
Table of Contents

Introduction	1
What Is COHERENT?	1
What is an Operating System?	1
Design Philosophy	2
Installation.	2
User Registration and Reaction Report	2
Technical Support	2
Help Us Help You	3
How To Use This Manual	3
Elementary Tutorials	3
Advanced Tutorials	4
The Lexicon	5
Where To Go From Here	5
Using the COHERENT System	7
How Do I Begin?	7
Logging in	7
Special Terminal Keys	8
Try Some COHERENT Commands	8
Giving Commands to COHERENT	9
help, man, apropos: Help with Commands	10
Shutting Down COHERENT and Rebooting.	11
Logging Out	12
Working With Files and Directories	12
File Names	12
Introduction to Directories	13
Path Names	13
ls, lc: Listing Your Directory.	14
cat: Print Contents of a File	15
more: List Files on the Screen	16
mkdir: Create a Directory	16
cd: Change Directory	16
pwd: Print Working Directory	16
mv, cp: Move and Copy Files	17
rm, rmdir: Remove Files and Directories	19
du, df: How Much Space?	19
ln: Link Files.	19
File Permissions.	20
chmod: Change File Permissions.	21
Creating and Mounting a File System	22
fdformat: Format a Floppy Disk	22
mkfs: Create a File System	22
mount: Mount a File System	23
Using a Newly Mounted File System	23
umount: Unmount a File System.	23
fsck: Check a File System	24
Devices, Files, and Drivers	24
Character-Special Files	25
tty Processing	25
A Tour Through the File System	25
General File System Layout	25
/bin	25
/dev	25
/drv	25
/etc	26
/lib	26
/usr	26

ii The COHERENT System

/u	26
Files: Conclusion	26
Introduction to COHERENT Commands	26
The Shell	27
Redirecting Input and Output	27
Pipes	27
Superuser	28
vsh: The Visual Shell	28
Manipulating Text Under COHERENT.	29
MicroEMACS: Text Screen Editor.	29
pr, prps, lp: Print Files	30
nroff, troff: Text Formatters	31
Miscellaneous Commands.	32
who: Who Is on the System	32
write: Electronic Dialogue	32
mail: Send an Electronic Letter.	32
msgs: Cumulative Message Board	34
grep: Find Patterns in Text Files	34
date: Print the Date.	35
passwd: Change Your Password	35
stty: Change Terminal Behavior	35
Scheduling Commands For Regular Execution	36
Managing Processes	37
ps: List Active Processes.	37
kill: Signal Processes	38
Programming Under COHERENT.	38
Basic Steps in COHERENT Programming.	38
Create the Program Source	39
cc: Compile the Program.	39
m4: Macro Processing	40
make: Build Larger Programs	40
db: Debug the Program	40
Administering the COHERENT System	41
Adding a New User	41
System Security.	41
Passwords	41
File Protection.	41
Encryption	42
Dumping and Saving Files.	42
System Accounting	42
ac: Login Accounting	42
sa: Processing Accounting.	43
Conclusion.	44
Introducing sh, the Bourne Shell.	45
Simple Commands	45
Special Characters	45
Running Commands in the Background	45
Scripts	46
.profile: Login Shell Script	47
Substitutions	47
File Name Substitution	47
Parameter Substitution	49
Shell Variable Substitution	50
Command Substitution	52
Special Shell Variables.	53
dot . : Read Commands	53
Values Returned by Commands	54
test: Condition Testing.	54
Executing Commands Conditionally.	54
Control Flow.	55
for: Execute a Loop	55

if: Execute Conditionally	56
while: Execute a Loop	57
until: Another Looping Construct	57
case: Serial Conditional Execution	57
Summary	58
Introduction to MicroEMACS	59
What is MicroEMACS?	59
Keystrokes: <ctrl>, <esc>	59
Becoming Acquainted with MicroEMACS	59
Beginning a Document	60
Moving the Cursor	61
Moving the Cursor Forward	61
Moving the Cursor Backwards	61
From Line to Line	61
Repetitive Motion	62
Moving Up and Down by a Screenful of Text	62
Moving to Beginning or End of Text	62
Saving Text and Quitting	62
Killing and Deleting	62
Deleting Vs. Killing	63
Erasing Text to the Right	63
Erasing Text to the Left	63
Erasing Lines of Text	63
Yanking Back (Restoring) Text	64
Quitting	64
Block Killing and Moving Text	64
Moving One Line of Text	64
Multiple Copying of Killed Text	64
Kill and Move a Block of Text	64
Capitalization and Other Tools	65
Capitalization and Lowercasing	65
Transpose Characters	66
Screen Redraw	66
Return Indent	66
Word Wrap	67
Search and Reverse Search	68
Search Forward	68
Reverse Search	68
Cancel a Command	69
Search and Replace	69
Saving Text and Exiting	70
Write Text to a New File	70
Save Text and Exit	70
Advanced Editing	70
Arguments	71
Arguments: Default Values	71
Selecting Values	71
Deleting With Arguments: An Exception	72
Buffers and Files	72
Definitions	72
File and Buffer Commands	72
Write and Rename Commands	72
Replace Text in a Buffer	73
Visiting Another Buffer	73
Move Text From One Buffer to Another	74
Checking Buffer Status	74
Renaming a Buffer	74
Delete a Buffer	74
Windows	75
Creating Windows and Moving Between Them	75
Enlarging and Shrinking Windows	76

iv The COHERENT System

Displaying Text Within a Window	76
One Buffer	77
Multiple Buffers	77
Moving and Copying Text Among Buffers	78
Checking Buffer Status	78
Saving Text From Windows	78
Keyboard Macros	78
Creating a Keyboard Macro	78
Execute a Macro Repeatedly	79
Replacing a Macro	79
Renaming a Macro	79
Renaming Macros: A Few Caveats	80
Setting the Initialization Macro	80
Flexible Key Bindings	80
Changing a Keybinding	80
Rebinding Metakeys	81
Save and Restore Keybindings	81
Sending Commands to COHERENT	82
Compiling and Debugging Through MicroEMACS	82
The MicroEMACS Help Facility	83
Where To Go From Here	83
Introduction to the ed Line Editor.	85
Why You Need an Editor	85
Learning To Use the Editor	85
General Topics.	85
ed, Files, and Text	86
Creating a File	86
Changing an Existing File	86
Working on Lines	87
Error Messages	87
Basic Editing Techniques	87
Creating a New File	87
Changing a File	88
Printing Lines	89
Abbreviating Line Numbers	90
How Many Lines?	90
Removing Lines	91
Abandoning Changes	92
Substituting Text Within a Line	92
Undoing Substitutions	93
Global Substitutions	94
Special Characters	94
Ranges of Substitution	94
Intermediate Editing	95
Relative Line Numbering	95
Changing Lines	96
Moving Blocks of Text	97
Copying Blocks of Text	98
String Searches	98
Remembered Search Arguments	99
Uses of Special Characters	99
Global Commands	100
Joining Lines	100
Splitting Lines	101
Marking Lines	101
Searching in Reverse Direction	103
Expert Editing	103
File Processing Commands	103
Patterns	104
Matching Many With One Character	105
Beginning and Ending of Lines	106

Replacing Matched Part	106
Replacing Parts of Matched String	106
Listing Funny Lines	108
Keeping Track of Current Line	108
When Current Line Is Changed	109
More About Global Commands	110
Issuing COHERENT Commands Within ed	110
For More Information.	110
Introduction to the sed Stream Editor	112
Getting to Know sed	112
Getting Started	112
Simple Commands	113
Substituting	113
Selecting Lines	114
p: Print Lines	115
Line Location	117
Add Lines of Text	118
Delete Lines	119
Change Lines	120
Include Lines From a File	120
Quit Processing	121
Next Line	121
Advanced sed Commands	122
Work Area	123
Add to Work Area	123
Print First Line	124
Save Work Area	125
Transform Characters	127
Command Control	128
{}: Command Grouping	128
!: All But	128
=: Print Line Number	128
Skipping Commands	129
t: Test Command	129
For More Information.	130
The C Language	131
Compiling C Programs under COHERENT	131
Try the Compiler	131
Phases of Compilation	131
Renaming Executable Files	132
Floating-Point Numbers	132
Compiling Multiple Source Files	133
Linking Without Compiling	133
Compiling Without Linking	133
Assembly-Language Files	133
Changing the Size of the Stack	134
Where To Go From Here	134
C for Beginners	134
Programming Languages and C.	134
Assembly and High-Level Languages	134
So, What Is C?.	135
Structured Programming	135
Writing a C Program	136
A Sample C Programming Session	136
Designing a Program	136
The main() Function	137
Open a File and Show Text	138
Accepting File Names.	139
Error Checking	140
Print a Portion of a File	142
Checking for the End of File.	144

Polling the Keyboard	145
For More Information.	147
Bibliography	147
Introduction to the awk Language	149
Example Files	149
Using awk	150
Command-line Options	150
Structure of an awk Program	150
Records and Fields	151
Patterns	151
Special Patterns.	151
Arithmetic Relational Expressions	152
Boolean Combinations of Expressions.	154
Patterns	155
Ranges of Patterns	156
Resetting Separators	157
Actions	159
awk Functions.	159
Printing with awk	160
Redirecting Output	163
Assignment of Variables	163
Field Variables.	164
Control Statements.	164
Arrays.	166
Initializing an Array	167
The for() Statement With Arrays	167
For More Information.	168
Introduction to lex, the Lexical Analyzer	169
How To Use lex	169
Translating Strings	169
Remove Blanks From Input	169
Trimming Blanks	170
lex Specification Form	170
Simple Form	170
Rules in lex	170
Statements in lex	171
Groups of Statements	172
Using the Same Action.	173
Patterns	173
Simple Patterns	173
Classes of Characters	174
Repetition	175
Choices and Grouping	176
Matching Non-Graphic Characters.	177
More Patterns	177
Line Context	177
Context Matching.	177
Macro Abbreviations	179
Context: Start Rules	179
Separate Contexts	180
More About Writing Actions.	181
ECHO.	181
Processing Overlapping Strings.	182
yylex	182
Header Section	183
Additional Routines.	183
Using lex With yacc.	183
Summary.	184
Introduction to yacc	185
Examples.	185
Phrases and Parentheses	185

Simple Expression Processing	187
Background	188
LR Parsing	188
Input Specification	188
Parser Operation	188
Form of yacc Programs.	189
Definitions	189
Rules	189
User Code	190
Rules	190
General Form of Rules	190
Suggested Style	190
Actions	191
Basic Action Statements.	191
Action Values	191
Structured Values	193
Handling Ambiguities	194
How yacc Reacts	195
Additional Control	195
Precedence.	196
Error Handling	197
Summary.	197
Helpful Hints	198
Where to Go From Here	198
bc Desk Calculator Language	199
Entry and Exit.	199
Example of Simple Use.	199
Simple Statements	200
Numbers with Fractions	202
The Scale of Numbers	202
Addition and Subtraction	202
Scale During Multiplication	203
Setting the Scale of Results	203
Scale for Divisions	203
Scale From Exponentiation	204
What Is the Current Scale?	204
The if Statement.	204
Using the if Statement	204
Comparisons.	204
Grouped Statements	205
Many Statements Per Line.	205
The while Statement	206
Abbreviations in the while Statement	207
The for Statement.	207
Three Parts of the for Statement	207
Similarities Between the for and while Statements	208
Functions in bc	208
Example of Function Use	208
Functions Using Other Functions	209
Functions That Call Themselves	209
The auto Statement.	210
Programs in a File	210
Using a Program From a File	210
Using Libraries	211
The bc Library.	211
Summary.	212
Introduction to the m4 Macro Processor.	213
Definitions and Syntax.	213
Defining Macros.	214
Input Control	215
Output Control	216

String Manipulation	217
Numeric Manipulation	218
COHERENT System Interface	219
Errors.	221
For More Information.	221
The make Programming Discipline	223
How Does make Work?.	223
Try make	224
Essential make	224
The makefile	224
Building a Simple makefile	225
Comments and Macros.	225
Setting the Time.	226
Building a Large Program	226
Command-Line Options	227
Other Command Line Features.	227
Advanced make	228
Default Rules	228
Source File Path.	229
Double-Colon Target Lines	229
Special Targets	230
Errors.	230
Exit Status.	230
Alternative Uses.	230
Where To Go From Here	231
nroff, The Text-Formatting Language	233
What is nroff?	233
nroff Input and Output	233
Printing nroff Output.	234
nroff Limitations	234
The ms Macro Package.	234
Using this Tutorial	235
The ms Macro Package.	235
Text and Commands	235
Command Names	236
Paragraphs.	237
Section Headings	240
Title Page.	241
Headers and Footers	242
Fonts	243
Special Characters	244
Footnotes.	244
Displays and Keeps.	244
Other Commands.	245
Introducing nroff's Primitives.	246
Page Format	246
Breaks	247
Fill and Adjust Modes	247
Defining Paragraphs	249
Centering.	249
Tabs.	249
Page Breaks	250
Macros and Traps.	250
What Is a Macro?	250
Introducing Traps.	252
Headers and Footers	252
Macro Arguments.	253
Double vs. Single Backslashes	254
Designing and Installing Macros	255
Strings	257
Strings Within Strings	258

Number Registers	258
Incrementing and Decrementing	260
Units of Measurement	261
Conditional Input	263
Environments and Diversions.	266
Buffers	268
Headers and Footers	269
More About Fonts.	269
Diversions	270
A Footnote Macro	272
Command Line Options	272
For Further Information	273
UUCP, Remote Communications Utility	275
Contents of This Tutorial	275
An Overview of UUCP	275
Implementations of UUCP	276
Programs	276
Files and Directories	277
Attaching a Modem to Your Computer	279
Selecting Site and Domain Names	280
Set Up a UUCP Site by Hand	280
port: Describe a Serial Port	280
dial: Describe a Modem	281
sys: Individual System Configuration	283
Simplifying a UUCP Configuration With uuinstall	287
Invoking uuinstall	287
The Port File	288
The Dial File	289
The sys File	290
Modifying an Existing Entry.	292
Configuring UUCP for Dial-in Access	292
Giving a Remote UUCP Site a Login	293
Configuring a Spooling Directory for Remote UUCP Access	293
Configuring UUCP Files	293
One Last, Loose Thread	294
Requesting Files From a Remote UUCP System	294
Sending Files to a Remote UUCP System	294
UUCP Administration	295
Networks	295
Services.	295
Available Networks	296
Debugging UUCP Problems	296
Define the Problem Exactly	296
Enabling and Disabling Ports	296
Stale Requests and Multiple Requests.	297
Problems With Lock Files	297
Enabling Ports, /etc/ttys Problems	297
Permission Problems	297
UUCP Cannot Find Its Own Files.	298
Modem Configuration	298
The Modem Does Not Respond	299
The Modem Responds But Does Not Dial	299
The Modem Dials But No Connection Made	299
The Modem Dials, Carrier Is Established, Nothing Else Happens.	299
uulog Shows Lost Packets.	300
uulog Shows Incorrect Response.	300
Files Refuse To Be Sent or Cannot Be Received	300
File Transfers Fail With imsg Statements.	300
Files are Being Lost.	300
Non-COHERENT UUCP Site Problems.	300
Where to Go From Here	300

The Lexicon		301
#	String-ize operator	303
##	Token-pasting operator	304
#define	Define an identifier as a macro	304
#elif	Include code conditionally	306
#else	Include code conditionally	306
#endif	End conditional inclusion of code	306
#if	Include code conditionally	306
#ifdef	Include code conditionally	307
#ifndef	Include code conditionally	307
#include	Read another file and include it	307
#line	Reset line number	308
#pragma	Perform implementation-specific preprocessing	308
#undef	Undefine a macro	309
__DATE__	Date of translation	309
__FILE__	Source file name	310
__LINE__	Current line within a source file	310
__STDC__	Mark a conforming translator	310
__TIME__	Time source file is translated	311
_exit()	Terminate a program	311
_getwd()	Get current working directory name	312
_tolower()	Convert characters to lower case	312
_toupper()	Convert characters to upper case	312
a.out.h	Include all COFF header files	314
abort()	End program immediately	314
abs()	Return the absolute value of an integer	314
ac	Summarize login accounting information	315
accept()	Accept a connection on a socket	315
access()	Check if a file can be accessed in a given mode	316
acct()	Enable/disable process accounting	317
acct.h	Format for process-accounting file	318
accton	Enable/disable process accounting	319
acos()	Calculate inverse cosine	319
add_history()	Add a line to history buffer	320
address		320
Administering COHERENT		321
alarm()	Set a timer	325
alias	Set an alias	326
aliases	File of users' aliases	326
alignment	Alignment or packing of fields within a structure	328
alloc.h	Define the allocator	328
alloca()	Dynamically allocate space on the stack	328
almanac	Print an almanac entry for this date	329
ANSI	Standards for information	329
apropos	Find manual pages on a given topic	330
ar	The librarian/archiver	330
ar.h	Format for archive files	331
arcoff.h	COFF archive-file header	332
arena		333
argc	Argument passed to main()	333
argv	Argument passed to main()	334
ARHEAD	Append options to beginning of ar command line	334
array		334
ARTAIL	Append options to end of ar command line	335
as	i80386 assembler	335
ASCII		366
asctime()	Convert time structure to ASCII string	369
asfix	Convert assembly-language programs into 80386 format	369
ASHEAD	Append options to beginning of as command line	370
asin()	Calculate inverse sine	370
ASKCC	Force prompting for CC names	370

assert()	Check assertion at run time.	370
assert.h	Define assert()	371
ASTAIL	Append options to end of as command line.	371
asy	Device driver for asynchronous serial lines.	371
asymkdev	Create nodes for asynchronous devices.	375
asypatch	Patch a kernel file for an asynchronous configuration.	375
at	Drivers for hard-disk partitions.	375
at	Execute commands at given time.	377
atan()	Calculate inverse tangent.	378
atan2()	Calculate inverse tangent.	378
ATclock	Read or set the AT realtime clock.	379
atexit()	Register a function to be called when the program exits.	379
atof()	Convert ASCII strings to floating point.	380
atoi()	Convert ASCII strings to integers.	380
atol()	Convert ASCII strings to long integers.	381
atrun	Execute commands at a preset time.	381
auto	Note an automatic variable.	382
awk	Pattern-scanning language.	382
backups	Strategies for backing up COHERENT.	384
bad	Maintain list of bad blocks.	389
badscan	Build bad block list.	389
banner	Print large letters.	390
basename	Strip path information from a file name.	390
bc	Interactive calculator with arbitrary precision.	390
bcmp()	Compare two chunks of memory.	392
bcopy()	Berkeley function to copy memory.	392
bind()	Bind a name to a socket.	393
bit		394
bit-fields		394
bit_count()	Count bits in a bit-mask.	394
bit map		395
block		395
boot	Boot block for hard-disk partition/nine-sector diskette.	395
boot.fha	Boot block for floppy disk.	396
booting	How booting works.	396
boottime	File that holds time system was last booted.	400
brc	Perform maintenance chores, single-user mode.	400
break	Exit from shell construct.	401
break	Exit from loop or switch statement.	401
brk()	Change size of data area.	401
bsearch()	Search an array.	401
buf.h	Buffer header.	403
buffer		403
build	Install COHERENT onto a hard disk.	403
builtin	Execute a command as a built-in command.	404
byte		404
byte ordering	Machine-dependent ordering of bytes.	404
bzero()	Initialize memory to NUL.	405
c	Print multi-column output.	406
C keywords		406
C language		407
C preprocessor		409
cabs()	Complex absolute value function.	412
cal	Print a calendar.	412
calendar	Reminder service.	412
calling conventions		413
calloc()	Allocate dynamic memory.	415
cancel	Cancel a print job.	416
canon.h	Portable layout of binary data.	416
captoinfo	Convert termcap data to terminfo form.	416
case	Execute commands conditionally according to pattern.	417

case	Introduce entry in switch statement	417
cast		418
cat	Concatenate the contents of a file to the standard output	418
caveat utilitor		418
cc	C compiler	418
cc0		432
cc1		432
cc2		432
cc3		432
CCHEAD	Append options to beginning of cc command line	432
CCTAIL	Append options to end of cc command line	432
cd	Change directory	433
CD-ROM	COHERENT support for read-only compact disk devices	433
cdmp	Dump COFF files into a readable form	434
cdplayer	Play audio CDs	435
cdrom.h	Definitions for CD-ROM drives	436
cdu31	Driver for the Sony CD-ROM drives	436
cdv	Interface to CD-ROM devices	436
cdview	Read a file from a CD-ROM	437
ceil()	Set numeric ceiling	437
cfgetispeed()	Get terminal input speed	438
cfgetospeed()	Get terminal output speed	438
cfsetispeed()	Set terminal input speed	438
cfsetospeed()	Set terminal output speed	439
cgrep	Pattern search for C source programs	439
char	Data type	441
chase	Highly amusing video game	441
chdir()	Change working directory	441
check	Check file system	442
checkerr	Check the mail system for errors	442
checklist	File systems to check when booting COHERENT	442
chgrp	Change the group owner of a file	442
chmod	Change the modes of a file	443
chmod()	Change file-protection modes	444
chmog	Change mode, owner, and group simultaneously	444
chown	Change the owner of files	445
chown()	Change ownership of a file	445
chreq	Change priority, lifetime, or printer for a job	445
chroot	Change root directory	446
chroot()	Change the root directory	446
chsize()	Change the size of a file	446
ckernit	Interactive inter-system communication and file transfer	447
clear	Clear the screen	451
clearerr()	Present stream status	451
clist.h	Character-list structures	451
clock	Read the system clock	451
clock()	Get processor time	452
close()	Close a file	452
closedir()	Close a directory stream	453
clri	Clear i-node	453
cmos	Device for reading CMOS	453
cmp	Compare bytes of two files	455
coff.h	Format for COFF objects	456
coffnlist()	Symbol table lookup, COFF format	457
coh_intro	Tour the COHERENT file system	458
coherent.h	Miscellaneous useful definitions	458
COHERENT	Principles of the COHERENT System	459
cohtune	Set a variable within a device driver	461
col	Remove reverse and half-line motions	462
comm	Print common lines	462
commands		462

compress	Compress a file	470
compression	Programs used to compress text	470
con.h	Configure device drivers	471
config	File that configures smail	471
config	File that configures UUCP	479
connect()	Connect to a socket.	481
console	Console device driver.	482
const	Qualify an identifier as not modifiable.	489
const.h	Declare machine-dependent constants	489
continue	Terminate current iteration of shell construct	489
continue	Force next iteration of a loop	489
controls	Data base for the lp print spooler	489
conv	Numeric base converter	492
core	Format of a core-dump file	493
core.h	Declare structure of a core file	494
cos()	Calculate cosine.	494
cosh()	Calculate hyperbolic cosine	494
cp	Copy a file	495
cpdir	Copy directory hierarchy.	496
cpio	Archiving/backup utility.	496
cpp	C preprocessor	496
CPPHEAD	Append options to beginning of cpp command line	500
CPPTAIL	Append options to end of cpp command line	500
creat()	Create/truncate a file	501
cron	Execute commands periodically	501
crontab	Copy a command file into the crontab directory	502
crypt	Encrypt/decrypt text	504
crypt()	Encryption using rotor algorithm.	505
ct	Controlling terminal driver	505
ctags	Generate tags and refs files for vi editor.	505
ctermid()	Name the terminal device that controls the current process.	506
ctime()	Convert system time to an ASCII string	506
ctype.h	Header file for data tests.	507
cu	UNIX-compatible communications utility	508
curses.h	Define functions and macros in curses library.	512
cut	Select portions of each line of its input	512
cvmail	Convert mail from COHERENT 3.X format to SV format	513
CWD	Current working directory.	513
d_passwd	Give passwords for devices	514
daemon	514
data formats	515
data types	515
date	Print/set the date and time	517
db	Assembler-level symbolic debugger	518
dbm.h	Header file for DBM routines	524
dbm_clearerr()	Clear an error condition on an NDBM data base.	524
dbm_close()	Close an NDBM data base.	524
dbm_delete()	Delete records from an NDBM data base	525
dbm_dirfno()	Return the file descriptor for an NDBM .dir file	525
dbm_error()	Check a NDBM data base for an error.	525
dbm_fetch()	Fetch a record from an NDBM data base	525
dbm_firstkey()	Retrieve the first key from an NDBM data base	526
dbm_nextdbm()	Retrieve the next key from an NDBM data base	526
dbm_open()	Open an NDBM data base.	526
dbm_pagfno()	Return the file descriptor for an NDBM .pag file	527
dbm_rdnly()	Set an NDBM data base into read-only mode.	527
dbm_store()	Store a record into an NDBM data base.	527
dbmclose()	Close a DBM data base	528
dbmopen()	Open a DBM data base.	528
dc	Desk calculator	528
dcheck	Check directory consistency.	529

dd	Convert the contents of a file	530
decvax_d()	Convert a double from IEEE to DECVAX format	531
decvax_f()	Convert a float from IEEE to DECVAX format	531
default	Default label in switch statement	531
defined	Perform an action if a macro is defined	532
deftty.h	Define default tty settings	532
delete()	Delete a record from a DBM data base	532
deroff	Remove text formatting control information	532
detab	Replace tab characters with spaces	533
device drivers		533
df	Measure free space on disk	536
dial	File that tells UUCP how to dial a system.	536
dialups	Name every device that may need an additional password.	539
diff	Compare two files.	539
diff3	Summarize differences among three files	540
difftime()	Calculate difference between two times	541
directors	Describe how to resolve local mail addresses.	541
directory		546
dirent.h	Define directory-related data elements	546
dirname	Extract a directory name.	546
dirs	Print the contents of the directory stack	547
disable	Disable a port	547
div()	Perform integer division	547
do	Introduce a loop.	548
domain	Set your system's mail domain	548
dos	Manipulate files on MS-DOS file systems	548
doscat	Concatenate a file on an MS-DOS file system	550
doscp	Copy files to/from an MS-DOS file system.	551
doscpdir	Copy a directory to/from an MS-DOS file system.	553
dosdel	Delete a file from an MS-DOS file system	554
dosdir	List contents of an MS-DOS directory	554
dosformat	Build an MS-DOS file system	555
doslabel	Label an MS-DOS floppy disk	556
dosls	List files on an MS-DOS file system.	556
dosmkdir	Create a directory in an MS-DOS file system	556
dosrm	Remove a file from an MS-DOS file system.	557
dosrmdir	Remove a directory from an MS-DOS file system	557
double	Data type.	558
dpac	De-fragment a COHERENT file system	558
drand48()	Return a 48-bit pseudo-random number as a double.	558
drvld.all	Load loadable drivers at boot time	559
du	Summarize disk usage	559
dump	File-system backup utility.	559
dumpdate	Print dump dates	560
dumpdir	Print the directory of a dump	560
dumptape.h	Define data structures used on dump tapes	561
dup()	Duplicate a file descriptor	561
dup2()	Duplicate a file descriptor	561
echo	Repeat/expand an argument	563
ecvt()	Convert floating-point numbers to strings	563
ed	Interactive line editor.	564
EDITOR	Name editor to use by default.	567
egrep	Extended pattern search.	567
else	Introduce a conditional statement	569
elvis	Clone of Berkeley-standard screen editor	569
elvprsv	Preserve the modified version of a file after a crash	581
elvrec	Recover the modified version of a file after a crash	581
em87	Perform/emulate hardware floating-point operations.	582
emacs	COHERENT screen editor	582
enable	Enable a port	582
endgrent()	Close group file	583

endhostent()	Close file /etc/hosts	583
endnetent()	Close network file	583
endprotoent()	Close protocols file	584
endpwent()	Close password file	584
endservent()	Close protocols file	584
endspent()	Close the shadow-password file	584
endutent()	Close the login logging file	584
enum	Declare a type and identifiers	585
ENV	File read to set environment	585
env	Execute a command in an environment	585
environ	Process environment	586
environmental variables		586
envp	Argument passed to main()	587
EOF	Indicate end of a file	587
epson	Prepare files for Epson printer	588
erand48()	Return a 48-bit pseudo-random number as a double	588
errno	External integer for return of error status	589
errno.h	Error numbers used by errno()	589
eval	Evaluate arguments	593
ex	Berkeley-style line editor	593
exec	Execute command directly	594
execel()	Execute a load module	594
execle()	Execute a load module	594
execlp()	Execute a load module	594
execlpe()	Execute a load module	595
execution		595
execv()	Execute a load module	596
execve()	Execute a load module	596
execvp()	Execute a load module	597
execvpe()	Execute a load module	598
exit	Exit from a shell	598
exit()	Terminate a program gracefully	598
EXIT_FAILURE	Indicate program failed to execute successfully	599
EXIT_SUCCESS	Indicate program executed successfully	599
exp()	Compute exponent	599
export	Add a shell variable to the environment	600
expr	Compute a command-line expression	600
extern	Declare storage class	601
fabs()	Compute absolute value	603
factor	Factor a number	603
false	Unconditional failure	603
fc	Edit and re-execute one or more previous commands	603
FCEDIT	Editor used by fc command	604
fclose()	Close a stream	604
fcntl()	Control open files	604
fcntl.h	Manifest constants for file-handling functions	605
fevt()	Convert floating-point numbers to strings	606
fd	Floppy disk driver	606
fd.h	Declare file-descriptor structure	608
fdformat	Low-level format a floppy disk	609
fdioctl.h	Control floppy-disk I/O	609
fdisk	Hard-disk partitioning utility	610
fdisk.h	Fixed-disk constants and structures	611
fdopen()	Open a stream for standard I/O	611
feof()	Discover stream status	612
ferror()	Discover stream status	613
fetch()	Fetch a record from a DBM data base	614
fflush()	Flush output stream's buffer	614
ffs()	Translate a bit mask into an integer value	615
fgetc()	Read character from stream	615
fgetpos()	Get value of file-position indicator	616

<code>fgets()</code>	Read line from stream	617
<code>fgetw()</code>	Read integer from stream	618
<code>field</code>		618
<code>file</code>	The way to access bits	619
<code>file</code>	Guess a file's type	619
<code>FILE</code>	Descriptor for a file stream	620
file descriptor		620
<code>fileno()</code>	Get file descriptor	620
<code>filsys.h</code>	Structures and constants for super block	621
<code>filter</code>		621
<code>find</code>	Search for files satisfying a pattern	621
<code>findmouse</code>	Examine a port to see if a mouse is plugged into it	623
<code>firstkey()</code>	Retrieve the first record from a DBM data base	623
<code>fixterm()</code>	Set the terminal into program mode	623
<code>float</code>	Data type	624
<code>float.h</code>	Define constants for floating-point numbers	627
<code>floor()</code>	Set a numeric floor	628
floppy disks		629
<code>fmap</code>	Measure fragmentation of the free list	632
<code>fmod()</code>	Calculate modulus for floating-point number	633
<code>fmt</code>	Adjust the length of lines in a file of text	633
<code>fnkey</code>	Set/print function keys for the console	633
<code>fnmatch()</code>	Match a string with a normal expression	634
<code>fnmatch.h</code>	Constants used with function <code>fnmatch()</code>	634
<code>fopen()</code>	Open a stream for standard I/O	634
<code>for</code>	Execute commands for tokens in list	636
<code>for</code>	Control a loop	636
<code>fork()</code>	Create a new process	636
<code>fortune</code>	Print randomly selected, hopefully humorous, text	637
<code>.forward</code>	Set a forwarding address for mail	637
<code>fpathconf()</code>	Get a file variable by file descriptor	638
<code>fperr.h</code>	Constants used with floating-point exception codes	639
<code>fprintf()</code>	Print formatted output into file stream	639
<code>fputc()</code>	Write character into file stream	640
<code>fputs()</code>	Write string into file stream	640
<code>fputw()</code>	Write an integer into a stream	641
<code>fread()</code>	Read data from file stream	641
<code>free()</code>	Return dynamic memory to free memory pool	641
<code>freemem</code>	Device that indicates amount of memory that is free	642
<code>freopen()</code>	Open file stream for standard I/O	642
<code>frexp()</code>	Separate fraction and exponent	643
<code>from</code>	Generate list of numbers, for use in loop	644
<code>fscanf()</code>	Format input from a file stream	644
<code>fsck</code>	Check and repair file systems interactively	645
<code>fseek()</code>	Seek on file stream	649
<code>fsetpos()</code>	Set file-position indicator	650
<code>fstat()</code>	Find attributes of an open file	651
<code>fstatfs()</code>	Get information about a file system	651
<code>ft</code>	Floppy-tape driver	652
<code>ftbad</code>	Manipulate bad-block list on a floppy-tape cartridge	653
<code>ftell()</code>	Return current position of file pointer	653
<code>ftime()</code>	Get the current time from the operating system	654
<code>ftok()</code>	Generate keys for interprocess communication	654
function		655
<code>fwrite()</code>	Write into file stream	655
<code>fwtable</code>	Build font-width table	655
<code>gawk</code>	Pattern-scanning and -processing language	657
<code>gcd()</code>	Set variable to greatest common divisor	669
<code>gcvt()</code>	Convert floating-point numbers to strings	670
<code>gdbm.h</code>	Header file for GDBM routines	670
<code>gdbm_close()</code>	Close a GDBM data base	671

<code>gdbm_delete()</code>	Delete a record from a GDBM data base	671
<code>gdbm_exists()</code>	Check whether a GDBM data base contains a given record	671
<code>gdbm_fetch()</code>	Retrieve a record from a GDBM data base	671
<code>gdbm_firstkey()</code>	Return the first record from a GDBM data base	672
<code>gdbm_nextkey()</code>	Return the next record from a GDBM data base	672
<code>gdbm_open()</code>	Open a GDBM data base.	673
<code>gdbm_reorganize()</code>	Reorganize a GDBM data base	674
<code>gdbm_setopt()</code>	Set GDBM options	674
<code>gdbm_store()</code>	Add records to a GDBM data base	675
<code>gdbm_strerror()</code>	Translate a GDBM error code into text	675
<code>gdbm_sync()</code>	Flush buffered GDBM data into its data base	675
<code>gdbmerrno.h</code>	Define error messages used by GDBM routines	676
<code>getc()</code>	Read character from file stream	677
<code>getchar()</code>	Read character from standard input	678
<code>getcwd()</code>	Get current working directory name	678
<code>getdents()</code>	Read directory entries	679
<code>getdtablesize()</code>	Get the number of files a process can open.	679
<code>getegid()</code>	Get effective group identifier	680
<code>getenv()</code>	Read environmental variable	680
<code>geteuid()</code>	Get effective user identifier	680
<code>getgid()</code>	Get real group identifier	681
<code>getgrent()</code>	Get group file information	681
<code>getgrgid()</code>	Get group file information, by group id	681
<code>getgrnam()</code>	Get group file information, by group name	682
<code>getgroups()</code>	Read the supplemental group-access list	682
<code>gethostbyaddr()</code>	Retrieve host information by address	682
<code>gethostbyname()</code>	Retrieve a host IP address by name	683
<code>gethostname()</code>	Get the name of the local host	683
<code>getlogin()</code>	Get login name	684
<code>getmap()</code>	De-archive Usenet map articles.	684
<code>getmsg()</code>	Get the next message from a stream.	684
<code>getnetbyaddr()</code>	Get a network entry by address.	686
<code>getnetbyname()</code>	Get a network entry by address.	686
<code>getnetent()</code>	Fetch a network entry	687
<code>getopt()</code>	Get option letter from argv	688
<code>getopts</code>	Parse command-line options	688
<code>getpass()</code>	Get password with prompting.	689
<code>getpeername()</code>	Get name of connected peer.	689
<code>getpgid()</code>	Get process-group identifier.	690
<code>getpid()</code>	Get process identifier.	690
<code>getppid()</code>	Get process identifier of parent process	690
<code>getprotobyname()</code>	Get protocol entry by protocol name	690
<code>getprotobynumber()</code>	Get protocol entry by protocol number	691
<code>getprotoent()</code>	Get protocol entry.	692
<code>getpw()</code>	Search password file	692
<code>getpwent()</code>	Get password file information.	692
<code>getpwnam()</code>	Get password file information, by name.	694
<code>getpwuid()</code>	Get password file information, by id	694
<code>gets()</code>	Read string from standard input	695
<code>getservbyname()</code>	Get a service entry by name.	695
<code>getservbyport()</code>	Get a service entry by port number	696
<code>getservent()</code>	Get a service entry	697
<code>getsockname()</code>	Get the name of a socket	697
<code>getsockopt()</code>	Read a socket option	698
<code>getspent()</code>	Get a shadow-password record	698
<code>getspnam()</code>	Get a shadow-password record, by user name	699
<code>gettimeofday()</code>	Berkeley time function	699
<code>getty</code>	Terminal initialization	699
<code>getuid()</code>	Get real user identifier	700
<code>getutent()</code>	Read an entry from a login logging file.	701
<code>getutid()</code>	Find a record in login logging file by login identifier.	701

getutline()	Find a record in login logging file by device.	702
getw()	Read word from file stream	702
GMT.		702
gmtime()	Convert system time to calendar structure	703
gnucpio.	Archiving/backup utility.	703
goto	Unconditionally jump within a function.	707
grep	Pattern search.	707
group	Define groups of users	708
grp.h	Declare group structure	709
gtar	Archiving/backup utility.	710
gtty()	Device-dependent control	715
guess	Extraordinarily amusing guessing game	715
gunzip	GNU utility to uncompress files	715
gzip	GNU utility to compress files	716
hai.	Host adapter-independent SCSI driver	719
hard disk.		722
hash.	Add a command to the shell's hash table.	725
hdioctl.h	Control hard-disk I/O	725
head.	Print the beginning of a file	726
header files.		726
help	Print concise description of command.	729
hmon	Monitor the COHERENT System	730
HOME.	User's home directory	733
hosts	Names and addresses of hosts on the local network	733
hosts.equiv.	Name equivalent hosts	733
hosts.lpd	Local system name and domain	734
hp	Prepare files for Hewlett-Packard LaserJet printer.	734
hpd	Spooler daemon for laser printer	734
hpr	Spool a job for printing on the laser printer	735
hpskip	Abort/restart current job on Hewlett-Packard LaserJet	736
hypot()	Compute hypotenuse of right triangle	736
i-node.	COHERENT system file identifier	738
icheck.	i-node consistency check	738
id	Print user and group IDs and names	739
idbld	Reconfigure the COHERENT kernel	739
ideinfo	Display information of an IDE hard-disk drive	739
idenable	Enable or disable a device driver	739
idle	Device that returns system's idle time.	740
idmkcohd	Build a new kernel	741
idtune.	Set a tunable system value	741
ieee_d()	Convert a double from DECVAX to IEEE format	742
ieee_f()	Convert a float from DECVAX to IEEE format	742
if.	Execute a command conditionally	742
if.	Introduce a conditional statement	743
IFS	Characters recognized as white space	743
index()	Find a character in a string	743
inet_addr()	Transform an IP address from text to binary	744
inet_network()	Transform an IP address from text to an integer.	744
inetd.conf.	Configure the Internet daemons	745
infocmp.	De-compile a terminfo file	745
init	System initialization	745
initgroups().	Initialize the supplementary group-access list	747
initialization		747
ino.h	Constants and structures for disk i-nodes	749
inode.h	Constants and structures for memory-resident i-nodes	749
install.	Install a software update onto COHERENT.	750
int	Data type.	751
interrupt		752
io.h	Constants and structures used by I/O	752
ioctl()	Device-dependent control	752
ipc.h.	Definitions for interprocess communications.	757

ipcrm	Remove an interprocess-communication memory item	757
ipcs	Display a snapshot of interprocess communications	758
IRQ	Interrupts on the IBM PC	760
isalnum()	Check if a character is a number or letter	761
isalpha()	Check if a character is a letter	761
isascii()	Check if a character is an ASCII character	761
isatty()	Check if a device is a terminal	762
iscntrl()	Check if a character is a control character	762
isdigit()	Check if a character is a numeral	762
isgraph()	Check if a character is printable	762
islower()	Check if a character is a lower-case letter	763
ispos()	Return if variable is positive or negative	763
isprint()	Check if a character is printable	763
ispunct()	Check if a character is a punctuation mark	763
isspace()	Check if a character prints white space	764
isupper()	Check if a character is an upper-case letter	764
isxdigit()	Check if a character is a hexadecimal numeral	764
itom()	Create a multiple-precision integer	765
j0()	Compute Bessel function	766
j1()	Compute Bessel function	767
jn()	Compute Bessel function	767
jobs	Print information about jobs	767
join	Join two data bases	767
rand48()	Return a 48-bit pseudo-random number as a long integer	768
kb.h	Define keys for loadable keyboard driver	769
kernel	Master program of the COHERENT system	769
keyboard	How COHERENT handles the console keyboard	772
kill	Signal a process	772
kill()	Kill a system process	773
ksh	The Korn shell	773
KSH_VERSION	List current version of Korn shell	789
.kshrc	Set personal environment for Korn shell	790
ktty.h	Kernel portion of tty structure	790
l	List directory's contents in long format	791
l.out.h	Format for COHERENT 286 objects	791
l3tol()	Convert file system block number to long integer	792
LASTERROR	Program that last generated an error	792
.lastlogin	Record of last login	792
Latin 1		792
lc	List directory's contents in columnar format	794
lcasep	Convert text to lower case	795
lcong48()	Initialize values from which 48-bit random numbers are computed	795
ld	Link relocatable object modules	795
ldexp()	Combine fraction and exponent	799
LDHEAD	Append options to beginning of ld command line	799
ldiv()	Perform long integer division	799
LDTAIL	Append options to end of ld command line	800
let	Evaluate an expression	800
lex	Lexical analyzer generator	800
Lexicon	Format of the COHERENT manual pages	802
lf	List directory's contents in columnar format	803
libc	Standard C library	803
libcurses	Library of screen-handling functions	810
libedit	Routines to gather and edit user input	822
libgdbm	Library for GNU DBM functions	823
libm	COHERENT mathematics library	825
libmisc	Library of miscellaneous functions	826
libmp	Library for multiple-precision mathematics	832
LIBPATH	Directories that hold compiler phases and libraries	834
libraries		835
libsocket	Library of communications routines	835

libterm	Functions to read termcap descriptions.	841
limits.h	Define numerical limits	842
lines	Highly amusing board game.	843
link()	Create a link.	843
listen()	Listen for a connection on a socket	844
lmail	Deliver mail on your local system	845
ln	Create a link to a file	845
localtime()	Convert system time to calendar structure	845
lockf()	Lock a file or a section of a file	847
log()	Compute natural logarithm	847
log10()	Compute common logarithm	848
login.	Log in a user.	849
login.	Set default values for logging in	852
loginlog.	Log of failed login attempts	852
logmsg	Hold COHERENT Login Message.	853
LOGNAME	Name user's identifier	853
long	Data type.	853
longjmp()	Perform a non-local goto.	853
look	Find matching lines in a sorted file	854
lp	Spool a file for printing.	854
lp	Driver for parallel ports	855
lpadmin.	Administer the lp print-spooler system	856
lpd.	Spooler daemon for line printer.	856
lpioctl.h	Definitions for line-printer I/O control	857
lpr	Spool a job for printing on the line printer	857
lpsched	Print jobs spooled with command lp; turn on printer daemon.	857
lpshut.	Turn off the printer daemon despooler	859
lpskip	Abort/restart current job on line printer	859
lpstat	Give status of printer or job.	859
lr	List subdirectories' contents in columnar format	860
lrand48()	Return a 48-bit pseudo-random number as a non-negative long integer	860
ls	List directory's contents	860
lseek()	Set read/write position.	861
lto30()	Convert long integer to file system block number	862
lvalue		862
lx	List directory's contents in columnar format.	863
m4.	Macro processor.	864
machine.h	Machine-dependent definitions	866
macro.		866
madd()	Add multiple-precision integers.	866
mail	Send or read mail.	866
mail	Electronic mail system.	868
mailq	Display information about spooled mail.	872
main()	Introduce program's main function	872
major number	Device numbering.	873
make	Program-building discipline.	873
makeboot.	Create a bootable floppy disk	878
makedepend.	Generate list of dependencies for a makefile	879
malloc()	Allocate dynamic memory	881
malloc.h	Definitions for memory-allocation functions	882
man	Manual macro package	882
man	Display Lexicon entries	884
manifest constant.		885
math.h	Declare mathematics functions.	885
MB_CUR_MAX.	Largest size of a multibyte character in current locale	885
mboot.	Master boot block for hard disk	885
mcd	Device driver for Mitsumi CD-ROM drives	886
mcmp()	Compare multiple-precision integers	886
mcopy()	Copy a multiple-precision integer	886
mdevice.	Describe drivers that can be linked into kernel	886
mdiv()	Divide multiple-precision integers	888

me	MicroEMACS screen editor	888
mem.	Physical memory file	894
memccpy()	Copy a region of memory up to a set character	894
memchr()	Search a region of memory for a character	895
memcmp()	Compare two regions.	896
memcpy()	Copy one region of memory into another	896
memmove()	Copy region of memory into area it overlaps	897
memok()	Test if the arena is corrupted	898
memset()	Fill an area with a character	898
mesg	Permit/deny messages from other users	899
min()	Read multiple-precision integer from stdin	899
minit().	Condition global or auto multiple-precision integer	899
minor number.	Device numbering.	900
mintfr()	Free a multiple-precision integer	900
mitom()	Reinitialize a multiple-precision integer	900
mkdbm	Build a data base for smail	900
mkdir	Create a directory.	901
mkdir()	Create a directory.	902
mkfifo()	Create a FIFO	902
mkfnames	Generate data base of user names	902
mkfs.	Make a new file system.	903
mkhpath	Build a pathalias data base from a hosts table.	905
mkline	Fold an alias file, paths file, or mailing list into one-line records	906
mklost+found	Make an enlarged lost+found directory	907
mknod	Make a special file or named pipe	907
mknod().	Create a special file.	908
mkpath	Create a pathalias output file	908
mksort	Sort the standard input, allowing arbitrarily long lines	909
mktemp()	Generate a temporary file name	910
mktime()	Turn broken-down time into calendar time.	910
MLP_COPIES	Set default number of copies to print	911
MLP_FORMLEN	Set default page length.	911
MLP_LIFE	Set default life for print jobs.	912
MLP_PRIORITY	Set default priority for print spooling	912
MLP_SPOOL	Pass user-specific information to print spooler.	912
mmu.h	Definitions for memory-management unit	912
mneg()	Negate multiple-precision integer.	913
mnttab	Mount table	913
mnttab.h	Structure for mount table	913
modem	913
modf().	Separate integral part and fraction.	917
modulus	918
mon.h.	Read profile output files	919
moo	Greatly amusing numeric guessing game	919
more	Display text one page at a time	919
motd	File that holds message of the day	921
mount.	Mount a file system.	921
mount.h	Define the mount table.	922
mount().	Mount a file system.	922
mount.all.	Mount file systems at boot time	923
mout().	Write multiple-precision integer to stdout	923
mprec.h.	Multiple-precision arithmetic	923
mrnd48()	Return a 48-bit pseudo-random number as a long integer.	923
ms.	Manuscript macro package	923
MS-DOS	That other operating system	925
msg	Kernel module for messages.	929
msg	Send a brief message to other users	929
msg.h	Definitions for message facility	930
msgctl().	Message control operations	930
msgget().	Create or get a message queue	931
msgrcv().	Receive a message	934

msgs	Read messages intended for all COHERENT users	935
msgsnd()	Send a message	936
msig.h	Machine-dependent signals	937
msqrt()	Compute square root of multiple-precision integer	937
msub()	Subtract multiple-precision integers	938
mtab.h	Currently mounted file systems	938
mtioctl.h	Magnetic-tape I/O control	938
mtoi()	Convert multiple-precision integer to integer	938
mtos()	Convert multiple-precision integer to string	938
mtune.	Define tunable kernel variables	939
mtype()	Return symbolic machine type	939
mtype.h	List processor code numbers	940
mult()	Multiply multiple-precision integers	940
mv.	Rename files or directories	940
mvdir	Rename a directory	940
mvfree()	Free multiple-precision integer	941
mwcbbs.	Download files from the Mark Williams bulletin board	941
n.out.h	Define n.out file structure	944
name space	C name-space rules	944
named pipe		946
nap()	Sleep briefly	947
ncheck	Print file names corresponding to i-node	947
ndbm.h	Header file for NDBM routines	947
netdb.h	Define structures used to describe networks	948
networks	Name remote networks	948
newaliases	Build the smail aliases data base from an ASCII source file	949
newgrp	Change to a new group	949
newusr	Add new user to COHERENT system	950
nextkey()	Retrieve the next record from a DBM data base	950
nm.	Print a program's symbol table	951
nohup.	Run a command immune to hangups and quits	951
nologin	Lock out logins	952
notmem()	Check whether memory is allocated	952
nptx.	Generate permutations of users' full names	953
nrand48()	Return a 48-bit pseudo-random number as a non-negative long integer	953
nroff.	Text-formatting language	953
NUL		963
NULL		963
null	The 'bit bucket'	963
nybble		963
object format.		964
od	Print an octal dump of a file	964
offsetof()	Offset of a field within a structure	964
open()	Open a file	965
opendir()	Open a directory stream	967
operator		968
PAGER	Specify Output Filter	971
param.h	Define machine-specific parameters	971
passwd	Set/change login password	971
passwd	Define system users	972
paste	Merge lines of files	972
patch	Patch a variable or flag within the kernel	973
PATH	Directories that hold executable files	975
path()	Path name for a file	975
path.h	Define/declare constants and functions used with PATH	976
pathalias	Generate a set of paths among computers	976
pathconf()	Get a file variable by path name	979
pathmerge	Merge sorted paths files	980
paths	Routing data base for mail	981
pattern		982
pause()	Wait for signal	982

pcfont	Prepare a PCL font for downloading via MLP	982
pclose()	Close a pipe	983
perror()	System call error messages	983
phone	Print numbers and addresses from phone directory.	984
pipe		984
pipe()	Open a pipe	984
pnmatch()	Match string pattern	986
pointer		986
poll()	Query several I/O devices	989
poll.h	Define structures/constants used with polling devices	990
popd	Pop an item from the directory stack	990
popen()	Open a pipe	991
port	File that describes ports for UUCP	991
portability		994
POSIX Standard.		994
pow()	Raise multiple-precision integer to power.	994
pow()	Compute a power of a number	995
pr	Paginate and print files	995
prep	Produce a word list	996
print.	Echo text onto the standard output	996
printer	How to attach and run a printer	997
printf()	Print formatted text.	1002
proc.h.	Define structures/constants used with processes.	1005
process		1005
prof	Print execution profile of a C program.	1005
profile.	Set default environment at login	1005
.profile	Execute commands at login	1006
Programming COHERENT.		1006
protocols	Name communications protocols.	1010
prps	Prepare files for PostScript-compatible printer	1011
ps	Print process status	1012
ps	Driver to return information about processes	1014
PS1	User's default prompt	1015
PS2	Prompt when user continues command onto additional lines	1015
PSfont.	Cook an Adobe font into PostScript format	1015
ptrace()	Trace process execution	1015
ptrace.h.	Perform process tracing	1016
pty.	Device driver for pseudoterminals	1017
pushd.	Push an item onto the directory stack.	1018
putc()	Write character into stream	1018
putchar()	Write a character onto the standard output	1019
putenv().	Add a string to the environment	1019
putmsg()	Place a message onto a stream	1020
putp()	Write a string into the standard window	1021
puts()	Write string onto standard output	1021
pututline()	Write a record into a logging file	1021
putw()	Write word into stream.	1022
pwd	Print the name of the current directory	1022
pwd.h.	Define password structure	1022
qfind	Quickly find all files with a given name	1024
qpac.	Map the file system.	1024
qsort().	Sort arrays in memory	1025
quot	Summarize file-system usage	1025
raise()	Let a process send a signal to itself	1027
ram	Driver for manipulating RAM	1028
ramdisk.	Script to create a RAM-disk	1029
rand()	Generate pseudo-random numbers	1030
RAND_MAX	Largest size of a pseudo-random number.	1031
random()	Return a random number	1031
random access.		1032
ranlib	Create index for object library.	1032

rc	Perform standard maintenance chores	1032
read-only memory		1033
read	Assign values to shell variables	1033
read()	Read from a file	1033
readdir()	Read a directory stream	1034
readline()	Read and edit a line of input	1034
readonly	Mark a shell variable as read only	1036
readonly	Storage class	1036
realloc()	Reallocate dynamic memory	1037
reboot	Reboot the COHERENT system	1037
recursion		1037
recv()	Receive a message from a connected socket	1038
recvfrom()	Receive a message from a socket	1039
ref	Display a C function header	1039
regcomp()	Compile a regular expression into a structure	1040
regerror()	Return an error message from a regular-expression function	1040
regexec()	Compare a string with a regular expression	1040
regexp.h	Header file for regular-expression functions	1041
register	Storage class	1042
register variable		1042
regsub()	Use regular expression to build a string	1042
remove()	Remove a file	1043
rename()	Rename a file	1043
reprint	Reprint a spooled print job	1044
resetterm()	Reset the terminal to its previous settings	1044
restor	Restore file system	1045
return	Return a value and control to calling function	1046
rev	Print text backwards	1047
rewind()	Reset file pointer	1047
rewinddir()	Rewind a directory stream	1047
rindex()	Find rightmost occurrence of a character in a string	1048
rm	Remove files	1048
rmail	Receive mail from remote sites	1049
rmdir	Remove directories	1050
rmdir()	Remove a directory	1050
root		1050
route	Show or reset a user's default printer	1050
routers	Rules for resolving mail addresses to remote systems	1051
rpow()	Raise multiple-precision integer to power	1053
RS-232	Serial port wiring	1053
rsmtplib	Run batched SMTP mail	1054
rubik	Play Rubik's cube	1055
runq	Periodically process the mail queue	1055
rvalue		1055
sa	Print a summary of process accounting	1056
savelog	Save a mail log	1057
sbrk()	Increase a program's data space	1058
scanf()	Accept and format input	1058
scat	Print text files one screenful at a time	1060
sched.h	Define constants used with scheduling	1062
script	Capture a terminal session into a file	1062
sdevice	Configure drivers included within kernel	1062
sdiv()	Divide multiple-precision integers	1063
SECONDS	Number of seconds since current shell started	1063
security		1063
sed	Stream editor	1064
seed48()	Initialize values from which 48-bit random numbers are computed	1067
seekdir()	Reset the position within a directory stream	1067
seg.h	Definitions used with segmentation	1067
select()	Check if devices are ready for activity	1067
sem	Kernel module for semaphores	1069

sem.h	Definitions used by semaphore facility	1070
semctl()	Control semaphore operations	1070
semget()	Create or get a set of semaphores	1071
semop()	Perform semaphore operations	1073
send()	Send a message to a socket	1075
sendto()	Send a message to a socket	1075
serialno	Hold the serial number of your system	1076
services	List supported TCP/IP services	1076
set	Set shell option flags and positional parameters	1077
setbuf()	Set alternative stream buffer	1078
setgid()	Set group id and user id	1078
setgrnt()	Rewind group file	1079
setgroups()	Set the supplemental group-access list	1079
sethostent()	Open and rewind file /etc/hosts	1080
setjmp()	Save machine state for non-local goto	1080
setjmp.h	Define setjmp() and longjmp()	1081
setnetent()	Open and rewind file /etc/networks	1081
setpgid()	Set the process-group identifier	1081
setpgrp()	Make a process a process-group leader	1082
setprotoent()	Open the protocols file	1082
setpwent()	Rewind password file	1082
setservent()	Open the services file	1082
setsid()	Set session identifier	1083
setsockopt()	Set a socket option	1083
setspent()	Rewind the shadow-password file	1084
setuid()	Set user identifier	1084
setupterm()	Initialize a terminal	1085
setutent()	Rewind the input stream for a login logging file	1085
setvbuf()	Set alternative file-stream buffer	1085
sgtty	General terminal interface	1086
sgtty.h	Definitions used to control terminal I/O	1090
sh	The Bourne shell	1090
shadow	File that holds restricted passwords	1101
shadow.h	Definitions used with shadow passwords	1102
SHELL	Name the default shell	1102
shellsort()	Sort arrays in memory	1102
shift	Shift positional parameters	1103
shm	Kernel module for shared memory	1103
shm.h	Definitions used with shared memory	1103
shmat()	Attach a shared-memory segment to a process	1103
shmctl()	Manipulate shared memory	1104
shmdt()	Detach a shared-memory segment from a process	1105
shmget()	Create or get shared-memory segment	1105
short	Data type	1109
shutdown	Shut down the COHERENT system	1109
shutdown()	Replace function to shut down system	1110
sigaction()	Perform detailed signal management	1110
sigaddset()	Add a signal to a set of signals	1111
sigdelset()	Delete a signal from a set	1111
sigemptyset()	Initialize a set of signals	1111
sigfillset()	Initialize a set of signals	1112
sighold()	Place a signal on hold	1112
sigignore()	Tell the system to ignore a signal	1112
sigismember()	Check if a signal is a member of a set	1113
siglongjmp()	Perform a non-local goto and restore signal mask	1113
signal()	Specify action to take upon receipt of a given signal	1113
signal.h	Define signals	1115
signame	Array of names of signals	1116
sigpause()	Pause until a given signal is received	1116
sigpending()	Examine signals that are blocked and pending	1117
sigprocmask()	Examine or change the signal mask	1117

sigrelse()	Release a signal for processing	1117
sigset()	Specify action to take upon receipt of a given signal	1118
sigsetjmp()	Save machine state and signal mask for non-local jump	1119
sigsuspend()	Install a signal mask and suspend process	1119
sin()	Calculate sine	1119
sinh()	Calculate hyperbolic sine	1120
size	Print size of an object file	1121
sizeof	Return size of a data element	1121
sleep	Stop executing for a specified time	1122
sleep()	Suspend execution for interval	1122
mail	Mail delivery system	1122
smtpd	SMTP daemon	1132
smult()	Multiply multiple-precision integers	1132
SOCKADDRLEN	Return length of an address	1132
socket()	Create a socket	1133
socket.h	Define constants and structures with sockets	1134
socketpair()	Create a pair of sockets	1134
sort	Sort lines of text	1134
spac	Sort a file system	1135
spell	Find spelling errors	1135
split	Split a text file into smaller files	1136
spow()	Raise multiple-precision integer to power	1137
sprintf()	Format output	1137
sqrt()	Compute square root	1137
srand()	Seed random number generator	1138
srand48()	Seed the 48-bit pseudo-random number routines	1139
srandom()	Seed the random-number generator	1139
srcpath	Find source files	1139
sscanf()	Format a string	1140
stack		1140
standard error		1140
standard input		1141
standard output		1141
stat()	Find file attributes	1141
stat.h	Definitions and declarations used to obtain file status	1142
statfs()	Get information about a file system	1143
static	Declare storage class	1143
stdarg.h	Header for variable numbers of arguments	1144
stddef.h	Header for standard definitions	1145
stderr		1145
stdin		1145
STDIO		1145
stdio.h	Declarations and definitions for I/O	1146
stdlib.h	Declare/define general functions	1147
stdout		1148
sticky bit		1148
stime()	Set the time	1148
storage class		1148
store()	Write a record into a DBM data base	1149
strcasecmp()	Case-insensitive string comparison	1149
strcasncmp()	Case-insensitive string comparison	1149
strcat()	Concatenate two strings	1149
strchr()	Find a character in a string	1150
strcmp()	Compare two strings	1150
strcoll()	Compare two strings, using locale-specific information	1150
strcpy()	Copy one string into another	1151
strcspn()	Return length a string excludes characters in another	1151
strdup()	Duplicate a string	1151
stream		1152
stream.h	Definitions for message facility	1152
STREAMS	COHERENT implementation of STREAMS	1152

strerror()	Translate an error number into a string.	1152
strftime()	Format locale-specific time	1153
string.h	Declarations for string library.	1154
strings	Print all character strings from a file.	1155
strip	Strip tables from executable file	1156
strlen()	Measure a string	1156
strncat()	Append one string onto another	1156
strncmp()	Compare two strings	1157
strncpy()	Copy one string into another	1157
stropts.h	User-level STREAMS routines.	1158
strpbrk()	Find first occurrence of a character from another string	1158
strchr()	Search for rightmost occurrence of a character in a string.	1159
strspn()	Return length a string includes characters in another	1159
strstr()	Find one string within another	1159
strtod()	Convert string to floating-point number.	1160
strtok()	Break a string into tokens.	1161
strtol()	Convert string to long integer.	1162
strtoul()	Convert string to unsigned long integer.	1162
struct	Data type.	1164
structure		1165
structure assignment		1165
strxfrm()	Transform a string using locale information	1165
stty	Set/print terminal modes	1166
stty()	Set terminal modes.	1170
stune	Set values of tunable kernel variables.	1170
su	Substitute user id, become superuser.	1171
sum	Print checksum of a file	1171
superuser		1171
swab()	Swap a pair of bytes	1171
switch	Test a variable against a table	1172
sync	Flush system buffers.	1172
sync()	Flush system buffers.	1173
sys	Data base for UUCP connections.	1173
sysconf()	Get configurable system variables	1186
sysi86()	Identify parts within Intel-based machines	1188
system()	Pass a command to the shell for execution	1189
tail	Print the end of a file	1190
tan()	Calculate tangent.	1190
tanh()	Calculate hyperbolic cosine	1191
tape	Magnetic-tape devices	1191
tape	Manipulate a tape device	1193
tar	Archiving/backup utility.	1194
tboot	Describe the tertiary bootstrap	1194
tcdrain()	Drain output to a device.	1195
tcfloor()	Control flow on a terminal device.	1195
tcfloor()	Flush data being exchanged with a terminal	1196
tcgetattr()	Get terminal attributes.	1197
tcsetattr()	Send a break to a terminal	1197
tee	Set terminal attributes.	1198
tee	Copy input to multiple output streams	1198
telldir()	Return the current position within a directory stream	1198
tempnam()	Generate a unique name for a temporary file.	1199
TERM	Name the default terminal type.	1199
term	Format of compiled terminfo file	1199
termcap	Terminal-description language	1200
terminal		1208
terminfo	Terminal-description language	1211
termio	General terminal interface.	1222
termio.h	Definitions used with terminal input and output	1228
termios	POSIX extended terminal interface.	1228
termios.h	Definitions used with POSIX extended terminal interface	1229

test	Evaluate conditional expression	1230
tgetent()	Read termcap entry	1232
tgetflag()	Get termcap Boolean entry	1232
tgetnum()	Get termcap numeric feature	1232
tgetstr()	Get termcap string entry	1233
tgoto()	Read/interpret termcap cursor-addressing string	1233
tic	Compile a terminfo description	1233
time		1234
time	Time the execution of a command	1236
time.h	Give time-description structure	1236
time()	Get current system time	1236
timeb.h	Define timeb structure	1236
timeout.h	Define the timer queue	1237
times	Print total user and system times	1237
times.h	Definitions used with times() system call	1237
times()	Obtain process execution times	1237
TIMEZONE	Time zone information	1238
TMPDIR	Directory that holds temporary files	1239
tmpfile()	Create a temporary file	1239
tmpnam()	Generate a unique name for a temporary file	1242
toascii()	Convert characters to ASCII	1242
tolower()	Convert characters to lower case	1243
touch	Update modification time of a file	1244
toupper()	Convert characters to upper case	1244
tparam()	Output a parameterized string	1244
tputs()	Read/decode leading padding information	1244
tr	Translate characters	1245
tr	Driver to read stored error messages	1245
transports	Describe mail transportation systems	1246
trap	Execute command on receipt of signal	1251
trigraph		1252
troff	Extended text-formatting language	1252
true	Unconditional success	1261
trustme	List of trusted users	1261
tsort	Topological sort	1261
ttt	Play 3-D tic-tac-toe	1262
tty	Print the user's terminal name	1262
tty.h	Define flags used with tty processing	1262
ttyname()	Identify a terminal	1262
ttys	Describe terminal ports	1263
ttyslot()	Return a terminal's line number	1264
ttystat	Get terminal status	1264
ttytype	Select a default terminal type for a port	1265
type checking		1265
type promotion		1266
typedef	Define a new data type	1266
types.h	Define system-specific data types	1266
typeset	Set/list variables and their attributes	1266
typo	Detect possible typographical and spelling errors	1267
tzset()	Set the local time zone	1267
ulimit()	Get/set limits for a process	1268
ulimit.h	Define manifest constants used by system call ulimit()	1268
umask	Set the file-creation mask	1269
umask()	Set file-creation mask	1269
umount	Unmount file system	1270
umount()	Unmount a file system	1270
unalias	Remove an alias	1270
uname	Print information about COHERENT	1270
uname()	Get the name and version of COHERENT	1271
uncompress	Uncompress a compressed file	1271
unctrl.h	Define macro unctrl()	1272

ungetc()	Return character to input stream	1272
union	Multiply declare a variable.	1272
uniq	Remove/count repeated lines in a sorted file	1273
unistd.h	Define constants for file-handling routines	1273
units	Convert measurements	1273
unlink()	Remove a file.	1274
unpack	GNU utility to uncompress files	1275
unset	Unset an environment variable or shell function.	1275
unsigned	Data type.	1275
until	Execute commands repeatedly	1275
unzip	Un-zip a zipped archive	1276
upac	De-fragment a file system without sorting	1276
update	Update file systems periodically	1277
uproc.h	Definitions used with user processes	1277
USER	Name user's identifier	1277
Using COHERENT		1277
usleep()	Sleep briefly	1278
usrtime	Times each user is permitted to log in.	1279
ustat()	Get statistics on a file system.	1280
utime()	Change file access and modification times	1281
utime.h	Declare system call utime()	1281
utmp	File that notes login events that are active	1281
utmp.h	Login accounting information.	1281
utmpname()	Manipulate a login logging file other than /etc/utmp.	1283
utsname.h	Define utsname structure	1283
uuchk	Check UUCP configuration.	1283
uucico	Communicate with a remote site.	1284
uuconv	Convert UUCP configuration files to Taylor format	1286
UUCP	Unattended communication with remote systems.	1286
uucp	Spool files for transmission to other systems.	1290
uucpname	Set the system's UUCP name	1291
uudecode	Decode a binary file sent from a remote system	1292
uuencode	Encode a binary file for transmission	1292
uuinstall	Install or modify UUCP	1293
uulog	Read a UUCP log.	1294
uumkdir	Create UUCP directories.	1294
uumvlog	Archive UUCP log files	1294
uuname	List UUCP names of known systems	1295
uupick	Pick up a file uploaded from a remote system	1295
uurmlock	Remove UUCP lock files.	1295
uusched	Call all systems that have jobs waiting for them.	1296
uustat	UUCP status inquiry and control.	1296
uuto	Send a file to a remote system	1299
uutouch	Touch a file to trigger UUCP poll	1299
uutry	Debugging script for UUCP.	1299
uux	Execute a command on a remote system	1299
uuxqt	Execute commands requested by a remote system	1302
va_arg()	Return pointer to next argument in argument list.	1304
va_end()	Tidy up after traversal of argument list	1304
va_start()	Point to beginning of argument list	1305
varargs.h	Declare/define routines for variable arguments	1305
vfprintf()	Print formatted text into stream	1306
vi	Clone of Berkeley-style screen editor.	1306
vidattr()	Set the terminal's video attributes.	1307
vidputs()	Write video attributes into a function	1307
view	Screen-oriented viewing utility	1307
virtual console	COHERENT system of multiple virtual consoles.	1308
void	Data type.	1309
volatile	Qualify an identifier as frequently changing	1310
vprintf()	Print formatted text into standard output stream	1310
vsh	Interactive graphical shell	1311

<code>vsprintf()</code>	Print formatted text into string	1326
<code>vtkb</code>	Non-configurable keyboard driver, virtual consoles	1327
<code>vtnkb</code>	Configurable keyboard driver, virtual consoles	1327
<code>wait</code>	Await completion of background process	1333
<code>wait.h</code>	Define wait routines	1333
<code>wait()</code>	Await completion of a child process	1333
<code>waitpid()</code>	Wait for a process to terminate	1334
<code>wall</code>	Send a message to all logged-in users	1334
<code>wc</code>	Count words, lines, and characters in text files	1335
<code>welcome</code>	Welcome a new user	1335
<code>whence</code>	List a command's type	1335
<code>whereis</code>	Locate source, binary, and manual files	1335
<code>which</code>	Locate executable files	1336
<code>while</code>	Execute commands repeatedly	1337
<code>while</code>	Introduce a loop	1337
<code>who</code>	Print who is logged in	1337
wildcards		1337
<code>write.</code>	Converse with another user	1338
<code>write()</code>	Write to a file	1338
<code>wtmp</code>	File that records past login events	1339
<code>xargs</code>	Execute a command with many arguments	1340
<code>xgcd()</code>	Extended greatest-common-divisor function	1340
<code>yacc</code>	Parser generator	1341
<code>yes.</code>	Print infinitely many responses	1342
<code>zcat</code>	Concatenate a compressed file	1343
<code>zcmp</code>	Compare compressed files	1343
<code>zdiff</code>	Compare two compressed files	1343
<code>zerop()</code>	Indicate if multi-precision integer is zero	1343
<code>zforce</code>	Force the suffix <code>.gz</code> onto every <code>gzip</code> file	1344
<code>zgrep</code>	Search compressed files for a regular expression	1344
<code>zip</code>	Zip files into a compressed archive	1344
<code>zmore</code>	Display compressed text one page at a time	1345
<code>znew</code>	Recompress <code>.Z</code> files to <code>.gz</code> files	1345