IEEE P1003.2 Draft 11.2 – September 1991

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

effort to contact the objector to resolve this and get a newly worded objection.) Further examples of unresponsive submittals are those not marked as either *Objection* or *Comment* ; those that do not identify the portion of the document that is being objected to (each objection must be separately labeled); those that object to material in a recirculation that has not changed and do not cite an unresolved objection; those that do not provide specific or general guidance on what changes would be required to resolve the objection.

(c) Finally, others are valid technical points, but they would result in decreasing the consensus of the Balloting Group. (This judgment is made based on other ballots and on the experiences of the working group through almost five years of work and fifteen drafts preceding this one.) These are referred to as *Unresolved Objections* . Summaries of unresolved objections and their reasons for rejection are maintained throughout the balloting process, are circulated to members of the Balloting Group for their consideration, and are presented to the IEEE Standards Board when the final draft is offered for approval. Unresolved objections are only circulated to the balloting group when they are presented by members of the balloting group or by parties of interest. Unsolicited correspondence from outside these two groups may result in draft changes, but are not recirculated to the balloting group members.

Please ensure that you correctly characterize your ballot by providing one of the following:

(1) Your IEEE member number

(2) Your IEEE Computer Society affiliate number

(3) If (1) or (2) don't apply, a statement that you are a ''Party of Interest''

Ballot Resolution

The general procedure for resolving ballots is:

(1) The balloting cuts off on 21 October 1991. This is a receipt date at the IEEE , not a postmark date. (Please do not telephone or FAX on 21 October 1991 and say that your specific comments will come later; late-arriving comments will not be considered as objections.) We will accept comments after that date, including direct e-mail to the working group officers or the Technical Reviewers, but they will be treated as comments only—not objections. And we don't guarantee a written response to these late submissions.

(2) The ballots are put online and distributed to the Technical Reviewers.

(3) If a ballot contains an objection, the balloter will be contacted individually by telephone, letter, or e-mail and the corrective action to be taken will be described (or negotiated). The personal contact will most likely not occur if the objection is very simple and obvious to fix or the balloter cannot be reached after a few reasonable attempts. Repeated failed

group

has already been closed. There are also a few ‘‘parties of interest’’ who are not 1
members of the IEEE or the Computer Society. Members of the Balloting Group 1
IEEE Standards Board

for approval. If the changes cause too many ballots to slip back into negative status, another resolution and recirculation cycle begins.

Balloting Guidelines

This section consists of guidelines on how to write and submit the most effective ballot possible. The activity of resolving balloting comments is difficult and time consuming. Poorly constructed comments can make that even worse.

We have found several things that can be done to a ballot that make our job more difficult than it needs to be, and likely will result in a less than optimal response to ballots that do not follow the form below. Thus it is to your advantage, as well as ours, for you to follow these recommendations and requirements.

If a ballot that significantly violates the guidelines described in this section comes to us, we will determine that the ballot is unresponsive, and simply ignore all the material in it.

Secondly, objections that don't contain a specification so that the correction to resolve the objection ''can be readily determined'' are also unresponsive and will be ignored.

(If we do recognize a ballot that is generally ''unresponsive,'' we will try to inform the balloter as soon as possible so he/she can correct it, but it is ultimately the balloter's responsibility to assure the ballot is responsive.)

Typesetting is not particularly useful to us. And please do not send handwritten ballots. Typewritten (or equivalent) is fine, and if some font information is lost it

204

205

206

on both sides. Each objection/comment/editorial comment should be sequentially numbered, not in individual ranges [i.e., not Objection #1, Comment #1]

Since we deal with the ballots online, there is no longer any requirement to put only one objection or section per page.

Don't format the ballot as a letter or document with its *own* section numbers. These are simply confusing. As shown below, it is best if you cause each objection and comment to have a sequential number that we can refer to amongst ourselves and to you over the phone. Number sequentially from 1 and count objections, comments, and editorial comments the same; don't number each in its own range. If you don't do this, we'll number them ourselves, but you won't know what numbers we're using.

Please precede each objection/comment with a little code line (if you don't, we'll have to do it ourselves):

@ *<section>*.*<clause>* *<code>* *<seqno>*

where:

@	At-sign in column 1 (which means no @'sin any other column 1's).
<section>	The major section (chapter or annex) number or letter in column 3. Use zero for Global or for something, like the frontmatter, that has no section or annex number.
<clause>	The clause number (second-level header). Please do not go deeper than these two levels. In the text of your objection or comment, go as deep as you can in describing the location, but this code line uses two levels only.
<code>	One of the following lowercase letters, preceded and followed by spaces: `o` Objection. `c` Comment or Editorial Comment.
<seqno>	A sequence number, counting all objections and comments in a single range.

Objection:

```
Balloter Name             (202)555-1212       page  x of nn.
E-Mail Address            FAX:   Fax Number
Balloter2 Name                   (303)555-1213
E-Mail Address2           FAX:   Fax Number2
------------------------------------------------------------------
@ x.y o seq#
<Seq#> Sect x.y OBJECTION. page xxx, line zzz:
```

`Problem:`

A clear statement of the problem that is observed, sufficient for others to understand the nature of the problem. Note that you should identify problems by section, page, and line numbers. This may seem redundant, but if you transpose a digit pair, we may get totally lost without a cross-check like this. Use the line number where the problem starts, not just where the section itself starts; we sometimes attempt to sort objections by line numbers to make editing more accurate. If you are referring to a range of lines, please don't say ‘‘lines 1000ff;’’ use a real range so we can tell where to stop looking. If you have access to the online versions of a balloting draft, please do not send in a ballot that refers to the page numbers in the `nroff` output version; use only the line and page numbers found in the printed draft or the online PostScript draft. We will really love you if you can manage to include enough context information in the problem statement (such as the name of the utility) so we can understand it without having the draft in our laps at the time. (It also helps you when we e-mail it back to you.) If you are objecting to an action in the Unresolved Objections List, use the section/page/line number reference for the appropriate place in the standard; don't refer to the UOL except to cite its number and for clarification of your points.

`Action:`

A precise statement of the actions to be taken on the document to resolve the objection above, which if taken verbatim will completely remove the objection.

If there is an acceptable range of actions, any of which will resolve the problem for you if taken exactly, please indicate all of them. If we accept any of these, your objection will be considered as resolved.

If the Action section is omitted or is vague in its solution, the objection will be reclassified as a nonbinding comment. The Technical Reviewers, being human, will give more attention to Actions that are well-described than ones that are vague or imprecise. The best ballots of all have very explicit directions to substitute, delete, or add text in a style consistent with the rest of the document, such as:

Delete the sentence on lines 101-102:

"The implementation shall not ... or standard error."

On line 245, change "shall not" to "should not".

After line 103, add:

```
-r    Reverse the order of bytes read from the file.
```

Some examples of poorly-constructed actions:

Remove all features of this command that are not supported by BSD.

Add -i.

Make this command more efficient and reliable.

Use some other flag that isn't so confusing.

I don't understand this section.

Specify a value--I don't care what.

Objection Example:

```
Hal Jespersen             (415) 364-3410            page  3 of 17.
UUCP: hlj@Posix.COM  FAX:   (415) 364-4498
-------------------------------------------------------------------
@ 2.6 o 23
23. Sect 2.6 OBJECTION. page 77, line 1217:

Problem:

The EDITOR environment variable is not used as stated
in my company.  This description would cause hundreds
of my shell scripts to break.

Action:

Change the first sentence on line 1217 to:

      The e-mail address of the editor of the user's
      favorite POSIX standard.

-----------------------
@ 3.1 o 24
24. Sect 3.1.6 OBJECTION. page 123, line 17:

Problem:

I support UO 3.01-999-6 concerning the objection to the                1
definition of "operator".                                              1
This definition would cause great hardship to the users                1
of the systems I develop.                                              1
I feel your rationale for rejection was inappropriate                  1
because you overlooked the following technical points [etc.]...        1

Action:

Change the term "operator" to "operation-symbol" in this
definition and globally throughout Section 3.
```

Comment:

```
----------------------------------------------------------------
@ x.z c seq#
<Seq#> Sect x.z COMMENT. page xxx, line zzz:
```

A statement of a problem that you might want to be resolved by the reviewer, but which does not in any way affect whether your ballot is negative or positive. The form for objections is not required, but it increases the probability that your comment will have an effect on the final document.

Although there may be questions to you or responses on the topic, no changes in the drafts are required by a comment, although it will be looked at to determine whether the concern should be addressed. It is possible to abuse this rule and label all of your comments as objections, but it is a significant disservice to the individuals who are volunteering their time to address your concerns.

Remember that any issue concerning the pages preceding page 1 (the Frontmatter), Rationale text with shaded margins, Annexes, NOTES in the text, footnotes, or examples will be treated as a nonbinding comment whether you label it that way or not, but it would help us if you'd label it correctly.

Editorial Comment:

```
----------------------------------------------------------------
@ x.z c seq#
<Seq#> Sect x.z EDITORIAL COMMENT. page xxx, line zzz:
```

These are for strictly editorial issues, where the technical meaning of the document is not changed. Examples are: typos; misspellings; English syntax or usage errors; appearances of lists or tables; arrangement of sections, clauses, and subclauses (except where the location of information changes the optionality of a feature). Marking these as comments but indicating that they are editorial speeds the process.

Please be aware that after balloting concludes the document will be subjected to more sets of editors at the IEEE and ISO who are empowered to make broad editorial changes and rewording (for example, to get the text ready for translation into French.)

Thank you for your cooperation in this important balloting process.

Hal Jespersen

Identifier Index

[	**test** — Evaluate expression {4.62}	631
ar	**ar** — Create and maintain library archives {6.1}	687
asa	**asa** — Interpret carriage-control characters {C.1}	813
awk	**awk** — Pattern scanning and processing language {4.1}	263
basename	**basename** — Return nondirectory portion of pathname {4.2}	297
bc	**bc** — Arbitrary-precision arithmetic language {4.3}	301
break	**break** — Exit from **for**, **while**, or **until** loop {3.14.1}	246
c89	**c89** — Compile Standard C programs {A.1}	726
case	**case** Conditional Construct {3.9.4.3}	227
cat	**cat** — Concatenate and print files {4.4}	318
cd	**cd** — Change working directory {4.5}	322
chgrp	**chgrp** — Change file group ownership {4.6}	326
chmod	**chmod** — Change file modes {4.7}	329
chown	**chown** — Change file ownership {4.8}	337
cksum	**cksum** — Write file checksums and sizes {4.9}	341
cmp	**cmp** — Compare two files {4.10}	347
colon	colon — Null utility {3.14.2}	247
comm	**comm** — Select or reject lines common to two files {4.11}	350
command	**command** — Execute a simple command {4.12}	354
confstr ()	C Binding for Get String-Valued Configurable Variables {B.10.1}	809
continue	**continue** — Continue **for**, **while**, or **until** loop {3.14.3}	248
cp	**cp** — Copy files {4.13}	359
cut	**cut** — Cut out selected fields of each line of a file {4.14}	368
date	**date** — Write the date and time {4.15}	373
dd	**dd** — Convert and copy a file {4.16}	379
diff	**diff** — Compare two files {4.17}	388
dirname	**dirname** — Return directory portion of pathname {4.18}	395
dot	**dot** — Execute commands in current environment {3.14.4}	248
echo	**echo** — Write arguments to standard output {4.19}	399
ed	**ed** — Edit text {4.20}	402
env	**env** — Set environment for command invocation {4.21}	419
eval	**eval** — Construct command by concatenating arguments {3.14.5}	249
exec	**exec** — Execute commands and open, close, and/or copy file descriptors {3.14.6}	250
exit	**exit** — Cause the shell to exit {3.14.7}	251
export	**export** — Set export attribute for variables {3.14.8}	252
expr	**expr** — Evaluate arguments as an expression {4.22}	423

false	**false** — Return false value {4.23}	428
find	**find** — Find files {4.24}	430
fnmatch ()	C Binding for Match Filename or Pathname {B.6}	794
fold	**fold** — Fold lines {4.25}	438
for	**for** Loop {3.9.4.2}	226
fort77	**fort77** — FORTRAN compiler {C.2}	817
fpathconf ()	C Binding for Get Numeric-Valued Configurable Variables {B.10.2}	811
getconf	**getconf** — Get configuration values {4.26}	442
getenv ()	C Binding for Access Environment Variables {B.4}	786
getopt ()	C Binding for Command Option Parsing {B.7}	796
getopts	**getopts** — Parse utility options {4.27}	447
glob ()	C Binding for Generate Pathnames Matching a Pattern {B.8}	799
glob_t	Description {B.8.2}	799
grep	**grep** — File pattern searcher {4.28}	452
head	**head** — Copy the first part of files {4.29}	459
id	**id** — Return user identity {4.30}	462
if	**if** Conditional Construct {3.9.4.4}	228
join	**join** — Relational database operator {4.31}	466
kill	**kill** — Terminate or signal processes {4.32}	471
lex	**lex** — Generate programs for lexical tasks {A.2}	736
ln	**ln** — Link files {4.33}	476
locale	**locale** — Get locale-specific information {4.34}	480
localedef	**localedef** — Define locale environment {4.35}	486
logger	**logger** — Log messages {4.36}	491
logname	**logname** — Return user's login name {4.37}	494
lp	**lp** — Send files to a printer {4.38}	496
ls	**ls** — List directory contents {4.39}	502
mailx	**mailx** — Process messages {4.40}	510
make	**make** — Maintain, update, and regenerate groups of programs {6.2}	695
mkdir	**mkdir** — Make directories {4.41}	514
mkfifo	**mkfifo** — Make FIFO special files {4.42}	518
mv	**mv** — Move files {4.43}	521
nohup	**nohup** — Invoke a utility immune to hangups {4.44}	526
od	**od** — Dump files in various formats {4.45}	530
paste	**paste** — Merge corresponding or subsequent lines of files {4.46}	538
pathchk	**pathchk** — Check pathnames {4.47}	543
pathconf ()	C Binding for Get Numeric-Valued Configurable Variables {B.10.2}	811
pax	**pax** — Portable archive interchange {4.48}	548
pclose ()	C Binding for Pipe Communications with Programs {B.3.2}	782
popen ()	C Binding for Pipe Communications with Programs {B.3.2}	782
pr	**pr** — Print files {4.49}	562

`printf`	**printf** — Write formatted output {4.50}	568
`pwd`	**pwd** — Return working directory name {4.51}	574
`read`	**read** — Read a line from standard input {4.52}	576
`readonly`	**readonly** — Set read-only attribute for variables {3.14.9}	253
regcomp ()	C Binding for Regular Expression Matching {B.5}	786
regerror ()	C Binding for Regular Expression Matching {B.5}	786
regexec ()	C Binding for Regular Expression Matching {B.5}	786
regex_t	Description {B.5.2}	786
regfree ()	C Binding for Regular Expression Matching {B.5}	786
regmatch_t	Description {B.5.2}	786
regoff_t	Description {B.5.2}	786
`return`	**return** — Return from a function {3.14.10}	254
`rm`	**rm** — Remove directory entries {4.53}	579
`rmdir`	**rmdir** — Remove directories {4.54}	584
`sed`	**sed** — Stream editor {4.55}	587
`set`	**set** — Set/unset options and positional parameters {3.14.11}	254
`sh`	**sh** — Shell, the standard command language interpreter {4.56}	597
`shift`	**shift** — Shift positional parameters {3.14.12}	258
`sleep`	**sleep** — Suspend execution for an interval {4.57}	603
`sort`	**sort** — Sort, merge, or sequence check text files {4.58}	605
`strip`	**strip** — Remove unnecessary information from executable files {6.3}	716
`stty`	**stty** — Set the options for a terminal {4.59}	613
sysconf ()	C Binding for Get Numeric-Valued Configurable Variables {B.10.2}	811
system ()	C Binding for Execute Command {B.3.1}	778
`tail`	**tail** — Copy the last part of a file {4.60}	623
`tee`	**tee** — Duplicate standard input {4.61}	628
`test`	**test** — Evaluate expression {4.62}	631
`touch`	**touch** — Change file access and modification times {4.63}	640
`tr`	**tr** — Translate characters {4.64}	645
`trap`	**trap** — Trap signals {3.14.13}	258
`true`	**true** — Return true value {4.65}	652
`tty`	**tty** — Return user's terminal name {4.66}	654
`umask`	**umask** — Get or set the file mode creation mask {4.67}	657
`uname`	**uname** — Return system name {4.68}	662
`uniq`	**uniq** — Report or filter out repeated lines in a file {4.69}	665
`unset`	**unset** — Unset values and attributes of variables and functions {3.14.14}	260
`wait`	**wait** — Await process completion {4.70}	669
`wc`	**wc** — Word, line, and byte count {4.71}	674
`while`	**until** Loop {3.9.4.6}	229
`while`	**while** Loop {3.9.4.5}	229

wordexp ()	C Binding for Perform Word Expansions {B.9}	804
wordexp_t	Description {B.9.2}	804
`xargs`	**`xargs`** — Construct argument list(s) and invoke utility {4.72}	678
`yacc`	**`yacc`** — Yet another compiler compiler {A.3}	750

Alphabetic Topical Index

A

A/2047 ... 597
/ ... 109
// ... 297, 300, 395, 398
[
  definition of ... 631
Abbreviations ... 50
ABCDEF ... 170, 742
ABC ... 572
absolute pathname
  definition of ... 26
access control
  additional ... 361
  alternate ... 361
Access Environment Variables ... 720, 844
ACK ... 68, 74
ACK ... 56, 618
ACM ... 346
Actions Equivalent to POSIX. 1 Functions ... 144
Actions ... 275
{ACTSIZE} ... 764
ACUTE ... 100-101
adb ... 833
ADD_ASSIGN ... 282, 286-287
address space
  definition of ... 26
ADD_ASSIGN ... 290
affirmative response ... 26, 92, 104, 360, 362-363, 432, 434, 485, 487, 521, 524, 555, 579-581
  definition of ... 26
A-F ... 317
Aim of Character Mnemonics ... 884
<alert>
  definition of ... 26
Algorithms ... 764
Allow Historical Conforming Applications ... 6
AM/PM ... 89
AM/PM ... 374
AND_IF ... 235-236
AND ... 282, 285-287
AND Lists ... 225
AND/OR lists ... 218, 222, 225
AND-OR ... 216
AND/OR ... 218
AND-OR ... 222
AND ... 146, 222, 225, 254-255, 268-269, 290, 296, 433, 525, 583
angle brackets
  definition of ... 26
ANSI ... 826
a.out ... 726-727, 732, 735, 816-818, 822-823
Append Command ... 408
Appending Redirected Output ... 210
APPEND ... 282, 284
APPEND ... 290
Application Conformance ... 15
application
  definition of ... 48
apply ... 681
appropriate privileges ... 26, 32-33, 41, 329, 336, 462, 486-487, 557, 559, 561
  definition of ... 26
ar
  — Create and maintain library archives ... 687, 843
    ... 687-690, 692-694, 728, 732, 818-819
  definition of ... 687
ARFLAGS ... 707-708
ARGC-1 ... 292
ARGC
  awk variable ... 271
ARGC ... 265, 271, 292-293
{ARG_MAX} ... 175-176, 179, 185, 216, 678, 682-683
{ARG_MAX} ... 723
Argument Processing with *getopt* () ... 798
argument
  definition of ... 26
argv ... 26

`ARGV`
  `awk` variable ... 271

ARGV ... 265, 271, 292-293

Arithmetic Expansion ... 205

Arithmetic Functions ... 278

Arithmetic Precision and Operations ... 145

ARPANET ... 513

`asa`
  — Interpret carriage control characters ... 846
  — Interpret carriage-control characters ... 813
    ... 813-816
  definition of ... 813

ASCII ... 74, 102

ASCII to EBCDIC Conversion ... 385

ASCII to IBM EBCDIC Conversion ... 386

ASCII ... 48-49, 58-59, 129-130, 132, 385-387, 536-538, 621, 693, 816

ASSIGNMENT ... 233

ASSIGNMENT _WORD ... 234

ASSIGNMENT _WORD ... 235, 237

ASSIGN _OP ... 303, 305

ASSIGN _OP ... 22, 306

asterisk
  definition of ... 26

Asynchronous Lists ... 224

`at` ... 833

AT&T ... 9-10, 557, 572

`awk`
  — Pattern scanning and processing language ... 263, 838
    ... 5, 25, 41, 44, 60, 129, 144, 155, 158, 161, 163, 165, 172, 178, 208, 263-267, 270-275, 277, 279, 281, 288, 290-291, 293-296, 317, 573, 748
  Arithmetic Functions ... 278
  definition of ... 263
  Escape Sequences ... 289
  Expressions ... 267
  Expressions in Decreasing Precedence ... 268
  Functions ... 277
  Grammar ... 281
  Input/Output and General Functions ... 280
  Lexical Conventions ... 288
  Output Statements ... 276
  Patterns ... 274
  Regular Expressions ... 273
  String Functions ... 278
  User-Defined Functions ... 281
  Variables and Special Variables ... 271

AWK ... 9, 293-294, 826

A-Z ... 650

B

B.3 ... 784

background process group
  definition of ... 27

background process
  definition of ... 26

background ... 26-27, 36

background ... 26-27, 35, 129, 159, 163, 193, 221, 224, 476, 617, 672-673, 829

backquote
  definition of ... 27

BACKREF ... 122-123

backslash
  definition of ... 27

`<backspace>`
  definition of ... 27

Balloting Instructions ... 919

`basename`
  — Return nondirectory portion of pathname ... 297, 838
    ... 298-300, 398

basename
  definition of ... 27

`basename`
  definition of ... 297

basic regular expression
  definition of ... 27

Basic Regular Expressions ... 112

`bc`
  — Arbitrary-precision arithmetic language ... 301, 838
    ... 25, 44, 155, 173-174, 301-303, 305, 308, 311-317, 573, 834
  definition of ... 301
  Grammar ... 303
  Lexical Conventions ... 305
  Operations ... 307
  Operators ... 307

{BC_BASE _MAX} ... 176, 308

BC_BASE _MAX ... 173-174

{BC_BASE _MAX} ... 174

BC_BASE _MAX ... 775

{BC_BASE _MAX} ... 775, 811

{BC_DIM _MAX} ... 175-176, 307

BC_DIM_MAX ... 173-174

{BC_DIM_MAX} ... 174

BC_DIM_MAX ... 775

{BC_DIM_MAX} ... 775, 811

{BC_SCALE_MAX} ... 175-176, 307

BC_SCALE_MAX ... 173-174

{BC_SCALE_MAX} ... 174

BC_SCALE_MAX ... 775

{BC_SCALE_MAX} ... 775, 811

{BC_STRING_MAX} ... 305

BC_STRING_MAX ... 173-174

{BC_STRING_MAX} ... 174

BC_STRING_MAX ... 775

{BC_STRING_MAX} ... 775, 811

BEGIN ... 282

BEGIN ... 264-265, 267, 272, 274-275, 289, 292, 294-295, 744

BEL ... 56, 618

Bibliography ... 825

`/bin` ... 109

blank line
  definition of ... 27

`<blank>`
  definition of ... 27

block special file ... 351
  definition of ... 27

BNF ... 10

BODY ... 756

braces
  definition of ... 27

bracket expression ... 113

brackets
  definition of ... 28

`break` ... 191, 228, 247, 275
  definition of ... 246

BRE [ERE] matching a single character
  definition of ... 111

BRE [ERE] matching multiple characters
  definition of ... 111

BRE Expression Anchoring ... 116

BRE Ordinary Characters ... 112

BRE Precedence ... 116-117

BRE Special Characters ... 112

BRE
  abbreviation ... 50

BRE/ERE Grammar Lexical Conventions ... 121

BRE/ERE ... 121

BRE ... 110-113, 115-117, 127-128, 136, 453-454

BREs Matching a Single Character or Collating Element ... 112

BREs Matching Multiple Characters ... 115

BRKINT ... 615

BSD/32V ... 596

BSD
  4.2 ... 387
  4.3 ... 315, 336, 340, 365, 393-394, 397, 436, 525, 529
  4.4 ... 693

BSD ... 4-5, 9, 47, 181, 187, 213, 314, 320-321, 325, 328, 335-336, 339-340, 345, 365, 378, 387, 394, 401-402, 416-418, 437-438, 458, 462, 465-466, 470, 479, 493, 501, 508-510, 514, 525, 536-538, 557-559, 567, 595-596, 621, 626-627, 630, 635-639, 644-645, 650-651, 681, 694, 708-710, 712, 714, 734, 749, 832, 837

BUFSIZ ... 177

Built-in Utilities ... 51, 830

built-in utility
  definition of ... 28

`builtin` ... 357

BUILTIN_FUNC_NAME ... 282, 286-287

built-in ... 21, 28, 30, 36, 45-46, 51-52, 176, 181-182, 192, 214, 217, 219-221, 226, 230-231, 241, 246, 248, 252-253, 255, 259, 261, 263, 273-274, 277, 282, 289-290, 293, 325, 354, 356

builtin ... 357

built-in ... 358, 419, 422, 431, 450, 452, 476, 527, 529, 547, 576, 578-579, 601-602, 653, 660, 672, 679, 695-696, 699, 703-704, 711-712, 830, 832

BUILTIN_FUNC_NAME ... 289

byte
  definition of ... 28

C

C Binding for Access Environment Variables ... 786, 845

C Binding for Command Option Parsing ... 796, 845

C Binding for Execute Command ... 778

C Binding for Generate Pathnames Matching a Pattern ... 799, 845

C Binding for Get Numeric-Valued Configurable Variables ... 811

C Binding for Get POSIX Configurable Variables ... 809, 845

C Binding for Get String-Valued Configurable Variables ... 809

C Binding for Locale Control ... 812, 846

C Binding for Match Filename or Pathname ... 794, 845

C Binding for Perform Word Expansions ... 804, 845

C Binding for Pipe Communications with Programs ... 782

C Binding for Regular Expression Matching ... 786, 845

C Binding for Shell Command Interface ... 778, 845

C Bindings for Numeric-Valued Configurable Variables ... 811

C Compile-Time Symbolic Constants ... 776

C Execution-Time Symbolic Constants ... 777

C Language Bindings Option ... 771, 845

C Language Definitions ... 772, 845

C Language Development Utilities Option ... 725, 844

C Macros for Symbolic Limits ... 775

C Numerical Limits ... 775, 845

C Shell ... 181, 202, 232, 245, 325, 475, 637

C Standard Operators and Functions ... 146

C Standard ... 3, 22, 28, 44-46, 48-50, 62, 71, 81, 86-87, 106, 129-131, 134, 144-146, 150, 154, 171, 175, 177, 206, 267-270, 275, 278, 288, 293-294, 296, 321, 378, 537, 547, 570, 572-573, 630, 677, 726, 733, 736, 740, 748, 750, 754, 772-774, 778-780, 782, 792-793
  definition of ... 50

`c89`
  — Compile Standard C programs ... 726, 845
    ... 4, 154, 533, 537, 712, 726-727, 729-735, 739, 745, 750, 753, 763-764, 766, 822, 824, 833, 835
  definition of ... 726

can
  definition of ... 23

CAN ... 68, 74

CAN ... 56, 618

carriage-control characters ... 814

`<carriage-return>`
  definition of ... 28

case conversion ... 67

`case` ... 191, 206, 216, 227-228, 234, 239, 243
  Conditional Construct ... 227
  definition of ... 227

`cat`
  — Concatenate and print files ... 318, 838
    ... 158, 251, 318-322, 595, 630
  definition of ... 318

C_BIND ... 179, 446, 730

`cc` ... 154, 712, 733-734

CCITT ... 513

CC
  variable ... 162

`cd`
  — Change working directory ... 322, 838
    ... 51-52, 107, 143, 220, 240-241, 322-325, 356
  definition of ... 322

C_DEV ... 179-180, 777, 811

CDPATH
  variable ... 107, 323-325

CFLAGS ... 706-708, 711, 713

Change Command ... 408

Changing the Current Working Directory ... 143

character attributes ... 67

character case conversion ... 67

character class expression ... 114

character class
  definition of ... 29

character classification ... 67

Character Mnemonics Classes ... 885

Character Mnemonics Guidelines ... 884

Character Set and Symbolic Names ... 54

Character Set Description File ... 55

Character Set ... 54, 830

character set
  definition of ... 48
  portable ... 54

character special file ... 351
  definition of ... 29

character
  definition of ... 28

charmap file ... 55-56, 58-60, 65, 73, 77, 99, 481, 483-484, 486-490, 621, 847, 850, 887, 912

CHARMAP ... 56, 99

CHAR ... 94

{CHAR_BIT} ... 46, 418

{CHAR_MAX} ... 86-87

CHARSET
  variable ... 849

CHARSYMBOL ... 94

CHARSYMBOL ... 93

chdir () ... 143

`chgrp`
  — Change file group ownership ... 326, 839
    ... 326-328, 340
  definition of ... 326

{CHILD_MAX} ... 176, 224, 672

{CHILD_MAX} ... 723

`chmod`
  — Change file modes ... 329, 839
    ... 3, 5, 329-330, 334-336, 436, 508, 515, 517-518, 657-658, 661
  definition of ... 329
  Grammar ... 333

chmod () ... 3, 32-33, 141, 334, 336

`chown`
  — Change file ownership ... 337, 839
    ... 337-340
  definition of ... 337

chown () ... 326, 337, 339-340

CH-1211 ... 12, 825

C_IDENTIFIER ... 759-760, 768

circumflex
  definition of ... 29

`cksum`
  — Write file checksums and block counts ... 839
  — Write file checksums and sizes ... 341
    ... 5, 341-343, 345-346, 709, 837
  definition of ... 341

Clean Up the Interfaces ... 4

{CLK_TCK} ... 723

CLK_TCK ... 443

CLOBBER ... 235, 238

CLOCAL ... 615

`cmp`
  — Compare two files ... 347, 839
    ... 347-350
  definition of ... 347

Code file ... 752

`col` ... 833

collating element
  definition of ... 29

collating symbol ... 114

`collating-element`
  Keyword ... 73

`collating-symbol`
  Keyword ... 77

Collation Order ... 78

collation sequence
  definition of ... 29

collation sequences
  defining ... 72

collation
  definition of ... 29

COLLELEMENT ... 94-96

COLLELEMENT ... 93

COLL_ELEM ... 122, 124

{COLL_WEIGHTS_MAX} ... 29, 72, 77, 79, 177, 489, 847

COLL_WEIGHTS_MAX ... 173-174

{COLL_WEIGHTS_MAX} ... 174

COLL_WEIGHTS_MAX ... 775

{COLL_WEIGHTS_MAX} ... 775, 811

COLLSYMBOL ... 94-96

colon — Null utility ... 247

`colon`
  definition of ... 247

column position
  definition of ... 30

COLUMNS
  variable ... 107, 503-504

Combination Modes ... 618

`comm`
  — Select or reject lines common to two files ... 350, 839
    ... 6, 350-353
  definition of ... 350

command language interpreter
  definition of ... 30

Command Option Parsing ... 721, 844

Command Search and Execution ... 219

Command Substitution ... 203

`command`
  — Execute a simple command ... 354
  — Select or reject lines common to two files ... 839
    ... 51-52, 107, 182, 215, 220, 354-358, 422, 529, 682

command
  definition of ... 30

`command`
  definition of ... 354

`command.c` ... 713

COMMENT_CHAR ... 711

compile C programs ... 726

compile FORTRAN programs ... 817

Compile-Time Symbolic Constants for Portability Specifications ... 776

Completing the Program ... 763

Compound Commands ... 226

Concepts Derived from the C Standard ... 144

Concurrent Execution of Processes ... 139

Configuration Values ... 173, 831

Conflicts ... 761

conformance document
  definition of ... 23

Conformance ... 12, 830

conformance ... 2-3, 11-17, 23-24, 53, 107, 138, 153, 159-160, 317, 417, 461, 501, 626, 687, 719, 725, 771, 813, 826, 830, 843-844

conforming application ... 6, 103, 128, 147, 149, 161, 166, 182, 598, 650, 693, 726, 733, 747, 817

Conforming Implementation Options ... 14

Conforming POSIX.2 Application Using Extensions ... 16

Conforming POSIX.2 Application ... 15

confstr () ... 443, 445, 722, 781, 793, 809-810
  definition of ... 809

`confstr`()
  name Values ... 809

Consequences of Shell Errors ... 214

`continue` ... 247-248, 275
  definition of ... 248

Control Character Set ... 56

Control Modes ... 614

control operator
  definition of ... 183

controlling terminal ... 138

Conventions ... 19, 830

`CONVFMT`
  `awk` variable ... 271

CONVFMT ... 269, 276, 295

Coordinated Universal Time (UTC) ... 378

Copy Command ... 414

`core` ... 713

Covered Coded Character Sets ... 884

`cp`
  — Copy files ... 359, 839
    ... 359-367, 480, 525, 558, 560, 584
  definition of ... 359

`cpio` ... 66, 552, 557-561, 692

CRC ... 341, 343, 345-346

CREAD ... 615

creat () ... 32, 387, 559, 640

`create` ... 547

`cron` ... 833

CSA ... 99

`csh` ... 529

CSMA/CD ... 825

`csplit` ... 833-834

_CS_PATH ... 107, 356-357, 443, 809

CS_PATH ... 107, 356-357, 443, 809

CSTOPB ... 615

csysconf () ... 810

`<ctype.h>` ... 730

current working directory ... 138
  definition of ... 30, 46

`cut`
  — Cut out selected fields of each line of a file ... 368, 839
    ... 176, 179, 368, 370-372, 441, 541-542
  definition of ... 368

{CUT_FIELD_MAX} ... 176

{CUT_LINE_MAX} ... 176

C_VERSION ... 774, 776

D

`da_DK`
  — (Example) Danish National Locale ... 850

Danish Locale Model ... 848

date formats ... 88

`date`
  — Write the date and time ... 373, 839
    ... 5, 88, 91, 106, 257, 373, 375-378, 505, 691
  definition of ... 373

DATE ... 377

DBL_MANT_DIG ... 537

DC1 . . . 68, 74

DC2 . . . 68, 74

DC3 . . . 68, 74

DC4 . . . 68, 74

`dc` . . . 315-316, 834

DC1 . . . 56, 618

DC2 . . . 56, 618

DC3 . . . 56, 618

DC4 . . . 56, 618

`dd`
  — Convert and copy a file . . . 379, 839
    . . . 5, 379-387, 559, 833, 837
  definition of . . . 379

DEAD
  variable . . . 107, 511, 514

Debugging the Parser . . . 764

DECIMAL_CHAR . . . 94

Declarations Section . . . 754

DECR . . . 282, 286-287

DECR . . . 290

Default Rules . . . 706

DEFAULT . . . 706

Definitions . . . 23, 830

Delete Command . . . 408

DEL . . . 68, 76

DEL . . . 56, 618

Dependencies on Other Standards . . . 138, 830

{DEPTH_MAX} . . . 176-178

`/dev` . . . 109

`/dev/null` . . . 109-110, 214, 224, 458, 511, 514, 805-806, 808

`/dev/tty` . . . 109-110, 139, 213, 550, 554, 561, 567

DGREAT . . . 235, 238

DIAERESIS . . . 100

`diff`
  — Compare two files . . . 388, 839
    . . . 5, 388-395, 682, 836
  `c` or `C` Output Format . . . 391
  Default Output Format . . . 390
  definition of . . . 388
  Directory Comparison Format . . . 390
  `e` Output Format . . . 391

DIGIT . . . 746

{DIGIT} . . . 746

directory entry . . . 30-31, 33, 36-38, 40, 141, 165, 476-477, 502, 579, 581, 583-586, 841
  definition of . . . 30

directory
  current working . . . 30
  definition of . . . 30
  empty . . . 31
  parent . . . 38
  root . . . 42
  working . . . 46

directory . . . 19-20, 30-34, 36, 38-39, 42, 46-47, 52, 103-105, 107, 109, 138-143, 155, 158-159, 164, 167, 176, 194-195, 197-198, 218, 221, 231, 240-241, 245, 248-249, 260, 297, 299-300, 322-326, 328-329, 331, 334-337, 339-340, 356-357, 359-360, 363-367, 371, 388, 390, 393-395, 398, 403-404, 430-431, 433, 436, 438, 445-446, 461, 476-477, 479-480, 502, 504, 506-509, 511, 514-517, 520-523, 525-527, 541, 543, 545, 547-550, 553, 557-561, 566, 574-575, 579-580, 582-585, 587, 598-599, 612, 631-632, 637, 639, 688, 698, 706, 726-727, 729, 733, 750, 767, 780, 795, 799-801, 810, 817-818, 820, 822, 834, 838-839, 841-842, 847

`dirname`
  — Return directory portion of pathname
    . . . 395, 839
    . . . 300, 396-398
  definition of . . . 395
  Examples . . . 398

DIS . . . 884

DIV_ASSIGN . . . 282, 286-287

DIV_ASSIGN . . . 290

DK-2900 . . . 847

DLE . . . 68, 74

DLE . . . 56, 618

DLESSDASH . . . 235, 238

DLESS . . . 235, 238

`do` . . . 226-227, 296

document
  conformance . . . 23

Documentation . . . 13

documentation
  system . . . 24

document . . . 1-3, 9-16, 21, 23-25, 35, 49, 55, 60, 63, 66, 125, 149, 152-153, 159-161, 168, 177, 182, 185, 188-189, 201, 203, 211, 213-214, 218, 228, 234, 239-240, 243, 257, 294, 321, 336, 353, 387-388, 417, 427-428, 437, 458, 470, 475, 480, 496, 501, 595-596, 611-613, 627, 639, 677, 693, 725, 733-734, 740, 747, 764, 768-769, 774, 797, 801, 822-823, 825-826, 836

dollar-sign
  definition of ... 30
`done` ... 226-227
`dot`
  — Execute commands in current environment ... 248
dot ... 31, 34, 39
`dot` ... 231, 248-249
dot
  definition of ... 30
`dot`
  definition of ... 248
dot-dot ... 31, 34, 38-39
  definition of ... 30
double-quote
  definition of ... 30
Double-Quotes ... 186
DOUBLE ... 707
DSEMI ... 235, 237
DUP _COUNT ... 122-125
Duplicating an Input File Descriptor ... 211
Duplicating an Output File Descriptor ... 212
DUP _COUNT ... 121

E

E.4 ... 7, 843-844, 846
[EACCES] ... 107
EBCDIC ... 58-59, 385-387
[ECHILD] ... 783-784
`echo`
  — Write arguments to standard output
    ... 399, 839
    ... 5, 47, 188
`echo:` ... 292
`echo` ... 399-402, 493, 509, 572-573, 679, 709, 715, 807
  definition of ... 399
ECHOE ... 616
ECHOK ... 617
ECHONL ... 617
ECHO ... 616, 744
ECMA ... 885
`ed`
  — Edit text ... 402, 839
    ... 6, 25, 44, 125-126, 141, 160-161, 177, 278, 388, 391, 402-405, 407, 410, 412, 416-418, 537, 551-552, 560, 595, 748, 834
  Addresses ... 405
  Commands ... 407
  definition of ... 402
  Regular Expressions ... 405
`ed.hup` ... 404
Edit Command ... 408
Edit Without Checking Command ... 409
Editorial Conventions ... 19
EDITOR
  variable ... 108
{ED_FILE _MAX} ... 160, 177, 417
{ED_LINE _MAX} ... 177, 417
effective group ID ... 31, 33, 36, 42, 44, 138-140, 463, 465, 597
  definition of ... 31
effective user ID ... 31, 34, 42, 45, 138-140, 329, 463-465, 597
  definition of ... 31
EFL ... 748
`egrep` ... 125, 127, 453, 457, 747
Eighth Edition UNIX ... 357
[EINTR]
EINTR ... 19, 781, 785
[EINVAL] ... 783, 810
`elif` ... 229
ELLIPSIS ... 94-96
`else` ... 229, 296
empty directory
  definition of ... 31
empty line
  definition of ... 31
empty string
  definition of ... 31
END ... 68, 76, 84, 88-89, 92, 95-99, 101-102, 282
END ... 56, 64, 99, 265, 267, 272, 274-275, 289, 292, 294, 486
[ENOENT] ... 357, 422, 529, 682
[ENOEXEC] ... 107, 219-220
ENQ ... 68, 74
ENQ ... 56, 618
entire regular expression
  definition of ... 110
`env`
  — Set environment for command invocation
    ... 419, 839
    ... 51, 215, 357, 419-422, 485, 529, 682
  definition of ... 419
`ENVIRON`
  `awk` variable ... 271