characters will not be given to a program until you type a "return" (or "new-line"), as described above in *Logging in.*

UNIX terminal input/output 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 output will have interspersed in it the input characters. However, whatever you type will be saved and interpreted in the 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.

On an input line from a terminal, the character @ "kills" all the characters typed before it, so typing mistakes can be repaired on a single line. The character # erases the last character typed. Successive uses of # will erase characters back to, but not beyond, the beginning of the line; @ and # can be typed into a program by preceding them with \ (thus, to erase a \, you need two #s).

The ASCII **DC3** (control-s) character can be used to temporarily stop output. It is useful with CRT terminals to prevent output from disappearing before it can be read. Output is resumed only when an ASCII **DC1** (control-q) is typed. These start/stop characters are not passed to any other program when used in this manner. On CB-UNIX only, output may also be stopped by typing the break or escape keys. In this case, typing another escape (or any other characters, for that matter) will cause output to be resumed.

The ASCII "delete" (a.k.a. "rubout") character is not passed to programs, but instead generates an *interrupt signal.* 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 *ed*(1), 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.

The *quit* signal is generated by typing the ASCII **FS** character. 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 as to whether you have a terminal with the "new-line" function, or whether it must be simulated with a "carriage-return" and "line-feed" pair. In the latter case, all *input* "carriage-return" characters are changed to "line-feed" characters (the standard line delimiter), and a "carriage-return" and "line-feed" pair is echoed to the terminal. If you get into the wrong mode, the *stty*(1) command will rescue you. *Stty* can also be used to change the default *erase* and *kill* characters mentioned above.

Tab characters are used freely in UNIX source programs. If your terminal does not have the tab function, you can arrange to have tab characters changed into spaces during output, and echoed as spaces during input. Again, the *stty*(1) command will set or reset this mode. The system assumes that tabs are set every eight character positions.

How to run a program. When you have successfully logged into UNIX, a program called the shell is listening to your terminal. The shell reads the lines you type, splits them into a command name and its arguments, and executes the command. A command is simply an executable program. Normally, the shell looks first in your current directory (see *The current directory* below) for a program with the given name, and if none is there, then in system directories. There is nothing special about system-provided commands except that they are kept in directories where the shell can find them. The command name is usually the first word on an input line to the shell; the command and its arguments are separated from one another by space and/or tab characters.

When a program terminates, the shell will ordinarily regain control and type a **$** at you to indicate that it is ready for another command. The shell has many other capabilities, which are described in detail in *sh*(1).

The current directory. UNIX has a file system arranged in a hierarchy of directories. When the system administrator gave you a user name, he or she also created a directory for you (ordinarily with the same name as your user name, and known as your *login* directory). When you log in, that directory becomes your *current* or *working* directory, and any file name you type is by default assumed to be in this directory. Because you are the owner of this directory, you have full permissions 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 respective owners, or by the system administrator. To change the current directory 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 /, which is 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 (e.g., **/usr/ae/filex** refers to file **filex** in directory **ae**, while **ae** is itself a subdirectory of **usr**, and **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 corresponding subdirectory (*without* a prefixed /). Without important exception, a path name may be used anywhere a file name is required.

Important commands that 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). Use *mkdir*(1) for making directories and *rmdir*(1) for destroying them.

For a fuller discussion of the file system, see the references cited at the beginning of the *INTRODUCTION* above. 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 *ed*(1). The four principal languages available under UNIX are C (see *cc*(1)), Fortran (see *f77*(1)), bs (a compiler/interpreter in the spirit of Basic, see *bs*(1)), and assembly language (see *as*(1)). After the program text has been entered with the editor and written into a file (whose name has the appropriate suffix), you can give the name of that file to the appropriate language processor as an argument. Normally, the output of the language processor will be left in a file in the current directory named **a.out** (if that output is precious, use *mv*(1) to give it a less vulnerable name). If the program is written in assembly language, you will probably need to load with it library subroutines (see *ld*(1)). Fortran and C call the loader automatically; programs written in *bs*(1) are interpreted and, therefore, do not need to be loaded.

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 **$** prompt.

If any execution (run-time) errors occur, you will need *adb*(1) to examine the remains of your program.

Your programs can receive arguments from the command line just as system programs do. See *exec*(2).

Text processing. Almost all text is entered through the editor *ed*(1). The commands most often used to write text on a terminal are *cat*(1), *pr*(1), and *nroff*(1). The *cat*(1) command simply dumps ASCII text on the terminal, with no processing at all. The *pr*(1) command paginates the text, supplies headings, and has a facility for multi-column output. *Nroff*(1) is an elaborate text formatting program, and requires careful forethought in entering both the text and the formatting commands into the input file; it produces output on a typewriter-like terminal. *Troff*(1) is very similar to *nroff*(1), but drives a Graphic Systems, Inc. phototypesetter. (It was used to typeset this manual.) There are several "macro" packages (especially the so-called *mm* package) that significantly ease the effort required to use *nroff*(1) and *troff*(1); Section 5 entries for these packages indicate where you can find their detailed descriptions.

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 or she next logs in. The corresponding entries in this manual also suggest how to respond to these two commands if you are their target.

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

Normally, the UNIX system runs in a smooth manner and users need not be concerned about details of system operation; however, when a hardware problem exists or when new features are being tested, users may be asked to log off so that problems can be corrected. If asked to log off please do so promptly as work done may be in danger of being lost or destroyed. Whenever possible, the system will give sufficient warning of an impending outage so that you may "gracefully" log off.

TABLE OF CONTENTS

1. Commands and Application Programs

intro	introduction to commands and application programs
abort	remove previously queued line printer jobs
ac	login accounting
accton	turn accounting on/off
adb	debugger
addsccs	add SCCS keywords to a file
admin	administer SCCS files
ar	archive and library maintainer
arcv	convert archives to new format
as	assembler
at	execute commands at a later time
atgex	convert ascii file to GEX format
awk	pattern scanning and processing language
banner	make headlines
basename	deliver portions of pathnames
bc	arbitrary-precision arithmetic language
bclk	reads and sets the battery clock
bd	binary dump of a file
bdiff	big diff
bdump	read from block device
bload	write on block device
bs	a compiler/interpreter for modest-sized programs
cal	print calendar
calendar	reminder service
cat	concatenate and print files
cb	C program beautifier
cc	C compiler
chdir	change working directory
check	file system consistency check and repair
chess	the game of chess
chghist	change the history entry of an SCCS delta
chkold	file system consistency chkold
chmod	change mode of file
chown	change owner of group of a file
chroot	change root directory for a command
clri	clear inode
cmp	compare two files
cmpfs	compare and archive file systems
cmt	insert the delta commentary for an initial SCCS delta
col	filter reverse line-feeds
comb	combine SCCS deltas
comm	select or reject lines common to two sorted files
cp	copy, link, or move files
cpio	copy file archives in and out
cpmv	copy move
cref	make cross reference listing
cron	clock daemon
crypt	encode/decode
ct	call terminal
cu	call another UNIX system
cubic	three dimensional tic-tac-toe
cut	cut out selected fields of each line of a file
date	print and set the date

dc	desk calculator
dcheck	file system directory consistency check
dd	convert and copy a file
dead	crash analysis
delta	to an SCCS file
deroff	remove nroff, troff, tbl and eqn constructs
df	disk free
diff	differential file comparator
diff3	3-way differential file comparison
diffmk	mark differences between files
dircmp	directory comparison
dmplfs	dump logical file system to tape
dsw	delete interactively
du	summarize disk usage
echo	echo arguments
ed	text editor
env	set environment for command execution
epoch	print and set system backup date
eqn	typeset mathematical text
errdead	extract error records from dump
errdemon	error-logging daemon
errpt	process a report of logged errors
expr	evaluate arguments as an expression
f77	FORTRAN 77 compiler
factor	factor a number, generate large primes
fed	edit associative memory for form letter
festoon	turgid memorandum composition
file	determine file type
file_log	log an input string in a logfile.
find	find files
flog	speed up a process
form	form letter generator
g_find	locate and identify a source file
gadd	add a file to SCCS
gadmin	admin a file in SCCS
gcat	send phototypesetter output to the HONEYWELL 6000
gcon	convert GEX file to HIS format
gdelta	delta a file from SCCS
gdiff	diff an SCCS file with named file
gdump	prints a gex graphic file
get	get a version of an SCCS file
getpc	get Program Counter data on running processes
getty	set terminal type, modes, speed, and line discipline
gex	Graphic EXerciser for Tektronix 4014
gget	get a file from SCCS
gls	list the directory $SCCSOURCE with input args appended
gmark	mark a subsystem of SCCS files.
gprt	prt a file in SCCS
graph	draw a graph
grep	search a file for a pattern
gsplit	filter to break gex files into pieces
gtty	get terminal line options
hatch	filter to hatch GEX files
help	ask for help
hex	translate binary file to ascii hexadecimal
hold	suspend printing of queued line printer jobs

hyphen . . . . find hyphenated words
id . . . . print user and group id
idump . . . . dump an inode
infect . . . . Give a virus to another UNIX system
init . . . . reinitialize line printer demon
init . . . . process control initialization
inode . . . . find inode on disk
install . . . . install commands
iostat . . . . report I/O and system statistics
join . . . . relational database operator
kasb . . . . assembler for the KMC11 microprocessor
kill . . . . send a signal to a process or process group
kunb . . . . un-assembler for the KMC11/DMC11 microprocessor
ld . . . . link editor
ldrxboot . . . . load floppy disk second level boot
lex . . . . generate programs for simple lexical tasks
lfcheck . . . . consistency check and repair
lfmount . . . . mount logical file system (LFS)
lfsync . . . . update modified LFS data
lfumount . . . . unmount the logical file system (LFS)
lfupdate . . . . update modified LFS data repetitively
line . . . . get line identification
lint . . . . a C program verifier
load . . . . load
login . . . . sign on
lorder . . . . find ordering relation for an object library
lpr . . . . line printer spooling program
ls . . . . list contents of directory
m4 . . . . macro processor
mail . . . . send mail to users or read mail
make . . . . maintain program groups
man . . . . print pages of this manual
mesg . . . . permit or deny messages
mhdump . . . . incremental file system dump
mhrestor . . . . incremental file system restore
mhstty . . . . set the options for a terminal
mkconf . . . . create configuration table and low core
mkdir . . . . make a directory
mkfs . . . . construct a file system
mkfst . . . . construct a file system on mag tape
mklfs . . . . construct a Logical File System (LFS)
mknod . . . . build special file
mkpt . . . . make proto
mm . . . . type out documents that use the PWB/MM macros
mmkdir . . . . made path names
mo . . . . nroff, nnroff mm interface for preprinted letterhead
moo . . . . guessing game
mount . . . . mount file system
move . . . . move a file and set the mode
mtm . . . . magnetic tape manipulation
mvdir . . . . move a directory
nar . . . . new format archive and library maintainer
ncheck . . . . generate names from i-numbers
newgrp . . . . log in to a new group
news . . . . print news items
nice . . . . run a command at specified priority

nm . . . print name list
nnroff . . . format text
nohup . . . run a command immune to hangups
nroff . . . format or typeset text
occ . . . old C compiler
od . . . octal dump
over . . . overstrike optimizer
pack . . . compress files
padm . . . program administration system
passwd . . . change login password
paste . . . merge same lines of several files or subsequent lines of one file
pcat . . . expand compressed file to standard output
pcstat . . . report statistics on output of getpc command
plot . . . graphics filters
pr . . . print file
prof . . . display profile data
prs . . . print an SCCS file
prt . . . print SCCS file
ps . . . process status
ptx . . . permuted index
pwd . . . working directory name
quot . . . summarize file system ownership
ratfor . . . rational FORTRAN dialect
readl . . . read one line
reboot . . . replace current UNIX with new program or system
reform . . . reformat text file
release . . . restore printing of queued line printer jobs
restrain . . . suspend printing of queued line printer jobs
rew . . . rewind tape
rm . . . remove files or directories
rmdel . . . remove a delta from an SCCS file
rsh . . . restricted shell (command interpreter)
rstlfs . . . restore logical file system from tape
sa . . . shell accounting
savdate . . . save and restore modification date
sccsclean . . . remove unwanted files in SCCS directories
sccsdiff . . . compare two versions of an SCCS file
sccstring . . . echo SCCS keywords to standard output
sdiff . . . side-by-side difference program
sed . . . stream editor
sema . . . semaphore operations
setpgrp . . . execute program with new process group
sh . . . shell, the standard/restricted command programming language
size . . . size of an object file
sleep . . . suspend execution for an interval
sno . . . SNOBOL interpreter
sort . . . sort or merge files
spell . . . find spelling errors
spline . . . interpolate smooth curve
split . . . split a file into pieces
spr . . . special print command
sprof . . . system profile
sps . . . detail process status
stack . . . stack trace from crash file
stamp . . . version stamp utility
start . . . restore printing of queued line printer jobs

startrek	clobber klingons
strip	remove symbols and relocation bits
stty	set teletype options
su	become super-user or another user
sum	sum and count blocks in a file
sync	update the super block
tail	deliver the last part of a file
talk	allow user to listen and talk to one or more other users
tar	tape archiver
tbl	format tables for nroff or troff
tc	phototypesetter simulator
tcmp	text comparison for crash dump
tee	pipe fitting
tek	graphics filters
telinit	user communication with init
test	condition evaluation command
time	time a command
tk	paginator for the Tektronix 4014
tkdump	prints a Tektronix file
tm	meditate
touch	change modification time of a file
tp	manipulate tape archive
tr	translate characters
true	provide truth values
tsort	topological sort
ttt	tic-tac-toe
tty	get the terminal's name
turbo	encabulator
typo	find possible typos
ucore	turn on or off the unique core dumping feature.
umount	dismount file system
uname	print name of current UNIX
unhex	translate hexed file to binary
uniq	report repeated lines in a file
units	conversion program
unpack	expand compressed files
update	periodically update the super block
updfs	update file system
uuclean	uucp spool directory clean-up
uucp	unix to unix copy
uunames	list names of UNIX systems known to uucp
uustat	uucp status inquiry and job control
uusub	monitor uucp network
uux	unix to unix command execution
val	validate SCCS file
vcrt	filter nroff output for virtual crts
vpmc	compiler for the virtual protocol machine
vpmsave	save and print VPM event traces
vpmset	connect VPM drivers and KMCs; load the KMC11-B.
wait	await completion of process
wall	write to all users
wc	word count
what	identify files
who	who is on the system
whodo	who is doing what
write	write to another user

wump . . . hunt the wumpus
x25pvc . . . install, remove, or get status for a PVC or BX.25 link
xargs . . . construct argument lists and execute command
xref . . . cross reference for C programs
yacc . . . yet another compiler-compiler

2. System Calls

intro . . . introduction to system calls
access . . . determine accessibility of file
acct . . . turn accounting on/off
alarm . . . schedule signal after specified time
break . . . change memory allocation
chdir . . . change working directory
chgrp:o . . . change group
chmod . . . change mode of file
chown . . . change owner and group of a file
chown:o . . . change owner
chroot . . . change root directory
close . . . close a file
creat . . . create a new file
dup . . . duplicate an open file descriptor
exec . . . execute a file
exec:o . . . execute a file
exit . . . terminate process
fcntl . . . file control
fork . . . spawn new process
fstat . . . get status of open file
ftime . . . get date and time
getcsw . . . read console switches
getgid:o . . . get group identification
getpid . . . get process identification
getu . . . get selected user block information
getuid . . . get user and group identity
getuid:o . . . get user identification
indir . . . indirect system call
ioctl . . . control device
kill . . . send signal to a process
link . . . link to a file
lseek . . . move read/write pointer
maus . . . multiple access user space operations
mdate:o . . . set modified date on file
message . . . send and receive messages
mknod . . . make a directory or a special file
mount . . . mount or remove file system
mpx . . . create and manipulate multiplexed files
nice . . . set program priority
open . . . open for reading or writing
pause . . . stop until signal
pipe . . . create a pipe
plock . . . lock process or text in memory
profil . . . execution time user profile
ptrace . . . process trace
read . . . read from file
reboot . . . transfer control to DEC rom and reboot
seek:o . . . move read/write pointer
sema . . . semaphore operations

setgid	set process group ID
setpgrp	set process group
setuid	set process user ID
shmem	shared memory operations
signal	catch or ignore signals
sprofil	turn on/off system profiling
stat	get file status
stat:o	get file status
stime	set time
stty:o	set and retrieve terminal modes
sync	update super-block
tell:o	get file offset
time	get date and time
times	get process times
ucore	enable/disable unique core dumping feature.
umask	set and get creation mask
umount	dismount file system
uname	get name of current UNIX system
unlink	remove directory entry
utime	update times in file
wait	wait for process to die
write	write on a file

3. Subroutines

intro	introduction to subroutines and libraries
a64l	convert between long and base-64 ASCII
abort	generate an IOT fault
abs	integer absolute value
alloc:o	core allocator
alrm	audible alarm
assert	program verification
atof	convert ASCII to numbers
atof:o	convert ASCII to numbers
bessel	bessel functions
call:o	create and execute a new process
calloc:o	core memory allocator
clearer:o	stream error reset
cnvtime	convert string to internal time
conns	connect to a remote system
conv	character translation
crypt	DES encryption
cspeed	convert baud to speed number
ctermid	generate file name for terminal
ctime	convert date and time to ASCII
ctime:o	convert date and time to ASCII
ctype	character classification
cuserid	character user ID
dtol:o	double precision integer to floating point conversion
ecvt	output conversion
end	last locations in program
exp	exponential, logarithm, power, square root
exprog:o	perform standard Shell execute sequence
fclose	close or flush a stream
ferror	stream status inquiries
floor	absolute value, floor, ceiling, remainder functions
fopen	open a stream

fopen:o ... open a stream
fpemul ... floating point interpreter
fread ... buffered binary input/output
frexp ... split into mantissa and exponent
fseek ... reposition a stream
gamma ... log gamma function
getc ... get character or word from stream
getc:o ... buffered input
getchar:o ... read character
getenv ... value for environment name
getgrent ... get group file entry
getlogin ... get login name
getopt ... get option letter from argv
getpass ... read a password
getpw ... get name from UID
getpwent ... get password file entry
gets ... get a string from a stream
getut ... access utmp file entry
hmul:o ... high-order product
hypot ... euclidean distance
itol:o ... integer to long integer conversion
l3tol ... convert between 3-byte integers and long integers
ldiv:o ... long division
lfs ... Logical File System operations
lib7 ... Version 7 library
libil ... CB UNIX Release 1 Conversion Library
lnxx:o ... return name of current terminal
locv:o ... long output conversion
lpdata ... decode line printer data files (printers and qmap)
lpropen ... open pipe to the line printer
ltod:o ... double precision integer to floating point conversion
ltoi:o ... long integer to integer conversion
malloc ... main memory allocator
malloc:o ... core memory allocator
mkdir:o ... make directory
mktemp ... make a unique file name
mktemp:o ... make temporary file name
mktmp:o ... make a temporary file
monitor ... prepare execution profile
msg ... old message veneer for sending and receiving messages.
nargs:o ... argument count
nlist ... get entries from name list
nlist:o ... get entries from name list
perror ... system error messages
plot ... graphics interface
popen ... initiate I/O to/from a process
printf ... formatted output conversion
printf:o ... formatted print
putc ... put character or word on a stream
putc:o ... buffered output
putchar:o ... write character
putpwent ... write password file entry
puts ... put a string on a stream
qsort ... quicker sort
rand ... random number generator
reset:o ... execute non-local goto

rmdir:o . . . remove directory
scanf . . . formatted input conversion
setbuf . . . assign buffering to a stream
setjmp . . . non-local goto
sinh . . . hyperbolic functions
sleep . . . stop execution for interval
ssignal . . . software signals
stdio . . . standard buffered input/output package
stdio:o . . . standard buffered input/output package
string . . . string operations
stty . . . set and retrieve the modes of a terminal
swab . . . swap bytes
system . . . issue a shell command
tell . . . get file offset
tmpnam . . . create a name for a temporary file
trig . . . trigonometric functions
ttyname . . . find name of a terminal
ttyslot . . . find the slot in the utmp file of the current user
ungetc . . . push character back into input stream
utindx . . . access routines for utmp file

4. Device Interfaces Special Files

intro . . . introduction to special files
dh . . . asynchronous multiplexers
dn . . . DN-11 ACU interface
err . . . error-logging interface
hp . . . RP04/RP05/RP06 moving-head disk
kl . . . KL-11 or DL-11 asynchronous interface
kmc . . . KMC11/DMC11 microprocessor
lp . . . line printer
mem . . . core memory
mt . . . TE16/TU16 magnetic tape interface
nc . . . network control
null . . . the null file
pcs . . . program counter sampling device
pipe . . . named pipe
rk . . . RK11/RK03 or RK05 disk
rootdev . . . root file system
rx . . . floppy disk
swapdev . . . location for swapping
tm . . . TM11/TU10 magnetic tape interface
trace . . . event-tracing driver
tty . . . general interface for terminals
vp . . . Versatec printer-plotter
vpm . . . Virtual Protocol Machine Protocol and Interface
vt . . . graphics interface
vtp . . . virtual terminal protocol
x25 . . . BX.25 network interface

5. File Formats, Tables, and Macros

intro . . . file format description
L-devices . . . auto-dialer device table
L-dialcodes . . . uucp system dialcodes
L.sys . . . table of connecting uucp systems
a.out . . . assembler and link editor output

acct . . . . accounting file

ar . . . . archive file format

cpio . . . . format of cpio archive

crontab . . . . table of chronological events to be executed.

d_passwd . . . . dial up password file

dialups . . . . list of dialup lines

dir . . . . format of directories

dump . . . . incremental dump tape format

errfile . . . . error-log file format

fs . . . . format of system volume

gettydefs . . . . speed and terminal settings used by getty

group . . . . group file

inittab . . . . script for the init process

inode . . . . format of an inode

issue . . . . issue identification file

lfs . . . . format of Logical File System disk area

manmac . . . . macros to print CB-UNIX manual sections

mountpts . . . . general user mount point table

mpxio . . . . multiplexed I/O

mtab . . . . mounted file system table

nar . . . . file format

passwd . . . . password file

plot . . . . graphics interface

powerfail . . . . commands to be executed following powerfail

printers . . . . defines printer options to /etc/lpd

profile . . . . setting up an environment at login time

qmap . . . . queue to printers map

sccsfile . . . . format of SCCS file

tp . . . . magnetic tape format

utmp . . . . utmp and wtmp entry formats

6. Unix System Explantions

intro . . . . introduction to UNIX system explanations

bproc . . . . UNIX startup

crash . . . . what to do when the system crashes

uemess . . . . description of UNIX console messages

7. Kinks and Conventions

intro . . . . introduction to miscellany

ascii . . . . map of ASCII character set

environ . . . . user environment

greek . . . . graphics for extended TTY-37 type-box

regexp . . . . regular expression compile and match routines

stat . . . . data returned by stat system call

types . . . . primitive system data types

8. Stand-Alone Utilities

intro . . . . introduction to stand-alone utilities

mmtest . . . . PDP 11/70 memory management test

sacopy . . . . stand-alone copy/verify

PERMUTED INDEX

mmtest: PDP	11/70 memory management test.	mmtest(8)
l3tol, ltol3: convert between	3-byte integers and long integers.	l3tol(3C)
diff3:	3-way differential file comparison.	diff3(1)
gex: Graphic EXerciser for Tektronix	4014.	gex(1G)
tk: paginator for the Tektronix	4014.	tk(1)
gcat: send phototypesetter output to the HONEYWELL	6000.	gcat(1C)
f77: FORTRAN	77 compiler.	f77(1)
	a64l, l64a: convert between long and base-64 ASCII.	a64l(3C)
	abort: generate an IOT fault.	abort(3C)
	abort: remove previously queued line printer jobs.	abort(1)
	abs: integer absolute value.	abs(3C)
abs: integer	absolute value.	abs(3C)
functions. floor, fabs, ceil, fmod:	absolute value, floor, ceiling, remainder	floor(3M)
	ac: login accounting.	ac(1)
	access: determine accessibility of file.	access(2)
utindx, utline:	access routines for utmp file.	utindx(3C)
freemaus, enabmaus, dismaus, switmaus: multiple	access user space operations. maus, getmaus,	maus(2)
getutline, pututline, setutent, endutent, utmpname:	access utmp file entry. getutent, getutid,	getut(3C)
access: determine	accessibility of file.	access(2)
ac: login	accounting.	ac(1)
acct:	accounting file.	acct(5)
acct: turn	accounting on/off.	acct(2)
accton: turn	accounting on/off.	accton(1)
sa: shell	accounting.	sa(1M)
	acct: accounting file.	acct(5)
	acct: turn accounting on/off.	acct(2)
	accton: turn accounting on/off.	accton(1)
sin, cos, tan, asin,	acos, atan, atan2: trigonometric functions.	trig(3M)
dn: DN-11	ACU interface.	dn(4)
	adb: debugger.	adb(1)
gadd:	add a file to SCCS.	gadd(1S)
addsccs:	add SCCS keywords to a file.	addsccs(1S)
	addsccs: add SCCS keywords to a file.	addsccs(1S)
gadmin:	admin a file in SCCS.	gadmin(1S)
	admin: administer SCCS files.	admin(1S)
admin:	administer SCCS files.	admin(1S)
padm: program	administration system.	padm(1S)
alrm: audible	alarm.	alrm(3C)
	alarm: schedule signal after specified time.	alarm(2)
	alloc: core allocator.	alloc:o(3C)
break, brk, sbrk: change memory	allocation.	break(2)
alloc: core	allocator.	alloc:o(3C)
calloc, cfree: core memory	allocator.	calloc:o(3S)
malloc, free, realloc, calloc: main memory	allocator.	malloc(3C)
malloc, free: core memory	allocator.	malloc:o(3C)
users. talk:	allow user to listen and talk to one or more other	talk(1)
	alrm: audible alarm.	alrm(3C)
dead: crash	analysis.	dead(1M)
	a.out: assembler and link editor output.	a.out(5)
gls: list the directory $SCCSOURCE with input args	appended.	gls(1S)
intro: introduction to commands and	application programs.	intro(1)
	ar: archive and library maintainer.	ar(1)
	ar: archive file format.	ar(5)
bc:	arbitrary-precision arithmetic language.	bc(1)
ar:	archive and library maintainer.	ar(1)
nar: new format	archive and library maintainer.	nar(1)
cpio: format of cpio	archive.	cpio(5)
ar:	archive file format.	ar(5)
cmpfs: compare and	archive file systems.	cmpfs(1M)
nar:	archive (library) file format.	nar(5)
tp: manipulate tape	archive.	tp(1)
tar: tape	archiver.	tar(1)
cpio: copy file	archives in and out.	cpio(1)
arcv: convert	archives to new format.	arcv(1M)
	arcv: convert archives to new format.	arcv(1M)
gls: list the directory $SCCSOURCE with input	args appended.	gls(1S)
nargs:	argument count.	nargs:o(3C)
xargs: construct	argument lists and execute command.	xargs(1)
expr: evaluate	arguments as an expression.	expr(1)
echo, fecho: echo	arguments.	echo(1)
getopt: get option letter from	argv.	getopt(3C)
bc: arbitrary-precision	arithmetic language.	bc(1)

expr: evaluate arguments	as an expression.	expr(1)
	as: assembler.	as(1)
a64l, l64a: convert between long and base-64	ASCII.	a64l(3C)
ascii: map of	ASCII character set.	ascii(7)
gmtime, asctime, timezone: convert date and time to	ASCII. ctime, localtime,	ctime(3C)
ctime: convert date and time to	ASCII.	ctime:o(3C)
atgex: convert	ascii file to GEX format.	atgex(1G)
hex: translate binary file to	ascii hexadecimal.	hex(1)
	ascii: map of ASCII character set.	ascii(7)
atof, atoi, atol: convert	ASCII to numbers.	atof(3C)
atof, atoi, atol: convert	ASCII to numbers.	atof:o(3C)
ctime, localtime, gmtime,	asctime, timezone: convert date and time to ASCII.	ctime(3C)
sin, cos, tan,	asin, acos, atan, atan2: trigonometric functions.	trig(3M)
help:	ask for help.	help(1S)
a.out:	assembler and link editor output.	a.out(5)
as:	assembler.	as(1)
kasb:	assembler for the KMC11 microprocessor.	kasb(1)
	assert: program verification.	assert(3X)
setbuf:	assign buffering to a stream.	setbuf(3S)
fed: edit	associative memory for form letter.	fed(1)
kl: KL-11 or DL-11	asynchronous interface.	kl(4)
dh, dz:	asynchronous multiplexers.	dh(4)
sin, cos, tan, asin, acos,	atan, atan2: trigonometric functions.	trig(3M)
sin, cos, tan, asin, acos, atan,	atan2: trigonometric functions.	trig(3M)
	atgex: convert ascii file to GEX format.	atgex(1G)
	atof, atoi, atol: convert ASCII to numbers.	atof(3C)
	atof, atoi, atol: convert ASCII to numbers.	atof:o(3C)
atof,	atoi, atol: convert ASCII to numbers.	atof(3C)
atof,	atoi, atol: convert ASCII to numbers.	atof:o(3C)
atof, atoi,	atol: convert ASCII to numbers.	atof(3C)
atof, atoi,	atol: convert ASCII to numbers.	atof:o(3C)
create and manipulate/ mpx, join, chan, extract,	attach, detach, connect, npgrp, ckill, mpxcall:	mpx(2)
alrm:	audible alarm.	alrm(3C)
L-devices:	auto-dialer device table.	L-devices(5)
wait:	await completion of process.	wait(1)
	awk: pattern scanning and processing language.	awk(1)
ungetc: push character	back into input stream.	ungetc(3S)
epoch: print and set system	backup date.	epoch(1M)
	banner: make headlines.	banner(1)
a64l, l64a: convert between long and	base-64 ASCII.	a64l(3C)
	basename, dirname: deliver portions of pathnames.	basename(1)
bclk, setbclk: reads and sets the	battery clock.	bclk(1M)
cspeed: convert	baud to speed number.	cspeed(3C)
	bc: arbitrary-precision arithmetic language.	bc(1)
	bclk, setbclk: reads and sets the battery clock.	bclk(1M)
	bd: binary dump of a file.	bd(1)
	bdiff: big diff.	bdiff(1)
	bdump: read from block device.	bdump(1M)
cb: C program	beautifier.	cb(1)
j0, j1, jn, y0, y1, yn:	bessel functions.	bessel(3M)
bd:	binary dump of a file.	bd(1)
hex: translate	binary file to ascii hexadecimal.	hex(1)
fread, fwrite: buffered	binary input/output.	fread(3S)
unhex: translate hexed file to	binary.	unhex(1)
strip: remove symbols and relocation	bits.	strip(1)
	bload: write on block device.	bload(1M)
bdump: read from	block device.	bdump(1M)
bload: write on	block device.	bload(1M)
getu: get selected user	block information.	getu(2)
semaphore operations. sema, p, v, test, post,	block, setsem, rdsem, lock, unlock, tlock, noulk:	sema(2)
sync: update the super	block.	sync(1M)
update: periodically update the super	block.	update(1)
sum: sum and count	blocks in a file.	sum(1)
ldrxboot: load floppy disk second level	boot.	ldrxboot(1)
	boot procedures: UNIX startup.	bproc(6)
	break, brk, sbrk: change memory allocation.	break(2)
gsplit: filter to	break gex files into pieces.	gsplit(1G)
break,	brk, sbrk: change memory allocation.	break(2)
programs.	bs: a compiler/interpreter for modest-sized	bs(1)
fread, fwrite:	buffered binary input/output.	fread(3S)
getc:	buffered input.	getc:o(3C)
stdio: standard	buffered input/output package.	stdio(3S)
stdio: standard	buffered input/output package.	stdio:o(3S)
putc:	buffered output.	putc:o(3C)
setbuf: assign	buffering to a stream.	setbuf(3S)
mknod:	build special file.	mknod(1M)

x25lnk: install, remove, or get status for a PVC or	BX.25 link. x25pvc.	x25pvc(1C)
X25:	BX.25 network interface.	x25(4)
swab: swap	bytes.	swab(3C)
cc, pcc:	C compiler.	cc(1)
occ: old	C compiler.	occ(1)
cb:	C program beautifier.	cb(1)
lint: a	C program verifier.	lint(1)
xref: cross reference for	C programs.	xref(1)
	cal: print calendar.	cal(1)
dc: desk	calculator.	dc(1)
cal: print	calendar.	cal(1)
	calendar: reminder service.	calendar(1)
cu:	call another UNIX system.	cu(1C)
indir: indirect system	call.	indir(2)
stat: data returned by stat system	call.	stat(7)
ct:	call terminal.	ct(1C)
	calloc, cfree: core memory allocator.	calloc:o(3S)
malloc, free, realloc,	calloc: main memory allocator.	malloc(3C)
intro: introduction to system	calls.	intro(2)
	cat: concatenate and print files.	cat(1)
signal:	catch or ignore signals.	signal(2)
	cb: C program beautifier.	cb(1)
libil:	CB UNIX Release 1 Conversion Library.	libi1(3X)
manmac: macros to print	CB-UNIX manual sections.	manmac(5)
	cc, pcc: C compiler.	cc(1)
chdir,	cd: change working directory.	chdir(1)
remainder functions. floor, fabs,	ceil, fmod: absolute value, floor, ceiling,	floor(3M)
floor, fabs, ceil, fmod: absolute value, floor,	ceiling, remainder functions.	floor(3M)
calloc,	cfree: core memory allocator.	calloc:o(3S)
ckill, mpxcall: create and manipulate/ mpx, join,	chan, extract, attach, detach, connect, npgrp,	mpx(2)
delta: make a delta	(change) to an SCCS file.	delta(1S)
ungetc: push	character back into input stream.	ungetc(3S)
isspace, ispunct, isprint, iscntrl, isascii:	character classification. /isdigit, isalnum,	ctype(3S)
getchar: read	character.	getchar:o(3C)
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)
putchar: write	character.	putchar:o(3C)
ascii: map of ASCII	character set.	ascii(7)
toupper, tolower, toascii:	character translation.	conv(3C)
cuserid:	character user ID.	cuserid(3S)
tr: translate	characters.	tr(1)
	chdir, cd: change working directory.	chdir(1)
	chdir: change working directory.	chdir(2)
check: file system consistency	check and repair.	check(1M)
lfcheck: logical file system (LFS) consistency	check and repair.	lfcheck(1)
dcheck: file system directory consistency	check.	dcheck(1M)
	check: file system consistency check and repair.	check(1M)
eqn, neqn,	checkeq: typeset mathematical text.	eqn(1)
chess: the game of	chess.	chess(1X)
	chess: the game of chess.	chess(1X)
	chghist: change the history entry of an SCCS delta.	chghist(1S)
	chgrp: change group.	chgrp:o(2)
chown,	chgrp: change owner of group of a file.	chown(1)
chkold: file system consistency	chkold.	chkold(1M)
	chkold: file system consistency chkold.	chkold(1M)
	chmod: change mode of file.	chmod(1)
	chmod: change mode of file.	chmod(2)
	chown: change owner and group of a file.	chown(2)
	chown: change owner.	chown:o(2)
	chown, chgrp: change owner of group of a file.	chown(1)
crontab: table of	chronological events to be executed..	crontab(5)
	chroot: change root directory.	chroot(2)
	chroot: change root directory for a command.	chroot(1M)
/chan, extract, attach, detach, connect, npgrp,	ckill, mpxcall: create and manipulate multiplexed/	mpx(2)
ispunct, isprint, iscntrl, isascii: character	classification. /isdigit, isalnum, isspace,	ctype(3S)
uuclean: uucp spool directory	clean-up.	uuclean(1M)
clri:	clear inode.	clri(1M)
	clearer: stream error reset.	clearer:o(3S)
ferror, feof,	clearerr, fileno: stream status inquiries.	ferror(3S)
startrek:	clobber klingons.	startrek(1X)
bclk, setbclk: reads and sets the battery	clock.	bclk(1M)
cron:	clock daemon.	cron(1)
close:	close a file.	close(2)
	close: close a file.	close(2)
fclose, fflush:	close or flush a stream.	fclose(3S)
	clri: clear inode.	clri(1M)

	cmp: compare two files.	cmp(1)
	cmpfs: compare and archive file systems.	cmpfs(1M)
SCCS delta.	cmt: insert the delta commentary for an initial	cmt(1S)
	cnvtime, gtime: convert string to internal time.	cnvtime(3C)
	col: filter reverse line-feeds.	col(1)
	comb: combine SCCS deltas.	comb(1S)
comb:	combine SCCS deltas.	comb(1S)
files.	comm: select or reject lines common to two sorted	comm(1)
nice: run a	command at specified priority.	nice(1)
chroot: change root directory for a	command.	chroot(1M)
env: set environment for	command execution.	env(1)
uux: unix to unix	command execution.	uux(1C)
nohup: run a	command immune to hangups.	nohup(1)
rsh: restricted shell	(command interpreter).	rsh(1)
pcstat: report statistics on output of getpc	command.	pcstat(1)
sh, rsh: shell, the standard/restricted	command programming language.	sh(1)
spr: special print	command.	spr(1)
system: issue a shell	command.	system(3S)
test: condition evaluation	command.	test(1)
time: time a	command.	time(1)
xargs: construct argument lists and execute	command.	xargs(1)
intro: introduction to	commands and application programs.	intro(1)
at: execute	commands at a later time.	at(1)
install: install	commands.	install(1M)
powerfail:	commands to be executed following powerfail.	powerfail(5)
cmt: insert the delta	commentary for an initial SCCS delta.	cmt(1S)
comm: select or reject lines	common to two sorted files.	comm(1)
telinit: user	communication with init.	telinit(1M)
diff: differential file	comparator.	diff(1)
cmpfs:	compare and archive file systems.	cmpfs(1M)
cmp:	compare two files.	cmp(1)
sccsdiff:	compare two versions of an SCCS file.	sccsdiff(1S)
diff3: 3-way differential file	comparison.	diff3(1)
dircmp: directory	comparison.	dircmp(1)
tcmp: text	comparison for crash dump.	tcmp(1M)
regexp: regular expression	compile and match routines.	regexp(7)
cc, pcc: C	compiler.	cc(1)
f77: FORTRAN 77	compiler.	f77(1)
vpmc:	compiler for the virtual protocol machine.	vpmc(1C)
occ: old C	compiler.	occ(1)
yacc: yet another	compiler-compiler.	yacc(1)
bs: a	compiler/interpreter for modest-sized programs.	bs(1)
wait: await	completion of process.	wait(1)
festoon: turgid memorandum	composition.	festoon(1)
pack:	compress files.	pack(1)
pcat: expand	compressed file to standard output.	pcat(1)
unpack: expand	compressed files.	unpack(1)
cat:	concatenate and print files.	cat(1)
test:	condition evaluation command.	test(1)
mkconf: create	configuration table and low core.	mkconf(1M)
mpx, join, chan, extract, attach, detach,	connect, npgrp, ckill, mpxcall: create and/	mpx(2)
conns:	connect to a remote system.	conns(3C)
vpmset, vpmstart:	connect VPM drivers and KMCs; load the KMC11-B..	vpmset(1C)
L.sys: table of	connecting uucp systems.	L.sys(5)
	conns: connect to a remote system.	conns(3C)
check: file system	consistency check and repair.	check(1M)
lfcheck: logical file system (LFS)	consistency check and repair.	lfcheck(1)
dcheck: file system directory	consistency check.	dcheck(1M)
chkold: file system	consistency chkold.	chkold(1M)
Unix Error Messages: description of UNIX	console messages.	uemess(6)
getcsw: read	console switches.	getcsw(2)
mkfs:	construct a file system.	mkfs(1M)
mkfst:	construct a file system on mag tape.	mkfst(1M)
mklfs:	construct a Logical File System (LFS).	mklfs(1)
xargs:	construct argument lists and execute command.	xargs(1)
deroff: remove nroff, troff, tbl and eqn	constructs.	deroff(1)
ls: list	contents of directory.	ls(1)
ioctl:	control device.	ioctl(2)
fcntl: file	control.	fcntl(2)
init: process	control initialization.	init(1M)
nc: network	control.	nc(4)
reboot: transfer	control to DEC rom and reboot.	reboot(2)
uustat: uucp status inquiry and job	control.	uustat(1C)
ltod: double precision integer to floating point	conversion. dtol.	dtol:o(3C)
ecvt, fcvt: output	conversion.	ecvt(3C)
itol: integer to long integer	conversion.	itol:o(3C)

libil: CB UNIX Release 1	Conversion Library.	libil(3X)
locv: long output	conversion.	locv:o(3C)
ltod: double precision integer to floating point	conversion.	ltod:o(3C)
ltoi: long integer to integer	conversion.	ltoi:o(3C)
printf, fprintf, sprintf: formatted output	conversion.	printf(3S)
units:	conversion program.	units(1)
scanf, fscanf, sscanf: formatted input	conversion.	scanf(3S)
dd:	convert and copy a file.	dd(1)
arcv:	convert archives to new format.	arcv(1M)
atgex:	convert ascii file to GEX format.	atgex(1G)
atof, atoi, atol:	convert ASCII to numbers.	atof(3C)
atof, atoi, atol:	convert ASCII to numbers.	atof:o(3C)
cspeed:	convert baud to speed number.	cspeed(3C)
l3tol, ltol3:	convert between 3-byte integers and long integers.	l3tol(3C)
a64l, l64a:	convert between long and base-64 ASCII.	a64l(3C)
ctime, localtime, gmtime, asctime, timezone:	convert date and time to ASCII.	ctime(3C)
ctime:	convert date and time to ASCII.	ctime:o(3C)
gcon:	convert GEX file to HIS format.	gcon(1G)
cnvtime, gtime:	convert string to internal time.	cnvtime(3C)
dd: convert and	copy a file.	dd(1)
cpio:	copy file archives in and out.	cpio(1)
cp, ln, mv:	copy, link, or move files.	cp(1)
cpmv:	copy move.	cpmv(1)
uucp, uulog, uuname: unix to unix	copy.	uucp(1C)
sacopy: stand-alone	copy/verify.	sacopy(8)
alloc:	core allocator.	alloc:o(3C)
ucore: turn on or off the unique	core dumping feature..	ucore(1)
ucore: enable/disable unique	core dumping feature..	ucore(2)
calloc, cfree:	core memory allocator.	calloc:o(3S)
malloc, free:	core memory allocator.	malloc:o(3C)
mem:	core memory.	mem(4)
mkconf: create configuration table and low	core.	mkconf(1M)
functions. sin,	cos, tan, asin, acos, atan, atan2: trigonometric	trig(3M)
sinh,	cosh, tanh: hyperbolic functions.	sinh(3M)
sum: sum and	count blocks in a file.	sum(1)
nargs: argument	count.	nargs:o(3C)
wc: word	count.	wc(1)
getpc: get Program	Counter data on running processes.	getpc(1)
pcs: program	counter sampling device.	pcs(4)
	cp, ln, mv: copy, link, or move files.	cp(1)
cpio: format of	cpio archive.	cpio(5)
	cpio: copy file archives in and out.	cpio(1)
	cpio: format of cpio archive.	cpio(5)
	cpmv: copy move.	cpmv(1)
dead:	crash analysis.	dead(1M)
tcmp: text comparison for	crash dump.	tcmp(1M)
stack: stack trace from	crash file.	stack(1)
	crash: what to do when the system crashes.	crash(6)
crash: what to do when the system	crashes.	crash(6)
	creat: create a new file.	creat(2)
tmpnam:	create a name for a temporary file.	tmpnam(3S)
creat:	create a new file.	creat(2)
pipe:	create a pipe.	pipe(2)
lcall, vcall:	create and execute a new process.	call:o(3C)
attach, detach, connect, npgrp, ckill, mpxcall:	create and manipulate multiplexed files. /extract,	mpx(2)
mkconf:	create configuration table and low core.	mkconf(1M)
umask: set and get	creation mask.	umask(2)
	cref: make cross reference listing.	cref(1)
	cron: clock daemon.	cron(1)
executed..	crontab: table of chronological events to be	crontab(5)
xref:	cross reference for C programs.	xref(1)
cref: make	cross reference listing.	cref(1)
vcrt: filter nroff output for virtual	crts.	vcrt(1)
	crypt: encode/decode.	crypt(1)
	crypt, setkey, encrypt: DES encryption.	crypt(3C)
	cspeed: convert baud to speed number.	cspeed(3C)
	ct: call terminal.	ct(1C)
	ctermid: generate file name for terminal.	ctermid(3S)
	ctime: convert date and time to ASCII.	ctime:o(3C)
convert date and time to ASCII.	ctime, localtime, gmtime, asctime, timezone:	ctime(3C)
	cu: call another UNIX system.	cu(1C)
	cubic: three dimensional tic-tac-toe.	cubic(1)
lnxx: return name of	current terminal.	lnxx:o(3C)
uname: get name of	current UNIX system.	uname(2)
uname: print name of	current UNIX.	uname(1)
reboot: replace	current UNIX with new program or system.	reboot(1M)

ttyslot: find the slot in the utmp file of the	current user.	ttyslot(3C)
spline: interpolate smooth	curve.	spline(1G)
	cuserid: character user ID.	cuserid(3S)
file.	cut: cut out selected fields of each line of a	cut(1)
cut:	cut out selected fields of each line of a file.	cut(1)
cron: clock	daemon.	cron(1)
errdemon: error-logging	daemon.	errdemon(1M)
lpdata: decode line printer	data files (printers and qmap).	lpdata(3C)
lfsync: update modified LFS	data.	lfsync(1)
getpc: get Program Counter	data on running processes.	getpc(1)
prof: display profile	data.	prof(1)
lfupdate: update modified LFS	data repetitively.	lfupdate(1)
stat:	data returned by stat system call.	stat(7)
types: primitive system	data types.	types(7)
join: relational	database operator.	join(1)
ftime: get	date and time.	ftime(2)
time: get	date and time.	time(2)
localtime, gmtime, asctime, timezone: convert	date and time to ASCII. ctime,	ctime(3C)
ctime: convert	date and time to ASCII.	ctime:o(3C)
date: print and set the	date.	date(1)
epoch: print and set system backup	date.	epoch(1M)
mdate: set modified	date on file.	mdate:o(2)
	date: print and set the date.	date(1)
savdate: save and restore modification	date.	savdate(1)
	dc: desk calculator.	dc(1)
	dcheck: file system directory consistency check.	dcheck(1M)
	dd: convert and copy a file.	dd(1)
	dead: crash analysis.	dead(1M)
adb:	debugger.	adb(1)
reboot: transfer control to	DEC rom and reboot.	reboot(2)
lpdata:	decode line printer data files (printers and qmap).	lpdata(3C)
printers:	defines printer options to /etc/lpd.	printers(5)
dsw:	delete interactively.	dsw(1)
basename, dirname:	deliver portions of pathnames.	basename(1)
tail:	deliver the last part of a file.	tail(1)
gdelta:	delta a file from SCCS.	gdelta(1S)
delta: make a	delta (change) to an SCCS file.	delta(1S)
chghist: change the history entry of an SCCS	delta.	chghist(1S)
insert the delta commentary for an initial SCCS	delta. cmt:	cmt(1S)
cmt: insert the	delta commentary for an initial SCCS delta.	cmt(1S)
rmdel: remove a	delta from an SCCS file.	rmdel(1S)
	delta: make a delta (change) to an SCCS file.	delta(1S)
comb: combine SCCS	deltas.	comb(1S)
init: reinitialize line printer	demon.	init(1)
mesg: permit or	deny messages.	mesg(1)
constructs.	deroff: remove nroff, troff, tbl and eqn	deroff(1)
crypt, setkey, encrypt:	DES encryption.	crypt(3C)
intro: file format	description.	intro(5)
Unix Error Messages:	description of UNIX console messages.	uemess(6)
dup: duplicate an open file	descriptor.	dup(2)
dc:	desk calculator.	dc(1)
manipulate/ mpx, join, chan, extract, attach,	detach, connect, npgrp, ckill, mpxcall: create and	mpx(2)
sps:	detail process status.	sps(1)
access:	determine accessibility of file.	access(2)
file:	determine file type.	file(1)
bdump: read from block	device.	bdump(1M)
bload: write on block	device.	bload(1M)
ioctl: control	device.	ioctl(2)
pcs: program counter sampling	device.	pcs(4)
L-devices: auto-dialer	device table.	L-devices(5)
	df: disk free.	df(1)
	dh, dz: asynchronous multiplexers.	dh(4)
d_passwd:	dial up password file.	d_passwd(5)
L-dialcodes: uucp system	dialcodes.	L-dialcodes(5)
ratfor: rational FORTRAN	dialect.	ratfor(1)
dialups: list of	dialup lines.	dialups(5)
	dialups: list of dialup lines.	dialups(5)
wait: wait for process to	die.	wait(2)
gdiff:	diff an SCCS file with named file.	gdiff(1S)
bdiff: big	diff.	bdiff(1)
	diff: differential file comparator.	diff(1)
	diff3: 3-way differential file comparison.	diff3(1)
sdiff: side-by-side	difference program.	sdiff(1)
diffmk: mark	differences between files.	diffmk(1)
diff:	differential file comparator.	diff(1)
diff3: 3-way	differential file comparison.	diff3(1)

	diffmk: mark differences between files.	diffmk(1)
cubic: three	dimensional tic-tac-toe.	cubic(1)
	dir: format of directories.	dir(5)
	dircmp: directory comparison.	dircmp(1)
dir: format of	directories.	dir(5)
rm, rmdir: remove files or	directories.	rm(1)
sccsclean: remove unwanted files in SCCS	directories.	sccsclean(1S)
chdir, cd: change working	directory.	chdir(1)
chdir: change working	directory.	chdir(2)
chroot: change root	directory.	chroot(2)
uuclean: uucp spool	directory clean-up.	uuclean(1M)
dircmp:	directory comparison.	dircmp(1)
dcheck: file system	directory consistency check.	dcheck(1M)
unlink: remove	directory entry.	unlink(2)
chroot: change root	directory for a command.	chroot(1M)
ls: list contents of	directory.	ls(1)
mkdir: make a	directory.	mkdir(1)
mkdir: make	directory.	mkdir:o(3C)
mvdir: move a	directory.	mvdir(1M)
pwd: working	directory name.	pwd(1)
mknod: make a	directory or a special file.	mknod(2)
rmdir: remove	directory.	rmdir:o(3C)
gls: list the	directory $SCCSOURCE with input args appended.	gls(1S)
basename,	dirname: deliver portions of pathnames.	basename(1)
getty: set terminal type, modes, speed, and line	discipline.	getty(1M)
lfs: format of Logical File System	disk area.	lfs(5)
df:	disk free.	df(1)
hp: RP04/RP05/RP06 moving-head	disk.	hp(4)
inode: find inode on	disk.	inode(1)
rk?: RK11/RK03 or RK05	disk.	rk(4)
rx?: floppy	disk.	rx(4)
ldrxboot: load floppy	disk second level boot.	ldrxboot(1)
du: summarize	disk usage.	du(1)
operations. maus, getmaus, freemaus, enabmaus,	dismaus, switmaus: multiple access user space	maus(2)
umount:	dismount file system.	umount(1)
umount:	dismount file system.	umount(2)
prof:	display profile data.	prof(1)
hypot: euclidean	distance.	hypot(3M)
ldiv: long	division.	ldiv:o(3C)
kl: KL-11 or	DL-11 asynchronous interface.	kl(4)
	dmplfs: dump logical file system to tape.	dmplfs(1)
	dn: DN-11 ACU interface.	dn(4)
dn:	DN-11 ACU interface.	dn(4)
mm: type out	documents that use the PWB/MM macros.	mm(1)
whodo: who is	doing what.	whodo(1M)
conversion. dtol, ltod:	double precision integer to floating point	dtol:o(3C)
conversion. ltod:	double precision integer to floating point	ltod:o(3C)
	d_passwd: dial up password file.	d_passwd(5)
graph:	draw a graph.	graph(1G)
trace: event-tracing	driver.	trace(4)
vpmset, vpmstart: connect VPM	drivers and KMCs; load the KMC11-B..	vpmset(1C)
	dsw: delete interactively.	dsw(1)
point conversion.	dtol, ltod: double precision integer to floating	dtol:o(3C)
	du: summarize disk usage.	du(1)
idump:	dump an inode.	idump(1M)
errdead: extract error records from	dump.	errdead(1M)
	dump: incremental dump tape format.	dump(5)
dmplfs:	dump logical file system to tape.	dmplfs(1)
mhdump: incremental file system	dump.	mhdump(1M)
od: octal	dump.	od(1)
bd: binary	dump of a file.	bd(1)
dump: incremental	dump tape format.	dump(5)
tcmp: text comparison for crash	dump.	tcmp(1M)
ucore: turn on or off the unique core	dumping feature..	ucore(1)
ucore: enable/disable unique core	dumping feature..	ucore(2)
	dup: duplicate an open file descriptor.	dup(2)
dup:	duplicate an open file descriptor.	dup(2)
dh,	dz: asynchronous multiplexers.	dh(4)
echo, fecho:	echo arguments.	echo(1)
	echo, fecho: echo arguments.	echo(1)
sccstring:	echo SCCS keywords to standard output.	sccstring(1S)
	ecvt, fcvt: output conversion.	ecvt(3C)
	ed: text editor.	ed(1)
end, etext,	edata: last locations in program.	end(3C)
fed:	edit associative memory for form letter.	fed(1)
ed: text	editor.	ed(1)

ld: link	editor.	ld(1)
a.out: assembler and link	editor output.	a.out(5)
sed: stream	editor.	sed(1)
grep.	egrep, fgrep: search a file for a pattern.	grep(1)
ucore:	enable/disable unique core dumping feature..	ucore(2)
space operations. maus, getmaus, freemaus,	enabmaus, dismaus, switmaus: multiple access user	maus(2)
turbo:	encabulator.	turbo(1)
crypt:	encode/decode.	crypt(1)
crypt, setkey,	encrypt: DES encryption.	crypt(3C)
crypt, setkey, encrypt: DES	encryption.	crypt(3C)
	end, etext, edata: last locations in program.	end(3C)
getgrent, getgrgid, getgrnam, setgrent,	endgrent: get group file entry.	getgrent(3C)
getpwent, getpwuid, getpwnam, setpwent,	endpwent: get password file entry.	getpwent(3C)
getutent, getutid, getutline, pututline, setutent,	endutent, utmpname: access utmp file entry.	getut(3C)
nlist: get	entries from name list.	nlist(3C)
nlist: get	entries from name list.	nlist:o(3C)
utmp, wtmp: utmp and wtmp	entry formats.	utmp(5)
getgrnam, setgrent, endgrent: get group file	entry. getgrent, getgrgid,	getgrent(3C)
getpwnam, setpwent, endpwent: get password file	entry. getpwent, getpwuid,	getpwent(3C)
setutent, endutent, utmpname: access utmp file	entry. getutent, getutid, getutline, pututline,	getut(3C)
chghist: change the history	entry of an SCCS delta.	chghist(1S)
putpwent: write password file	entry.	putpwent(3C)
unlink: remove directory	entry.	unlink(2)
	env: set environment for command execution.	env(1)
	environ: user environment.	environ(7)
profile: setting up an	environment at login time.	profile(5)
environ: user	environment.	environ(7)
env: set	environment for command execution.	env(1)
getenv: value for	environment name.	getenv(3C)
	epoch: print and set system backup date.	epoch(1M)
deroff: remove nroff, troff, tbl and	eqn constructs.	deroff(1)
	eqn, neqn, checkeq: typeset mathematical text.	eqn(1)
	err: error-logging interface.	err(4)
	errdead: extract error records from dump.	errdead(1M)
	errdemon: error-logging daemon.	errdemon(1M)
	errfile: error-log file format.	errfile(5)
perror, sys_errlist, sys_nerr,	errno: system error messages.	perror(3C)
messages. Unix	Error Messages: description of UNIX console	uemess(6)
perror, sys_errlist, sys_nerr, errno: system	error messages.	perror(3C)
errdead: extract	error records from dump.	errdead(1M)
clearer: stream	error reset.	clearer:o(3S)
errfile:	error-log file format.	errfile(5)
errdemon:	error-logging daemon.	errdemon(1M)
err:	error-logging interface.	err(4)
errpt: process a report of logged	errors.	errpt(1M)
spell, spellin, spellout: find spelling	errors.	spell(1)
	errpt: process a report of logged errors.	errpt(1M)
plot: openpl	et al.: graphics interface.	plot(3X)
printers: defines printer options to	/etc/lpd.	printers(5)
end,	etext, edata: last locations in program.	end(3C)
hypot:	euclidean distance.	hypot(3M)
expr:	evaluate arguments as an expression.	expr(1)
test: condition	evaluation command.	test(1)
vpmsnap, vpmtrace, vpmfmt: save and print VPM	event traces. vpmsave,	vpmsave(1C)
crontab: table of chronological	events to be executed..	crontab(5)
trace:	event-tracing driver.	trace(4)
	exec, execl, execv, exect: execute a file.	exec:o(2)
execute a file.	execl, execv, execle, execve, execlp, execvp:	exec(2)
exec,	execl, execv, exect: execute a file.	exec:o(2)
execl, execv,	execle, execve, execlp, execvp: execute a file.	exec(2)
execl, execv, execle, execve,	execlp, execvp: execute a file.	exec(2)
exec, execl, execv,	exect: execute a file.	exec:o(2)
execl, execv, execle, execve, execlp, execvp:	execute a file.	exec(2)
exec, execl, execv, exect:	execute a file.	exec:o(2)
lcall, vcall: create and	execute a new process.	call:o(3C)
xargs: construct argument lists and	execute command.	xargs(1)
at:	execute commands at a later time.	at(1)
reset:	execute non-local goto.	reset:o(3C)
setpgrp:	execute program with new process group.	setpgrp(1)
exprog: perform standard Shell	execute sequence.	exprog:o(3C)
crontab: table of chronological events to be	executed..	crontab(5)
powerfail: commands to be	executed following powerfail.	powerfail(5)
env: set environment for command	execution.	env(1)
sleep: suspend	execution for an interval.	sleep(1)
sleep: stop	execution for interval.	sleep(3C)
monitor: prepare	execution profile.	monitor(3C)

profil:	execution time user profile.	profil(2)
uux: unix to unix command	execution.	uux(1C)
file. execl,	execv, execle, execve, execlp, execvp: execute a	exec(2)
exec, execl,	execv, exect: execute a file.	exec:o(2)
execl, execv, execle,	execve, execlp, execvp: execute a file.	exec(2)
execl, execv, execle, execve, execlp,	execvp: execute a file.	exec(2)
gex: Graphic	EXerciser for Tektronix 4014.	gex(1G)
	exit: terminate process.	exit(2)
square root.	exp, log, pow, sqrt: exponential, logarithm, power,	exp(3M)
pcat:	expand compressed file to standard output.	pcat(1)
unpack:	expand compressed files.	unpack(1)
intro: introduction to UNIX system	explanations.	intro(6)
frexp, ldexp, modf: split into mantissa and	exponent.	frexp(3C)
exp, log, pow, sqrt:	exponential, logarithm, power, square root.	exp(3M)
	expr: evaluate arguments as an expression.	expr(1)
regexp: regular	expression compile and match routines.	regexp(7)
expr: evaluate arguments as an	expression.	expr(1)
	exprog: perform standard Shell execute sequence.	exprog:o(3C)
greek: graphics for	extended TTY-37 type-box.	greek(7)
mpxcall: create and manipulate/ mpx, join, chan,	extract, attach, detach, connect, npgrp, ckill,	mpx(2)
errdead:	extract error records from dump.	errdead(1M)
	f77: FORTRAN 77 compiler.	f77(1)
remainder functions. floor,	fabs, ceil, fmod: absolute value, floor, ceiling,	floor(3M)
factor, primes:	factor a number, generate large primes.	factor(1)
primes.	factor, primes: factor a number, generate large	factor(1)
true,	false: provide truth values.	true(1)
abort: generate an IOT	fault.	abort(3C)
	fclose, fflush: close or flush a stream.	fclose(3S)
	fcntl: file control.	fcntl(2)
ecvt,	fcvt: output conversion.	ecvt(3C)
fopen, freopen,	fdopen: open a stream.	fopen(3S)
ucore: turn on or off the unique core dumping	feature..	ucore(1)
ucore: enable/disable unique core dumping	feature..	ucore(2)
echo,	fecho: echo arguments.	echo(1)
	fed: edit associative memory for form letter.	fed(1)
ferror,	feof, clearerr, fileno: stream status inquiries.	ferror(3S)
inquiries.	ferror, feof, clearerr, fileno: stream status	ferror(3S)
	festoon: turgid memorandum composition.	festoon(1)
fclose,	fflush: close or flush a stream.	fclose(3S)
getc, getchar,	fgetc, getw: get character or word from stream.	getc(3S)
gets,	fgets: get a string from a stream.	gets(3S)
grep, egrep,	fgrep: search a file for a pattern.	grep(1)
access: determine accessibility of	file.	access(2)
acct: accounting	file.	acct(5)
addsccs: add SCCS keywords to a	file.	addsccs(1S)
move: move a	file and set the mode.	move(1)
cpio: copy	file archives in and out.	cpio(1)
bd: binary dump of a	file.	bd(1)
chmod: change mode of	file.	chmod(1)
chmod: change mode of	file.	chmod(2)
chown, chgrp: change owner of group of a	file.	chown(1)
chown: change owner and group of a	file.	chown(2)
close: close a	file.	close(2)
diff: differential	file comparator.	diff(1)
diff3: 3-way differential	file comparison.	diff3(1)
fcntl:	file control.	fcntl(2)
creat: create a new	file.	creat(2)
cut: cut out selected fields of each line of a	file.	cut(1)
dd: convert and copy a	file.	dd(1)
delta: make a delta (change) to an SCCS	file.	delta(1S)
dup: duplicate an open	file descriptor.	dup(2)
	file: determine file type.	file(1)
d_passwd: dial up password	file.	d_passwd(5)
getgrgid, getgrnam, setgrent, endgrent: get group	file entry. getgrent,	getgrent(3C)
getpwnam, setpwent, endpwent: get password	file entry. getpwent, getpwuid,	getpwent(3C)
setutent, endutent, utmpname: access utmp	file entry. /getutid, getutline, pututline,	getut(3C)
putpwent: write password	file entry.	putpwent(3C)
execv, execle, execve, execlp, execvp: execute a	file. execl,	exec(2)
exec, execl, execv, exect: execute a	file.	exec:o(2)
grep, egrep, fgrep: search a	file for a pattern.	grep(1)
ar: archive	file format.	ar(5)
intro:	file format description.	intro(5)
errfile: error-log	file format.	errfile(5)
nar: archive (library)	file format.	nar(5)
gdelta: delta a	file from SCCS.	gdelta(1S)
gget: get a	file from SCCS.	gget(1S)

fstat: get status of open	file.	fstat(2)
gdiff: diff an SCCS file with named	file.	gdiff(1S)
gdump: prints a gex graphic	file.	gdump(1G)
get: get a version of an SCCS	file.	get(1S)
g_find: locate and identify a source	file.	g_find(1S)
group: group	file.	group(5)
gadmin: admin a	file in SCCS.	gadmin(1S)
gprt: prt a	file in SCCS.	gprt(1S)
split: split a	file into pieces.	split(1)
issue: issue identification	file.	issue(5)
link: link to a	file.	link(2)
mdate: set modified date on	file.	mdate:o(2)
mknod: build special	file.	mknod(1M)
mknod: make a directory or a special	file.	mknod(2)
mktmp: make a temporary	file.	mktmp:o(3C)
ctermid: generate	file name for terminal.	ctermid(3S)
mktemp: make a unique	file name.	mktemp(3C)
mktemp: make temporary	file name.	mktemp:o(3C)
null: the null	file.	null(4)
ttyslot: find the slot in the utmp	file of the current user.	ttyslot(3C)
tell: get	file offset.	tell(3C)
tell: get	file offset.	tell:o(2)
passwd: password	file.	passwd(5)
lines of several files or subsequent lines of one	file. paste: merge same	paste(1)
pr: print	file.	pr(1)
prs: print an SCCS	file.	prs(1S)
prt: print SCCS	file.	prt(1S)
read: read from	file.	read(2)
reform: reformat text	file.	reform(1S)
rmdel: remove a delta from an SCCS	file.	rmdel(1S)
sccsdiff: compare two versions of an SCCS	file.	sccsdiff(1S)
sccsfile: format of SCCS	file.	sccsfile(5)
size: size of an object	file.	size(1)
stack: stack trace from crash	file.	stack(1)
stat, fstat: get	file status.	stat(2)
stat: get	file status.	stat:o(2)
sum: sum and count blocks in a	file.	sum(1)
check:	file system consistency check and repair.	check(1M)
chkold:	file system consistency chkold.	chkold(1M)
dcheck:	file system directory consistency check.	dcheck(1M)
lfs: format of Logical	File System disk area.	lfs(5)
mhdump: incremental	file system dump.	mhdump(1M)
rstlfs: restore logical	file system from tape.	rstlfs(1)
lfcheck: logical	file system (LFS) consistency check and repair.	lfcheck(1)
lfmount: mount logical	file system (LFS).	lfmount(1)
lfumount: unmount the logical	file system (LFS).	lfumount(1)
mklfs: construct a Logical	File System (LFS).	mklfs(1)
mkfs: construct a	file system.	mkfs(1M)
mount: mount	file system.	mount(1)
mount, umount: mount or remove	file system.	mount(2)
mkfst: construct a	file system on mag tape.	mkfst(1M)
lfs: Logical	File System operations.	lfs(3C)
quot: summarize	file system ownership.	quot(1M)
mhrestor: incremental	file system restore.	mhrestor(1M)
rootdev: root	file system.	rootdev(4)
mtab: mounted	file system table.	mtab(5)
dmplfs: dump logical	file system to tape.	dmplfs(1)
umount: dismount	file system.	umount(1)
umount: dismount	file system.	umount(2)
updfs: update	file system.	updfs(1M)
cmpfs: compare and archive	file systems.	cmpfs(1M)
tail: deliver the last part of a	file.	tail(1)
tkdump: prints a Tektronix	file.	tkdump(1G)
tmpnam: create a name for a temporary	file.	tmpnam(3S)
hex: translate binary	file to ascii hexadecimal.	hex(1)
unhex: translate hexed	file to binary.	unhex(1)
atgex: convert ascii	file to GEX format.	atgex(1G)
gcon: convert GEX	file to HIS format.	gcon(1G)
gadd: add a	file to SCCS.	gadd(1S)
pcat: expand compressed	file to standard output.	pcat(1)
touch: change modification time of a	file.	touch(1)
file: determine	file type.	file(1)
uniq: report repeated lines in a	file.	uniq(1)
utime: update times in	file.	utime(2)
utindx, utline: access routines for utmp	file.	utindx(3C)
val: validate SCCS	file.	val(1S)

gdiff: diff an SCCS	file with named file.	gdiff(1S)
write: write on a	file.	write(2)
	file_log: log an input string in a logfie..	fiie_log(1S)
ferror, feof, clearerr,	fileno: stream status inquiries.	ferror(3S)
admin: administer SCCS	files.	admin(1S)
cat: concatenate and print	files.	cat(1)
cmp: compare two	files.	cmp(1)
comm: select or reject lines common to two sorted	files.	comm(1)
cp, ln, mv: copy, link, or move	files.	cp(1)
diffmk: mark differences between	files.	diffmk(1)
find: find	files.	find(1)
gmark: mark a subsystem of SCCS	files..	gmark(1S)
hatch: filter to hatch GEX	files.	hatch(1G)
sccsclean: remove unwanted	files in SCCS directories.	sccsclean(1S)
gsplit: filter to break gex	files into pieces.	gsplit(1G)
intro: introduction to special	files.	intro(4)
ckill, mpxcall: create and manipulate multiplexed	files. /extract, attach, detach, connect, npgrp,	mpx(2)
rm, rmdir: remove	files or directories.	rm(1)
paste: merge same lines of several	files or subsequent lines of one file.	paste(1)
pack: compress	files.	pack(1)
lpdata: decode line printer data	files (printers and qmap).	lpdata(3C)
sort: sort or merge	files.	sort(1)
unpack: expand compressed	files.	unpack(1)
what: identify	files.	what(1S)
	filesystem: format of system volume.	fs(5)
vcrt:	filter nroff output for virtual crts.	vcrt(1)
col:	filter reverse line-feeds.	col(1)
gsplit:	filter to break gex files into pieces.	gsplit(1G)
hatch:	filter to hatch GEX files.	hatch(1G)
plot: graphics	filters.	plot(1G)
tek, vplot, t300, t300s, t450: graphics	filters.	tek(1)
find:	find files.	find(1)
	find: find files.	find(1)
hyphen:	find hyphenated words.	hyphen(1)
inode:	find inode on disk.	inode(1)
ttyname, isatty:	find name of a terminal.	ttyname(3C)
lorder:	find ordering relation for an object library.	lorder(1)
typo:	find possible typos.	typo(1)
spell, spellin, spellout:	find spelling errors.	spell(1)
ttyslot:	find the slot in the utmp file of the current user.	ttyslot(3C)
tee: pipe	fitting.	tee(1)
dtol, ltod: double precision integer to	floating point conversion.	dtol:o(3C)
ltod: double precision integer to	floating point conversion.	ltod:o(3C)
fpemul:	floating point interpreter.	fpemul(3C)
	flog: speed up a process.	flog(1)
floor, fabs, ceil, fmod: absolute value,	floor, ceiling, remainder functions.	floor(3M)
ceiling, remainder functions.	floor, fabs, ceil, fmod: absolute value, floor,	floor(3M)
rx?:	floppy disk.	rx(4)
ldrxboot: load	floppy disk second level boot.	ldrxboot(1)
fclose, fflush: close or	flush a stream.	fclose(3S)
functions. floor, fabs, ceil,	fmod: absolute value, floor, ceiling, remainder	floor(3M)
powerfail: commands to be executed	following powerfail.	powerfail(5)
	fopen, freopen, fdopen: open a stream.	fopen(3S)
	fopen, freopen: open a stream.	fopen:o(3S)
	fork: spawn new process.	fork(2)
	form: form letter generator.	form(1)
fed: edit associative memory for	form letter.	fed(1)
form:	form letter generator.	form(1)
ar: archive file	format.	ar(5)
nar: new	format archive and library maintainer.	nar(1)
arcv: convert archives to new	format.	arcv(1M)
atgex: convert ascii file to GEX	format.	atgex(1G)
intro: file	format description.	intro(5)
dump: incremental dump tape	format.	dump(5)
errfile: error-log file	format.	errfile(5)
gcon: convert GEX file to HIS	format.	gcon(1G)
nar: archive (library) file	format.	nar(5)
inode:	format of an inode.	inode(5)
cpio:	format of cpio archive.	cpio(5)
dir:	format of directories.	dir(5)
lfs:	format of Logical File System disk area.	lfs(5)
sccsfile:	format of SCCS file.	sccsfile(5)
filesystem:	format of system volume.	fs(5)
nroff, troff:	format or typeset text.	nroff(1)
tbl:	format tables for nroff or troff.	tbl(1)
nnroff:	format text.	nnroff(1)

tp: magnetic tape	format.	tp(5)
utmp, wtmp: utmp and wtmp entry	formats.	utmp(5)
scanf, fscanf, sscanf:	formatted input conversion.	scanf(3S)
printf, fprintf, sprintf:	formatted output conversion.	printf(3S)
printf:	formatted print.	printf:o(3C)
f77:	FORTRAN 77 compiler.	f77(1)
ratfor: rational	FORTRAN dialect.	ratfor(1)
	fpemul: floating point interpreter.	fpemul(3C)
printf,	fprintf, sprintf: formatted output conversion.	printf(3S)
putc, putchar,	fputc, putw: put character or word on a stream.	putc(3S)
puts,	fputs: put a string on a stream.	puts(3S)
	fread, fwrite: buffered binary input/output.	fread(3S)
malloc,	free, realloc, calloc: main memory allocator.	malloc(3C)
access user space operations. maus, getmaus,	freemaus, enabmaus, dismaus, switmaus: multiple	maus(2)
fopen,	freopen, fdopen: open a stream.	fopen(3S)
fopen,	freopen: open a stream.	fopen:o(3S)
exponent.	frexp, ldexp, modf: split into mantissa and	frexp(3C)
gets, fgets: get a string	from a stream.	gets(3S)
rmdel: remove a delta	from an SCCS file.	rmdel(1S)
getopt: get option letter	from argv.	getopt(3C)
bdump: read	from block device.	bdump(1M)
stack: stack trace	from crash file.	stack(1)
errdead: extract error records	from dump.	errdead(1M)
read: read	from file.	read(2)
ncheck: generate names	from i-numbers.	ncheck(1M)
nlist: get entries	from name list.	nlist(3C)
nlist: get entries	from name list.	nlist:o(3C)
gdelta: delta a file	from SCCS.	gdelta(1S)
gget: get a file	from SCCS.	gget(1S)
getc, getchar, fgetc, getw: get character or word	from stream.	getc(3S)
rstlfs: restore logical file system	from tape.	rstlfs(1)
getpw: get name	from UID.	getpw(3C)
scanf,	fscanf, sscanf: formatted input conversion.	scanf(3S)
	fseek, ftell, rewind: reposition a stream.	fseek(3S)
stat,	fstat: get file status.	stat(2)
	fstat: get status of open file.	fstat(2)
fseek,	ftell, rewind: reposition a stream.	fseek(3S)
	ftime: get date and time.	ftime(2)
gamma: log gamma	function.	gamma(3M)
j0, j1, jn, y0, y1, yn: bessel	functions.	bessel(3M)
fmod: absolute value, floor, ceiling, remainder	functions. floor, fabs, ceil,	floor(3M)
sinh, cosh, tanh: hyperbolic	functions.	sinh(3M)
cos, tan, asin, acos, atan, atan2: trigonometric	functions. sin,	trig(3M)
fread,	fwrite: buffered binary input/output.	fread(3S)
	gadd: add a file to SCCS.	gadd(1S)
	gadmin: admin a file in SCCS.	gadmin(1S)
moo: guessing	game.	moo(1X)
chess: the	game of chess.	chess(1X)
gamma: log	gamma function.	gamma(3M)
	gamma: log gamma function.	gamma(3M)
6000.	gcat: send phototypesetter output to the HONEYWELL	gcat(1C)
	gcon: convert GEX file to HIS format.	gcon(1G)
	gdelta: delta a file from SCCS.	gdelta(1S)
	gdiff: diff an SCCS file with named file.	gdiff(1S)
	gdump: prints a gex graphic file.	gdump(1G)
abort:	generate an IOT fault.	abort(3C)
ctermid:	generate file name for terminal.	ctermid(3S)
factor, primes: factor a number,	generate large primes.	factor(1)
ncheck:	generate names from i-numbers.	ncheck(1M)
lex:	generate programs for simple lexical tasks.	lex(1)
form: form letter	generator.	form(1)
rand, srand: random number	generator.	rand(3C)
	getc: buffered input.	getc:o(3C)
from stream.	getc, getchar, fgetc, getw: get character or word	getc(3S)
stream. getc,	getchar, fgetc, getw: get character or word from	getc(3S)
	getchar: read character.	getchar:o(3C)
	getcsw: read console switches.	getcsw(2)
getuid, getgid, geteuid,	getegid: get user and group identity.	getuid(2)
	getenv: value for environment name.	getenv(3C)
getuid, getgid,	geteuid, getegid: get user and group identity.	getuid(2)
	getgid: get group identification.	getgid:o(2)
identity. getuid,	getgid, geteuid, getegid: get user and group	getuid(2)
get group file entry.	getgrent, getgrgid, getgrnam, setgrent, endgrent:	getgrent(3C)
file entry. getgrent,	getgrgid, getgrnam, setgrent, endgrent: get group	getgrent(3C)
getgrent, getgrgid,	getgrnam, setgrent, endgrent: get group file entry.	getgrent(3C)
	getlogin: get login name.	getlogin(3C)

multiple access user space operations. maus,	getmaus, freemaus, enabmaus, dismaus, switmaus:	maus(2)
	getopt: get option letter from argv.	getopt(3C)
	getpass: read a password.	getpass(3C)
pcstat: report statistics on output of	getpc command.	pcstat(1)
processes.	getpc: get Program Counter data on running	getpc(1)
	getpid, getppid: get process identification.	getpid(2)
getpid,	getppid: get process identification.	getpid(2)
	getpw: get name from UID.	getpw(3C)
get password file entry.	getpwent, getpwuid, getpwnam, setpwent, endpwent:	getpwent(3C)
entry. getpwent, getpwuid,	getpwnam, setpwent, endpwent: get password file	getpwent(3C)
password file entry. getpwent,	getpwuid, getpwnam, setpwent, endpwent: get	getpwent(3C)
	gets, fgets: get a string from a stream.	gets(3S)
gettydefs: speed and terminal settings used by	getty.	gettydefs(5)
discipline.	getty: set terminal type, modes, speed, and line	getty(1M)
getty.	gettydefs: speed and terminal settings used by	gettydefs(5)
	getu: get selected user block information.	getu(2)
	getuid: get user identification.	getuid:o(2)
group identity.	getuid, getgid, geteuid, getegid: get user and	getuid(2)
endutent, utmpname: access utmp file entry.	getutent, getutid, getutline, pututline, setutent,	getut(3C)
utmpname: access utmp file entry. getutent,	getutid, getutline, pututline, setutent, endutent,	getut(3C)
access utmp file entry. getutent, getutid,	getutline, pututline, setutent, endutent, utmpname:	getut(3C)
getc, getchar, fgetc,	getw: get character or word from stream.	getc(3S)
gcon: convert	GEX file to HIS format.	gcon(1G)
hatch: filter to hatch	GEX files.	hatch(1G)
gsplit: filter to break	gex files into pieces.	gsplit(1G)
atgex: convert ascii file to	GEX format.	atgex(1G)
	gex: Graphic EXerciser for Tektronix 4014.	gex(1G)
gdump: prints a	gex graphic file.	gdump(1G)
	g_find: locate and identify a source file.	g_find(1S)
	gget: get a file from SCCS.	gget(1S)
appended.	gls: list the directory $SCCSOURCE with input args	gls(1S)
	gmark: mark a subsystem of SCCS files..	gmark(1S)
ASCII. ctime, localtime,	gmtime, asctime, timezone: convert date and time to	ctime(3C)
reset: execute non-local	goto.	reset:o(3C)
setjmp, longjmp: non-local	goto.	setjmp(3C)
	gprt: prt a file in SCCS.	gprt(1S)
	graph: draw a graph.	graph(1G)
graph: draw a	graph.	graph(1G)
gex:	Graphic EXerciser for Tektronix 4014.	gex(1G)
gdump: prints a gex	graphic file.	gdump(1G)
plot:	graphics filters.	plot(1G)
tek, vplot, t300, t300s, t450:	graphics filters.	tek(1)
greek:	graphics for extended TTY-37 type-box.	greek(7)
plot: openpl et al.:	graphics interface.	plot(3X)
plot:	graphics interface.	plot(5)
vt:	graphics interface.	vt(4)
	greek: graphics for extended TTY-37 type-box.	greek(7)
	grep, egrep, fgrep: search a file for a pattern.	grep(1)
chgrp: change	group.	chgrp:o(2)
getgrgid, getgrnam, setgrent, endgrent: get	group file entry. getgrent,	getgrent(3C)
group:	group file.	group(5)
	group: group file.	group(5)
id: print user and	group id.	id(1)
setgid: set process	group ID.	setgid(2)
getgid: get	group identification.	getgid:o(2)
getuid, getgid, geteuid, getegid: get user and	group identity.	getuid(2)
kill: send a signal to a process or process	group.	kill(1)
newgrp: log in to a new	group.	newgrp(1)
chown, chgrp: change owner of	group of a file.	chown(1)
chown: change owner and	group of a file.	chown(2)
setpgrp: execute program with new process	group.	setpgrp(1)
setpgrp: set process	group.	setpgrp(2)
make: maintain program	groups.	make(1)
ssignal,	gsignal: software signals.	ssignal(3C)
	gsplit: filter to break gex files into pieces.	gsplit(1G)
cnvtime,	gtime: convert string to internal time.	cnvtime(3C)
	gtty: get terminal line options.	gtty(1)
stty,	gtty: set and retrieve terminal modes.	stty:o(2)
stty,	gtty: set and retrieve the modes of a terminal.	stty(3C)
moo:	guessing game.	moo(1X)
nohup: run a command immune to	hangups.	nohup(1)
	hatch: filter to hatch GEX fiies.	hatch(1G)
hatch: filter to	hatch GEX files.	hatch(1G)
banner: make	headlines.	banner(1)
	help: ask for help.	help(1S)
help: ask for	help.	help(1S)

	hex: translate binary file to ascii hexadecimal.	hex(1)
hex: translate binary file to ascii	hexadecimal.	hex(1)
unhex: translate	hexed file to binary.	unhex(1)
hmul:	high-order product.	hmul:o(3C)
	hmul: high-order product.	hmul:o(3C)
	hold: suspend printing of queued line printer jobs.	hold(1)
gcat: send phototypesetter output to the	HONEYWELL 6000.	gcat(1C)
	hp: RP04/RP05/RP06 moving-head disk.	hp(4)
wump:	hunt the wumpus.	wump(1X)
sinh, cosh, tanh:	hyperbolic functions.	sinh(3M)
	hyphen: find hyphenated words.	hyphen(1)
hyphen: find	hyphenated words.	hyphen(1)
	hypot: euclidean distance.	hypot(3M)
cuserid: character user	ID.	cuserid(3S)
id: print user and group	id.	id(1)
	id: print user and group id.	id(1)
setgid: set process group	ID.	setgid(2)
setuid: set process user	ID.	setuid(2)
issue: issue	identification file.	issue(5)
getgid: get group	identification.	getgid:o(2)
getpid, getppid: get process	identification.	getpid(2)
getuid: get user	identification.	getuid:o(2)
line: get line	identification.	line(1)
g_find: locate and	identify a source file.	g_find(1S)
what:	identify files.	what(1S)
getgid, geteuid, getegid: get user and group	identity. getuid,	getuid(2)
	idump: dump an inode.	idump(1M)
signal: catch or	ignore signals.	signal(2)
nohup: run a command	immune to hangups.	nohup(1)
dump:	incremental dump tape format.	dump(5)
mhdump:	incremental file system dump.	mhdump(1M)
mhrestor:	incremental file system restore.	mhrestor(1M)
ptx: permuted	index.	ptx(1)
	indir: indirect system call.	indir(2)
indir:	indirect system call.	indir(2)
	infect: Give a virus to another UNIX system.	infect(1)
	init: process control initialization.	init(1M)
inittab: script for the	init process.	inittab(5)
	init: reinitialize line printer demon.	init(1)
telinit: user communication with	init.	telinit(1M)
cmt: insert the delta commentary for an	initial SCCS delta.	cmt(1S)
init: process control	initialization.	init(1M)
popen, pclose:	initiate I/O to/from a process.	popen(3S)
	inittab: script for the init process.	inittab(5)
clri: clear	inode.	clri(1M)
	inode: find inode on disk.	inode(1)
	inode: format of an inode.	inode(5)
idump: dump an	inode.	idump(1M)
inode: format of an	inode.	inode(5)
inode: find	inode on disk.	inode(1)
gls: list the directory $SCCSOURCE with	input args appended.	gls(1S)
scanf, fscanf, sscanf: formatted	input conversion.	scanf(3S)
getc: buffered	input.	getc:o(3C)
ungetc: push character back into	input stream.	ungetc(3S)
file_log: log an	input string in a logfile..	file_log(1S)
fread, fwrite: buffered binary	input/output.	fread(3S)
stdio: standard buffered	input/output package.	stdio(3S)
stdio: standard buffered	input/output package.	stdio:o(3S)
ferror, feof, clearerr, fileno: stream status	inquiries.	ferror(3S)
uustat: uucp status	inquiry and job control.	uustat(1C)
delta. cmt:	insert the delta commentary for an initial SCCS	cmt(1S)
install:	install commands.	install(1M)
	install: install commands.	install(1M)
link. x25pvc, x25lnk:	install, remove, or get status for a PVC or BX.25	x25pvc(1C)
abs:	integer absolute value.	abs(3C)
itol: integer to long	integer conversion.	itol:o(3C)
ltoi: long integer to	integer conversion.	ltoi:o(3C)
dtol, ltod: double precision	integer to floating point conversion.	dtol:o(3C)
ltod: double precision	integer to floating point conversion.	ltod:o(3C)
ltoi: long	integer to integer conversion.	ltoi:o(3C)
itol:	integer to long integer conversion.	itol:o(3C)
l3tol, ltol3: convert between 3-byte	integers and long integers.	l3tol(3C)
ltol3: convert between 3-byte integers and long	integers. l3tol,	l3tol(3C)
dsw: delete	interactively.	dsw(1)
dn: DN-11 ACU	interface.	dn(4)
err: error-logging	interface.	err(4)

mo, mo90, nmo, nmo90: nroff, nnroff mm	interface for preprinted letterhead.	mo(1)
tty: general	interface for terminals.	tty(4)
kl: KL-11 or DL-11 asynchronous	interface.	kl(4)
mt?: TE16/TU16 magnetic tape	interface.	mt(4)
plot: openpl et al.: graphics	interface.	plot(3X)
plot: graphics	interface.	plot(5)
tm?: TM11/TU10 magnetic tape	interface.	tm(4)
vpm, vpb: Virtual Protocol Machine Protocol and	Interface.	vpm(4)
vt: graphics	interface.	vt(4)
X25: BX.25 network	interface.	x25(4)
cnvtime, gtime: convert string to	internal time.	cnvtime(3C)
spline:	interpolate smooth curve.	spline(1G)
fpemul: floating point	interpreter.	fpemul(3C)
rsh: restricted shell (command	interpreter).	rsh(1)
sno: SNOBOL	interpreter.	sno(1)
sleep: suspend execution for an	interval.	sleep(1)
sleep: stop execution for	interval.	sleep(3C)
	intro: file format description.	intro(5)
programs.	intro: introduction to commands and application	intro(1)
	intro: introduction to miscellany.	intro(7)
	intro: introduction to special files.	intro(4)
	intro: introduction to stand-alone utilities.	intro(8)
	intro: introduction to subroutines and libraries.	intro(3)
	intro: introduction to system calls.	intro(2)
	intro: introduction to UNIX system explanations.	intro(6)
intro:	introduction to commands and application programs.	intro(1)
intro:	introduction to miscellany.	intro(7)
intro:	introduction to special files.	intro(4)
intro:	introduction to stand-alone utilities.	intro(8)
intro:	introduction to subroutines and libraries.	intro(3)
intro:	introduction to system calls.	intro(2)
intro:	introduction to UNIX system explanations.	intro(6)
ncheck: generate names from	i-numbers.	ncheck(1M)
iostat: report	I/O and system statistics.	iostat(1M)
mpxio: multiplexed	I/O.	mpxio(5)
popen, pclose: initiate	I/O to/from a process.	popen(3S)
	ioctl: control device.	ioctl(2)
	iostat: report I/O and system statistics.	iostat(1M)
abort: generate an	IOT fault.	abort(3C)
isascii:/ isalpha, isupper, islower, isdigit,	isalnum, isspace, ispunct, isprint, iscntrl,	ctype(3S)
isspace, ispunct, isprint, iscntrl, isascii:/	isalpha, isupper, islower, isdigit, isalnum,	ctype(3S)
isalnum, isspace, ispunct, isprint, iscntrl,	isascii: character classification. /isdigit,	ctype(3S)
ttyname,	isatty: find name of a terminal.	ttyname(3C)
/isdigit, isalnum, isspace, ispunct, isprint,	iscntrl, isascii: character classification.	ctype(3S)
iscntrl, isascii:/ isalpha, isupper, islower,	isdigit, isalnum, isspace, ispunct, isprint,	ctype(3S)
isprint, iscntrl, isascii:/ isalpha, isupper,	islower, isdigit, isalnum, isspace, ispunct,	ctype(3S)
islower, isdigit, isalnum, isspace, ispunct,	isprint, iscntrl, isascii: character/ /isupper,	ctype(3S)
/isupper, islower, isdigit, isalnum, isspace,	ispunct, isprint, iscntrl, isascii: character/	ctype(3S)
isalpha, isupper, islower, isdigit, isalnum,	isspace, ispunct, isprint, iscntrl, isascii:/	ctype(3S)
system:	issue a shell command.	system(3S)
issue:	issue identification file.	issue(5)
	issue: issue identification file.	issue(5)
ispunct, isprint, iscntrl, isascii:/ isalpha,	isupper, islower, isdigit, isalnum, isspace,	ctype(3S)
news: print news	items.	news(1)
	itol: integer to long integer conversion.	itol:o(3C)
	j0, j1, jn, y0, y1, yn: bessel functions.	bessel(3M)
j0,	j1, jn, y0, y1, yn: bessel functions.	bessel(3M)
j0, j1,	jn, y0, y1, yn: bessel functions.	bessel(3M)
npgrp, ckill, mpxcall: create and manipulate/ mpx,	join, chan, extract, attach, detach, connect,	mpx(2)
	join: relational database operator.	join(1)
	kasb: assembler for the KMC11 microprocessor.	kasb(1)
addsccs: add SCCS	keywords to a file.	addsccs(1S)
sccstring: echo SCCS	keywords to standard output.	sccstring(1S)
	kill: send a signal to a process or process group.	kill(1)
	kill: send signal to a process.	kill(2)
	kl: KL-11 or DL-11 asynchronous interface.	kl(4)
kl:	KL-11 or DL-11 asynchronous interface.	kl(4)
startrek: clobber	klingons.	startrek(1X)
	kmc: KMC11/DMC11 microprocessor.	kmc(4)
kasb: assembler for the	KMC11 microprocessor.	kasb(1)
vpmstart: connect VPM drivers and KMCs; load the	KMC11-B.. vpmset,	vpmset(1C)
kmc:	KMC11/DMC11 microprocessor.	kmc(4)
kunb: un-assembler for the	KMC11/DMC11 microprocessor.	kunb(1)
vpmset, vpmstart: connect VPM drivers and	KMCs; load the KMC11-B..	vpmset(1C)
microprocessor.	kunb: un-assembler for the KMC11/DMC11	kunb(1)
long integers.	l3tol, ltol3: convert between 3-byte integers and	l3tol(3C)

a64l,	l64a: convert between long and base-64 ASCII.	a64l(3C)
awk: pattern scanning and processing	language.	awk(1)
bc: arbitrary-precision arithmetic	language.	bc(1)
shell, the standard/restricted command programming	language. sh, rsh:	sh(1)
	lcall, vcall: create and execute a new process.	call:o(3C)
	ld: link editor.	ld(1)
	L-devices: auto-dialer device table.	L-devices(5)
frexp,	ldexp, modf: split into mantissa and exponent.	frexp(3C)
	L-dialcodes: uucp system dialcodes.	L-dialcodes(5)
	ldiv: long division.	ldiv:o(3C)
	ldrxboot: load floppy disk second level boot.	ldrxboot(1)
fed: edit associative memory for form	letter.	fed(1)
getopt: get option	letter from argv.	getopt(3C)
form: form	letter generator.	form(1)
nmo90: nroff, nnroff mm interface for preprinted	letterhead. mo, mo90, nmo,	mo(1)
ldrxboot: load floppy disk second	level boot.	ldrxboot(1)
	lex: generate programs for simple lexical tasks.	lex(1)
lex: generate programs for simple	lexical tasks.	lex(1)
check and repair.	lfcheck: logical file system (LFS) consistency	lfcheck(1)
	lfmount: mount logical file system (LFS).	lfmount(1)
lfcheck: logical file system	(LFS) consistency check and repair.	lfcheck(1)
lfsync: update modified	LFS data.	lfsync(1)
lfupdate: update modified	LFS data repetitively.	lfupdate(1)
	lfs: format of Logical File System disk area.	lfs(5)
lfmount: mount logical file system	(LFS).	lfmount(1)
lfumount: unmount the logical file system	(LFS).	lfumount(1)
	lfs: Logical File System operations.	lfs(3C)
mklfs: construct a Logical File System	(LFS).	mklfs(1)
	lfsync: update modified LFS data.	lfsync(1)
	lfumount: unmount the logical file system (LFS).	lfumount(1)
	lfupdate: update modified LFS data repetitively.	lfupdate(1)
	lib7: Version 7 library.	lib7(3X)
	libil: CB UNIX Release 1 Conversion Library.	libil(3X)
intro: introduction to subroutines and	libraries.	intro(3)
nar: archive	(library) file format.	nar(5)
lib7: Version 7	library.	lib7(3X)
libil: CB UNIX Release 1 Conversion	Library.	libil(3X)
lorder: find ordering relation for an object	library.	lorder(1)
ar: archive and	library maintainer.	ar(1)
nar: new format archive and	library maintainer.	nar(1)
getty: set terminal type, modes, speed, and	line discipline.	getty(1M)
	line: get line identification.	line(1)
line: get	line identification.	line(1)
cut: cut out selected fields of each	line of a file.	cut(1)
gtty: get terminal	line options.	gtty(1)
lpdata: decode	line printer data files (printers and qmap).	lpdata(3C)
init: reinitialize	line printer demon.	init(1)
abort: remove previously queued	line printer jobs.	abort(1)
hold: suspend printing of queued	line printer jobs.	hold(1)
release: restore printing of queued	line printer jobs.	release(1)
restrain: suspend printing of queued	line printer jobs.	restrain(1)
start: restore printing of queued	line printer jobs.	start(1)
lp:	line printer.	lp(4)
lpropen: open pipe to the	line printer.	lpropen(3C)
lpr:	line printer spooling program.	lpr(1)
readl: read one	line.	readl(1)
col: filter reverse	line-feeds.	col(1)
comm: select or reject	lines common to two sorted files.	comm(1)
dialups: list of dialup	lines.	dialups(5)
uniq: report repeated	lines in a file.	uniq(1)
merge same lines of several files or subsequent	lines of one file. paste:	paste(1)
file. paste: merge same	lines of several files or subsequent lines of one	paste(1)
ld:	link editor.	ld(1)
a.out: assembler and	link editor output.	a.out(5)
	link: link to a file.	link(2)
cp, ln, mv: copy,	link, or move files.	cp(1)
link:	link to a file.	link(2)
install, remove, or get status for a PVC or BX.25	link. x25pvc, x25lnk:	x25pvc(1C)
	lint: a C program verifier.	lint(1)
ls:	list contents of directory.	ls(1)
uunames:	list names of UNIX systems known to uucp.	uunames(1C)
nlist: get entries from name	list.	nlist(3C)
nlist: get entries from name	list.	nlist:o(3C)
nm: print name	list.	nm(1)
dialups:	list of dialup lines.	dialups(5)
appended. gls:	list the directory $SCCSOURCE with input args	gls(1S)

talk: allow user to	listen and talk to one or more other users.	talk(1)
cref: make cross reference	listing.	cref(1)
xargs: construct argument	lists and execute command.	xargs(1)
cp,	ln, mv: copy, link, or move files.	cp(1)
	lnxx: return name of current terminal.	lnxx:o(3C)
ldrxboot:	load floppy disk second level boot.	ldrxboot(1)
	load: load.	load(1M)
load:	load.	load(1M)
vpmset, vpmstart: connect VPM drivers and KMCs;	load the KMC11-B..	vpmset(1C)
and time to ASCII. ctime,	localtime, gmtime, asctime, timezone: convert date	ctime(3C)
g_find:	locate and identify a source file.	g_find(1S)
swapdev:	location for swapping.	swapdev(4)
end, etext, edata: last	locations in program.	end(3C)
plock:	lock process or text in memory.	plock(2)
sema, p, v, test, post, block, setsem, rdsem,	lock, unlock, tlock, noulk: semaphore operations.	sema(2)
	locv: long output conversion.	locv:o(3C)
file_log:	log an input string in a logfile..	file_log(1S)
gamma:	log gamma function.	gamma(3M)
newgrp:	log in to a new group.	newgrp(1)
square root. exp,	log, pow, sqrt: exponential, logarithm, power,	exp(3M)
exp, log, pow, sqrt: exponential,	logarithm, power, square root.	exp(3M)
file_log: log an input string in a	logfile..	file_log(1S)
errpt: process a report of	logged errors.	errpt(1M)
lfs: format of	Logical File System disk area.	lfs(5)
rstlfs: restore	logical file system from tape.	rstlfs(1)
repair. lfcheck:	logical file system (LFS) consistency check and	lfcheck(1)
lfmount: mount	logical file system (LFS).	lfmount(1)
lfumount: unmount the	logical file system (LFS).	lfumount(1)
mklfs: construct a	Logical File System (LFS).	mklfs(1)
lfs:	Logical File System operations.	lfs(3C)
dmplfs: dump	logical file system to tape.	dmplfs(1)
ac:	login accounting.	ac(1)
getlogin: get	login name.	getlogin(3C)
passwd: change	login password.	passwd(1)
	login: sign on.	login(1)
profile: setting up an environment at	login time.	profile(5)
a64l, l64a: convert between	long and base-64 ASCII.	a64l(3C)
ldiv:	long division.	ldiv:o(3C)
itol: integer to	long integer conversion.	itol:o(3C)
ltoi:	long integer to integer conversion.	ltoi:o(3C)
l3tol, ltol3: convert between 3-byte integers and	long integers.	l3tol(3C)
locv:	long output conversion.	locv:o(3C)
setjmp,	longjmp: non-local goto.	setjmp(3C)
library.	lorder: find ordering relation for an object	lorder(1)
mkconf: create configuration table and	low core.	mkconf(1M)
	lp: line printer.	lp(4)
and qmap).	lpdata: decode line printer data files (printers	lpdata(3C)
	lpr: line printer spooling program.	lpr(1)
	lpropen: open pipe to the line printer.	lpropen(3C)
	ls: list contents of directory.	ls(1)
	lseek: move read/write pointer.	lseek(2)
	L.sys: table of connecting uucp systems.	L.sys(5)
conversion. dtol,	ltod: double precision integer to floating point	dtol:o(3C)
conversion.	ltod: double precision integer to floating point	ltod:o(3C)
	ltoi: long integer to integer conversion.	ltoi:o(3C)
integers. l3tol,	ltol3: convert between 3-byte integers and long	l3tol(3C)
	m4: macro processor.	m4(1)
vpm, vpb: Virtual Protocol	Machine Protocol and Interface.	vpm(4)
vpmc: compiler for the virtual protocol	machine.	vpmc(1C)
m4:	macro processor.	m4(1)
mm: type out documents that use the PWB/MM	macros.	mm(1)
manmac:	macros to print CB-UNIX manual sections.	manmac(5)
mkfst: construct a file system on	mag tape.	mkfst(1M)
tp:	magnetic tape format.	tp(5)
mt?: TE16/TU16	magnetic tape interface.	mt(4)
tm?: TM11/TU10	magnetic tape interface.	tm(4)
mtm:	magnetic tape manipulation.	mtm(1)
mail, rmail: send mail to users or read	mail.	mail(1)
	mail, rmail: send mail to users or read mail.	mail(1)
mail, rmail: send	mail to users or read mail.	mail(1)
malloc, free, realloc, calloc:	main memory allocator.	malloc(3C)
make:	maintain program groups.	make(1)
ar: archive and library	maintainer.	ar(1)
nar: new format archive and library	maintainer.	nar(1)
delta:	make a delta (change) to an SCCS file.	delta(1S)
mkdir:	make a directory.	mkdir(1)

mknod:	make a directory or a special file.	mknod(2)
mktmp:	make a temporary file.	mktmp:o(3C)
mktemp:	make a unique file name.	mktemp(3C)
cref:	make cross reference listing.	cref(1)
mkdir:	make directory.	mkdir:o(3C)
banner:	make headlines.	banner(1)
	make: maintain program groups.	make(1)
mkpt:	make proto.	mkpt(1M)
mktemp:	make temporary file name.	mktemp:o(3C)
	malloc, free: core memory allocator.	malloc:o(3C)
allocator.	malloc, free, realloc, calloc: main memory	malloc(3C)
	man: print pages of this manual.	man(1)
mmtest: PDP 11/70 memory	management test.	mmtest(8)
detach, connect, npgrp, ckill, mpxcall: create and	manipulate multiplexed files. /extract, attach,	mpx(2)
tp:	manipulate tape archive.	tp(1)
mtm: magnetic tape	manipulation.	mtm(1)
	manmac: macros to print CB-UNIX manual sections.	manmac(5)
frexp, ldexp, modf: split into	mantissa and exponent.	frexp(3C)
man: print pages of this	manual.	man(1)
manmac: macros to print CB-UNIX	manual sections.	manmac(5)
ascii:	map of ASCII character set.	ascii(7)
qmap: queue to printers	map.	qmap(5)
gmark:	mark a subsystem of SCCS files..	gmark(1S)
diffmk:	mark differences between files.	diffmk(1)
umask: set and get creation	mask.	umask(2)
regexp: regular expression compile and	match routines.	regexp(7)
eqn, neqn, checkeq: typeset	mathematical text.	eqn(1)
switmaus: multiple access user space operations.	maus, getmaus, freemaus, enabmaus, dismaus,	maus(2)
menab, mdisab, msend, mrecv,	mctl: send and receive messages.	message(2)
	mdate: set modified date on file.	mdate:o(2)
messages. menab,	mdisab, msend, mrecv, mctl: send and receive	message(2)
tm:	meditate.	tm(1)
	mem: core memory.	mem(4)
festoon: turgid	memorandum composition.	festoon(1)
break, brk, sbrk: change	memory allocation.	break(2)
calloc, cfree: core	memory allocator.	calloc:o(3S)
malloc, free, realloc, calloc: main	memory allocator.	malloc(3C)
malloc, free: core	memory allocator.	malloc:o(3C)
fed: edit associative	memory for form letter.	fed(1)
mmtest: PDP 11/70	memory management test.	mmtest(8)
mem: core	memory.	mem(4)
smcreat, smopen, smclose, smget, smput: shared	memory operations.	shmem(2)
plock: lock process or text in	memory.	plock(2)
messages.	menab, mdisab, msend, mrecv, mctl: send and receive	message(2)
sort: sort or	merge files.	sort(1)
lines of one file. paste:	merge same lines of several files or subsequent	paste(1)
	mesg: permit or deny messages.	mesg(1)
/send, sendw, recv, recvw, msgstat, msgctl: old	message veneer for sending and receiving messages..	msg(3)
Unix Error	Messages: description of UNIX console messages.	uemess(6)
mesg: permit or deny	messages.	mesg(1)
menab, mdisab, msend, mrecv, mctl: send and receive	messages.	message(2)
old message veneer for sending and receiving	messages.. /sendw, recv, recvw, msgstat, msgctl:	msg(3)
perror, sys_errlist, sys_nerr, errno: system error	messages.	perror(3C)
Unix Error Messages: description of UNIX console	messages.	uemess(6)
	mhdump: incremental file system dump.	mhdump(1M)
	mhrestor: incremental file system restore.	mhrestor(1M)
	mhstty: set the options for a terminal.	mhstty(1)
kasb: assembler for the KMC11	microprocessor.	kasb(1)
kmc: KMC11/DMC11	microprocessor.	kmc(4)
kunb: un-assembler for the KMC11/DMC11	microprocessor.	kunb(1)
	mkconf: create configuration table and low core.	mkconf(1M)
	mkdir: make a directory.	mkdir(1)
	mkdir: make directory.	mkdir:o(3C)
	mkfs: construct a file system.	mkfs(1M)
	mkfst: construct a file system on mag tape.	mkfst(1M)
	mklfs: construct a Logical File System (LFS).	mklfs(1)
	mknod: build special file.	mknod(1M)
	mknod: make a directory or a special file.	mknod(2)
	mkpt: make proto.	mkpt(1M)
	mktemp: make a unique file name.	mktemp(3C)
	mktemp: make temporary file name.	mktemp:o(3C)
	mktmp: make a temporary file.	mktmp:o(3C)
mo, mo90, nmo, nmo90: nroff, nnroff	mm interface for preprinted letterhead.	mo(1)
	mm: type out documents that use the PWB/MM macros.	mm(1)
	mmkdir: made path names.	mmkdir(1)
	mmtest: PDP 11/70 memory management test.	mmtest(8)

for preprinted letterhead.	mo, mo90, nmo, nmo90: nroff, nnroff mm interface	mo(1)
preprinted letterhead. mo,	mo90, nmo, nmo90: nroff, nnroff mm interface for	mo(1)
move: move a file and set the	mode.	move(1)
chmod: change	mode of file.	chmod(1)
chmod: change	mode of file.	chmod(2)
stty, gtty: set and retrieve the	modes of a terminal.	stty(3C)
getty: set terminal type,	modes, speed, and line discipline.	getty(1M)
stty, gtty: set and retrieve terminal	modes.	stty:o(2)
bs: a compiler/interpreter for	modest-sized programs.	bs(1)
frexp, ldexp,	modf: split into mantissa and exponent.	frexp(3C)
savdate: save and restore	modification date.	savdate(1)
touch: change	modification time of a file.	touch(1)
mdate: set	modified date on file.	mdate:o(2)
lfsync: update	modified LFS data.	lfsync(1)
lfupdate: update	modified LFS data repetitively.	lfupdate(1)
	monitor: prepare execution profile.	monitor(3C)
uusub:	monitor uucp network.	uusub(1M)
	moo: guessing game.	moo(1X)
mount:	mount file system.	mount(1)
lfmount:	mount logical file system (LFS).	lfmount(1)
	mount: mount file system.	mount(1)
mount, umount:	mount or remove file system.	mount(2)
mountpts: general user	mount point table.	mountpts(5)
	mount, umount: mount or remove file system.	mount(2)
mtab:	mounted file system table.	mtab(5)
	mountpts: general user mount point table.	mountpts(5)
mvdir:	move a directory.	mvdir(1M)
move:	move a file and set the mode.	move(1)
cpmv: copy	move.	cpmv(1)
cp, ln, mv: copy, link, or	move files.	cp(1)
	move: move a file and set the mode.	move(1)
lseek:	move read/write pointer.	lseek(2)
seek:	move read/write pointer.	seek:o(2)
hp: RP04/RP05/RP06	moving-head disk.	hp(4)
npgrp, ckill, mpxcall: create and manipulate/	mpx, join, chan, extract, attach, detach, connect,	mpx(2)
/extract, attach, detach, connect, npgrp, ckill,	mpxcall: create and manipulate multiplexed files.	mpx(2)
	mpxio: multiplexed I/O.	mpxio(5)
menab, mdisab, msend,	mrecv, mctl: send and receive messages.	message(2)
menab, mdisab,	msend, mrecv, mctl: send and receive messages.	message(2)
msgstat, msgctl: old message veneer for sending/	msg, msgenab, msgdisab, send, sendw, recv, recvw,	msg(3)
/msgdisab, send, sendw, recv, recvw, msgstat,	msgctl: old message veneer for sending and/	msg(3)
msgctl: old message veneer for/ msg, msgenab,	msgdisab, send, sendw, recv, recvw, msgstat,	msg(3)
msgstat, msgctl: old message veneer for/ msg,	msgenab, msgdisab, send, sendw, recv, recvw,	msg(3)
msg, msgenab, msgdisab, send, sendw, recv, recvw,	msgstat, msgctl: old message veneer for sending and/	msg(3)
	mt?: TE16/TU16 magnetic tape interface.	mt(4)
	mtab: mounted file system table.	mtab(5)
	mtm: magnetic tape manipulation.	mtm(1)
getmaus, freemaus, enabmaus, dismaus, switmaus:	multiple access user space operations. maus,	maus(2)
npgrp, ckill, mpxcall: create and manipulate	multiplexed files. /attach, detach, connect,	mpx(2)
mpxio:	multiplexed I/O.	mpxio(5)
dh, dz: asynchronous	multiplexers.	dh(4)
cp, ln,	mv: copy, link, or move files.	cp(1)
	mvdir: move a directory.	mvdir(1M)
gdiff: diff an SCCS file with	named file.	gdiff(1S)
pipe:	named pipe.	pipe(4)
	nar: archive (library) file format.	nar(5)
	nar: new format archive and library maintainer.	nar(1)
	nargs: argument count.	nargs:o(3C)
	nc: network control.	nc(4)
	ncheck: generate names from i-numbers.	ncheck(1M)
eqn,	neqn, checkeq: typeset mathematical text.	eqn(1)
nc:	network control.	nc(4)
X25: BX.25	network interface.	x25(4)
uusub: monitor uucp	network.	uusub(1M)
	newgrp: log in to a new group.	newgrp(1)
news: print	news items.	news(1)
	news: print news items.	news(1)
	nice: run a command at specified priority.	nice(1)
	nice: set program priority.	nice(2)
	nlist: get entries from name list.	nlist(3C)
	nlist: get entries from name list.	nlist:o(3C)
	nm: print name list.	nm(1)
preprinted letterhead. mo, mo90,	nmo, nmo90: nroff, nnroff mm interface for	mo(1)
letterhead. mo, mo90, nmo,	nmo90: nroff, nnroff mm interface for preprinted	mo(1)
	nnroff: format text.	nnroff(1)
mo, mo90, nmo, nmo90: nroff,	nnroff mm interface for preprinted letterhead.	mo(1)

	nohup: run a command immune to hangups.	nohup(1)
reset: execute	non-local goto.	reset:o(3C)
setjmp, longjmp:	non-local goto.	setjmp(3C)
post, block, setsem, rdsem, lock, unlock, tlock,	noulk: semaphore operations. sema, p, v, test,	sema(2)
mpx, join, chan, extract, attach, detach, connect,	npgrp, ckill, mpxcall: create and manipulate/	mpx(2)
letterhead. mo, mo90, nmo, nmo90:	nroff, nnroff mm interface for preprinted	mo(1)
tbl: format tables for	nroff or troff.	tbl(1)
vcrt: filter	nroff output for virtual crts.	vcrt(1)
	nroff, troff: format or typeset text.	nroff(1)
deroff: remove	nroff, troff, tbl and eqn constructs.	deroff(1)
null: the	null file.	null(4)
	null: the null file.	null(4)
factor, primes: factor a	number, generate large primes.	factor(1)
size: size of an	object file.	size(1)
lorder: find ordering relation for an	object library.	lorder(1)
	occ: old C compiler.	occ(1)
od:	octal dump.	od(1)
	od: octal dump.	od(1)
tell: get file	offset.	tell(3C)
tell: get file	offset.	tell:o(2)
acct: turn accounting	on/off.	acct(2)
accton: turn accounting	on/off.	accton(1)
sprofil: turn	on/off system profiling.	sprofil(2)
fopen, freopen, fdopen:	open a stream.	fopen(3S)
fopen, freopen:	open a stream.	fopen:o(3S)
dup: duplicate an	open file descriptor.	dup(2)
fstat: get status of	open file.	fstat(2)
open:	open for reading or writing.	open(2)
	open: open for reading or writing.	open(2)
lpropen:	open pipe to the line printer.	lpropen(3C)
plot:	openpl et al.: graphics interface.	plot(3X)
lfs: Logical File System	operations.	lfs(3C)
dismaus, switmaus: multiple access user space	operations. maus, getmaus, freemaus, enabmaus,	maus(2)
sema: semaphore	operations.	sema(1)
rdsem, lock, unlock, tlock, noulk: semaphore	operations. sema, p, v, test, post, block, setsem,	sema(2)
smopen, smclose, smget, smput: shared memory	operations. smcreat,	shmem(2)
strcpy, strncpy, strlen, strchr, strrchr: string	operations. strcat, strncat, strcmp, strncmp,	string(3C)
join: relational database	operator.	join(1)
over: overstrike	optimizer.	over(1)
getopt: get	option letter from argv.	getopt(3C)
mhstty: set the	options for a terminal.	mhstty(1)
gtty: get terminal line	options.	gtty(1)
stty: set teletype	options.	stty(1)
printers: defines printer	options to /etc/lpd.	printers(5)
lorder: find	ordering relation for an object library.	lorder(1)
a.out: assembler and link editor	output.	a.out(5)
ecvt, fcvt:	output conversion.	ecvt(3C)
locv: long	output conversion.	locv:o(3C)
printf, fprintf, sprintf: formatted	output conversion.	printf(3S)
vcrt: filter nroff	output for virtual crts.	vcrt(1)
pcstat: report statistics on	output of getpc command.	pcstat(1)
pcat: expand compressed file to standard	output.	pcat(1)
putc: buffered	output.	putc:o(3C)
sccstring: echo SCCS keywords to standard	output.	sccstring(1S)
gcat: send phototypesetter	output to the HONEYWELL 6000.	gcat(1C)
over:	overstrike optimizer.	over(1)
chown: change	owner and group of a file.	chown(2)
chown: change	owner.	chown:o(2)
chown, chgrp: change	owner of group of a file.	chown(1)
quot: summarize file system	ownership.	quot(1M)
unlock, tlock, noulk: semaphore operations. sema,	p, v, test, post, block, setsem, rdsem, lock,	sema(2)
	pack: compress files.	pack(1)
stdio: standard buffered input/output	package.	stdio(3S)
stdio: standard buffered input/output	package.	stdio:o(3S)
	padm: program administration system.	padm(1S)
man: print	pages of this manual.	man(1)
tk:	paginator for the Tektronix 4014.	tk(1)
	passwd: change login password.	passwd(1)
	passwd: password file.	passwd(5)
d_passwd: dial up	password file.	d_passwd(5)
getpwuid, getpwnam, setpwent, endpwent: get	password file entry. getpwent,	getpwent(3C)
putpwent: write	password file entry.	putpwent(3C)
passwd:	password file.	passwd(5)
getpass: read a	password.	getpass(3C)
passwd: change login	password.	passwd(1)
subsequent lines of one file.	paste: merge same lines of several files or	paste(1)

mmkdir: made	path names.	mmkdir(1)
basename, dirname: deliver portions of	pathnames.	basename(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(2)
	pcat: expand compressed file to standard output.	pcat(1)
cc,	pcc: C compiler.	cc(1)
popen,	pclose: initiate I/O to/from a process.	popen(3S)
	pcs: program counter sampling device.	pcs(4)
command.	pcstat: report statistics on output of getpc	pcstat(1)
mmtest:	PDP 11/70 memory management test.	mmtest(8)
exprog:	perform standard Shell execute sequence.	exprog:o(3C)
update:	periodically update the super block.	update(1)
mesg:	permit or deny messages.	mesg(1)
ptx:	permuted index.	ptx(1)
messages.	perror, sys_errlist, sys_nerr, errno: system error	perror(3C)
gcat: send	phototypesetter output to the HONEYWELL 6000.	gcat(1C)
tc:	phototypesetter simulator.	tc(1)
gsplit: filter to break gex files into	pieces.	gsplit(1G)
split: split a file into	pieces.	split(1)
	pipe: create a pipe.	pipe(2)
tee:	pipe fitting.	tee(1)
	pipe: named pipe.	pipe(4)
pipe: create a	pipe.	pipe(2)
pipe: named	pipe.	pipe(4)
lpropen: open	pipe to the line printer.	lpropen(3C)
	plock: lock process or text in memory.	plock(2)
	plot: graphics filters.	plot(1G)
	plot: graphics interface.	plot(5)
	plot: openpl et al.: graphics interface.	plot(3X)
lseek: move read/write	pointer.	lseek(2)
seek: move read/write	pointer.	seek:o(2)
	popen, pclose: initiate I/O to/from a process.	popen(3S)
basename, dirname: deliver	portions of pathnames.	basename(1)
noulk: semaphore operations. sema, p, v, test,	post, block, setsem, rdsem, lock, unlock, tlock,	sema(2)
root. exp, log,	pow, sqrt: exponential, logarithm, power, square	exp(3M)
exp, log, pow, sqrt: exponential, logarithm,	power, square root.	exp(3M)
powerfail.	powerfail: commands to be executed following	powerfail(5)
powerfail: commands to be executed following	powerfail.	powerfail(5)
	pr: print file.	pr(1)
dtol, ltod: double	precision integer to floating point conversion.	dtol:o(3C)
ltod: double	precision integer to floating point conversion.	ltod:o(3C)
monitor:	prepare execution profile.	monitor(3C)
mo90, nmo, nmo90: nroff, nnroff mm interface for	preprinted letterhead. mo,	mo(1)
abort: remove	previously queued line printer jobs.	abort(1)
factor,	primes: factor a number, generate large primes.	factor(1)
factor, primes: factor a number, generate large	primes.	factor(1)
types:	primitive system data types.	types(7)
prs:	print an SCCS file.	prs(1S)
epoch:	print and set system backup date.	epoch(1M)
date:	print and set the date.	date(1)
cal:	print calendar.	cal(1)
manmac: macros to	print CB-UNIX manual sections.	manmac(5)
spr: special	print command.	spr(1)
pr:	print file.	pr(1)
cat: concatenate and	print files.	cat(1)
nm:	print name list.	nm(1)
uname:	print name of current UNIX.	uname(1)
news:	print news items.	news(1)
man:	print pages of this manual.	man(1)
printf: formatted	print.	printf:o(3C)
prt:	print SCCS file.	prt(1S)
id:	print user and group id.	id(1)
vpmsave, vpmsnap, vpmtrace, vpmfmt: save and	print VPM event traces.	vpmsave(1C)
lpdata: decode line	printer data files (printers and qmap).	lpdata(3C)
init: reinitialize line	printer demon.	init(1)
abort: remove previously queued line	printer jobs.	abort(1)
hold: suspend printing of queued line	printer jobs.	hold(1)
release: restore printing of queued line	printer jobs.	release(1)
restrain: suspend printing of queued line	printer jobs.	restrain(1)
start: restore printing of queued line	printer jobs.	start(1)
lp: line	printer.	lp(4)
lpropen: open pipe to the line	printer.	lpropen(3C)
printers: defines	printer options to /etc/lpd.	printers(5)
lpr: line	printer spooling program.	lpr(1)
vp: Versatec	printer-plotter.	vp(4)

lpdata: decode line printer data files	(printers and qmap).	lpdata(3C)
	printers: defines printer options to /etc/lpd.	printers(5)
qmap: queue to	printers map.	qmap(5)
	printf: formatted print.	printf:o(3C)
conversion.	printf, fprintf, sprintf: formatted output	printf(3S)
hold: suspend	printing of queued line printer jobs.	hold(1)
release: restore	printing of queued line printer jobs.	release(1)
restrain: suspend	printing of queued line printer jobs.	restrain(1)
start: restore	printing of queued line printer jobs.	start(1)
gdump:	prints a gex graphic file.	gdump(1G)
tkdump:	prints a Tektronix file.	tkdump(1G)
nice: run a command at specified	priority.	nice(1)
nice: set program	priority.	nice(2)
errpt:	process a report of logged errors.	errpt(1M)
lcall, vcall: create and execute a new	process.	call:o(3C)
init:	process control initialization.	init(1M)
exit: terminate	process.	exit(2)
flog: speed up a	process.	flog(1)
fork: spawn new	process.	fork(2)
setgid: set	process group ID.	setgid(2)
kill: send a signal to a process or	process group.	kill(1)
setpgrp: execute program with new	process group.	setpgrp(1)
setpgrp: set	process group.	setpgrp(2)
getpid, getppid: get	process identification.	getpid(2)
inittab: script for the init	process.	inittab(5)
kill: send signal to a	process.	kill(2)
kill: send a signal to a	process or process group.	kill(1)
plock: lock	process or text in memory.	plock(2)
popen, pclose: initiate I/O to/from a	process.	popen(3S)
ps:	process status.	ps(1)
sps: detail	process status.	sps(1)
times: get	process times.	times(2)
wait: wait for	process to die.	wait(2)
ptrace:	process trace.	ptrace(2)
setuid: set	process user ID.	setuid(2)
wait: await completion of	process.	wait(1)
getpc: get Program Counter data on running	processes.	getpc(1)
awk: pattern scanning and	processing language.	awk(1)
m4: macro	processor.	m4(1)
hmul: high-order	product.	hmul:o(3C)
	prof: display profile data.	prof(1)
	profil: execution time user profile.	profil(2)
prof: display	profile data.	prof(1)
monitor: prepare execution	profile.	monitor(3C)
profil: execution time user	profile.	profil(2)
	profile: setting up an environment at login time.	profile(5)
Sprof: system	profile.	sprof(1M)
sprofil: turn on/off system	profiling.	sprofil(2)
sh, rsh: shell, the standard/restricted command	programming language.	sh(1)
mkpt: make	proto.	mkpt(1M)
vpm, vpb: Virtual Protocol Machine	Protocol and Interface.	vpm(4)
vpm, vpb: Virtual	Protocol Machine Protocol and Interface.	vpm(4)
vpmc: compiler for the virtual	protocol machine.	vpmc(1C)
vtp: virtual terminal	protocol.	vtp(4)
true, false:	provide truth values.	true(1)
	prs: print an SCCS file.	prs(1S)
gprt:	prt a file in SCCS.	gprt(1S)
	prt: print SCCS file.	prt(1S)
	ps: process status.	ps(1)
	ptrace: process trace.	ptrace(2)
	ptx: permuted index.	ptx(1)
ungetc:	push character back into input stream.	ungetc(3S)
	putc: buffered output.	putc:o(3C)
on a stream.	putc, putchar, fputc, putw: put character or word	putc(3S)
stream. putc,	putchar, fputc, putw: put character or word on a	putc(3S)
	putchar: write character.	putchar:o(3C)
	putpwent: write password file entry.	putpwent(3C)
	puts, fputs: put a string on a stream.	puts(3S)
utmp file entry. getutent, getutid, getutline,	pututline, setutent, endutent, utmpname: access	getut(3C)
putc, putchar, fputc,	putw: put character or word on a stream.	putc(3S)
x25lnk: install, remove, or get status for a	PVC or BX.25 link. x25pvc,	x25pvc(1C)
mm: type out documents that use the	PWB/MM macros.	mm(1)
	pwd: working directory name.	pwd(1)
decode line printer data files (printers and	qmap). lpdata:	lpdata(3C)
	qmap: queue to printers map.	qmap(5)
	qsort: quicker sort.	qsort(3C)

qmap:	queue to printers map.	qmap(5)
abort: remove previously	queued line printer jobs.	abort(1)
hold: suspend printing of	queued line printer jobs.	hold(1)
release: restore printing of	queued line printer jobs.	release(1)
restrain: suspend printing of	queued line printer jobs.	restrain(1)
start: restore printing of	queued line printer jobs.	start(1)
qsort:	quicker sort.	qsort(3C)
	quot: summarize file system ownership.	quot(1M)
	rand, srand: random number generator.	rand(3C)
rand, srand:	random number generator.	rand(3C)
	ratfor: rational FORTRAN dialect.	ratfor(1)
ratfor:	rational FORTRAN dialect.	ratfor(1)
operations. sema, p, v, test, post, block, setsem,	rdsem, lock, unlock, tlock, noulk: semaphore	sema(2)
getpass:	read a password.	getpass(3C)
getchar:	read character.	getchar:o(3C)
getcsw:	read console switches.	getcsw(2)
bdump:	read from block device.	bdump(1M)
read:	read from file.	read(2)
mail, rmail: send mail to users or	read mail.	mail(1)
readl:	read one line.	readl(1)
	read: read from file.	read(2)
open: open for	reading or writing.	open(2)
	readl: read one line.	readl(1)
bclk, setbclk:	reads and sets the battery clock.	bclk(1M)
lseek: move	read/write pointer.	lseek(2)
seek: move	read/write pointer.	seek:o(2)
malloc, free,	realloc, calloc: main memory allocator.	malloc(3C)
reboot: transfer control to DEC rom and	reboot.	reboot(2)
system.	reboot: replace current UNIX with new program or	reboot(1M)
	reboot: transfer control to DEC rom and reboot.	reboot(2)
menab, mdisab, msend, mrecv, mctl: send and	receive messages.	message(2)
msgstat, msgctl: old message veneer for sending and	receiving messages.. /send, sendw, recv, recvw,	msg(3)
errdead: extract error	records from dump.	errdead(1M)
for sending/ msg, msgenab, msgdisab, send, sendw,	recv, recvw, msgstat, msgctl: old message veneer	msg(3)
sending/ msg, msgenab, msgdisab, send, sendw, recv,	recvw, msgstat, msgctl: old message veneer for	msg(3)
xref: cross	reference for C programs.	xref(1)
cref: make cross	reference listing.	cref(1)
	reform: reformat text file.	reform(1S)
reform:	reformat text file.	reform(1S)
routines.	regexp: regular expression compile and match	regexp(7)
regexp:	regular expression compile and match routines.	regexp(7)
init:	reinitialize line printer demon.	init(1)
comm: select or	reject lines common to two sorted files.	comm(1)
lorder: find ordering	relation for an object library.	lorder(1)
join:	relational database operator.	join(1)
libi1: CB UNIX	Release 1 Conversion Library.	libi1(3X)
jobs.	release: restore printing of queued line printer	release(1)
strip: remove symbols and	relocation bits.	strip(1)
fabs, ceil, fmod: absolute value, floor, ceiling,	remainder functions. floor,	floor(3M)
calendar:	reminder service.	calendar(1)
conns: connect to a	remote system.	conns(3C)
rmdel:	remove a delta from an SCCS file.	rmdel(1S)
unlink:	remove directory entry.	unlink(2)
rmdir:	remove directory.	rmdir:o(3C)
mount, umount: mount or	remove file system.	mount(2)
rm, rmdir:	remove files or directories.	rm(1)
deroff:	remove nroff, troff, tbl and eqn constructs.	deroff(1)
x25pvc, x25lnk: install,	remove, or get status for a PVC or BX.25 link.	x25pvc(1C)
abort:	remove previously queued line printer jobs.	abort(1)
strip:	remove symbols and relocation bits.	strip(1)
sccsclean:	remove unwanted files in SCCS directories.	sccsclean(1S)
check: file system consistency check and	repair.	check(1M)
logical file system (LFS) consistency check and	repair. lfcheck:	lfcheck(1)
uniq: report	repeated lines in a file.	uniq(1)
lfupdate: update modified LFS data	repetitively.	lfupdate(1)
reboot:	replace current UNIX with new program or system.	reboot(1M)
iostat:	report I/O and system statistics.	iostat(1M)
errpt: process a	report of logged errors.	errpt(1M)
uniq:	report repeated lines in a file.	uniq(1)
pcstat:	report statistics on output of getpc command.	pcstat(1)
fseek, ftell, rewind:	reposition a stream.	fseek(3S)
clearer: stream error	reset.	clearer:o(3S)
	reset: execute non-local goto.	reset:o(3C)
rstlfs:	restore logical file system from tape.	rstlfs(1)
mhrestor: incremental file system	restore.	mhrestor(1M)
savdate: save and	restore modification date.	savdate(1)

release:	restore printing of queued line printer jobs.	release(1)
start:	restore printing of queued line printer jobs.	start(1)
jobs.	restrain: suspend printing of queued line printer	restrain(1)
rsh:	restricted shell (command interpreter).	rsh(1)
stty, gtty: set and	retrieve terminal modes.	stty:o(2)
stty, gtty: set and	retrieve the modes of a terminal.	stty(3C)
lnxx:	return name of current terminal.	lnxx:o(3C)
stat: data	returned by stat system call.	stat(7)
col: filter	reverse line-feeds.	col(1)
	rew: rewind tape.	rew(1)
fseek, ftell,	rewind: reposition a stream.	fseek(3S)
rew:	rewind tape.	rew(1)
	rk?: RK11/RK03 or RK05 disk.	rk(4)
rk?: RK11/RK03 or	RK05 disk.	rk(4)
rk?:	RK11/RK03 or RK05 disk.	rk(4)
	rm, rmdir: remove files or directories.	rm(1)
mail,	rmail: send mail to users or read mail.	mail(1)
	rmdel: remove a delta from an SCCS file.	rmdel(1S)
	rmdir: remove directory.	rmdir:o(3C)
rm,	rmdir: remove files or directories.	rm(1)
reboot: transfer control to DEC	rom and reboot.	reboot(2)
chroot: change	root directory.	chroot(2)
chroot: change	root directory for a command.	chroot(1M)
pow, sqrt: exponential, logarithm, power, square	root. exp, log,	exp(3M)
rootdev:	root file system.	rootdev(4)
	rootdev: root file system.	rootdev(4)
utindx, utline: access	routines for utmp file.	utindx(3C)
regexp: regular expression compile and match	routines.	regexp(7)
hp:	RP04/RP05/RP06 moving-head disk.	hp(4)
	rsh: restricted shell (command interpreter).	rsh(1)
programming language. sh,	rsh: shell, the standard/restricted command	sh(1)
	rstlfs: restore logical file system from tape.	rstlfs(1)
nice:	run a command at specified priority.	nice(1)
nohup:	run a command immune to hangups.	nohup(1)
getpc: get Program Counter data on	running processes.	getpc(1)
	rx?: floppy disk.	rx(4)
	sa: shell accounting.	sa(1M)
	sacopy: stand-alone copy/verify.	sacopy(8)
pcs: program counter	sampling device.	pcs(4)
	savdate: save and restore modification date.	savdate(1)
vpmsave, vpmsnap, vpmtrace, vpmfmt:	save and print VPM event traces.	vpmsave(1C)
savdate:	save and restore modification date.	savdate(1)
break, brk,	sbrk: change memory allocation.	break(2)
	scanf, fscanf, sscanf: formatted input conversion.	scanf(3S)
awk: pattern	scanning and processing language.	awk(1)
chghist: change the history entry of an	SCCS delta.	chghist(1S)
cmt: insert the delta commentary for an initial	SCCS delta.	cmt(1S)
comb: combine	SCCS deltas.	comb(1S)
sccsclean: remove unwanted files in	SCCS directories.	sccsclean(1S)
delta: make a delta (change) to an	SCCS file.	delta(1S)
get: get a version of an	SCCS file.	get(1S)
prs: print an	SCCS file.	prs(1S)
prt: print	SCCS file.	prt(1S)
rmdel: remove a delta from an	SCCS file.	rmdel(1S)
sccsdiff: compare two versions of an	SCCS file.	sccsdiff(1S)
sccsfile: format of	SCCS file.	sccsfile(5)
val: validate	SCCS file.	val(1S)
gdiff: diff an	SCCS file with named file.	gdiff(1S)
admin: administer	SCCS files.	admin(1S)
gmark: mark a subsystem of	SCCS files..	gmark(1S)
gadd: add a file to	SCCS.	gadd(1S)
gadmin: admin a file in	SCCS.	gadmin(1S)
gdelta: delta a file from	SCCS.	gdelta(1S)
gget: get a file from	SCCS.	gget(1S)
gprt: prt a file in	SCCS.	gprt(1S)
addsccs: add	SCCS keywords to a file.	addsccs(1S)
sccstring: echo	SCCS keywords to standard output.	sccstring(1S)
directories.	sccsclean: remove unwanted files in SCCS	sccsclean(1S)
	sccsdiff: compare two versions of an SCCS file.	sccsdiff(1S)
	sccsfile: format of SCCS file.	sccsfile(5)
gls: list the directory	$SCCSOURCE with input args appended.	gls(1S)
	sccstring: echo SCCS keywords to standard output.	sccstring(1S)
alarm:	schedule signal after specified time.	alarm(2)
inittab:	script for the init process.	inittab(5)
	sdiff: side-by-side difference program.	sdiff(1)
grep, egrep, fgrep:	search a file for a pattern.	grep(1)

manmac: macros to print CB-UNIX manual	sections.	manmac(5)
	sed: stream editor.	sed(1)
	seek: move read/write pointer.	seek:o(2)
comm:	select or reject lines common to two sorted files.	comm(1)
cut: cut out	selected fields of each line of a file.	cut(1)
getu: get	selected user block information.	getu(2)
unlock, tlock, noulk: semaphore operations.	sema, p, v, test, post, block, setsem, rdsem, lock,	sema(2)
	sema: semaphore operations.	sema(1)
sema:	semaphore operations.	sema(1)
block, setsem, rdsem, lock, unlock, tlock, noulk:	semaphore operations. sema, p, v, test, post,	sema(2)
kill:	send a signal to a process or process group.	kill(1)
menab, mdisab, msend, mrecv, mctl:	send and receive messages.	message(2)
mail, rmail:	send mail to users or read mail.	mail(1)
gcat:	send phototypesetter output to the HONEYWELL 6000.	gcat(1C)
message veneer for sending/ msg, msgenab, msgdisab,	send, sendw, recv, recvw, msgstat, msgctl: old	msg(3)
kill:	send signal to a process.	kill(2)
recvw, msgstat, msgctl: old message veneer for	sending and receiving messages.. /sendw, recv,	msg(3)
veneer for sending/ msg, msgenab, msgdisab, send,	sendw, recv, recvw, msgstat, msgctl: old message	msg(3)
exprog: perform standard Shell execute	sequence.	exprog:o(3C)
bclk,	setbclk: reads and sets the battery clock.	bclk(1M)
	setbuf: assign buffering to a stream.	setbuf(3S)
	setgid: set process group ID.	setgid(2)
getgrent, getgrgid, getgrnam,	setgrent, endgrent: get group file entry.	getgrent(3C)
	setjmp, longjmp: non-local goto.	setjmp(3C)
crypt,	setkey, encrypt: DES encryption.	crypt(3C)
	setpgrp: execute program with new process group.	setpgrp(1)
	setpgrp: set process group.	setpgrp(2)
getpwent, getpwuid, getpwnam,	setpwent, endpwent: get password file entry.	getpwent(3C)
bclk, setbclk: reads and	sets the battery clock.	bclk(1M)
semaphore/ sema, p, v, test, post, block,	setsem, rdsem, lock, unlock, tlock, noulk:	sema(2)
profile:	setting up an environment at login time.	profile(5)
gettydefs: speed and terminal	settings used by getty.	gettydefs(5)
	setuid: set process user ID.	setuid(2)
entry. getutent, getutid, getutline, pututline,	setutent, endutent, utmpname: access utmp file	getut(3C)
programming language.	sh, rsh: shell, the standard/restricted command	sh(1)
smcreat, smopen, smclose, smget, smput:	shared memory operations.	shmem(2)
sa:	shell accounting.	sa(1M)
rsh: restricted	shell (command interpreter).	rsh(1)
system: issue a	shell command.	system(3S)
exprog: perform standard	Shell execute sequence.	exprog:o(3C)
language. sh, rsh:	shell, the standard/restricted command programming	sh(1)
sdiff:	side-by-side difference program.	sdiff(1)
login:	sign on.	login(1)
alarm: schedule	signal after specified time.	alarm(2)
	signal: catch or ignore signals.	signal(2)
pause: stop until	signal.	pause(2)
kill: send	signal to a process.	kill(2)
kill: send a	signal to a process or process group.	kill(1)
signal: catch or ignore	signals.	signal(2)
ssignal, gsignal: software	signals.	ssignal(3C)
lex: generate programs for	simple lexical tasks.	lex(1)
tc: phototypesetter	simulator.	tc(1)
trigonometric functions.	sin, cos, tan, asin, acos, atan, atan2:	trig(3M)
	sinh, cosh, tanh: hyperbolic functions.	sinh(3M)
size:	size of an object file.	size(1)
	size: size of an object file.	size(1)
	sleep: stop execution for interval.	sleep(3C)
	sleep: suspend execution for an interval.	sleep(1)
ttyslot: find the	slot in the utmp file of the current user.	ttyslot(3C)
smcreat, smopen,	smclose, smget, smput: shared memory operations.	shmem(2)
memory operations.	smcreat, smopen, smclose, smget, smput: shared	shmem(2)
smcreat, smopen, smclose,	smget, smput: shared memory operations.	shmem(2)
spline: interpolate	smooth curve.	spline(1G)
operations. smcreat,	smopen, smclose, smget, smput: shared memory	shmem(2)
smcreat, smopen, smclose, smget,	smput: shared memory operations.	shmem(2)
	sno: SNOBOL interpreter.	sno(1)
sno:	SNOBOL interpreter.	sno(1)
ssignal, gsignal:	software signals.	ssignal(3C)
sort:	sort or merge files.	sort(1)
qsort: quicker	sort.	qsort(3C)
	sort: sort or merge files.	sort(1)
tsort: topological	sort.	tsort(1)
comm: select or reject lines common to two	sorted files.	comm(1)
g_find: locate and identify a	source file.	g_find(1S)
enabmaus, dismaus, switmaus: multiple access user	space operations. maus, getmaus, freemaus,	maus(2)
fork:	spawn new process.	fork(2)

nice: run a command at	specified priority.	nice(1)
alarm: schedule signal after	specified time.	alarm(2)
getty: set terminal type, modes,	speed, and line discipline.	getty(1M)
gettydefs:	speed and terminal settings used by getty.	gettydefs(5)
cspeed: convert baud to	speed number.	cspeed(3C)
flog:	speed up a process.	flog(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)
spell, spellin,	spellout: find spelling errors.	spell(1)
	spline: interpolate smooth curve.	spline(1G)
split:	split a file into pieces.	split(1)
frexp, ldexp, modf:	split into mantissa and exponent.	frexp(3C)
	split: split a file into pieces.	split(1)
uuclean: uucp	spool directory clean-up.	uuclean(1M)
lpr: line printer	spooling program.	lpr(1)
	spr: special print command.	spr(1)
printf, fprintf,	sprintf: formatted output conversion.	printf(3S)
	Sprof: system profile.	sprof(1M)
	sprofil: turn on/off system profiling.	sprofil(2)
	sps: detail process status.	sps(1)
exp, log, pow,	sqrt: exponential, logarithm, power, square root.	exp(3M)
exp, log, pow, sqrt: exponential, logarithm, power,	square root.	exp(3M)
rand,	srand: random number generator.	rand(3C)
scanf, fscanf,	sscanf: formatted input conversion.	scanf(3S)
	ssignal, gsignal: software signals.	ssignal(3C)
	stack: stack trace from crash file.	stack(1)
stack:	stack trace from crash file.	stack(1)
stamp: version	stamp utility.	stamp(1)
	stamp: version stamp utility.	stamp(1)
sacopy:	stand-alone copy/verify.	sacopy(8)
intro: introduction to	stand-alone utilities.	intro(8)
stdio:	standard buffered input/output package.	stdio(3S)
stdio:	standard buffered input/output package.	stdio:o(3S)
pcat: expand compressed file to	standard output.	pcat(1)
sccstring: echo SCCS keywords to	standard output.	sccstring(1S)
exprog: perform	standard Shell execute sequence.	exprog:o(3C)
sh, rsh: shell, the	standard/restricted command programming language.	sh(1)
jobs.	start: restore printing of queued line printer	start(1)
	startrek: clobber klingons.	startrek(1X)
boot procedures: UNIX	startup.	bproc(6)
	stat: data returned by stat system call.	stat(7)
	stat, fstat: get file status.	stat(2)
	stat: get file status.	stat:o(2)
stat: data returned by	stat system call.	stat(7)
iostat: report I/O and system	statistics.	iostat(1M)
pcstat: report	statistics on output of getpc command.	pcstat(1)
x25pvc, x25lnk: install, remove, or get	status for a PVC or BX.25 link.	x25pvc(1C)
ferror, feof, clearerr, fileno: stream	status inquiries.	ferror(3S)
uustat: uucp	status inquiry and job control.	uustat(1C)
fstat: get	status of open file.	fstat(2)
ps: process	status.	ps(1)
sps: detail process	status.	sps(1)
stat, fstat: get file	status.	stat(2)
stat: get file	status.	stat:o(2)
	stdio: standard buffered input/output package.	stdio(3S)
	stdio: standard buffered input/output package.	stdio:o(3S)
	stime: set time.	stime(2)
sleep:	stop execution for interval.	sleep(3C)
pause:	stop until signal.	pause(2)
strlen, strchr, strrchr: string operations.	strcat, strncat, strcmp, strncmp, strcpy, strncpy,	string(3C)
strncat, strcmp, strncmp, strcpy, strncpy, strlen,	strchr, strrchr: string operations. strcat,	string(3C)
strrchr: string operations. strcat, strncat,	strcmp, strncmp, strcpy, strncpy, strlen, strchr,	string(3C)
operations. strcat, strncat, strcmp, strncmp,	strcpy, strncpy, strlen, strchr, strrchr: string	string(3C)
sed:	stream editor.	sed(1)
clearer:	stream error reset.	clearer:o(3S)
fclose, fflush: close or flush a	stream.	fclose(3S)
fopen, freopen, fdopen: open a	stream.	fopen(3S)
fopen, freopen: open a	stream.	fopen:o(3S)
fseek, ftell, rewind: reposition a	stream.	fseek(3S)
getchar, fgetc, 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)
setbuf: assign buffering to a	stream.	setbuf(3S)
ferror, feof, clearerr, fileno:	stream status inquiries.	ferror(3S)

ungetc: push character back into input	stream.	ungetc(3S)
gets, fgets: get a	string from a stream.	gets(3S)
file_log: log an input	string in a logfile..	file_log(1S)
puts, fputs: put a	string on a stream.	puts(3S)
strncmp, strcpy, strncpy, strlen, strchr, strrchr:	string operations. strcat, strncat, strcmp,	string(3C)
cnvtime, gtime: convert	string to internal time.	cnvtime(3C)
	strip: remove symbols and relocation bits.	strip(1)
strcat, strncat, strcmp, strncmp, strcpy, strncpy,	strlen, strchr, strrchr: string operations.	string(3C)
strchr, strrchr: string operations. strcat,	strncat, strcmp, strncmp, strcpy, strncpy, strlen,	string(3C)
string operations. strcat, strncat, strcmp,	strncmp, strcpy, strncpy, strlen, strchr, strrchr:	string(3C)
strcat, strncat, strcmp, strncmp, strcpy,	strncpy, strlen, strchr, strrchr: string/	string(3C)
strcmp, strncmp, strcpy, strncpy, strlen, strchr,	strrchr: string operations. strcat, strncat,	string(3C)
	stty, gtty: set and retrieve terminal modes.	stty:o(2)
terminal.	stty, gtty: set and retrieve the modes of a	stty(3C)
	stty: set teletype options.	stty(1)
	su: become super-user or another user.	su(1)
intro: introduction to	subroutines and libraries.	intro(3)
paste: merge same lines of several files or	subsequent lines of one file.	paste(1)
gmark: mark a	subsystem of SCCS files..	gmark(1S)
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(1M)
sync: update the	super block.	sync(1M)
update: periodically update the	super block.	update(1)
sync: update	super-block.	sync(2)
su: become	super-user or another user.	su(1)
sleep:	suspend execution for an interval.	sleep(1)
hold:	suspend printing of queued line printer jobs.	hold(1)
restrain:	suspend printing of queued line printer jobs.	restrain(1)
	swab: swap bytes.	swab(3C)
swab:	swap bytes.	swab(3C)
	swapdev: location for swapping.	swapdev(4)
swapdev: location for	swapping.	swapdev(4)
getcsw: read console	switches.	getcsw(2)
maus, getmaus, freemaus, enabmaus, dismaus,	switmaus: multiple access user space operations.	maus(2)
strip: remove	symbols and relocation bits.	strip(1)
	sync: update super-block.	sync(2)
	sync: update the super block.	sync(1M)
messages. perror,	sys_errlist, sys_nerr, errno: system error	perror(3C)
perror, sys_errlist,	sys_nerr, errno: system error messages.	perror(3C)
tek, vplot,	t300, t300s, t450: graphics filters.	tek(1)
tek, vplot, t300,	t300s, t450: graphics filters.	tek(1)
tek, vplot, t300, t300s,	t450: graphics filters.	tek(1)
mkconf: create configuration	table and low core.	mkconf(1M)
L-devices: auto-dialer device	table.	L-devices(5)
mountpts: general user mount point	table.	mountpts(5)
mtab: mounted file system	table.	mtab(5)
crontab:	table of chronological events to be executed..	crontab(5)
L.sys:	table of connecting uucp systems.	L.sys(5)
tbl: format	tables for nroff or troff.	tbl(1)
	tail: deliver the last part of a file.	tail(1)
other users.	talk: allow user to listen and talk to one or more	talk(1)
talk: allow user to listen and	talk to one or more other users.	talk(1)
functions. sin, cos,	tan, asin, acos, atan, atan2: trigonometric	trig(3M)
sinh, cosh,	tanh: hyperbolic functions.	sinh(3M)
tp: manipulate	tape archive.	tp(1)
tar:	tape archiver.	tar(1)
dmplfs: dump logical file system to	tape.	dmplfs(1)
dump: incremental dump	tape format.	dump(5)
tp: magnetic	tape format.	tp(5)
mt?: TE16/TU16 magnetic	tape interface.	mt(4)
tm?: TM11/TU10 magnetic	tape interface.	tm(4)
mtm: magnetic	tape manipulation.	mtm(1)
mkfst: construct a file system on mag	tape.	mkfst(1M)
rew: rewind	tape.	rew(1)
rstlfs: restore logical file system from	tape.	rstlfs(1)
	tar: tape archiver.	tar(1)
lex: generate programs for simple lexical	tasks.	lex(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)
	tcmp: text comparison for crash dump.	tcmp(1M)
mt?:	TE16/TU16 magnetic tape interface.	mt(4)
	tee: pipe fitting.	tee(1)
	tek, vplot, t300, t300s, t450: graphics filters.	tek(1)

gex: Graphic EXerciser for	Tektronix 4014.	gex(1G)
tk: paginator for the	Tektronix 4014.	tk(1)
tkdump: prints a	Tektronix file.	tkdump(1G)
stty: set	teletype options.	stty(1)
	telinit: user communication with init.	telinit(1M)
	tell: get file offset.	tell(3C)
	tell: get file offset.	tell:o(2)
mktmp: make a	temporary file.	mktmp:o(3C)
mktemp: make	temporary file name.	mktemp:o(3C)
tmpnam: create a name for a	temporary file.	tmpnam(3S)
ct: call	terminal.	ct(1C)
ctermid: generate file name for	terminal.	ctermid(3S)
gtty: get	terminal line options.	gtty(1)
lnxx: return name of current	terminal.	lnxx:o(3C)
mhstty: set the options for a	terminal.	mhstty(1)
stty, gtty: set and retrieve	terminal modes.	stty:o(2)
vtp: virtual	terminal protocol.	vtp(4)
gettydefs: speed and	terminal settings used by getty.	gettydefs(5)
stty, gtty: set and retrieve the modes of a	terminal.	stty(3C)
ttyname, isatty: find name of a	terminal.	ttyname(3C)
getty: set	terminal type, modes, speed, and line discipline.	getty(1M)
tty: get the	terminal's name.	tty(1)
tty: general interface for	terminals.	tty(4)
exit:	terminate process.	exit(2)
	test: condition evaluation command.	test(1)
mmtest: PDP 11/70 memory management	test.	mmtest(8)
tlock, noulk: semaphore operations. sema, p, v,	test, post, block, setsem, rdsem, lock, unlock,	sema(2)
tcmp:	text comparison for crash dump.	tcmp(1M)
ed:	text editor.	ed(1)
eqn, neqn, checkeq: typeset mathematical	text.	eqn(1)
reform: reformat	text file.	reform(1S)
plock: lock process or	text in memory.	plock(2)
nnroff: format	text.	nnroff(1)
nroff, troff: format or typeset	text.	nroff(1)
cubic: three dimensional	tic-tac-toe.	cubic(1)
ttt:	tic-tac-toe.	ttt(1X)
time:	time a command.	time(1)
alarm: schedule signal after specified	time.	alarm(2)
at: execute commands at a later	time.	at(1)
cnvtime, gtime: convert string to internal	time.	cnvtime(3C)
ftime: get date and	time.	ftime(2)
	time: get date and time.	time(2)
touch: change modification	time of a file.	touch(1)
profile: setting up an environment at login	time.	profile(5)
stime: set	time.	stime(2)
	time: time a command.	time(1)
time: get date and	time.	time(2)
gmtime, asctime, timezone: convert date and	time to ASCII. ctime, localtime,	ctime(3C)
ctime: convert date and	time to ASCII.	ctime:o(3C)
profil: execution	time user profile.	profil(2)
	times: get process times.	times(2)
utime: update	times in file.	utime(2)
times: get process	times.	times(2)
ctime, localtime, gmtime, asctime,	timezone: convert date and time to ASCII.	ctime(3C)
	tk: paginator for the Tektronix 4014.	tk(1)
	tkdump: prints a Tektronix file.	tkdump(1G)
v, test, post, block, setsem, rdsem, lock, unlock,	tlock, noulk: semaphore operations. sema, p,	sema(2)
	tm: meditate.	tm(1)
	tm?: TM11/TU10 magnetic tape interface.	tm(4)
tm?:	TM11/TU10 magnetic tape interface.	tm(4)
	tmpnam: create a name for a temporary file.	tmpnam(3S)
toupper, tolower,	toascii: character translation.	conv(3C)
popen, pclose: initiate I/O	to/from a process.	popen(3S)
toupper,	tolower, toascii: character translation.	conv(3C)
tsort:	topological sort.	tsort(1)
	touch: change modification time of a file.	touch(1)
	toupper, tolower, toascii: character translation.	conv(3C)
	tp: magnetic tape format.	tp(5)
	tp: manipulate tape archive.	tp(1)
	tr: translate characters.	tr(1)
	trace: event-tracing driver.	trace(4)
stack: stack	trace from crash file.	stack(1)
ptrace: process	trace.	ptrace(2)
vpmsnap, vpmtrace, vpmfmt: save and print VPM event	traces. vpmsave,	vpmsave(1C)
reboot:	transfer control to DEC rom and reboot.	reboot(2)
hex:	translate binary file to ascii hexadecimal.	hex(1)

tr:	translate characters.	tr(1)
unhex:	translate hexed file to binary.	unhex(1)
toupper, tolower, toascii: character	translation.	conv(3C)
sin, cos, tan, asin, acos, atan, atan2:	trigonometric functions.	trig(3M)
nroff,	troff: format or typeset text.	nroff(1)
deroff: remove nroff,	troff, tbl and eqn constructs.	deroff(1)
tbl: format tables for nroff or	troff.	tbl(1)
	true, false: provide truth values.	true(1)
true, false: provide	truth values.	true(1)
	tsort: topological sort.	tsort(1)
	ttt: tic-tac-toe.	ttt(1X)
	tty: general interface for terminals.	tty(4)
	tty: get the terminal's name.	tty(1)
greek: graphics for extended	TTY-37 type-box.	greek(7)
	ttyname, isatty: find name of a terminal.	ttyname(3C)
current user.	ttyslot: find the slot in the utmp file of the	ttyslot(3C)
	turbo: encabulator.	turbo(1)
festoon:	turgid memorandum composition.	festoon(1)
file: determine file	type.	file(1)
getty: set terminal	type, modes, speed, and line discipline.	getty(1M)
mm:	type out documents that use the PWB/MM macros.	mm(1)
greek: graphics for extended TTY-37	type-box.	greek(7)
	types: primitive system data types.	types(7)
types: primitive system data	types.	types(7)
eqn, neqn, checkeq:	typeset mathematical text.	eqn(1)
nroff, troff: format or	typeset text.	nroff(1)
	typo: find possible typos.	typo(1)
typo: find possible	typos.	typo(1)
	ucore: enable/disable unique core dumping feature..	ucore(2)
feature..	ucore: turn on or off the unique core dumping	ucore(1)
getpw: get name from	UID.	getpw(3C)
	umask: set and get creation mask.	umask(2)
	umount: dismount file system.	umount(1)
	umount: dismount file system.	umount(2)
mount,	umount: mount or remove file system.	mount(2)
	uname: get name of current UNIX system.	uname(2)
	uname: print name of current UNIX.	uname(1)
kunb:	un-assembler for the KMC11/DMC11 microprocessor.	kunb(1)
	ungetc: push character back into input stream.	ungetc(3S)
	unhex: translate hexed file to binary.	unhex(1)
	uniq: report repeated lines in a file.	uniq(1)
ucore: turn on or off the	unique core dumping feature..	ucore(1)
ucore: enable/disable	unique core dumping feature..	ucore(2)
mktemp: make a	unique file name.	mktemp(3C)
	units: conversion program.	units(1)
	unlink: remove directory entry.	unlink(2)
sema, p, v, test, post, block, setsem, rdsem, lock,	unlock, tlock, noulk: semaphore operations.	sema(2)
lfumount:	unmount the logical file system (LFS).	lfumount(1)
	unpack: expand compressed files.	unpack(1)
sccsclean: remove	unwanted files in SCCS directories.	sccsclean(1S)
updfs:	update file system.	updfs(1M)
lfsync:	update modified LFS data.	lfsync(1)
lfupdate:	update modified LFS data repetitively.	lfupdate(1)
	update: periodically update the super block.	update(1)
sync:	update super-block.	sync(2)
sync:	update the super block.	sync(1M)
update: periodically	update the super block.	update(1)
utime:	update times in file.	utime(2)
	updfs: update file system.	updfs(1M)
du: summarize disk	usage.	du(1)
id: print	user and group id.	id(1)
getuid, getgid, geteuid, getegid: get	user and group identity.	getuid(2)
getu: get selected	user block information.	getu(2)
telinit:	user communication with init.	telinit(1M)
environ:	user environment.	environ(7)
cuserid: character	user ID.	cuserid(3S)
setuid: set process	user ID.	setuid(2)
getuid: get	user identification.	getuid:o(2)
mountpts: general	user mount point table.	mountpts(5)
profil: execution time	user profile.	profil(2)
enabmaus, dismaus, switmaus: multiple access	user space operations. maus, getmaus, freemaus,	maus(2)
su: become super-user or another	user.	su(1)
talk: allow	user to listen and talk to one or more other users.	talk(1)
find the slot in the utmp file of the current	user. ttyslot:	ttyslot(3C)
write: write to another	user.	write(1)
mail, rmail: send mail to	users or read mail.	mail(1)

allow user to listen and talk to one or more other	users. talk:	talk(1)
wall: write to all	users.	wall(1)
intro: introduction to stand-alone	utilities.	intro(8)
stamp: version stamp	utility.	stamp(1)
	utime: update times in file.	utime(2)
	utindx, utline: access routines for utmp file.	utindx(3C)
utindx,	utline: access routines for utmp file.	utindx(3C)
utmp, wtmp:	utmp and wtmp entry formats.	utmp(5)
pututline, setutent, endutent, utmpname: access	utmp file entry. getutent, getutid, getutline,	getut(3C)
ttyslot: find the slot in the	utmp file of the current user.	ttyslot(3C)
utindx, utline: access routines for	utmp file.	utindx(3C)
	utmp, wtmp: utmp and wtmp entry formats.	utmp(5)
getutid, getutline, pututline, setutent, endutent,	utmpname: access utmp file entry. getutent,	getut(3C)
	uuclean: uucp spool directory clean-up.	uuclean(1M)
uusub: monitor	uucp network.	uusub(1M)
uuclean:	uucp spool directory clean-up.	uuclean(1M)
uustat:	uucp status inquiry and job control.	uustat(1C)
L-dialcodes:	uucp system dialcodes.	L-dialcodes(5)
L.sys: table of connecting	uucp systems.	L.sys(5)
	uucp, uulog, uuname: unix to unix copy.	uucp(1C)
uunames: list names of UNIX systems known to	uucp.	uunames(1C)
uucp,	uulog, uuname: unix to unix copy.	uucp(1C)
uucp, uulog,	uuname: unix to unix copy.	uucp(1C)
	uunames: list names of UNIX systems known to uucp.	uunames(1C)
	uustat: uucp status inquiry and job control.	uustat(1C)
	uusub: monitor uucp network.	uusub(1M)
	uux: unix to unix command execution.	uux(1C)
tlock, noulk: semaphore operations. sema, p,	v, test, post, block, setsem, rdsem, lock, unlock,	sema(2)
	val: validate SCCS file.	val(1S)
val:	validate SCCS file.	val(1S)
abs: integer absolute	value.	abs(3C)
floor, fabs, ceil, fmod: absolute	value, floor, ceiling, remainder functions.	floor(3M)
getenv:	value for environment name.	getenv(3C)
true, false: provide truth	values.	true(1)
lcall,	vcall: create and execute a new process.	call:o(3C)
	vcrt: filter nroff output for virtual crts.	vcrt(1)
sendw, recv, recvw, msgstat, msgctl: old message	veneer for sending and receiving messages.. /send,	msg(3)
assert: program	verification.	assert(3X)
lint: a C program	verifier.	lint(1)
vp:	Versatec printer-plotter.	vp(4)
lib7:	Version 7 library.	lib7(3X)
get: get a	version of an SCCS file.	get(1S)
stamp:	version stamp utility.	stamp(1)
sccsdiff: compare two	versions of an SCCS file.	sccsdiff(1S)
vcrt: filter nroff output for	virtual crts.	vcrt(1)
vpm, vpb:	Virtual Protocol Machine Protocol and Interface.	vpm(4)
vpmc: compiler for the	virtual protocol machine.	vpmc(1C)
vtp:	virtual terminal protocol.	vtp(4)
infect: Give a	virus to another UNIX system.	infect(1)
filesystem: format of system	volume.	fs(5)
	vp: Versatec printer-plotter.	vp(4)
Interface. vpm,	vpb: Virtual Protocol Machine Protocol and	vpm(4)
tek,	vplot, t300, t300s, t450: graphics filters.	tek(1)
vpmset, vpmstart: connect	VPM drivers and KMCs; load the KMC11-B..	vpmset(1C)
vpmsave, vpmsnap, vpmtrace, vpmfmt: save and print	VPM event traces.	vpmsave(1C)
Interface.	vpm, vpb: Virtual Protocol Machine Protocol and	vpm(4)
	vpmc: compiler for the virtual protocol machine.	vpmc(1C)
vpmsave, vpmsnap, vpmtrace,	vpmfmt: save and print VPM event traces.	vpmsave(1C)
VPM event traces.	vpmsave, vpmsnap, vpmtrace, vpmfmt: save and print	vpmsave(1C)
load the KMC11-B..	vpmset, vpmstart: connect VPM drivers and KMCs;	vpmset(1C)
traces. vpmsave,	vpmsnap, vpmtrace, vpmfmt: save and print VPM event	vpmsave(1C)
KMC11-B.. vpmset,	vpmstart: connect VPM drivers and KMCs; load the	vpmset(1C)
vpmsave, vpmsnap,	vpmtrace, vpmfmt: save and print VPM event traces.	vpmsave(1C)
	vt: graphics interface.	vt(4)
	vtp: virtual terminal protocol.	vtp(4)
	wait: await completion of process.	wait(1)
wait:	wait for process to die.	wait(2)
	wait: wait for process to die.	wait(2)
	wall: write to all users.	wall(1)
	wc: word count.	wc(1)
whodo:	who is doing what.	whodo(1M)
who:	who is on the system.	who(1)
	who: who is on the system.	who(1)
	whodo: who is doing what.	whodo(1M)
chdir, cd: change	working directory.	chdir(1)
chdir: change	working directory.	chdir(2)

pwd:	working directory name.	pwd(1)
putchar:	write character.	putchar:o(3C)
write:	write on a file.	write(2)
bload:	write on block device.	bload(1M)
putpwent:	write password file entry.	putpwent(3C)
wall:	write to all users.	wall(1)
write:	write to another user.	write(1)
	write: write on a file.	write(2)
	write: write to another user.	write(1)
open: open for reading or	writing.	open(2)
utmp, wtmp: utmp and	wtmp entry formats.	utmp(5)
utmp,	wtmp: utmp and wtmp entry formats.	utmp(5)
	wump: hunt the wumpus.	wump(1X)
wump: hunt the	wumpus.	wump(1X)
	X25: BX.25 network interface.	x25(4)
BX.25 link. x25pvc,	x25lnk: install, remove, or get status for a PVC or	x25pvc(1C)
a PVC or BX.25 link.	x25pvc, x25lnk: install, remove, or get status for	x25pvc(1C)
command.	xargs: construct argument lists and execute	xargs(1)
	xref: cross reference for C programs.	xref(1)
j0, j1, jn,	y0, y1, yn: bessel functions.	bessel(3M)
j0, j1, jn, y0,	y1, yn: bessel functions.	bessel(3M)
	yacc: yet another compiler-compiler.	yacc(1)
j0, j1, jn, y0, y1,	yn: bessel functions.	bessel(3M)