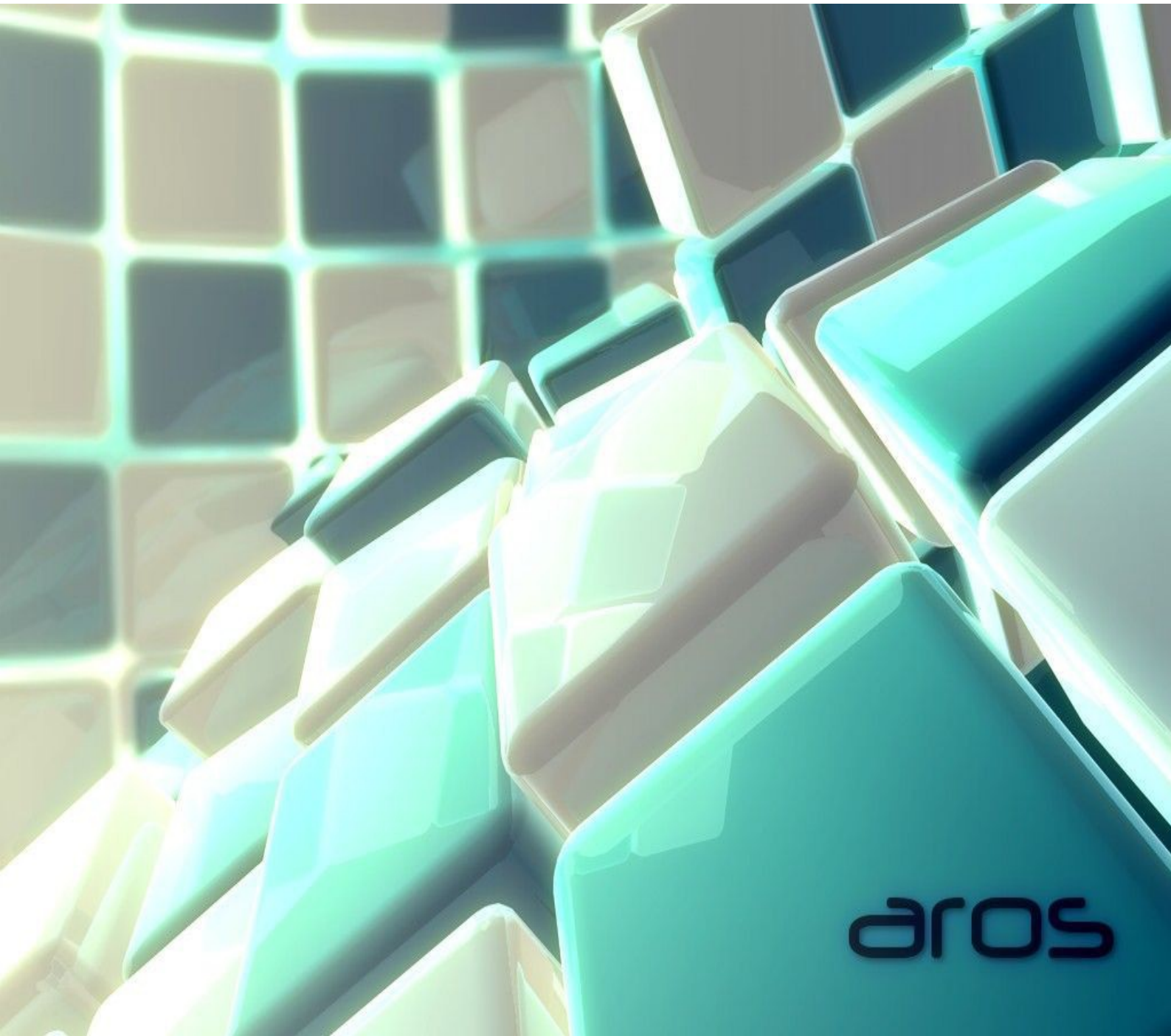
AROS GNU Tools



This manual includes documentation distributed under different terms (above all: GPL, LGPL and Creative Commons), taken from various sources but, mainly, from online manuals and referring websites. Informations provided in this manual can be wrong, outdated or partially not applicable to AROS due to differences between POSIX compatible OS-es and AROS itself.

ABC-Shell introduction chapter is an extract from
http://www.cs.mun.ca/~michael/pdksh/pdksh-man.html
some of the informations provided there may not apply to ABC-Shell, even if this one is based on pdksh.

Informations provided here have been included for reference. All copyrights retained by respective owners.

TABLE OF CONTENTS

Indice generale

HOW TO USE THIS MANUAL

Welcome

This manual contains all available man pages for development shell utilities included in AROS' Development/bin drawer. All these commands have been ported from their POSIX counterparts, so their sysopsis will always be "Unix-Like command".

They are not meant to be used in AROS or AmigaOS-centric shell scripts, because even their availability on AROS is not assured: you can find them in nightly builds as long as they get compiled, but distributors might decide to cut them off to save space. Users, after all, are not supposed to use them. Icaros Desktop, on the other hand, forces the user to install them even if development files have been unchecked during OS installation, because some of its scripts work only if these programs are installed. In a nutshell: if you plan to include them in a script, be always sure that users will have them installed, or your program might fail.

Keep these simple notes always in mind if you're planning to use them in a script. Many times AROS already provides a native command which does exactly the same job. For instance, you should never use 'kill' when AROS command that kills processes is 'break'. You should use 'dir' instead of 'ls', and so on. There are also commands which share the same name, but use a different syntax. One of them, for example, is the 'cut' command: when writing a script be always sure to use the right one.

Did you know...
This image regards some more curious aspect of AROS itself or related subject.

Warning:
This kind of image is intended to get your attention regarding some procedure or behaviour which might require caution from your part.

NOTE:
This image regards something that you should know in order to perform an operation on the best possible manner.

Example/Tutorial
This underlines an example of an integrated or complex operation performed under AROS.

Hardware related:
It pinpoints some important points regarding hardware support, behaviour or configuration.

Other useful or reference documentation

You should also read the following manuals:

AROS Shell Manual

AROS User Manual

Also it's recommended that you read other more actively maintained documentation, which is kept in http://www.aros.org/.

ABC-SHELL

AROS-DOS features a CLI (Command Line Interface) which shares the same advantages and usage of the AmigaDOS one.
And like AmigaDOS it combines all the best aspects of Unix and DOS Command Line Interfaces, with some exclusive features.

AROS shell, however, might not be enough to build your sources, nor to support GNU tools completely. In this case, you can use the SH command in an AROS shell to call ABC-Shell, a bourne shell compatible environment based on pdksh.

Did you know that AmigaOS was the first pre-emptive multi-tasking operating system for the "Personal Computer" concept?

Launched in 1985 the Amiga 1000 sported the AmigaOS which unlike it's 16-bit counterparts (IBM pc, Apple Mac and Atari st) was a true pre-emptive multi-tasking operating system taking advantage of the Custom Chip architecture.

The latest AmigaOS offering during the Commodore Amiga era was the 3.1 version in 1993 and offered a more stable and flexible OS that endured for many years and got different additions. This is regarded as the main fact that kept Amiga community active even after the Commodore demise.

Synopsis

sh [**+-abCefhikmnprsuvxX**] [**+-o** *option*] [[**-c** *command-string* [*command-name*] | **-s** | *file*] [*argument* ...]]

Shell Startup

The following options can be specified only on the command line:

-c *command-string*

the shell executes the command(s) contained in *command-string*

-i

interactive mode - see below

-l

login shell - see below interactive mode - see below

-s

the shell reads commands from standard input; all non-option arguments are positional parameters

-r

restricted mode - see below

In addition to the above, the options described in the **set** built-in command can also be used on the command line.

If neither the **-c** nor the **-s** options are specified, the first non-option argument specifies the name of a file the shell reads commands from; if there are no non-option arguments, the shell reads commands from standard input. The name of the shell (*i.e.*, the contents of the **$0**) parameter is determined as follows: if the **-c** option is used and there is a non-option argument, it is used as the name; if commands are being read from a file, the file is used as the name; otherwise the name the shell was called with (*i.e.*, argv[0]) is used.

A shell is **interactive** if the **-i** option is used or if both standard input and standard error are attached to a tty. An interactive shell has job control enabled (if available), ignores the INT, QUIT and TERM signals, and prints prompts before reading input (see **PS1** and **PS2** parameters). For non-interactive shells, the **trackall** option is on by default (see **set** command below).

A shell is **restricted** if the **-r** option is used or if either the basename of the name the shell is invoked with or the **SHELL** parameter match the pattern *r*sh (*e.g.*, rsh, rksh, rpdksh, *etc.*). The following restrictions come into effect after the shell processes any profile and **$ENV** files:

- the **cd** command is disabled
- the **SHELL**, **ENV** and **PATH** parameters can't be changed
- command names can't be specified with absolute or relative paths
- the **-p** option of the **command** built-in can't be used
- redirections that create files can't be used (*i.e.*, **>**, **>|**, **>>**, **<>**)

A shell is **privileged** if the **-p** option is used or if the real user-id or group-id does not match the effective user-id or group-id (see *getuid*(2), *getgid*(2)). A privileged shell does not process $HOME/.profile nor the **ENV** parameter (see below), instead the file /etc/suid_profile is processed. Clearing the privileged option causes the shell to set its effective user-id (group-id) to its real user-id (group-id).

If the basename of the name the shell is called with (*i.e.*, argv[0]) starts with **-** or if the **-l** option is used, the shell is assumed to be a login shell and the shell reads and executes the contents of **/etc/profile** and **$HOME/.profile** if they exist and are readable.

If the **ENV** parameter is set when the shell starts (or, in the case of login shells, after any profiles are processed), its value is subjected to parameter, command, arithmetic and tilde substitution and the resulting file (if any) is read and executed. If **ENV** parameter is not set (and not null) and pdksh was compiled with the **DEFAULT_ENV** macro defined, the file named in that macro is included (after the above mentioned substitutions have been performed).

The exit status of the shell is 127 if the command file specified on the command line could not be opened, or non-zero if a fatal syntax error occurred during the execution of a script. In the absence of fatal errors, the exit status is that of the last command executed, or zero, if no command is executed.

Command Syntax

The shell begins parsing its input by breaking it into *word*s. Words, which are sequences of

characters, are delimited by unquoted *white-space* characters (space, tab and newline) or *meta-characters* (**<**, **>**, **|**, **;**, **&**, **(** and **)**). Aside from delimiting words, spaces and tabs are ignored, while newlines usually delimit commands. The meta-characters are used in building the following tokens: **<**, **<&**, **<<**, **>**, **>&**, **>>**, *etc.* are used to specify redirections (see Input/Output Redirection below); **|** is used to create pipelines; **|&** is used to create co-processes (see Co-Processes below); **;** is used to separate commands; **&** is used to create asynchronous pipelines; **&&** and **||** are used to specify conditional execution; **;;** is used in **case** statements; **((** .. **))** are used in arithmetic expressions; and lastly, **(** .. **)** are used to create subshells.

White-space and meta-characters can be quoted individually using backslash (****), or in groups using double (**"**) or single (**'**) quotes. Note that the following characters are also treated specially by the shell and must be quoted if they are to represent themselves: ****, **"**, **'**, **#**, **$**, **`**, **~**, **{**, **}**, *, **?** and **[**. The first three of these are the above mentioned quoting characters (see Quoting below); **#**, if used at the beginning of a word, introduces a comment - everything after the **#** up to the nearest newline is ignored; **$** is used to introduce parameter, command and arithmetic substitutions (see Substitution below); **`** introduces an old-style command substitution (see Substitution below); **~** begins a directory expansion (see Tilde Expansion below); **{** and **}** delimit *csh*(1) style alternations (see Brace Expansion below); and, finally, *, **?** and **[** are used in file name generation (see File Name Patterns below).

As words and tokens are parsed, the shell builds commands, of which there are two basic types: *simple-commands*, typically programs that are executed, and *compound-commands*, such as **for** and **if** statements, grouping constructs and function definitions.

A simple-command consists of some combination of parameter assignments (see Parameters below), input/output redirections (see Input/Output Redirections below), and command words; the only restriction is that parameter assignments come before any command words. The command words, if any, define the command that is to be executed and its arguments. The command may be a shell built-in command, a function or an *external command*, *i.e.*, a separate executable file that is located using the **PATH** parameter (see Command Execution below). Note that all command constructs have an *exit status*: for external commands, this is related to the status returned by *wait*(2) (if the command could not be found, the exit status is 127, if it could not be executed, the exit status is 126); the exit status of other command constructs (built-in commands, functions, compound-commands, pipelines, lists, *etc.*) are all well defined and are described where the construct is described. The exit status of a command consisting only of parameter assignments is that of the last command substitution performed during the parameter assignment or zero if there were no command substitutions.

Commands can be chained together using the **|** token to form *pipelines*, in which the standard output of each command but the last is piped (see *pipe*(2)) to the standard input of the following command. The exit status of a pipeline is that of its last command. A pipeline may be prefixed by the **!** reserved word which causes the exit status of the pipeline to be logically complemented: if the original status was 0 the complemented status will be 1, and if the original status was not 0, then the complemented status will be 0.

Lists of commands can be created by separating pipelines by any of the following tokens:

&&, **||**, **&**, **|&** and **;**. The first two are for conditional execution: *cmd1* **&&** *cmd2* executes *cmd2* only if the exit status of *cmd1* is zero; **||** is the opposite - *cmd2* is executed only if the exit status of *cmd1* is non-zero. **&&** and **||** have equal precedence which is higher than that of **&**, **|&** and **;**, which also have equal precedence. The **&** token causes the preceding command to be executed asynchronously, that is, the shell starts the command, but does not wait for it to complete (the shell does keep track of the status of asynchronous commands - see Job Control below). When an asynchronous command is started when job control is disabled (*i.e.*, in most scripts), the command is started with signals INT and QUIT ignored and with input redirected from /dev/null (however, redirections specified in the asynchronous command have precedence). The **|&** operator starts a *co-process* which is special kind of asynchronous process (see Co-Processes below). Note that a command must follow the **&&** and **||** operators, while a command need not follow **&**, **|&** and **;**. The exit status of a list is that of the last command executed, with the exception of asynchronous lists, for which the exit status is 0.

Compound commands are created using the following reserved words - these words are only recognized if they are unquoted and if they are used as the first word of a command (*i.e.*, they can't be preceded by parameter assignments or redirections):

case	**else**	**function**	**then**	**!**
do	**esac**	**if**	**time**	**[[**
done	**fi**	**in**	**until**	**{**
elif	**for**	**select**	**while**	**}**

Note: Some shells (but not this one) execute control structure commands in a subshell when one or more of their file descriptors are redirected, so any environment changes inside them may fail. To be portable, the **exec** statement should be used instead to redirect file descriptors before the control structure.

In the following compound command descriptions, command lists (denoted as *list*) that are followed by reserved words must end with a semi-colon, a newline or a (syntactically correct) reserved word. For example,

{ echo foo; echo bar; }
{ echo foo; echo bar<newline>}
{ { echo foo; echo bar; } }

are all valid, but

{ echo foo; echo bar }

is not.

(*list* **)**

Execute *list* in a subshell. There is no implicit way to pass environment changes from a subshell back to its parent.

{ *list* **}**

Compound construct; *list* is executed, but not in a subshell. Note that **{** and **}** are reserved words, not meta-characters.

case *word* **in** [[**(**] *pattern* [**|** *pattern*] ... **)** *list* **;;**] ... **esac**

The **case** statement attempts to match *word* against the specified *pattern*s; the *list* associated with the first successfully matched pattern is executed. Patterns used in **case** statements are the same as those used for file name patterns except that the restrictions regarding **.** and / are dropped. Note that any unquoted space before and after a pattern is stripped; any space with a pattern must be quoted. Both the word and the patterns are subject to parameter, command, and arithmetic substitution as well as tilde substitution. For historical reasons, open and close braces may be used instead of **in** and **esac** (*e.g.*, **case $foo { *) echo bar; }**). The exit status of a **case** statement is that of the executed *list*; if no *list* is executed, the exit status is zero.

for *name* [**in** *word ... term*] **do** *list* **done**

where *term* is either a newline or a **;**. For each *word* in the specified word list, the parameter *name* is set to the word and *list* is executed. If **in** is not used to specify a word list, the positional parameters (**"$1"**, **"$2"**, *etc.*) are used instead. For historical reasons, open and close braces may be used instead of **do** and **done** (*e.g.*, **for i; { echo $i; }**). The exit status of a **for** statement is the last exit status of *list*; if *list* is never executed, the exit status is zero.

if *list* **then** *list* [**elif** *list* **then** *list*] ... [**else** *list*] **fi**

If the exit status of the first *list* is zero, the second *list* is executed; otherwise the *list* following the **elif**, if any, is executed with similar consequences. If all the lists following the **if** and **elif**s fail (*i.e.*, exit with non-zero status), the *list* following the **else** is executed. The exit status of an **if** statement is that of non-conditional *list* that is executed; if no non-conditional *list* is executed, the exit status is zero.

select *name* [**in** *word ... term*] **do** *list* **done**

where *term* is either a newline or a **;**. The **select** statement provides an automatic method of presenting the user with a menu and selecting from it. An enumerated list of the specified *words* is printed on standard error, followed by a prompt (**PS3**, normally `**#?** '). A number corresponding to one of the enumerated words is then read from standard input, *name* is set to the selected word (or is unset if the selection is not valid), **REPLY** is set to what was read (leading/trailing space is stripped), and *list* is executed. If a blank line (*i.e.*, zero or more **IFS** characters) is entered, the menu is re-printed without executing *list*. When *list* completes, the enumerated list is printed if **REPLY** is null, the prompt is printed and so on. This process is continues until an end-of-file is read, an interrupt is received or a break statement is executed inside the loop. If **in** *word* **...** is omitted, the positional parameters are used (*i.e.*, **"$1"**, **"$2"**, *etc.*). For historical reasons, open and close braces may be used instead of **do** and **done** (*e.g.*, **select i; { echo $i; }**). The exit status of a **select** statement is zero if a break statement is used to exit the loop, non-zero otherwise.

until *list* **do** *list* **done**

This works like **while**, except that the body is executed only while the exit status of the first *list* is non-zero.

while *list* **do** *list* **done**

A **while** is a prechecked loop. Its body is executed as often as the exit status of the first *list* is zero. The exit status of a **while** statement is the last exit status of the *list* in the body of the loop; if the body is not executed, the exit status is zero.

function *name* **{** *list* **}**

Defines the function *name*. See Functions below. Note that redirections specified after a function definition are performed whenever the function is executed, not when the function definition is executed.

name **()** *command*

Mostly the same as **function**. See Functions below.

time [**-p**] [*pipeline*]

The **time** reserved word is described in the Command Execution section.

((*expression* **))**

The arithmetic expression *expression* is evaluated; equivalent to **let "***expression***"**. See Arithmetic Expressions and the **let** command below.

[[*expression* **]]**

Similar to the **test** and **[** ... **]** commands (described later), with the following exceptions:

- Field splitting and file name generation are not performed on arguments.
- The **-a** (and) and **-o** (or) operators are replaced with **&&** and ||, respectively.
- Operators (*e.g.*, **-f**, **=**, **!**, *etc.*) must be unquoted.
- The second operand of **!=** and = expressions are patterns (*e.g.*, the comparison in

  [[foobar = f*r]]

  succeeds).
- There are two additional binary operators: **<** and **>** which return true if their first string operand is less than, or greater than, their second string operand, respectively.
- The single argument form of **test**, which tests if the argument

has non-zero length, is not valid - explicit operators must be always be used, *e.g.*, instead of

[*str* **]**

use

[[-n *str* **]]**

- Parameter, command and arithmetic substitutions are performed as expressions are evaluated and lazy expression evaluation is used for the **&&** and || operators. This means that in the statement

[[-r foo && $(< foo) = b*r]]

the **$(< foo)** is evaluated if and only if the file **foo** exists and is readable.

Quoting

Quoting is used to prevent the shell from treating characters or words specially. There are three methods of quoting: First, **** quotes the following character, unless it is at the end of a line, in which case both the **** and the newline are stripped. Second, a single quote (**'**) quotes everything up to the next single quote (this may span lines). Third, a double quote (**"**) quotes all characters, except **$**, **`** and ****, up to the next unquoted double quote. **$** and **`** inside double quotes have their usual meaning (*i.e.*, parameter, command or arithmetic substitution) except no field splitting is carried out on the results of double-quoted substitutions. If a **** inside a double-quoted string is followed by ****, **$**, **`** or **"**, it is replaced by the second character; if it is followed by a newline, both the **** and the newline are stripped; otherwise, both the **** and the character following are unchanged.

Note: see POSIX Mode below for a special rule regarding sequences of the form **"…`…\"…`.."**.

Aliases

There are two types of aliases: normal command aliases and tracked aliases. Command aliases are normally used as a short hand for a long or often used command. The shell expands command aliases (*i.e.*, substitutes the alias name for its value) when it reads the first word of a command. An expanded alias is re-processed to check for more aliases. If a command alias ends in a space or tab, the following word is also checked for alias expansion. The alias expansion process stops when a word that is not an alias is found, when a quoted word is found or when an alias word that is currently being expanded is found.

The following command aliases are defined automatically by the shell:

autoload='typeset -fu'
functions='typeset -f'
hash='alias -t'
history='fc -l'
integer='typeset -i'
local='typeset'
login='exec login'
newgrp='exec newgrp'
nohup='nohup '
r='fc -e -'
stop='kill -STOP'
suspend='kill -STOP $$'
type='whence -v'

Tracked aliases allow the shell to remember where it found a particular command. The first time the shell does a path search for a command that is marked as a tracked alias, it saves the full path of the command. The next time the command is executed, the shell checks the saved path to see that it is still valid, and if so, avoids repeating the path search. Tracked aliases can be listed and created using **alias -t**. Note that changing the **PATH** parameter clears the saved paths for all tracked aliases. If the **trackall** option is set (*i.e.*, **set -o trackall** or **set -h**), the shell tracks all commands. This option is set automatically for non-interactive shells. For interactive shells, only the following commands are automatically tracked: **cat**, **cc**, **chmod**, **cp**, **date**, **ed**, **emacs**, **grep**, **ls**, **mail**, **make**, **mv**, **pr**, **rm**, **sed**, **sh**, **vi** and **who**.

Substitution

The first step the shell takes in executing a simple-command is to perform substitutions on the words of the command. There are three kinds of substitution: parameter, command and arithmetic. Parameter substitutions, which are described in detail in the next section, take the form **$name** or **${**...**}**; command substitutions take the form **$(***command***)** or **`***command***`**; and arithmetic substitutions take the form **$((***expression***))**.

If a substitution appears outside of double quotes, the results of the substitution are generally subject to word or field splitting according to the current value of the **IFS** parameter. The **IFS** parameter specifies a list of characters which are used to break a string up into several words; any characters from the set space, tab and newline that appear in the IFS characters are called *IFS white space*. Sequences of one or more IFS white space characters, in combination with zero or one non-IFS white space characters delimit a field. As a special case, leading and trailing IFS white space is stripped (*i.e.*, no leading or trailing empty field is created by it); leading or trailing non-IFS white space does create an empty field. Example: if **IFS** is set to `<space>:', the sequence of characters `<space>A<space>:<space><space>B::D' contains four fields: `A', `B', `' and `D'. Note that if the **IFS** parameter is set to the null string, no field splitting is done; if the parameter is unset, the default value of space, tab and newline is used.

The results of substitution are, unless otherwise specified, also subject to brace expansion and file name expansion (see the relevant sections below).

A command substitution is replaced by the output generated by the specified command, which is run in a subshell. For **$(***command***)** substitutions, normal quoting rules are used when *command* is parsed, however, for the **`***command***`** form, a **** followed by any of **$**, **`** or **** is stripped (a **** followed by any other character is unchanged). As a special case in command substitutions, a command of the form **<** *file* is interpreted to mean substitute the contents of *file* ($(< foo) has the same effect as $(cat foo), but it is carried out more efficiently because no process is started).
NOTE: **$(***command***)** expressions are currently parsed by finding the matching parenthesis, regardless of quoting. This will hopefully be fixed soon.

Arithmetic substitutions are replaced by the value of the specified expression. For example, the command **echo $((2+3*4))** prints 14. See Arithmetic Expressions for a description of an *expression*.

Parameters

Parameters are shell variables; they can be assigned values and their values can be accessed using a parameter substitution. A parameter name is either one of the special single punctuation or digit character parameters described below, or a letter followed by zero or more letters or digits (`_' counts as a letter). The later form can be treated as arrays by appending an array index of the form: **[***expr***]** where *expr* is an arithmetic expression. Array indicies are currently limited to the range 0 through 1023, inclusive. Parameter substitutions take the form **$***name*, **${***name***}** or **${***name***[***expr***]}**, where *name* is a parameter name. If substitution is performed on a parameter (or an array parameter element) that is not set, a null string is substituted unless the **nounset** option (**set -o nounset** or **set -u**) is set, in which case an error occurs.

Parameters can be assigned values in a number of ways. First, the shell implicitly sets some parameters like **#**, **PWD**, etc.; this is the only way the special single character parameters are set. Second, parameters are imported from the shell's environment at startup. Third, parameters can be assigned values on the command line, for example, `**FOO=bar**' sets the parameter FOO to bar; multiple parameter assignments can be given on a single command line and they can be followed by a simple-command, in which case the assignments are in effect only for the duration of the command (such assignments are also exported, see below for implications of this). Note that both the parameter name and the **=** must be unquoted for the shell to recognize a parameter assignment. The fourth way of setting a parameter is with the **export**, **readonly** and **typeset** commands; see their descriptions in the Command Execution section. Fifth, **for** and **select** loops set parameters as well as the **getopts**, **read** and **set -A** commands. Lastly, parameters can be assigned values using assignment operators inside arithmetic expressions (see Arithmetic Expressions below) or using the **${***name***=***value***}** form of parameter substitution (see below).

Parameters with the export attribute (set using the **export** or **typeset -x** commands, or

by parameter assignments followed by simple commands) are put in the environment (see *environ*(5)) of commands run by the shell as *name*=*value* pairs. The order in which parameters appear in the environment of a command is unspecified. When the shell starts up, it extracts parameters and their values from its environment and automatically sets the export attribute for those parameters.

Modifiers can be applied to the **${***name***}** form of parameter substitution:

${*name***:-***word***}**

if *name* is set and not null, it is substituted, otherwise *word* is substituted.

${*name***:+***word***}**

if *name* is set and not null, *word* is substituted, otherwise nothing is substituted.

${*name***:=***word***}**

if *name* is set and not null, it is substituted, otherwise it is assigned *word* and the resulting value of *name* is substituted.

${*name***:?***word***}**

if *name* is set and not null, it is substituted, otherwise *word* is printed on standard error (preceded by *name*:) and an error occurs (normally causing termination of a shell script, function or .-script). If word is omitted the string `parameter null or not set' is used instead.

In the above modifiers, the **:** can be omitted, in which case the conditions only depend on *name* being set (as opposed to set and not null). If *word* is needed, parameter, command, arithmetic and tilde substitution are performed on it; if *word* is not needed, it is not evaluated.

The following forms of parameter substitution can also be used:

${#*name***}**

The number of positional parameters if *name* is *****, **@** or is not specified, or the length of the string value of parameter *name*.

${#***name***[*]}**, **${#*name***[@]}**

The number of elements in the array *name*.

${***name***#***pattern***}**, **${*name***##***pattern***}**

If *pattern* matches the beginning of the value of parameter *name*, the matched text is deleted from the result of substitution. A single # results in the shortest match, two #'s results in the longest match.

${***name***%***pattern***}**, **${*name***%%***pattern***}**

Like **${**..#..**}** substitution, but it deletes from the end of the value.

The following special parameters are implicitly set by the shell and cannot be set directly using assignments:

!

Process id of the last background process started. If no background processes have been started, the parameter is not set.

#

The number of positional parameters (*i.e.*, **$1**, **$2**, *etc.*).

$

The process ID of the shell, or the PID of the original shell if it is a subshell.

-

The concatenation of the current single letter options (see **set** command below for list of options).

?

The exit status of the last non-asynchronous command executed. If the last command was killed by a signal, **$?** is set to 128 plus the signal number.

0

The name the shell was invoked with (*i.e.*, **argv[0]**), or the **command-name** if it was invoked with the **-c** option and the **command-name** was supplied, or the *file* argument, if it was supplied. If the **posix** option is not set, **$0** is the name of the current function or script.

1 ... **9**

The first nine positional parameters that were supplied to the shell, function or **.**-script. Further positional parameters may be accessed using **${***number***}**.

*****

All positional parameters (except parameter 0), *i.e.*, **$1 $2 $3**.... If used outside of double quotes, parameters are separate words (which are subjected to word splitting); if used within double quotes, parameters are separated by the first character of the **IFS** parameter (or the empty string if **IFS** is null).

@

Same as **$***, unless it is used inside double quotes, in which case a separate word is generated for each positional parameter - if there are no positional parameters, no word is generated ("$@" can be used to access arguments, verbatim, without loosing null arguments or splitting arguments with spaces).

The following parameters are set and/or used by the shell:

_ *(underscore)*

When an external command is executed by the shell, this parameter is set in the environment of the new process to the path of the executed command. In interactive use, this parameter is also set in the parent shell to the last word of the previous command. When **MAILPATH** messages are evaluated, this parameter contains the name of the file that changed (see **MAILPATH** parameter below).

CDPATH

Search path for the **cd** built-in command. Works the same way as **PATH** for those directories not beginning with / in **cd** commands. Note that if CDPATH is set and does not contain **.** nor an empty path, the current directory is not searched.

COLUMNS

Set to the number of columns on the terminal or window. Currently set to the **cols** value as reported by *stty*(1) if that value is non-zero. This parameter is used by the interactive line editing modes, and by **select**, **set -o** and **kill -l** commands to format information in columns.

EDITOR

If the **VISUAL** parameter is not set, this parameter controls the command line editing mode for interactive shells. See **VISUAL** parameter below for how this works.

ENV

If this parameter is found to be set after any profile files are executed, the expanded value is used as a shell start-up file. It typically contains function and alias definitions.

ERRNO

Integer value of the shell's errno variable - indicates the reason the last system call failed.

Not implemented yet.

EXECSHELL

If set, this parameter is assumed to contain the shell that is to be used to execute commands that *execve*(2) fails to execute and which do not start with a `**#!** *shell*' sequence.

FCEDIT

The editor used by the **fc** command (see below).

FPATH

Like **PATH**, but used when an undefined function is executed to locate the file defining the function. It is also searched when a command can't be found using **PATH**. See Functions below for more information.

HISTFILE

The name of the file used to store history. When assigned to, history is loaded from the specified file. Also, several invocations of the shell running on the same machine will share history if their **HISTFILE** parameters all point at the same file.

NOTE: if HISTFILE isn't set, no history file is used. This is different from the original Korn shell, which uses **$HOME/.sh_history**; in future, pdksh may also use a default history file.

HISTSIZE

The number of commands normally stored for history, default 128.

HOME

The default directory for the **cd** command and the value substituted for an unqualified ~ (see Tilde Expansion below).

IFS

Internal field separator, used during substitution and by the **read** command, to split values into distinct arguments; normally set to space, tab and newline. See Substitution above for details.
Note: this parameter is not imported from the environment when the shell is started.

KSH_VERSION

The version of shell and the date the version was created (readonly). See also the version commands in Emacs Editing Mode and Vi Editing Mode sections, below.

LINENO

The line number of the function or shell script that is currently being executed.

LINES

Set to the number of lines on the terminal or window.

Not implemented yet.

MAIL

If set, the user will be informed of the arrival of mail in the named file. This parameter is ignored if the **MAILPATH** parameter is set.

MAILCHECK

How often, in seconds, the shell will check for mail in the file(s) specified by **MAIL** or **MAILPATH**. If 0, the shell checks before each prompt. The default is 600 (10 minutes).

MAILPATH

A list of files to be checked for mail. The list is colon separated, and each file may be followed by a **?** and a message to be printed if new mail has arrived. Command, parameter and arithmetic substitution is performed on the message, and, during substitution, the parameter **$_** contains the name of the file. The default message is **you have mail in $_**.

OLDPWD

The previous working directory. Unset if **cd** has not successfully changed directories since the shell started, or if the shell doesn't know where it is.

OPTARG

When using **getopts**, it contains the argument for a parsed option, if it requires one.

OPTIND

The index of the last argument processed when using **getopts**. Assigning 1 to this parameter causes **getopts** to process arguments from the beginning the next time it is invoked.

PATH

A colon separated list of directories that are searched when looking for commands and **.**'d files. An empty string resulting from a leading or trailing colon, or two adjacent colons is treated as a `.', the current directory.

POSIXLY_CORRECT

If set, this parameter causes the **posix** option to be enabled. See POSIX Mode below.

PPID

The process ID of the shell's parent (readonly).

PS1

PS1 is the primary prompt for interactive shells. Parameter, command and arithmetic substitutions are performed, and **!** is replaced with the current command number (see **fc** command below). A literal ! can be put in the prompt by placing !! in PS1. Note that since the command line editors try to figure out how long the prompt is (so they know how far it is to edge of the screen), escape codes in the prompt tend to mess things up. You can tell the shell not to count certain sequences (such as escape codes) by prefixing your prompt with a non-printing character (such as control-A) followed by a carriage return and then delimiting the escape codes with this non-printing character. If you don't have any non-printing characters, you're out of luck... BTW, don't blame me for this hack; it's in the original ksh. Default is `**$** ' for non-root users, `# ' for root..

PS2

Secondary prompt string, by default `> ', used when more input is needed to complete a command.

PS3

Prompt used by **select** statement when reading a menu selection. Default is `**#?** '.

PS4

Used to prefix commands that are printed during execution tracing (see **set -x** command below). Parameter, command and arithmetic substitutions are performed before it is printed. Default is `+ '.

PWD

The current working directory. Maybe unset or null if shell doesn't know where it is.

RANDOM

A simple random number generator. Every time **RANDOM** is referenced, it is assigned the next number in a random number series. The point in the series can be set by assigning a number to **RANDOM** (see *rand*(3)).

REPLY

Default parameter for the **read** command if no names are given. Also used in **select** loops to store the value that is read from standard input.

SECONDS

The number of seconds since the shell started or, if the parameter has been assigned an integer value, the number of seconds since the assignment plus the value that was assigned.

TMOUT

If set to a positive integer in an interactive shell, it specifies the maximum number of seconds the shell will wait for input after printing the primary prompt (**PS1**). If the time is exceeded, the shell exits.

TMPDIR

The directory shell temporary files are created in. If this parameter is not set, or does not contain the absolute path of a writable directory, temporary files are created in **/tmp**.

VISUAL

If set, this parameter controls the command line editing mode for interactive shells. If the last component of the path specified in this parameter contains the string **vi**, **emacs** or **gmacs**, the vi, emacs or gmacs (Gosling emacs) editing mode is enabled, respectively.

Tilde Expansion

Tilde expansion, which is done in parallel with parameter substitution, is done on words starting with an unquoted **~**. The characters following the tilde, up to the first **/**, if any, are assumed to be a login name. If the login name is empty, **+** or **-**, the value of the **HOME**, **PWD**, or **OLDPWD** parameter is substituted, respectively. Otherwise, the password file is searched for the login name, and the tilde expression is substituted with the user's home directory. If the login name is not found in the password file or if any quoting or parameter substitution occurs in the login name, no substitution is performed.

In parameter assignments (those preceding a simple-command or those occurring in the arguments of **alias**, **export**, **readonly**, and **typeset**), tilde expansion is done after any unquoted colon (**:**), and login names are also delimited by colons.

The home directory of previously expanded login names are cached and re-used. The **alias -d** command may be used to list, change and add to this cache (*e.g.*, `alias -d fac=/usr/local/facilities; cd ~fac/bin').

Brace Expansion (alternation)

Brace expressions, which take the form

prefix{*str*1,...,*str*N}*suffix*

are expanded to N words, each of which is the concatenation of *prefix*, *str*i and *suffix* (*e.g.*, `a{c,b{X,Y},d}e' expands to four word: ace, abXe, abYe, and ade). As noted in the example, brace expressions can be nested and the resulting words are not sorted. Brace expressions must contain an unquoted comma (**,**) for expansion to occur (*i.e.*, **{}** and **{foo}** are not expanded). Brace expansion is carried out after parameter substitution and before file name generation.

File Name Patterns

A file name pattern is a word containing one or more unquoted **?** or ***** characters or **[..]** sequences. Once brace expansion has been performed, the shell replaces file name patterns with the sorted names of all the files that match the pattern (if no files match, the word is left unchanged). The pattern elements have the following meaning:

?

matches any single character.

*****

matches any sequence of characters.

[..**]**

matches any of the characters inside the brackets. Ranges of characters can be specified by separating two characters by a **-**, *e.g.*, **[a0-9]** matches the letter **a** or any digit. In order to represent itself, a **-** must either be quoted or the first or last character in the character list. Similarly, a **]** must be quoted or the first character in the list if it is represent itself instead of the end of the list. Also, a **!** appearing at the start of the list has special meaning (see below), so to represent itself it must be quoted or appear later in the list.

[!..**]**

like **[**..**]**, except it matches any character not inside the brackets.

(pattern***|** ... **|***pattern***)**

matches any string of characters that matches zero or more occurances of the

specified patterns. Example: the pattern ***(foo|bar)** matches the strings `', `foo', `bar', `foobarfoo', *etc.*.

+(*pattern*| ... |*pattern***)**

matches any string of characters that matches one or more occurances of the specified patterns. Example: the pattern **+(foo|bar)** matches the strings `foo', `bar', `foobarfoo', *etc.*.

?(*pattern*| ... |*pattern***)**

matches the empty string or a string that matches one of the specified patterns. Example: the pattern **?(foo|bar)** only matches the strings `', `foo' and `bar'.

@(*pattern*| ... |*pattern***)**

matches a string that matches one of the specified patterns. Example: the pattern **@(foo|bar)** only matches the strings `foo' and `bar'.

!(*pattern*| ... |*pattern***)**

matches any string that does not match one of the specified patterns. Examples: the pattern **!(foo|bar)** matches all strings except `foo' and `bar'; the pattern **!(*)** matches no strings; the pattern **!(?)*** matches all strings (think about it).

Note that pdksh currently never matches **.** and **..**, but the original ksh, Bourne sh and bash do, so this may have to change (too bad).

Note that none of the above pattern elements match either a period (**.**) at the start of a file name or a slash (**/**), even if they are explicitly used in a **[..]** sequence; also, the names **.** and **..** are never matched, even by the pattern **.***.

If the **markdirs** option is set, any directories that result from file name generation are marked with a trailing **/**.

The POSIX character classes (*i.e.*, **[:***class-name***:]** inside a **[..]** expression) are not yet implemented.

Input/Output Redirection

When a command is executed, its standard input, standard output and standard error (file descriptors 0, 1 and 2, respectively) are normally inherited from the shell. Three exceptions to this are commands in pipelines, for which standard input and/or standard output are those set up by the pipeline, asynchronous commands created when job control is disabled, for which standard input is initially set to be from **/dev/null**, and commands for which any of the following redirections have been specified:

>file

standard output is redirected to *file*. If *file* does not exist, it is created; if it does exist, is a regular file and the **noclobber** option is set, an error occurs, otherwise the file is truncated. Note that this means the command *cmd* < *foo* > *foo* will open *foo* for reading and then truncate it when it opens it for writing, before *cmd* gets a chance to actually read *foo*.

>| *file*

same as **>**, except the file is truncated, even if the **noclobber** option is set.

>> *file*

same as **>**, except the file an existing file is appended to instead of being truncated. Also, the file is opened in append mode, so writes always go to the end of the file (see *open*(2)).

< *file*

standard input is redirected from *file*, which is opened for reading.

<> *file*

same as **<**, except the file is opened for reading and writing.

<< *marker*

after reading the command line containing this kind of redirection (called a here document), the shell copies lines from the command source into a temporary file until a line matching *marker* is read. When the command is executed, standard input is redirected from the temporary file. If *marker* contains no quoted characters, the contents of the temporary file are processed as if enclosed in double quotes each time the command is executed, so parameter, command and arithmetic substitutions are performed, along with backslash (\) escapes for **$**, `, \ and **\newline**. If multiple here documents are used on the same command line, they are saved in order.

<<- *marker*

same as **<<**, except leading tabs are stripped from lines in the here document.

<& *fd*

standard input is duplicated from file descriptor *fd*. *fd* can be a single digit, indicating the number of an existing file descriptor, the letter **p**, indicating the file descriptor associated with the output of the current co-process, or the character **-**, indicating standard input is to be closed.

>& *fd*

same as **<&**, except the operation is done on standard output.

In any of the above redirections, the file descriptor that is redirected (*i.e.*, standard input or standard output) can be explicitly given by preceding the redirection with a single digit. Parameter, command and arithmetic substitutions, tilde substitutions and (if the shell is interactive) file name generation are all performed on the *file*, *marker* and *fd* arguments of redirections. Note however, that the results of any file name generation are only used if a single file is matched; if multiple files match, the word with the unexpanded file name generation characters is used. Note that in restricted shells, redirections which can create files cannot be used.

For simple-commands, redirections may appear anywhere in the command, for compound-commands (**if** statements, *etc.*), any redirections must appear at the end. Redirections are processed after pipelines are created and in the order they are given, so

cat /foo/bar 2>&1 > /dev/null | cat -n

will print an error with a line number prepended to it.

Arithmetic Expressions

Integer arithmetic expressions can be used with the **let** command, inside **$((..))** expressions, inside array references (*e.g.*, *name***[***expr***]**), as numeric arguments to the **test** command, and as the value of an assignment to an integer parameter.

Expression may contain alpha-numeric parameter identifiers, array references, and integer constants and may be combined with the following C operators (listed and grouped in increasing order of precedence).

Unary

+ - ! ~ ++ --

Binary

,
= *= /= %= += -= <<= >>= &= ^= |=
||
&&
|
^
&
== !=
< <= >= >
<< >>
+ -
*** / %**

Ternary

?: (precedence is immediately higher than assignment)

Grouping

( )

Integer constants may be specified with arbitrary bases using the notation *base***#***number*, where *base* is a decimal integer specifying the base, and *number* is a number in the specified base.

The operators are evaluated as follows:

unary +

result is the argument (included for completeness).

unary **-**

negation.

!

logical not; the result is 1 if argument is zero, 0 if not.

~

arithmetic (bit-wise) not.

++

increment; must be applied to a parameter (not a literal or other expression) - the parameter is incremented by 1. When used as a prefix operator, the result is the incremented value of the parameter, when used as a postfix operator, the result is the original value of the parameter.

++

similar to ++, except the paramter is decremented by 1.

,

separates two arithmetic expressions; the left hand side is evaluated first, then the right. The result is value of the expression on the right hand side.

=

assignment; variable on the left is set to the value on the right.

***= /= %= += -= <<= >>= &= ^= |=**

assignment operators; <*var*> <*op*>= <*expr*> is the same as <*var*> = <*var*> <*op*> **(** <*expr*> **)**.

||

logical or; the result is 1 if either argument is non-zero, 0 if not. The right argument is evaluated only if the left argument is zero.

&&

logical and; the result is 1 if both arguments are non-zero, 0 if not. The right argument is evaluated only if the left argument is non-zero.

|

arithmetic (bit-wise) or.

^

arithmetic (bit-wise) exclusive-or.

&

arithmetic (bit-wise) and.

==

equal; the result is 1 if both arguments are equal, 0 if not.

!=

not equal; the result is 0 if both arguments are equal, 1 if not.

<

less than; the result is 1 if the left argument is less than the right, 0 if not.

<= >= >

less than or equal, greater than or equal, greater than. See <.

<< >>

shift left (right); the result is the left argument with its bits shifted left

(right) by the amount given in the right argument.

+ - * /

addition, subtraction, multiplication, and division.

%

remainder; the result is the remainder of the division of the left argument by the right. The sign of the result is unspecified if either argument is negative.

<arg1> **?** *<arg2>* **:** *<arg3>*

if *<arg1>* is non-zero, the result is *<arg2>*, otherwise *<arg3>*.

Co-Processes

A co-process, which is a pipeline created with the **|&** operator, is an asynchronous process that the shell can both write to (using **print -p**) and read from (using **read -p**). The input and output of the co-process can also be manipulated using **>&p** and **<&p** redirections, respectively. Once a co-process has been started, another can't be started until the co-process exits, or until the co-process input has been redirected using an **exec** *n***>&p** redirection. If a co-process's input is redirected in this way, the next co-process to be started will share the output with the first co-process, unless the output of the initial co-process has been redirected using an **exec** *n***<&p** redirection.

Some notes concerning co-processes:

- the only way to close the co-process input (so the co-process reads an end-of-file) is to redirect the input to a numbered file descriptor and then close that file descriptor (*e.g.*, **exec 3>&p;exec 3>&-**).
- in order for co-processes to share a common output, the shell must keep the write portion of the output pipe open. This means that end of file will not be detected until all co-processes sharing the co-process output have exited (when they all exit, the shell closes its copy of the pipe). This can be avoided by redirecting the output to a numbered file descriptor (as this also causes the shell to close its copy). Note that this behaviour is slightly different from the original Korn shell which closes its copy of the write portion of the co-processs output when the most recently started co-process (instead of when all sharing co-processes) exits.
- **print -p** will ignore SIGPIPE signals during writes if the signal is not being trapped or ignored; the same is not true if the co-process input has been duplicated to another file descriptor and **print -u***n* is used.

Functions

Functions are defined using either Korn shell **function** *name* syntax or the Bourne/POSIX

shell *name***()** syntax (see below for the difference between the two forms). Functions are like **.**-scripts in that they are executed in the current environment, however, unlike **.**-scripts, shell arguments (*i.e.*, positional parameters, **$1**, *etc.*) are never visible inside them. When the shell is determining the location of a command, functions are searched after special built-in commands, and before regular and non-regular built-ins, and before the **PATH** is searched.

An existing function may be deleted using **unset -f** *function-name*. A list of functions can be obtained using **typeset +f** and the function definitions can be listed using **typeset -f**. **autoload** (which is an alias for **typeset -fu**) may be used to create undefined functions; when an undefined function is executed, the shell searches the path specified in the **FPATH** parameter for a file with the same name as the function, which, if found is read and executed. If after executing the file, the named function is found to be defined, the function is executed, otherwise, the normal command search is continued (*i.e.*, the shell searches the regular built-in command table and **PATH**). Note that if a command is not found using **PATH**, an attempt is made to autoload a function using **FPATH** (this is an undocumented feature of the original Korn shell).

Functions can have two attributes, trace and export, which can be set with **typeset -ft** and **typeset -fx**, respectively. When a traced function is executed, the shell's **xtrace** option is turned on for the functions duration, otherwise the **xtrace** option is turned off. The export attribute of functions is currently not used. In the original Korn shell, exported functions are visible to shell scripts that are executed.

Since functions are executed in the current shell environment, parameter assignments made inside functions are visible after the function completes. If this is not the desired effect, the **typeset** command can be used inside a function to create a local parameter. Note that special parameters (*e.g.*, **$$**, **$!**) can't be scoped in this way.

The exit status of a function is that of the last command executed in the function. A function can be made to finish immediately using the **return** command; this may also be used to explicitly specify the exit status.

Functions defined with the **function** reserved word are treated differently in the following ways from functions defined with the **()** notation:

- the **$0** parameter is set to the name of the function (Bourne-style functions leave **$0** untouched).
- parameter assignments preceeding function calls are not kept in the shell environment (executing Bourne-style functions will keep assignments).
- **OPTIND** is saved/reset and restored on entry and exit from the function so **getopts** can be used properly both inside and outside the function (Bourne-style functions leave **OPTIND** untouched, so using **getopts** inside a function interferes with using **getopts** outside the function). In the future, the following differences will also be added:
- A separate trap/signal environment will be used during the execution of functions. This will mean that traps set inside a function will not affect the shell's traps and signals that are not ignored in the shell (but may be trapped) will have their default effect in a function.
- The EXIT trap, if set in a function, will be executed after the function returns.

Command Execution

After evaluation of command line arguments, redirections and parameter assignments, the type of command is determined: a special built-in, a function, a regular built-in or the name of a file to execute found using the **PATH** parameter. The checks are made in the above order. Special built-in commands differ from other commands in that the **PATH** parameter is not used to find them, an error during their execution can cause a non-interactive shell to exit and parameter assignments that are specified before the command are kept after the command completes. Just to confuse things, if the posix option is turned off (see **set** command below) some special commands are very special in that no field splitting, file globing, brace expansion nor tilde expansion is preformed on arguments that look like assignments. Regular built-in commands are different only in that the **PATH** parameter is not used to find them.

Addr2line

Usage: `addr2line [option(s)] [addr(s)]`

Synopsis Unix-like command

Location Development:bin

Function Convert addresses into line number/file name pairs. If no addresses are specified on the command line, they will be read from stdin

Inputs The options are:

@<file>	Read options from <file>
-a --addresses	Show addresses
-b --target=<bfdname>	Set the binary file format
-e --exe=<executable>	Set the input file name (default is a.out)
-i --inlines	Unwind inlined functions
-j --section=<name>	Read section-relative offsets instead of addresses
-p --pretty-print	Make the output easier to read for humans
-s --basenames	Strip directory names
-f --functions	Show function names
-C --demangle[=style]	Demangle function names
-h --help	Display this information
-v --version	Display the program's version

Note addr2line: supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Ar

Usage: `ar [emulation options] [-]{dmpqrstx}[abcfilNoPsSuvV] [member-name] [count] archive-file file...`
`or: ar -M [<mri-script]`

Synopsis Unix-like command

Location Development:bin

Function Maintains the indexed libraries used by the linkage editor.

Inputs commands:

d	- delete file(s) from the archive
m[ab]	- move file(s) in the archive

```
p             - print file(s) found in the archive
q[f]          - quick append file(s) to the archive
r[ab][f][u]   - replace existing or insert new file(s) into the archive
s             - act as ranlib
t             - display contents of archive
x[o]          - extract file(s) from the archive

command specific modifiers:
[a]           - put file(s) after [member-name]
[b]           - put file(s) before [member-name] (same as [i])
[D]           - use zero for timestamps and uids/gids
[N]           - use instance [count] of name
[f]           - truncate inserted file names
[P]           - use full path names when matching
[o]           - preserve original dates
[u]           - only replace files that are newer than current archive
                contents

generic modifiers:
[c]           - do not warn if the library had to be created
[s]           - create an archive index (cf. ranlib)
[S]           - do not build a symbol table
[T]           - make a thin archive
[v]           - be verbose
[V]           - display the version number
@<file>       - read options from <file>
```

As

Usage: `as [option...] [asmfile...]`

Synopsis Unix-like command

Location Development:bin

Function The portable GNU assembler

Inputs Options:

```
-a[sub-option...]          turn on listings
                           Sub-options [default hls]:
                           c      omit false conditionals
                           d      omit debugging directives
                           g      include general info
                           h      include high-level source
                           l      include assembly
                           m      include macro expansions
                           n      omit forms processing
                           s      include symbols
                           =FILE  list to FILE (must be last sub-option)
--alternate                initially turn on alternate macro syntax
```

```
  --compress-debug-sections
                      compress DWARF debug sections using zlib
  --nocompress-debug-sections
                      don't compress DWARF debug sections
  -D                   produce assembler debugging messages
  --debug-prefix-map OLD=NEW
                      map OLD to NEW in debug information
  --defsym SYM=VAL        define symbol SYM to given value
  --execstack             require executable stack for this object
  --noexecstack           don't require executable stack for this object
  --size-check=[error|warning]
                            ELF .size directive check (default --size-
check=error)
  -f                  skip whitespace and comment preprocessing
  -g --gen-debug          generate debugging information
  --gstabs               generate STABS debugging information
  --gstabs+               generate STABS debug info with GNU extensions
  --gdwarf-2              generate DWARF2 debugging information
  --hash-size=<value>     set the hash table size close to <value>
  --help                 show this message and exit
  --target-help           show target specific options
  -I DIR                 add DIR to search list for .include directives
  -J                    don't warn about signed overflow
  -K                    warn when differences altered for long displacements
  -L,--keep-locals        keep local symbols (e.g. starting with `L')
  -M,--mri               assemble in MRI compatibility mode
  --MD FILE               write dependency information in FILE (default none)
  -nocpp                 ignored
  -o OBJFILE              name the object-file output OBJFILE (default a.out)
  -R                    fold data section into text section
  --reduce-memory-overheads
                      prefer smaller memory use at the cost of longer
                      assembly times
  --statistics           print various measured statistics from execution
  --strip-local-absolute  strip local absolute symbols
  --traditional-format    Use same format as native assembler when
possible
  --version              print assembler version number and exit
  -W  --no-warn           suppress warnings
  --warn                 don't suppress warnings
  --fatal-warnings        treat warnings as errors
  -w                    ignored
  -X                    ignored
  -Z                    generate object file even after errors
  --listing-lhs-width     set the width in words of the output data column of
                      the listing
  --listing-lhs-width2    set the width in words of the continuation lines
                      of the output data column; ignored if smaller than
                      the width of the first line
  --listing-rhs-width     set the max width in characters of the lines from
                      the source file
  --listing-cont-lines    set the maximum number of continuation lines used
                      for the output data column of the listing
  @FILE                  read options from FILE
  -Q                    ignored
  -V                    print assembler version number
```

```
  -k                  ignored
  -n                  Do not optimize code alignment
  -q                  quieten some warnings
  -s                  ignored
  --32/--64           generate 32bit/64bit code
  --divide            ignored
  -march=CPU[,+EXTENSION...]
                      generate code for CPU and EXTENSION, CPU is one of:
                       generic32, generic64, i386, i486, i586, i686,
                       pentium, pentiumpro, pentiumii, pentiumiii, pentium4,
                       prescott, nocona, core, core2, corei7, l1om, k6,
                       k6_2, athlon, opteron, k8, amdfam10, bdver1
                      EXTENSION is combination of:
                       8087, 287, 387, no87, mmx, nommx, sse, sse2, sse3,
                       ssse3, sse4.1, sse4.2, sse4, nosse, avx, noavx, vmx,
                       smx, xsave, xsaveopt, aes, pclmul, fsgsbase, rdrnd,
                       f16c, fma, fma4, xop, lwp, movbe, ept, clflush, nop,
                       syscall, rdtscp, 3dnow, 3dnowa, padlock, svme, sse4a,
                       abm
  -mtune=CPU          optimize for CPU, CPU is one of:
                      generic32, generic64, i8086, i186, i286, i386, i486,
                      i586, i686, pentium, pentiumpro, pentiumii,
                      pentiumiii, pentium4, prescott, nocona, core, core2,
                      corei7, l1om, k6, k6_2, athlon, opteron, k8,
                      amdfam10, bdver1
  -msse2avx           encode SSE instructions with VEX prefix
  -msse-check=[none|error|warning]
                      check SSE instructions
  -mavxscalar=[128|256]   encode scalar AVX instructions with specific
vector
                      length
  -mmnemonic=[att|intel]  use AT&T/Intel mnemonic
  -msyntax=[att|intel]    use AT&T/Intel syntax
  -mindex-reg         support pseudo index registers
  -mnaked-reg         don't require `%' prefix for registers
  -mold-gcc           support old (<= 2.8.1) versions of gcc
```

Awk

Usage: `awk [POSIX or GNU style options] -f progfile [--] file ...`
`or: awk [POSIX or GNU style options] [--] 'program' file ...`

Synopsis Unix-like command

Location Development:bin

Function Finds lines in files that match a pattern and performs specified actions on those lines.

Inputs

POSIX options:	GNU long options:
-f progfile	--file=progfile
-F fs	--field-separator=fs
-v var=val	--assign=var=val

-m[fr] val
-W compat --compat
-W copyleft --copyleft
-W copyright --copyright
-W dump-variables[=file] --dump-variables[=file]
-W gen-po --gen-po
-W help --help
-W lint[=fatal] --lint[=fatal]
-W lint-old --lint-old
-W non-decimal-data --non-decimal-data
-W profile[=file] --profile[=file]
-W posix --posix
-W re-interval --re-interval
-W source=program-text --source=program-text
-W traditional --traditional
-W usage --usage
-W version --version

To report bugs, see node `Bugs' in `gawk.info', which is
section `Reporting Problems and Bugs' in the printed version.

gawk is a pattern scanning and processing language.
By default it reads standard input and writes standard output.

Example
```
gawk '{ sum += $1 }; END { print sum }' file
gawk -F: '{ print $1 }' /etc/passwd
```

Base64

Usage: `base64 [OPTION] [FILE]`

Synopsis Unix-like command

Location Development:bin

Function Base64 encode or decode FILE, or standard input, to standard output.

Inputs
-w, --wrap=COLS
Wrap encoded lines after COLS character (default 76).
Use 0 to disable line wrapping.
-d, --decode
Decode data.
-i, --ignore-garbage
When decoding, ignore non-alphabet characters.

--help
Display this help and exit.
--version

Output version information and exit.

With no FILE, or when FILE is -, read standard input.

The data are encoded as described for the base64 alphabet in RFC 3548. When decoding, the input may contain newlines in addition to the bytes of the formal base64 alphabet. Use --ignore-garbage to attempt to recover from any other non-alphabet bytes in the encoded stream.

Basename

Usage: `basename NAME [SUFFIX]`
or: `basename OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print NAME with any leading directory components removed. If specified, also remove a trailing SUFFIX.

Inputs

--help	display this help and exit
--version	output version information and exit

Examples

```
basename /usr/bin/sort       Output "sort".
basename include/stdio.h .h  Output "stdio".
```

Bc

Usage: `bc [ -hlwsqv ] [long-options] [ file ... ]`

Synopsis Unix-like command

Location Development:bin

Function bc is a language that supports arbitrary precision numbers with interactive execution of statements. There are some similarities in the syntax to the C programming language. A standard math library is available by command line option. If requested, the math library is defined before processing any files. bc starts by processing code from all the files listed on the command line in the order listed. After all files have been processed, bc reads from the standard input. All code is executed as it is read. (If a file contains a command to halt the processor, bc will never read from the

standard input.)

Inputs

```
-h  --help         print this usage and exit
-i  --interactive  force interactive mode
-l  --mathlib      use the predefine math routnes
-q  --quiet        don't print initial banner
-s  --standard     non-standard bc constructs are errors
-w  --warn         warn about non-standard bc constructs
-v  --version      print version information and exit
```

Notes A complete version of BC manual is available here:
http://www.gnu.org/software/bc/manual/html_mono/bc.html#SEC1

Bison

Usage: `bison [OPTION]... FILE`

Synopsis Unix-like command

Location Development:bin

Function GNU bison generates parsers for LALR(1) grammars.

Inputs If a long option shows an argument as mandatory, then it is mandatory for the equivalent short option also. Similarly for optional arguments.

```
Operation modes:
  -h, --help                 display this help and exit
  -V, --version              output version information and exit
  -y, --yacc                 emulate POSIX yacc

Parser:
  -S, --skeleton=FILE        specify the skeleton to use
  -t, --debug                instrument the parser for debugging
      --locations            enable locations computation
  -p, --name-prefix=PREFIX   prepend PREFIX to the external
symbols
  -l, --no-lines             don't generate `#line' directives
  -n, --no-parser            generate the tables only
  -k, --token-table          include a table of token names

Output:
  -d, --defines              also produce a header file
  -r, --report=THINGS        also produce details on the automaton
  -v, --verbose              same as `--report=state'
  -b, --file-prefix=PREFIX   specify a PREFIX for output files
  -o, --output=FILE          leave output to FILE
  -g, --graph                also produce a VCG description of the
```

THINGS is a list of comma separated words that can include:

`state'	describe the states
`itemset'	complete the core item sets with their closure
`lookahead'	explicitly associate lookaheads to items
`solved'	describe shift/reduce conflicts solving
`all'	include all the above information
`none'	disable the report

Bzip2

Usage: `bzip2 [ -cdfkqstvzVL123456789 ] [ filenames ... ]`
`bunzip2 [ -fkvsVL ] [ filenames ... ]`
`bzcat [ -s ] [ filenames ... ]`

Synopsis Unix-like command

Location Development:bin

Function bzip2 compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. bzip2 expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name original_name.bz2. Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. bzip2 and bunzip2 will by default not overwrite existing files. If you want this to happen, specify the -f flag.

Inputs -c --stdout
Compress or decompress to standard output.

-d --decompress
Force decompression. bzip2, bunzip2 and bzcat are really the same program, and the decision about what actions to take is done on the basis of which name is used. This flag overrides that mechanism, and forces bzip2 to decompress.

-z --compress
The complement to -d: forces compression, regardless of the invokation name.

-t --test
Check integrity of the specified file(s), but don't decompress them. This really performs a trial decompression and throws away the result.

-f --force
Force overwrite of output files. Normally, bzip2 will not overwrite existing output files. Also forces bzip2 to break hard links to files, which it otherwise wouldn't do. bzip2 normally declines to decompress files which don't have the correct magic header bytes. If forced (-f), however, it will pass such files through unmodified. This is how GNU gzip behaves.

-k --keep
Keep (don't delete) input files during compression or decompression.

-s --small
Reduce memory usage, for compression, decompression and testing. Files are decompressed and tested using a modified algorithm which only requires 2.5 bytes per block byte. This means any file can be decompressed in 2300k of memory, albeit at about half the normal speed. During compression, -s selects a block size of 200k, which limits memory use to around the same figure, at the expense of your compression ratio. In short, if your machine is low on memory (8 megabytes or less), use -s for everything. See MEMORY MANAGEMENT below.

-q --quiet
Suppress non-essential warning messages. Messages pertaining to I/O errors and other critical events will not be suppressed.

-v --verbose
Verbose mode -- show the compression ratio for each file processed. Further -v's increase the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.

-L --license -V --version
Display the software version, license terms and conditions.

-1 (or --fast) to -9 (or --best)
Set the block size to 100 k, 200 k ... 900 k when compressing. Has no effect when decompressing. See MEMORY MANAGEMENT below. The --fast and --best aliases are primarily for GNU gzip compatibility. In particular, --fast doesn't make things significantly faster. And --best merely selects the default behaviour.

--
Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash, for example: bzip2 -- -myfilename.

Notes

A complete manual for bzip2recover and bzip2 is available here: http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.html

Bzip2recover

Usage: `bzip2recover filename`

Synopsis Unix-like command

Location Development:bin

Function Searches for blocks in .bz2 files, and writes each block out into its own .bz2 file. You can then use bzip2 -t to test the integrity of the resulting files, and decompress those which are undamaged.

bzip2 compresses files in blocks, usually 900kbytes long. Each block is handled independently. If a media or transmission error causes a multi-block .bz2 file to become damaged, it may be possible to recover data from the undamaged blocks in the file. The compressed representation of each block is delimited by a 48-bit pattern, which makes it possible to find the block boundaries with reasonable certainty. Each block also carries its own 32-bit CRC, so damaged blocks can be distinguished from undamaged ones.

bzip2recover takes a single argument, the name of the damaged file, and writes a number of files rec0001file.bz2, rec0002file.bz2, etc, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing -- for example, bzip2 -dc rec*file.bz2 > recovered_data -- lists the files in the correct order. bzip2recover should be of most use dealing with large .bz2 files, as these will contain many blocks. It is clearly futile to use it on damaged single-block files, since a damaged block cannot be recovered. If you wish to minimise any potential data loss through media or transmission errors, you might consider compressing with a smaller block size.

Notes A complete manual for bzip2recover and bzip2 is available here: http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.html

C++

Usage: `c++ [options] file...`

Synopsis Unix-like command

Location Development:bin

Function Compiles a human-readable source code into a machine-readable

executable

Inputs

Options:

-pass-exit-codes
    Exit with highest error code from a phase
--help
    Display this information
--target-help
    Display target specific command line options
(Use '-v --help' to display command line options of sub-processes)
-dumpspecs
    Display all of the built in spec strings
-dumpversion
    Display the version of the compiler
-dumpmachine
    Display the compiler's target processor
-print-search-dirs
    Display the directories in the compiler's search path
-print-libgcc-file-name
    Display the name of the compiler's companion library
-print-file-name=<lib>
    Display the full path to library <lib>
-print-prog-name=<prog>
    Display the full path to compiler component <prog>
-print-multi-directory
    Display the root directory for versions of libgcc
-print-multi-lib
    Display the mapping between command line options and
     multiple library search directories
-print-multi-os-directory
    Display the relative path to OS libraries
-Wa,<options>
    Pass comma-separated <options> on to the assembler
-Wp,<options>
    Pass comma-separated <options> on to the preprocessor
-Wl,<options>
    Pass comma-separated <options> on to the linker
-Xassembler <arg>
    Pass <arg> on to the assembler
-Xpreprocessor <arg>
    Pass <arg> on to the preprocessor
-Xlinker <arg>
    Pass <arg> on to the linker
-combine
    Pass multiple source files to compiler at once
-save-temps
    Do not delete intermediate files
-pipe
    Use pipes rather than intermediate files
-time
    Time the execution of each subprocess

-specs=<file>
Override built-in specs with the contents of <file>
-std=<standard>
Assume that the input sources are for <standard>
--sysroot=<directory>
Use <directory> as the root directory for headers
and libraries
-B <directory>
Add <directory> to the compiler's search paths
-b <machine>
Run gcc for target <machine>, if installed
-V <version>
Run gcc version number <version>, if installed
-v
Display the programs invoked by the compiler
-###
Like -v but options quoted and commands not executed
-E
Preprocess only; do not compile, assemble or link
-S
Compile only; do not assemble or link
-c
Compile and assemble, but do not link
-o <file>
Place the output into <file>
-x <language>
Specify the language of the following input files
Permissible languages include: c c++ assembler none
'none' means revert to the default behavior of
guessing the language based on the file's extension

Options starting with -g, -f, -m, -O, -W, or --param are
automatically passed on to the various sub-processes invoked by
c++. In order to pass other options on to these processes the
-W<letter> options must be used.

C++filt

Usage: `c++filt [options] [mangled names]`

Synopsis Unix-like command

Location Development:bin

Function Demangle C++ and Java symbols

Inputs Options are:
[-_|--strip-underscore] Ignore first leading underscore

[-n|--no-strip-underscore]	Do not ignore a leading underscore (default)
[-p|--no-params]	Do not display function arguments
[-i|--no-verbose]	Do not show implementation details (if any)
[-t|--types]	Also attempt to demangle type encodings
[-s|--format {none,auto,gnu,lucid,arm,hp,edg,gnu-v3,java,gnat}]	
[@<file>]	Read extra options from <file>
[-h|--help]	Display this information
[-v|--version]	Show the version information

Demangled names are displayed to stdout.
If a name cannot be demangled it is just echoed to stdout.
If no names are provided on the command line, stdin is read.
Report bugs to <http://www.sourceware.org/bugzilla/>.

Cat

Usage: `cat [OPTION] [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Concatenate FILE(s), or standard input, to standard output.

Inputs

-A, --show-all	equivalent to -vET
-b, --number-nonblank	number nonblank output lines
-e	equivalent to -vE
-E, --show-ends	display $ at end of each line
-n, --number	number all output lines
-s, --squeeze-blank	never more than one single blank line
-t	equivalent to -vT
-T, --show-tabs	display TAB characters as ^I
-u	(ignored)
-v, --show-nonprinting	use ^ and M- notation, except for LFD and TAB
--help	display this help and exit
--version	output version information and exit

With no FILE, or when FILE is -, read standard input.

Example `cat f - g`
Output f's contents, then standard input, then g's contents.

`cat`
Copy standard input to standard output.

Chmod

Usage: `chmod [OPTION]... MODE[,MODE]... FILE...`
`or: chmod [OPTION]... OCTAL-MODE FILE...`
`or: chmod [OPTION]... --reference=RFILE FILE...`

Synopsis Unix-like command

Location Development:bin

Function Change the mode of each FILE to MODE.

Inputs

-c, --changes	like verbose but report only when a change is made
--no-preserve-root	do not treat `/' specially (the default)
--preserve-root	fail to operate recursively on `/'
-f, --silent, --quiet	suppress most error messages
-v, --verbose	output a diagnostic for every file processed
--reference=RFILE	use RFILE's mode instead of MODE values
-R, --recursive	change files and directories recursively
--help	display this help and exit
--version	output version information and exit

Each MODE is of the form `[ugoa]*([-+=]([rwxXst]*|[ugo]))+'.

Chown

Usage: `chown [OPTION]... [OWNER][:[GROUP]] FILE...`
or: `chown [OPTION]... --reference=RFILE FILE...`

Synopsis Unix-like command

Location Development:bin

Function Change the owner and/or group of each FILE to OWNER and/or GROUP. With --reference, change the owner and group of each FILE to those of RFILE.

Inputs

-c, --changes like verbose but report only when a change is made

--dereference
affect the referent of each symbolic link (this is the default), rather than the symbolic link itself

-h, --no-dereference
affect each symbolic link instead of any referenced file (useful only on systems that can change the ownership of a symlink)

--from=CURRENT_OWNER:CURRENT_GROUP
change the owner and/or group of each file only if its current owner and/or group match those specified here. Either may be omitted, in which case a match is not required for the omitted attribute.

--no-preserve-root
do not treat `/' specially (the default)

--preserve-root
fail to operate recursively on `/'

-f, --silent, --quiet
suppress most error messages

--reference=RFILE
use RFILE's owner and group rather than specifying OWNER:GROUP values

-R, --recursive operate on files and directories recursively

-v, --verbose output a diagnostic for every file processed

The following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one takes effect.

-H if a command line argument is a symbolic link to a directory, traverse it

-L traverse every symbolic link to a directory encountered

-P do not traverse any symbolic links (default)

--help display this help and exit

--version output version information and exit

Owner is unchanged if missing. Group is unchanged if missing, but changed to login group if implied by a `:' following a symbolic OWNER. OWNER and GROUP may be numeric as well as symbolic.

Example

```
chown root /u
```
Change the owner of /u to "root".

```
chown root:staff /u
```
Likewise, but also change its group to "staff".

```
chown -hR root /u
```

Change the owner of /u and subfiles to "root".

Notes It won't produce results on native AROS and native AROS' filesystems

Cksum

Usage: `cksum [FILE]...`
or: `cksum [OPTION]`

Synopsis Unix-like command

Location Development:bin

Function Print CRC checksum and byte counts of each FILE.

Inputs
--help display this help and exit
--version output version information and exit

Cmp

Usage: `cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]`

Synopsis Unix-like command

Location Development:bin

Function Compare two files byte by byte.

Inputs
-b --print-bytes
Print differing bytes.
-i SKIP --ignore-initial=SKIP
Skip the first SKIP bytes of input.
-i SKIP1:SKIP2 --ignore-initial=SKIP1:SKIP2
Skip the first SKIP1 bytes of FILE1 and the first SKIP2 bytes of FILE2.
-l --verbose
Output byte numbers and values of all differing bytes.
-n LIMIT --bytes=LIMIT
Compare at most LIMIT bytes.
-s --quiet --silent
Output nothing; yield exit status only.
-v --version

Output version info.

--help

Output this help.

SKIP1 and SKIP2 are the number of bytes to skip in each file.
SKIP values may be followed by the following multiplicative suffixes:
kB 1000, K 1024, MB 1,000,000, M 1,048,576,
GB 1,000,000,000, G 1,073,741,824, and so on for T, P, E, Z, Y.

If a FILE is `-' or missing, read standard input.

Comm

Usage: `comm [OPTION]... FILE1 FILE2`

Synopsis Unix-like command

Location Development:bin

Function Compare sorted files FILE1 and FILE2 line by line.

Inputs With no options, produce three-column output. Column one contains lines unique to FILE1, column two contains lines unique to FILE2, and column three contains lines common to both files.

-1 suppress lines unique to FILE1
-2 suppress lines unique to FILE2
-3 suppress lines that appear in both files
--help display this help and exit
--version output version information and exit

Cp

Usage: `cp [OPTION]... [-T] SOURCE DEST`
or: `cp [OPTION]... SOURCE... DIRECTORY`
or: `cp [OPTION]... -t DIRECTORY SOURCE...`

Synopsis Unix-like command

Location Development:bin

Function Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY

Inputs Mandatory arguments to long options are mandatory for short options too.

```
-a, --archive
      same as -dpPR
 --backup[=CONTROL]
      make a backup of each existing destination file
-b
      like --backup but does not accept an argument
--copy-contents
     copy contents of special files when recursive
-d
     same as --no-dereference --preserve=link
-f, --force
     if an existing destination file cannot be opened, remove it
     and try again
-i, --interactive
     prompt before overwrite
-H
    follow command-line symbolic links
-l, --link
    link files instead of copying
-L, --dereference
    always follow symbolic links
-P, --no-dereference
    never follow symbolic links
-p
    same as --preserve=mode, ownership, timestamps
--preserve[=ATTR_LIST]
    preserve the specified attributes (default: mode, ownership,
    timestamps), if possible. Additional attributes: links, all
--no-preserve=ATTR_LIST
    don't preserve the specified attributes
--parents
    use full source file name under DIRECTORY
-R, -r, --recursive
    copy directories recursively
--remove-destination
    remove each existing destination file before
    attempting to open it (contrast with --force)
--sparse=WHEN
    control creation of sparse files
--strip-trailing-slashes
    remove any trailing slashes from each SOURCE argument
-s, --symbolic-link
    make symbolic links instead of copying
-S, --suffix=SUFFIX
    override the usual backup suffix
-t, --target-directory=DIRECTORY
   copy all SOURCE arguments into DIRECTORY
-T, --no-target-directory
   treat DEST as a normal file
-u, --update
     copy only when the SOURCE file is newer than the destination
```

file or when the destination file is missing
-v, --verbose
explain what is being done
-x, --one-file-system
stay on this file system
--help
display this help and exit
--version
output version information and exit

By default, sparse SOURCE files are detected by a crude heuristic and the corresponding DEST file is made sparse as well. That is the behavior selected by --sparse=auto. Specify --sparse=always to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes.

Use --sparse=never to inhibit creation of sparse files.

The backup suffix is `~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the --backup option or through the VERSION_CONTROL environment variable. Here are the values:

none, off
never make backups (even if --backup is given)
numbered, t
make numbered backups
existing, nil
numbered if numbered backups exist, simple otherwise
simple, never
always make simple backups

As a special case, cp makes a backup of SOURCE when the force and backup options are given and SOURCE and DEST are the same name for an existing, regular file.

Cpp

Usage: `cpp [options] file...`

Synopsis Unix-like command

Location Development:bin

Function The C preprocessor, often known as cpp, is a macro processor that is used automatically by the C compiler to transform your program before compilation. It is called a macro processor because it allows you to define macros, which are brief abbreviations for longer

constructs.

Inputs

Options:

```
  -pass-exit-codes       Exit with highest error code from a phase
  --help                 Display this information
  --target-help          Display target specific command line options
  (Use '-v --help' to display command line options of sub-processes)
  -dumpspecs             Display all of the built in spec strings
  -dumpversion           Display the version of the compiler
  -dumpmachine           Display the compiler's target processor
  -print-search-dirs     Display the directories in the compiler's
                         search path
  -print-libgcc-file-name
                         Display the name of the compiler's
                         companion library
  -print-file-name=<lib>
                         Display the full path to library <lib>
  -print-prog-name=<prog>
                         Display the full path to compiler component
                         <prog>
  -print-multi-directory Display the root directory for versions of
libgcc
  -print-multi-lib       Display the mapping between command line
                         options and multiple library search
                         directories
  -print-multi-os-directory
                         Display the relative path to OS libraries
  -Wa,<options>          Pass comma-separated <options> on to the
                         assembler
  -Wp,<options>          Pass comma-separated <options> on to
the
                         preprocessor
  -Wl,<options>          Pass comma-separated <options> on to
the
                         linker
  -Xassembler <arg>      Pass <arg> on to the assembler
  -Xpreprocessor <arg>
                         Pass <arg> on to the preprocessor
  -Xlinker <arg>         Pass <arg> on to the linker
  -combine               Pass multiple source files to compiler at once
  -save-temps            Do not delete intermediate files
  -pipe                  Use pipes rather than intermediate files
  -time                  Time the execution of each subprocess
  -specs=<file>          Override built-in specs with the contents of
                         <file>
  -std=<standard>        Assume that the input sources are for
                         <standard>
  --sysroot=<directory>
                         Use <directory> as the root directory for
                         headers and libraries
  -B <directory>         Add <directory> to the compiler's search
```

	paths
-b <machine>	Run gcc for target <machine>, if installed
-V <version>	Run gcc version number <version>, if installed
-v	Display the programs invoked by the compiler
-###	Like -v but options quoted and commands not executed
-E	Preprocess only; do not compile, assemble or link
-S	Compile only; do not assemble or link
-c	Compile and assemble, but do not link
-o <file>	Place the output into <file>
-x <language>	Specify the language of the following input files Permissible languages include: c c++ assembler none 'none' means revert to the default behavior of guessing the language based on the file's extension

Options starting with -g, -f, -m, -O, -W, or --param are automatically passed on to the various sub-processes invoked by cpp. In order to pass other options on to these processes the -W<letter> options must be used.

Notes A more complete manual for cpp is available here: http://linux.about.com/library/cmd/blcmdl1_cpp.htm

Csplit

Usage: `csplit [OPTION]... FILE PATTERN...`

Synopsis Unix-like command

Location Development:bin

Function Output pieces of FILE separated by PATTERN(s) to files `xx00', `xx01', ..., and output byte counts of each piece to standard output.

Inputs Mandatory arguments to long options are mandatory for short options too.

-b, --suffix-format=FORMAT	use sprintf FORMAT instead of %02d
-f, --prefix=PREFIX	use PREFIX instead of `xx'
-k, --keep-files	do not remove output files on errors
-n, --digits=DIGITS	use specified number of digits instead of 2
-s, --quiet, --silent	do not print counts of output file sizes

-z, --elide-empty-files	remove empty output files
--help	display this help and exit
--version	output version information and exit

Read standard input if FILE is -. Each PATTERN may be:

INTEGER	copy up to but not including specified line number
/REGEXP/[OFFSET]	copy up to but not including a matching line
%REGEXP%[OFFSET]	skip to, but not including a matching line
{INTEGER}	repeat the previous pattern specified number of times
{*}	repeat the previous pattern as many times as possible

A line OFFSET is a required `+' or `-' followed by a positive integer.

Example `1.sys> list | grep Device`

Searches for the "Device" word in the output of List command. If exists, it show only the occurring line

Curl

Usage: `curl [options] [URL...]`

Synopsis Unix-like command

Location Development:bin

Function Curl is a tool to transfer data from or to a server, using one of the supported protocols (HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction. curl offers a busload of useful tricks like proxy support, user authentication, FTP upload, HTTP post, SSL connections, cookies, file transfer resume and more.

Inputs Options: (H) means HTTP/HTTPS only, (F) means FTP only

```
    --anyauth       Pick "any" authentication method (H)
 -a/--append        Append to target file when uploading (F/SFTP)
    --basic         Use HTTP Basic Authentication (H)
    --cacert <file> CA certificate to verify peer against (SSL)
    --capath <directory> CA directory to verify peer against (SSL)
 -E/--cert <cert[:passwd]> Client certificate file and password (SSL)
    --cert-type <type> Certificate file type (DER/PEM/ENG) (SSL)
```

```
   --ciphers <list> SSL ciphers to use (SSL)
   --compressed    Request compressed response (using deflate or
                   gzip)
-K/--config <file> Specify which config file to read
   --connect-timeout <seconds> Maximum time allowed for
                       connection
-C/--continue-at <offset> Resumed transfer offset
-b/--cookie <name=string/file> Cookie string or file to read
                       cookies from (H)
-c/--cookie-jar <file> Write cookies to this file after operation (H)
   --create-dirs   Create necessary local directory hierarchy
   --crlf          Convert LF to CRLF in upload
-d/--data <data>   HTTP POST data (H)
   --data-ascii <data>  HTTP POST ASCII data (H)
   --data-binary <data> HTTP POST binary data (H)
   --data-urlencode <name=data/name@filename> HTTP POST
                   data url encoded (H)
   --digest        Use HTTP Digest Authentication (H)
   --disable-eprt  Inhibit using EPRT or LPRT (F)
   --disable-epsv  Inhibit using EPSV (F)
-D/--dump-header <file> Write the headers to this file
   --egd-file <file> EGD socket path for random data (SSL)
   --engine <eng>  Crypto engine to use (SSL). "--engine list" for
                   list
-f/--fail          Fail silently (no output at all) on HTTP errors (H)
-F/--form <name=content> Specify HTTP multipart POST data (H)
   --form-string <name=string> Specify HTTP multipart POST data
                   (H)
   --ftp-account <data> Account data to send when requested by
                   server (F)
   --ftp-alternative-to-user <cmd> String to replace "USER [name]"
                   (F)
   --ftp-create-dirs Create the remote dirs if not present (F)
   --ftp-method [multicwd/nocwd/singlecwd] Control CWD usage
                   (F)
   --ftp-pasv      Use PASV/EPSV instead of PORT (F)
-P/--ftp-port <address> Use PORT with address instead of PASV
                   (F)
   --ftp-skip-pasv-ip Skip the IP address for PASV (F)
   --ftp-ssl       Try SSL/TLS for ftp transfer (F)
   --ftp-ssl-ccc   Send CCC after authenticating (F)
   --ftp-ssl-ccc-mode [active/passive] Set CCC mode (F)
   --ftp-ssl-control Require SSL/TLS for ftp login, clear for transfer
                   (F)
   --ftp-ssl-reqd  Require SSL/TLS for ftp transfer (F)
-G/--get           Send the -d data with a HTTP GET (H)
-g/--globoff       Disable URL sequences and ranges using {} and []
-H/--header <line> Custom header to pass to server (H)
-I/--head          Show document info only
-h/--help          This help text
   --hostpubmd5 <md5> Hex encoded MD5 string of the host
```

```
                  public key. (SSH)
 -0/--http1.0       Use HTTP 1.0 (H)
   --ignore-content-length  Ignore the HTTP Content-Length
header
 -i/--include       Include protocol headers in the output (H/F)
 -k/--insecure      Allow connections to SSL sites without certs (H)
   --interface <interface> Specify network interface/address to use
 -4/--ipv4          Resolve name to IPv4 address
 -6/--ipv6          Resolve name to IPv6 address
 -j/--junk-session-cookies Ignore session cookies read from file (H)
   --keepalive-time <seconds> Interval between keepalive probes
   --key <key>      Private key file name (SSL/SSH)
   --key-type <type> Private key file type (DER/PEM/ENG) (SSL)
   --krb <level>    Enable Kerberos with specified security level (F)
   --libcurl <file> Dump libcurl equivalent code of this command
                    line
   --limit-rate <rate> Limit transfer speed to this rate
 -l/--list-only     List only names of an FTP directory (F)
   --local-port <num>[-num] Force use of these local port numbers
 -L/--location      Follow Location: hints (H)
   --location-trusted Follow Location: and send auth to other hosts
                    (H)
 -M/--manual        Display the full manual
   --max-filesize <bytes> Maximum file size to download (H/F)
   --max-redirs <num> Maximum number of redirects allowed (H)
 -m/--max-time <seconds> Maximum time allowed for the transfer
   --negotiate      Use HTTP Negotiate Authentication (H)
 -n/--netrc         Must read .netrc for user name and password
   --netrc-optional Use either .netrc or URL; overrides -n
 -N/--no-buffer     Disable buffering of the output stream
   --no-keepalive   Disable keepalive use on the connection
   --no-sessionid   Disable SSL session-ID reusing (SSL)
   --ntlm           Use HTTP NTLM authentication (H)
 -o/--output <file> Write output to <file> instead of stdout
   --pass  <pass>   Pass phrase for the private key (SSL/SSH)
   --post301        Do not switch to GET after following a 301
                    redirect (H)
   --post302        Do not switch to GET after following a 302
                    redirect (H)
 -#/--progress-bar  Display transfer progress as a progress bar
 -x/--proxy <host[:port]> Use HTTP proxy on given port
   --proxy-anyauth Pick "any" proxy authentication method (H)
   --proxy-basic    Use Basic authentication on the proxy (H)
   --proxy-digest   Use Digest authentication on the proxy (H)
   --proxy-negotiate Use Negotiate authentication on the proxy (H)
   --proxy-ntlm     Use NTLM authentication on the proxy (H)
 -U/--proxy-user <user[:password]> Set proxy user and password
 -p/--proxytunnel   Operate through a HTTP proxy tunnel (using
                    CONNECT)
   --pubkey <key>   Public key file name (SSH)
 -Q/--quote <cmd>   Send command(s) to server before file
```

```
                    transfer (F/SFTP)
   --random-file <file> File for reading random data from (SSL)
 -r/--range <range> Retrieve only the bytes within a range
   --raw           Pass HTTP "raw", without any transfer decoding
(H)
 -e/--referer       Referer URL (H)
 -O/--remote-name   Write output to a file named as the remote file
   --remote-name-all Use the remote file name for all URLs
 -R/--remote-time   Set the remote file's time on the local output
 -X/--request <command> Specify request command to use
   --retry <num>   Retry request <num> times if transient
                   problems occur
   --retry-delay <seconds> When retrying, wait this many seconds
                   between each
   --retry-max-time <seconds> Retry only within this period
 -S/--show-error    Show error. With -s, make curl show errors
when
                   they occur
 -s/--silent        Silent mode. Don't output anything
   --socks4 <host[:port]> SOCKS4 proxy on given host + port
   --socks4a <host[:port]> SOCKS4a proxy on given host + port
   --socks5 <host[:port]> SOCKS5 proxy on given host + port
   --socks5-hostname <host[:port]> SOCKS5 proxy, pass host
                  name to proxy
 -Y/--speed-limit   Stop transfer if below speed-limit for 'speed-time'
                  secs
 -y/--speed-time    Time needed to trig speed-limit abort. Defaults
                  to 30
 -2/--sslv2         Use SSLv2 (SSL)
 -3/--sslv3         Use SSLv3 (SSL)
   --stderr <file> Where to redirect stderr. - means stdout
   --tcp-nodelay   Use the TCP_NODELAY option
 -t/--telnet-option <OPT=val> Set telnet option
 -z/--time-cond <time> Transfer based on a time condition
 -1/--tlsv1         Use TLSv1 (SSL)
   --trace <file>  Write a debug trace to the given file
   --trace-ascii <file> Like --trace but without the hex output
   --trace-time    Add time stamps to trace/verbose output
 -T/--upload-file <file> Transfer <file> to remote site
   --url <URL>     Set URL to work with
 -B/--use-ascii     Use ASCII/text transfer
 -u/--user <user[:password]> Set server user and password
 -A/--user-agent <string> User-Agent to send to server (H)
 -v/--verbose       Make the operation more talkative
 -V/--version       Show version number and quit
 -w/--write-out <format> What to output after completion
 -q                 If used as the first parameter disables .curlrc
```

Cut

Usage: `cut [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Print selected parts of lines from each FILE to standard output.

Inputs Mandatory arguments to long options are mandatory for short options too.

```
  -b, --bytes=LIST          select only these bytes
  -c, --characters=LIST     select only these characters
  -d, --delimiter=DELIM     use DELIM instead of TAB for field
                            delimiter
  -f, --fields=LIST         select only these fields;  also print any line
                            that contains no delimiter character, unless
                            the -s option is specified
  -n                        (ignored)
      --complement          complement the set of selected bytes,
                            characters or fields.
  -s, --only-delimited      do not print lines not containing delimiters
      --output-delimiter=STRING
                            use STRING as the output delimiter
                            the default is to use the input delimiter
      --help                display this help and exit
      --version             output version information and exit
```

Use one, and only one of -b, -c or -f. Each LIST is made up of one range, or many ranges separated by commas. Selected input is written in the same order that it is read, and is written exactly once. Each range is one of:

```
  N          N'th byte, character or field, counted from 1
  N-         from N'th byte, character or field, to end of line
  N-M        from N'th to M'th (included) byte, character or field
  -M         from first to M'th (included) byte, character or field
```

With no FILE, or when FILE is -, read standard input.

Date

Usage: `date [OPTION]... [+FORMAT]`
`or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]`

Synopsis Unix-like command

Location Development:bin

Function Display the current time in the given FORMAT, or set the system date.

Inputs

```
-d, --date=STRING      display time described by STRING, not
                        `now'
-f, --file=DATEFILE     like --date once for each line of DATEFILE
-r, --reference=FILE    display the last modification time of FILE
-R, --rfc-2822          output date and time in RFC 2822 format.
                       Example: Mon, 07 Aug 2006 12:34:56 -0600
    --rfc-3339=TIMESPEC
                       output date and time in RFC 3339 format.
                       TIMESPEC=`date', `seconds', or `ns' for
                       date and time to the indicated precision.
                       Date and time components are separated by
                       a single space: 2006-08-07 12:34:56-06:00
-s, --set=STRING       set time described by STRING
-u, --utc, --universal print or set Coordinated Universal Time
    --help             display this help and exit
    --version          output version information and exit
```

FORMAT controls the output. The only valid option for the second form specifies Coordinated Universal Time. Interpreted sequences are:

```
%%   a literal %
%a   locale's abbreviated weekday name (e.g., Sun)
%A   locale's full weekday name (e.g., Sunday)
%b   locale's abbreviated month name (e.g., Jan)
%B   locale's full month name (e.g., January)
%c   locale's date and time (e.g., Thu Mar  3 23:05:25 2005)
%C   century; like %Y, except omit last two digits (e.g., 21)
%d   day of month (e.g, 01)
%D   date; same as %m/%d/%y
%e   day of month, space padded; same as %_d
%F   full date; same as %Y-%m-%d
%g   last two digits of year of ISO week number (see %G)
%G   year of ISO week number (see %V); normally useful only
       with %V
%h   same as %b
%H   hour (00..23)
%I   hour (01..12)
%j   day of year (001..366)
%k   hour ( 0..23)
%l   hour ( 1..12)
%m   month (01..12)
%M   minute (00..59)
%n   a newline
%N   nanoseconds (000000000..999999999)
%p   locale's equivalent of either AM or PM; blank if not known
%P   like %p, but lower case
```

%r locale's 12-hour clock time (e.g., 11:11:04 PM)
%R 24-hour hour and minute; same as %H:%M
%s seconds since 1970-01-01 00:00:00 UTC
%S second (00..60)
%t a tab
%T time; same as %H:%M:%S
%u day of week (1..7); 1 is Monday
%U week number of year, with Sunday as first day of week (00..53)
%V ISO week number, with Monday as first day of week (01..53)
%w day of week (0..6); 0 is Sunday
%W week number of year, with Monday as first day of week (00..53)
%x locale's date representation (e.g., 12/31/99)
%X locale's time representation (e.g., 23:13:48)
%y last two digits of year (00..99)
%Y year
%z +hhmm numeric timezone (e.g., -0400)
%:z +hh:mm numeric timezone (e.g., -04:00)
%::z +hh:mm:ss numeric time zone (e.g., -04:00:00)
%:::z numeric time zone with : to necessary precision (e.g., -04, +05:30)
%Z alphabetic time zone abbreviation (e.g., EDT)

By default, date pads numeric fields with zeroes.
The following optional flags may follow `%':

- (hyphen) do not pad the field
_ (underscore) pad with spaces
0 (zero) pad with zeros
^ use upper case if possible
# use opposite case if possible

After any flags comes an optional field width, as a decimal number; then an optional modifier, which is either
E to use the locale's alternate representations if available, or
O to use the locale's alternate numeric symbols if available.

Example `1.sys> list | grep Device`

Searches for the "Device" word in the output of List command. If exists, it show only the occurring line

Dc

Usage: `dc [OPTION] [file ...]`

Synopsis Unix-like command

Location Development:bin

Function DC is a reverse-polish desk calculator which supports unlimited precision arithmetic. It also allows you to define and call macros. Normally DC reads from the standard input; if any command arguments are given to it, they are filenames, and DC reads and executes the contents of the files instead of reading from standard input. All normal output is to standard output; all error messages are written to standard error.

A reverse-polish calculator stores numbers on a stack. Entering a number pushes it on the stack. Arithmetic operations pop arguments off the stack and push the results.

To enter a number in DC, type the digits, with an optional decimal point. Exponential notation is not supported. To enter a negative number, begin the number with `_'. `-' cannot be used for this, as it is a binary operator for subtraction instead. To enter two numbers in succession, separate them with spaces or newlines. These have no meaning as commands.

Inputs
-e, --expression=EXPR evaluate expression
-f, --file=FILE evaluate contents of file
-h, --help display this help and exit
-V, --version output version information and exit

Note A complete manual for dc is available here: http://www.gnu.org/software/bc/manual/dc-1.05/html_chapter/dc_toc.html

Dd

Usage: `dd [OPERAND]...`
or: `dd OPTION`

Synopsis Unix-like command

Location Development:bin

Function Copy a file, converting and formatting according to the operands.

Inputs
bs=BYTES force ibs=BYTES and obs=BYTES
cbs=BYTES convert BYTES bytes at a time
conv=CONVS convert the file as per the comma separated symbol list
count=BLOCKS copy only BLOCKS input blocks
ibs=BYTES read BYTES bytes at a time
if=FILE read from FILE instead of stdin

iflag=FLAGS read as per the comma separated symbol list
obs=BYTES write BYTES bytes at a time
of=FILE write to FILE instead of stdout
oflag=FLAGS write as per the comma separated symbol list
seek=BLOCKS skip BLOCKS obs-sized blocks at start of output
skip=BLOCKS skip BLOCKS ibs-sized blocks at start of input
status=noxfer suppress transfer statistics

Options are:

--help display this help and exit
--version output version information and exit

Further information

BLOCKS and BYTES may be followed by the following multiplicative suffixes: xM M, c 1, w 2, b 512, kB 1000, K 1024, MB 1000*1000, M 1024*1024, GB 1000*1000*1000, G 1024*1024*1024, and so on for T, P, E, Z, Y.

Each CONV symbol may be:

ascii from EBCDIC to ASCII
ebcdic from ASCII to EBCDIC
ibm from ASCII to alternate EBCDIC
block pad newline-terminated records with spaces to cbs-size
unblock replace trailing spaces in cbs-size records with newline
lcase change upper case to lower case
nocreat do not create the output file
excl fail if the output file already exists
notrunc do not truncate the output file
ucase change lower case to upper case
swab swap every pair of input bytes
noerror continue after read errors
sync pad every input block with NULs to ibs-size; when used with block or unblock, pad with spaces rather than NULs
fdatasync physically write output file data before finishing
fsync likewise, but also write metadata

Each FLAG symbol may be:

append append mode (makes sense only for output; conv=notrunc suggested)
dsync use synchronized I/O for data
sync likewise, but also for metadata
nonblock use non-blocking I/O

Sending a INFO signal to a running `dd' process makes it print I/O statistics to standard error and then resume copying.

```
$ dd if=/dev/zero of=/dev/null& pid=$!
$ kill -INFO $pid; sleep 1; kill $pid
18335302+0 records in
```

18335302+0 records out
9387674624 bytes (9.4 GB) copied, 34.6279 seconds, 271 MB/s

Derb

Usage:
```
derb [ -h, -?, --help ] [ -V, --version ]
     [ -v, --verbose ] [ -e, --encoding encoding ] [ --bom ]
     [ -t, --truncate [ size ] ]
     [ -s, --sourcedir source ] [ -d, --destdir destination ]
     [ -i, --icudatadir directory ] [ -c, --to-stdout ]
     [ -A, --suppressAliases]
     bundle ...
```

Synopsis Unix-like command

Location Development:bin

Function derb reads the compiled resource bundle files passed on the command line and write them back in text form. The resulting text files have a .txt extension while compiled resource bundle source files typically have a .res extension.

It is customary to name the resource bundles by their locale name, i.e. to use a local identifier for the bundle filename, e.g. ja_JP.res for Japanese (Japan) data, or root.res for the root bundle. This is especially important for derb since the locale name is not accessible directly from the compiled resource bundle, and to know which locale to ask for when opening the bundle. derb will produce a file whose base name is either the value of the -l, --locale option, or the same as the base name of the compiled resource file itself. If the --to-stdout, -c option is used, however, the text will be written on the standard output.

Inputs

-h, -?, --help
Print help about usage and exit.

-V, --version
Print the version of derb and exit.

-v, --verbose
Display extra informative messages during execution.

-e, --encoding encoding
Set the encoding used to write output files to encoding. The default encoding is the invariant (subset of ASCII or EBCDIC) codepage for the system (see section INVARIANT CHARACTERS). The choice of the encoding does not affect the data, just their representation. Characters that cannot be represented in the encoding will be represented using \uhhhh escape sequences.

--bom Write a byte order mark (BOM) at the beginning of the file.

-l, --locale locale

Set the locale for the resource bundle, which is used both in the generated text and as the base name of the output file.

-t, --truncate [size]
Truncate individual resources (strings or binary data) to size bytes. The default if size is not specified is 80 bytes.

-s, --sourcedir source
Set the source directory to source. The default source directory is the current directory. If - is passed for source, then the bundle will be looked for in its default location, specified by the ICU_DATA environment variable (or defaulting to the location set when ICU was built if ICU_DATA is not set).

-d, --destdir destination
Set the destination directory to destination. The default destination directory is specified by the environment variable ICU_DATA or is the location set when ICU was built if ICU_DATA is not set.

-i, --icudatadir directory
Look for any necessary ICU data files in directory. For example, when processing collation overrides, the file ucadata.dat must be located. The default ICU data directory is specified by the environment variable ICU_DATA.

-c, --to-stdout
Write the disassembled bundle on standard output instead of into a file.

Note
When the option --bom is used, the character U+FEFF is written in the destination encoding regardless of whether it is a Unicode transformation format (UTF) or not. This option should only be used with an UTF encoding, as byte order marks are not meaningful for other encodings.

Invariant characters
The invariant character set consists of the following set of characters, expressed as a standard POSIX regular expression: [a-z]|[A-Z]|[0-9]|_|-|+|-|*|/. This is the set which is guaranteed to be available regardless of code page.

Environment
ICU_DATA
Specifies the directory containing ICU data. Defaults to ${prefix}/share/icu/52.1/. Some tools in ICU depend on the presence of the trailing slash. It is thus important to make sure that it is present if ICU_DATA is set.

Authors
Vladimir Weinstein
Yves Arrouye

Diff

Usage: `diff [OPTION]... FILES`

Synopsis Unix-like command

Location Development:bin

Function Compare files line by line.

Inputs
-i --ignore-case
Ignore case differences in file contents.
--ignore-file-name-case
Ignore case when comparing file names.
--no-ignore-file-name-case
Consider case when comparing file names.
-E --ignore-tab-expansion
Ignore changes due to tab expansion.
-b --ignore-space-change
Ignore changes in the amount of white space.
-w --ignore-all-space
Ignore all white space.
-B --ignore-blank-lines
Ignore changes whose lines are all blank.
-I RE --ignore-matching-lines=RE
Ignore changes whose lines all match RE.
--strip-trailing-cr
Strip trailing carriage return on input.
-a --text
Treat all files as text.

-c -C NUM --context[=NUM]
Output NUM (default 3) lines of copied context.
-u -U NUM --unified[=NUM]
Output NUM (default 3) lines of unified context.
--label LABEL
Use LABEL instead of file name.
-p --show-c-function
Show which C function each change is in.
-F RE --show-function-line=RE
Show the most recent line matching RE.
-q --brief
Output only whether files differ.
-e --ed
Output an ed script.
--normal
Output a normal diff.
-n --rcs
Output an RCS format diff.
-y --side-by-side
Output in two columns.
-W NUM --width=NUM
Output at most NUM (default 130) print columns.
--left-column
Output only the left column of common lines.

```
  --suppress-common-lines
          Do not output common lines.
-D NAME  --ifdef=NAME
          Output merged file to show `#ifdef NAME' diffs.
--GTYPE-group-format=GFMT
          Similar, but format GTYPE input groups with GFMT.
--line-format=LFMT
          Similar, but format all input lines with LFMT.
--LTYPE-line-format=LFMT
          Similar, but format LTYPE input lines with LFMT.
          LTYPE is `old', `new', or `unchanged'.  GTYPE is LTYPE
          or `changed'.

  GFMT may contain:
    %<  lines from FILE1
    %>  lines from FILE2
    %=  lines common to FILE1 and FILE2
    %[-][WIDTH][.[PREC]]{doxX}LETTER
          printf-style spec for LETTER
      LETTERs are as follows for new group, lower case for old
          group:
        F  first line number
        L  last line number
        N  number of lines = L-F+1
        E  F-1
        M  L+1

  LFMT may contain:
    %L  contents of line
    %l  contents of line, excluding any trailing newline
    %[-][WIDTH][.[PREC]]{doxX}n  printf-style spec for input line
          number

  Either GFMT or LFMT may contain:
    %%  %
    %c'C'  the single character C
    %c'\OOO'  the character with octal code OOO

-l  --paginate
          Pass the output through `pr' to paginate it.
-t  --expand-tabs
          Expand tabs to spaces in output.
-T  --initial-tab
          Make tabs line up by prepending a tab.
--tabsize=NUM
          Tab stops are every NUM (default 8) print columns.

-r  --recursive
          Recursively compare any subdirectories found.
-N  --new-file
          Treat absent files as empty.
```

--unidirectional-new-file
Treat absent first files as empty.
-s --report-identical-files
Report when two files are the same.
-x PAT --exclude=PAT
Exclude files that match PAT.
-X FILE --exclude-from=FILE
Exclude files that match any pattern in FILE.
-S FILE --starting-file=FILE
Start with FILE when comparing directories.
--from-file=FILE1
Compare FILE1 to all operands. FILE1 can be a directory.
--to-file=FILE2
Compare all operands to FILE2. FILE2 can be a directory.

--horizon-lines=NUM
Keep NUM lines of the common prefix and suffix.
-d --minimal
Try hard to find a smaller set of changes.
--speed-large-files
Assume large files and many scattered small changes.

-v --version
Output version info.
--help
Output this help.

FILES are `FILE1 FILE2' or `DIR1 DIR2' or `DIR FILE...' or `FILE... DIR'. If --from-file or --to-file is given, there are no restrictions on FILES. If a FILE is `-', read standard input.

Diff3

Usage: `diff3 [OPTION]... MYFILE OLDFILE YOURFILE`

Synopsis Unix-like command

Location Development:bin

Function Compare three files line by line.

Inputs
-e --ed
Output unmerged changes from OLDFILE to YOURFILE into MYFILE.
-E --show-overlap
Output unmerged changes, bracketing conflicts.
-A --show-all
Output all changes, bracketing conflicts.

-x --overlap-only
Output overlapping changes.
-X
Output overlapping changes, bracketing them.
-3 --easy-only
Output unmerged nonoverlapping changes.

-m --merge
Output merged file instead of ed script (default -A).
-L LABEL --label=LABEL
Use LABEL instead of file name.
-i
Append `w' and `q' commands to ed scripts.
-a --text
Treat all files as text.
-T --initial-tab
Make tabs line up by prepending a tab.
--diff-program=PROGRAM
Use PROGRAM to compare files.

-v --version
Output version info.
--help
Output this help.

If a FILE is `-', read standard input.

Dir

Usage: `dir [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort.

Inputs Mandatory arguments to long options are mandatory for short options too.
-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
with -l, print the author of each file
-b, --escape
print octal escapes for nongraphic characters
--block-size=SIZE

```
          use SIZE-byte blocks
-B, --ignore-backups
          do not list implied entries ending with ~
-c
           with -lt: sort by, and show, ctime (time of last
           modification of file status information) with -l: show
           ctime and sort by name otherwise: sort by ctime
-C
           list entries by columns
   --color[=WHEN]
          control whether color is used to distinguish file types.
          WHEN may be `never', `always', or `auto'
-d, --directory
          list directory entries instead of contents, and do not
          dereference symbolic links
-D, --dired
          generate output designed for Emacs' dired mode
-f
          do not sort, enable -aU, disable -ls --color
-F, --classify
          append indicator (one of */=>@|) to entries
   --file-type
          likewise, except do not append `*'
   --format=WORD
          across -x, commas -m, horizontal -x, long -l,
          single-column -1, verbose -l, vertical -C
   --full-time
          like -l --time-style=full-iso
-g
          like -l, but do not list owner
   --group-directories-first
          group directories before files
-G, --no-group
          in a long listing, don't print group names
-h, --human-readable
          with -l, print sizes in human readable format
           (e.g., 1K 234M 2G)
   --si
          likewise, but use powers of 1000 not 1024
-H, --dereference-command-line
          follow symbolic links listed on the command line
   --dereference-command-line-symlink-to-dir
          follow each command line symbolic link
          that points to a directory
   --hide=PATTERN
          do not list implied entries matching shell PATTERN
          (overridden by -a or -A)
   --indicator-style=WORD
          append indicator with style WORD to entry names:
          none (default), slash (-p),
          file-type (--file-type), classify (-F)
```

```
-i, --inode
              print the index number of each file
-I, --ignore=PATTERN
               do not list implied entries matching shell PATTERN
-k
              like --block-size=1K
-l
              use a long listing format
-L, --dereference
               when showing file information for a symbolic
               link, show information for the file the link
              references rather than for the link itself
-m
              fill width with a comma separated list of entries
-n, --numeric-uid-gid
              like -l, but list numeric user and group IDs
-N, --literal
              print raw entry names (don't treat e.g. control
              characters specially)
-o
              like -l, but do not list group information
-p, --indicator-style=slash
              append / indicator to directories
-q, --hide-control-chars
              print ? instead of non graphic characters
    --show-control-chars
              show non graphic characters as-is (default
              unless program is `ls' and output is a terminal)
-Q, --quote-name
              enclose entry names in double quotes
    --quoting-style=WORD
              use quoting style WORD for entry names:
              literal, locale, shell, shell-always, c, escape
-r, --reverse
              reverse order while sorting
-R, --recursive
              list subdirectories recursively
-s, --size
              print the size of each file, in blocks
-S
              sort by file size
    --sort=WORD
              sort by WORD instead of name: none -U,
              extension -X, size -S, time -t, version -v
    --time=WORD
              with -l, show time as WORD instead of modification
              time: atime -u, access -u, use -u, ctime -c,
              or status -c; use specified time as sort key
              if --sort=time
    --time-style=STYLE
              with -l, show times using style STYLE:
```

full-iso, long-iso, iso, locale, +FORMAT.
FORMAT is interpreted like `date'; if FORMAT is
FORMAT1<newline>FORMAT2, FORMAT1 applies to
non-recent files and FORMAT2 to recent files;
if STYLE is prefixed with `posix-', STYLE
takes effect only outside the POSIX locale

-t
sort by modification time

-T, --tabsize=COLS
assume tab stops at each COLS instead of 8

-u
with -lt: sort by, and show, access time
with -l: show access time and sort by name
otherwise: sort by access time

-U
do not sort; list entries in directory order

-v
sort by version

-w, --width=COLS
assume screen width instead of current value

-x
list entries by lines instead of by columns

-X
sort alphabetically by entry extension

-1
list one file per line

--help
display this help and exit

--version
output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: kB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z, Y.

By default, color is not used to distinguish types of files. That is equivalent to using --color=none. Using the --color option without the optional WHEN argument is equivalent to using --color=always. With --color=auto, color codes are output only if standard output is connected to a terminal (tty). The environment variable LS_COLORS can influence the colors, and can be set easily by the dircolors command.

Notes Exit status is 0 if OK, 1 if minor problems, 2 if serious trouble.

Dircolors

Usage: `dircolors [OPTION]... [FILE]`

Synopsis Unix-like command

Location Development:bin

Function Output commands to set the LS_COLORS environment variable.

Inputs Determine format of output:
-b, --sh, --bourne-shell
output Bourne shell code to set LS_COLORS
-c, --csh, --c-shell
output C shell code to set LS_COLORS
-p, --print-database
output defaults
--help
display this help and exit
--version
output version information and exit

If FILE is specified, read it to determine which colors to use for which file types and extensions. Otherwise, a precompiled database is used. For details on the format of these files, run `dircolors --print-database'.

Dirname

Usage: `dirname NAME`
or: `dirname OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print NAME with its trailing /component removed

Inputs if NAME contains no /'s, output `.' (meaning the current directory).

--help display this help and exit
--version output version information and exit

Example `dirname /usr/bin/sort`
Output "/usr/bin".

`dirname stdio.h`
Output ".".

Du

Usage: `du [OPTION]... [FILE]...`

or: du [OPTION]... --files0-from=F

Synopsis Unix-like command

Location Development:bin

Function Summarize disk usage of each FILE, recursively for directories.

Inputs Mandatory arguments to long options are mandatory for short options too.

-a, --all
write counts for all files, not just directories
--apparent-size
print apparent sizes, rather than disk usage; although the apparent size is usually smaller, it may be larger due to holes in (`sparse') files, internal fragmentation, indirect blocks, and the like
-B, --block-size=SIZE
use SIZE-byte blocks
-b, --bytes
equivalent to `--apparent-size --block-size=1'
-c, --total
produce a grand total
-D, --dereference-args
dereference FILEs that are symbolic links
--files0-from=F
summarize disk usage of the NUL-terminated file names specified in file F
-H
like --si, but also evokes a warning; will soon change to be equivalent to --dereference-args (-D)
-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)
--si
like -h, but use powers of 1000 not 1024
-k
like --block-size=1K
-l, --count-links
count sizes many times if hard linked
-m
like --block-size=1M
-L, --dereference
dereference all symbolic links
-P, --no-dereference
don't follow any symbolic links (this is the default)
-0, --null
end each output line with 0 byte rather than newline
-S, --separate-dirs
do not include size of subdirectories
-s, --summarize
display only a total for each argument

-x, --one-file-system
skip directories on different file systems
-X FILE, --exclude-from=FILE
Exclude files that match any pattern in FILE.
--exclude=PATTERN
Exclude files that match PATTERN.
--max-depth=N
print the total for a directory (or file, with --all)
only if it is N or fewer levels below the command
line argument; --max-depth=0 is the same as
--summarize
--time
show time of the last modification of any file in the
directory, or any of its subdirectories
--time=WORD
show time as WORD instead of modification time:
atime, access, use, ctime or status
--time-style=STYLE
show times using style STYLE:
full-iso, long-iso, iso, +FORMAT
FORMAT is interpreted like `date'
--help
display this help and exit
--version
output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: kB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z, Y.

Echo

Usage: `echo [OPTION]... [STRING]...`

Synopsis Unix-like command

Location Development:bin

Function Echo the STRING(s) to standard output.

Inputs
-n do not output the trailing newline
-e enable interpretation of backslash escapes
-E disable interpretation of backslash escapes (default)
--help display this help and exit
--version output version information and exit

If -e is in effect, the following sequences are recognized:

\0NNN the character whose ASCII code is NNN (octal)

\\ backslash
\a alert (BEL)
\b backspace
\c suppress trailing newline
\f form feed
\n new line
\r carriage return
\t horizontal tab
\v vertical tab

Note AROS shell has its own version of echo, which usually supersedes the version described here. Please refer to AROS shell's main documentation for details about the options it supports.

Elfedit

Usage: `elfedit <option(s)> elffile(s)`

Synopsis Unix-like command

Location Development:bin

Function Update the ELF header of ELF files

Inputs The options are:

--input-mach <machine>	Set input machine type to <machine>
--output-mach <machine>	Set output machine type to <machine>
--input-type <type>	Set input file type to <type>
--output-type <type>	Set output file type to <type>
--input-osabi <osabi>	Set input OSABI to <osabi>
--output-osabi <osabi>	Set output OSABI to <osabi>
-h --help	Display this information
-v --version	Display the version number of elfedit

Env

Usage: `env [OPTION]... [-] [NAME=VALUE]... [COMMAND [ARG]...]`

Synopsis Unix-like command

Location Development:bin

Function Set each NAME to VALUE in the environment and run COMMAND.

Inputs

-i, --ignore-environment	start with an empty environment
-u, --unset=NAME	remove variable from the environment
--help	display this help and exit
--version	output version information and exit

A mere - implies -i. If no COMMAND, print the resulting environment.

Expand

Usage: `expand [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Convert tabs in each FILE to spaces, writing to standard output.

Inputs With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.

-i, --initial	do not convert tabs after non blanks
-t, --tabs=NUMBER	have tabs NUMBER characters apart, not 8
-t, --tabs=LIST	use comma separated list of explicit tab positions
--help	display this help and exit
--version	output version information and exit

Expr

Usage: `expr EXPRESSION`
or: `expr OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print the value of EXPRESSION to standard output.

Inputs A blank line below
separates increasing precedence groups. EXPRESSION may be:

ARG1 | ARG2	ARG1 if it is neither null nor 0, otherwise ARG2
ARG1 & ARG2	ARG1 if neither argument is null or 0, otherwise

0
ARG1 < ARG2 ARG1 is less than ARG2
ARG1 <= ARG2 ARG1 is less than or equal to ARG2
ARG1 = ARG2 ARG1 is equal to ARG2
ARG1 != ARG2 ARG1 is unequal to ARG2
ARG1 >= ARG2 ARG1 is greater than or equal to ARG2
ARG1 > ARG2 ARG1 is greater than ARG2
ARG1 + ARG2 arithmetic sum of ARG1 and ARG2
ARG1 - ARG2 arithmetic difference of ARG1 and ARG2
ARG1 * ARG2 arithmetic product of ARG1 and ARG2
ARG1 / ARG2 arithmetic quotient of ARG1 divided by ARG2
ARG1 % ARG2 arithmetic remainder of ARG1 divided by ARG2
STRING : REGEXP
anchored pattern match of REGEXP in STRING
match STRING REGEXP
same as STRING : REGEXP
substr STRING POS LENGTH
substring of STRING, POS counted from 1
index STRING CHARS
index in STRING where any CHARS is found, or
0
length STRING length of STRING
+ TOKEN interpret TOKEN as a string, even if it is a
keyword like `match' or an operator like `/'
(EXPRESSION) value of EXPRESSION

Beware that many operators need to be escaped or quoted for shells. Comparisons are arithmetic if both ARGs are numbers, else lexicographical. Pattern matches return the string matched between \(and \) or null; if \(and \) are not used, they return the number of characters matched or 0.

Exit status is 0 if EXPRESSION is neither null nor 0, 1 if EXPRESSION is null or 0, 2 if EXPRESSION is syntactically invalid, and 3 if an error occurred.

Faad

Usage: `faad [options] infile.aac`

Synopsis Unix-like command

Location Development:bin

Function *** REPLACE ME ***

Inputs Options:
-h Shows this help screen.
-i Shows info about the input file.

-a X Write MPEG-4 AAC ADTS output file.
-t Assume old ADTS format.
-o X Set output filename.
-f X Set output format. Valid values for X are:
1: Microsoft WAV format (default).
2: RAW PCM data.
-b X Set output sample format. Valid values for X are:
1: 16 bit PCM data (default).
2: 24 bit PCM data.
3: 32 bit PCM data.
4: 32 bit floating point data.
5: 64 bit floating point data.
-s X Force the samplerate to X (for RAW files).
-l X Set object type. Supported object types:
1: Main object type.
2: LC (Low Complexity) object type.
4: LTP (Long Term Prediction) object type.
23: LD (Low Delay) object type.
-d Down matrix 5.1 to 2 channels
-w Write output to stdio instead of a file.
-g Disable gapless decoding.
-q Quiet - suppresses status messages.

Example
faad infile.aac
faad infile.mp4
faad -o outfile.wav infile.aac
faad -w infile.aac > outfile.wav
faad -a outfile.aac infile.aac

Factor

Usage: `factor [NUMBER]...`
or: `factor OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print the prime factors of each NUMBER.

Inputs Print the prime factors of all specified integer NUMBERs. If no arguments are specified on the command line, they are read from standard input.

Options are:
--help display this help and exit
--version output version information and exit

False

Usage: `false [ignored command line arguments]`
or: `false OPTION`

Synopsis Unix-like command

Location Development:bin

Function Exit with a status code indicating failure.

Inputs
--help display this help and exit
--version output version information and exit

Note Icaros Desktop and other distribution provide a coding language called 'false'. Obviously, this is not the "same" *false* executable.

Find

Usage: `find [path...] [expression]`

Synopsis Unix-like command

Location Development:bin

Function find - search for files in a directory hierarchy

Inputs EXPRESSIONS
The expression is made up of options (which affect overall operation rather than the processing of a specific file, and always return true), tests (which return a true or false value), and actions (which have side effects and return a true or false value), all separated by operators. -and is assumed where the operator is omitted. If the expression contains no actions other than -prune, -print is performed on all files for which the expression is true.

OPTIONS
All options always return true. They always take effect, rather than being processed only when their place in the expression is reached. Therefore, for clarity, it is best to place them at the beginning of the expression.

-daystart
Measure times (for -amin, -atime, -cmin, -ctime, -mmin, and -mtime) from the beginning of today rather than from 24 hours

ago.

-depth
Process each directory's contents before the directory itself.

-follow
Dereference symbolic links. Implies -noleaf.

-help, --help
Print a summary of the command-line usage of find and exit.

-maxdepth levels
Descend at most levels (a non-negative integer) levels of directories below the command line arguments. `-maxdepth 0' means only apply the tests and actions to the command line arguments.

-mindepth levels
Do not apply any tests or actions at levels less than levels (a non-negative integer). `-mindepth 1' means process all files e except the command line arguments.

-mount
Don't descend directories on other filesystems. An alternate name for -xdev, for compatibility with some other versions of find.

-noleaf
Do not optimize by assuming that directories contain 2 fewer subdirectories than their hard link count. This option is needed when searching filesystems that do not follow the Unix directory-link convention, such as CD-ROM or MS-DOS filesystems or AFS volume mount points. Each directory on a normal Unix filesystem has at least 2 hard links: its name and its `.' entry. Additionally, its subdirectories (if any) each have a `..' entry linked to that directory. When find is examining a directory, after it has statted 2 fewer subdirectories than the directory's link count, it knows that the rest of the entries in the directory are non-directories (`leaf' files in the directory tree). If only the files' names need to be examined, there is no need to stat them; this gives a significant increase in search speed.

-version, --version
Print the find version number and exit.

-xdev
Don't descend directories on other filesystems.

TESTS

Numeric arguments can be specified as

+n
for greater than n,

-n
for less than n,

n
for exactly n.

-amin n
File was last accessed n minutes ago.

-anewer file

File was last accessed more recently than file was modified. -anewer is affected by -follow only if -follow comes before -anewer on the command line.

-atime n
File was last accessed n*24 hours ago.

-cmin n
File's status was last changed n minutes ago.

-cnewer file
File's status was last changed more recently than file was modified. -cnewer is affected by -follow only if -follow comes before -cnewer on the command line.

-ctime n
File's status was last changed n*24 hours ago.

-empty
File is empty and is either a regular file or a directory.

-false
Always false.

-fstype type
File is on a filesystem of type type. The valid filesystem types vary among different versions of Unix; an incomplete list of filesystem types that are accepted on some version of Unix or another is: ufs, 4.2, 4.3, nfs, tmp, mfs, S51K, S52K. You can use -printf with the %F directive to see the types of your filesystems.

-gid n
File's numeric group ID is n.

-group gname
File belongs to group gname (numeric group ID allowed).

-ilname pattern
Like -lname, but the match is case insensitive.

-iname pattern
Like -name, but the match is case insensitive. For example, the patterns `fo*' and `F??' match the file names `Foo', `FOO', `foo', `fOo', etc.

-inum n
File has inode number n.

-ipath pattern
Like -path, but the match is case insensitive.

-iregex pattern
Like -regex, but the match is case insensitive.

-links n
File has n links.

-lname pattern
File is a symbolic link whose contents match shell pattern pattern. The metacharacters do not treat `/' or `.' specially.

-mmin n
File's data was last modified n minutes ago.

-mtime n
File's data was last modified n*24 hours ago.

-name pattern
Base of file name (the path with the leading directories removed) matches shell pattern pattern. The metacharacters

(`*', `?', and `[]') do not match a `.' at the start of the base name. To ignore a directory and the files under it, use -prune; see an example in the description of -path.

-newer file
File was modified more recently than file. -newer is affected by -follow only if -follow comes before -newer on the command line.

-nouser
No user corresponds to file's numeric user ID.

-nogroup
No group corresponds to file's numeric group ID.

-path pattern
File name matches shell pattern pattern. The metacharacters do not treat `/' or `.' specially; so, for example,
find . -path './sr*sc'
will print an entry for a directory called './src/misc' (if one exists). To ignore a whole directory tree, use -prune rather than checking every file in the tree. For example, to skip the directory `src/emacs' and all files and directories under it, and print the names of the other files found, do something like this:
find . -path './src/emacs' -prune -o -print

-perm mode
File's permission bits are exactly mode (octal or symbolic). Symbolic modes use mode 0 as a point of departure.

-perm -mode
All of the permission bits mode are set for the file.

-perm +mode
Any of the permission bits mode are set for the file.

-regex pattern
File name matches regular expression pattern. This is a match on the whole path, not a search. For example, to match a file named `./fubar3', you can use the regular expression `.*bar.' or `.*b.*3', but not `b.*r3'.

-size n[bckw]
File uses n units of space. The units are 512-byte blocks by default or if `b' follows n, bytes if `c' follows n, kilobytes if `k' follows n, or 2-byte words if `w' follows n. The size does not count indirect blocks, but it does count blocks in sparse files that are not actually allocated.

-true
Always true.

-type c
File is of type c:

b
block (buffered) special

c
character (unbuffered) special

d
directory

p

named pipe (FIFO)

f
regular file

l
symbolic link

s
socket

D
door (Solaris)

-uid n
File's numeric user ID is n.

-used n
File was last accessed n days after its status was last changed.

-user uname
File is owned by user uname (numeric user ID allowed).

-xtype c
The same as -type unless the file is a symbolic link. For symbolic links: if -follow has not been given, true if the file is a link to a file of type c; if -follow has been given, true if c is `l'. In other words, for symbolic links, -xtype checks the type of the file that -type does not check.

ACTIONS

-exec command ;
Execute command; true if 0 status is returned. All following arguments to find are taken to be arguments to the command until an argument consisting of `;' is encountered. The string `{}' is replaced by the current file name being processed everywhere it occurs in the arguments to the command, not just in arguments where it is alone, as in some versions of find. Both of these constructions might need to be escaped (with a `\') or quoted to protect them from expansion by the shell. The command is executed in the starting directory.

-fls file
True; like -ls but write to file like -fprint.

-fprint file
True; print the full file name into file file. If file does not exist when find is run, it is created; if it does exist, it is truncated. The file names ``/dev/stdout'' and ``/dev/stderr'' are handled specially; they refer to the standard output and standard error output, respectively.

-fprint0 file
True; like -print0 but write to file like -fprint.

-fprintf file format
True; like -printf but write to file like -fprint.

-ok command ;
Like -exec but ask the user first (on the standard input); if the response does not start with `y' or `Y', do not run the command, and return false.

-print
True; print the full file name on the standard output, followed by a newline.

-print0
True; print the full file name on the standard output, followed by a null character. This allows file names that contain newlines to be correctly interpreted by programs that process the find output.

-printf format
True; print format on the standard output, interpreting `\' escapes and `%' directives. Field widths and precisions can be specified as with the `printf' C function. Unlike -print, -printf does not add a newline at the end of the string. The escapes and directives are:

\a
Alarm bell.

\b
Backspace.

\c
Stop printing from this format immediately and flush the output.

\f
Form feed.

\n
Newline.

\r
Carriage return.

\t
Horizontal tab.

\v
Vertical tab.

\\
A literal backslash (`\').

\NNN
The character whose ASCII code is NNN (octal).

A `\' character followed by any other character is treated as an ordinary character, so they both are printed.

%%
A literal percent sign.

%a
File's last access time in the format returned by the C `ctime' function.

%Ak
File's last access time in the format specified by k, which is either `@' or a directive for the C `strftime' function. The possible values for k are listed below; some of them might not be available on all systems, due to differences in `strftime' between systems.

@
: seconds since Jan. 1, 1970, 00:00 GMT.

Time fields:

H
: hour (00..23)

I
: hour (01..12)

k
: hour (0..23)

l
: hour (1..12)

M
: minute (00..59)

p
: locale's AM or PM

r
: time, 12-hour (hh:mm:ss [AP]M)

S
: second (00..61)

T
: time, 24-hour (hh:mm:ss)

X
: locale's time representation (H:M:S)

Z
: time zone (e.g., EDT), or nothing if no time zone is determinable

Date fields:

a
: locale's abbreviated weekday name (Sun..Sat)

A
: locale's full weekday name, variable length (Sunday..Saturday)

b
: locale's abbreviated month name (Jan..Dec)

B
: locale's full month name, variable length (January..December)

c
: locale's date and time (Sat Nov 04 12:02:33 EST 1989)

d
: day of month (01..31)

D
: date (mm/dd/yy)

h
: same as b

j
day of year (001..366)
m
month (01..12)
U
week number of year with Sunday as first day of week (00..53)
w
day of week (0..6)
W
week number of year with Monday as first day of week (00..53)
x
locale's date representation (mm/dd/yy)
y
last two digits of year (00..99)
Y
year (1970...)

%b
File's size in 512-byte blocks (rounded up).
%c
File's last status change time in the format returned by the C `ctime' function.
%Ck
File's last status change time in the format specified by k, which is the same as for %A.
%d
File's depth in the directory tree; 0 means the file is a command line argument.
%f
File's name with any leading directories removed (only the last element).
%F
Type of the filesystem the file is on; this value can be used for -fstype.
%g
File's group name, or numeric group ID if the group has no name.
%G
File's numeric group ID.
%h
Leading directories of file's name (all but the last element).
%H
Command line argument under which file was found.
%i
File's inode number (in decimal).
%k
File's size in 1K blocks (rounded up).
%l
Object of symbolic link (empty string if file is not a

symbolic link).

%m
File's permission bits (in octal).

%n
Number of hard links to file.

%p
File's name.

%P
File's name with the name of the command line argument under which it was found removed.

%s
File's size in bytes.

%t
File's last modification time in the format returned by the C `ctime' function.

%Tk
File's last modification time in the format specified by k, which is the same as for %A.

%u
File's user name, or numeric user ID if the user has no name.

%U
File's numeric user ID.

A `%' character followed by any other character is discarded (but the other character is printed).

-prune
If -depth is not given, true; do not descend the current directory.
If -depth is given, false; no effect.

-ls
True; list current file in `ls -dils' format on standard output. The block counts are of 1K blocks, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

OPERATORS
Listed in order of decreasing precedence:

(expr)
Force precedence.

! expr
True if expr is false.

-not expr
Same as ! expr.

expr1 expr2
And (implied); expr2 is not evaluated if expr1 is false.

expr1 -a expr2
Same as expr1 expr2.

expr1 -and expr2

Same as expr1 expr2.

expr1 -o expr2

Or; expr2 is not evaluated if expr1 is true.

expr1 -or expr2

Same as expr1 -o expr2.

expr1 , expr2

List; both expr1 and expr2 are always evaluated. The value of expr1 is discarded; the value of the list is the value of expr2.

Flac

Usage: Encoding: `flac [<general-options>] [<encoding/format-options>] [INPUTFILE [...]]`
Decoding: `flac -d [<general-options>] [<format-options>] [FLACFILE [...]]`
Testing: `flac -t [<general-options>] [FLACFILE [...]]`
Analyzing: `flac -a [<general-options>] [<analysis-options>] [FLACFILE [...]]`

Synopsis Unix-like command

Location Development:bin

Function Command-line FLAC encoder/decoder

Inputs general options:

-v, --version Show the flac version number
-h, --help Show this screen
-H, --explain Show detailed explanation of usage and options
-d, --decode Decode (the default behavior is to encode)
-t, --test Same as -d except no decoded file is written
-a, --analyze Same as -d except an analysis file is written
-c, --stdout Write output to stdout
-s, --silent Do not write runtime encode/decode statistics
--totally-silent Do not print anything, including errors
--no-utf8-convert Do not convert tags from local charset to UTF-8
-w, --warnings-as-errors Treat all warnings as errors
-f, --force Force overwriting of output files
-o, --output-name=FILENAME Force the output file name
--output-prefix=STRING Prepend STRING to output names
--delete-input-file Deletes after a successful encode/decode
--keep-foreign-metadata

```
                    Save/restore WAVE or AIFF non-audio chunks
    --skip={#|mm:ss.ss}
                    Skip the given initial samples for each input
    --until={#|[+|-]mm:ss.ss}
                    Stop at the given sample for each input file
    --ogg           Use Ogg as transport layer
    --serial-number
                    Serial number to use for the FLAC stream

analysis options:
    --residual-text     Include residual signal in text output
    --residual-gnuplot
                    Generate gnuplot files of residual distribution

decoding options:
  -F, --decode-through-errors
                    Continue decoding through stream errors
    --cue=[#.#][-[#.#]]
                    Set the beginning and ending cuepoints to
                    decode

encoding options:
  -V, --verify          Verify a correct encoding
    --lax               Allow encoder to generate non-Subset files
    --sector-align      Align multiple files on sector boundaries
    --replay-gain       Calculate ReplayGain & store in FLAC tags
    --cuesheet=FILENAME
                    Import cuesheet and store in CUESHEET block
    --picture=SPECIFICATION
                    Import picture and store in PICTURE block
  -T, --tag=FIELD=VALUE
                    Add a FLAC tag; may appear multiple times
    --tag-from-file=FIELD=FILENAME
                    Like --tag but gets value from file
  -S, --seekpoint={#|X|#x|#s}
                    Add seek point(s)
  -P, --padding=    Write a PADDING block of length #
  -0, --compression-level-0, --fast
                    Synonymous with -l 0 -b 1152 -r 3
  -1, --compression-level-1
                    Synonymous with -l 0 -b 1152 -M -r 3
  -2, --compression-level-2
                    Synonymous with -l 0 -b 1152 -m -r 3
  -3, --compression-level-3
                    Synonymous with -l 6 -b 4096 -r 4
  -4, --compression-level-4
                    Synonymous with -l 8 -b 4096 -M -r 4
  -5, --compression-level-5
                    Synonymous with -l 8 -b 4096 -m -r 5
  -6, --compression-level-6
                    Synonymous with -l 8 -b 4096 -m -r 6
```

```
  -7, --compression-level-7
                        Synonymous with -l 8 -b 4096 -m -e -r 6
  -8, --compression-level-8, --best
                        Synonymous with -l 12 -b 4096 -m -e -r 6
  -b, --blocksize=#     Specify blocksize in samples
  -m, --mid-side        Try mid-side coding for each frame
  -M, --adaptive-mid-side
                        Adaptive mid-side coding for all frames
  -e, --exhaustive-model-search
                        Do exhaustive model search (expensive!)
  -A, --apodization="function"
                        Window audio data with given the function
  -l, --max-lpc-order=#
                        Max LPC order; 0 => only fixed predictors
  -p, --qlp-coeff-precision-search
                        Exhaustively search LP coeff quantization
  -q, --qlp-coeff-precision=#
                        Specify precision in bits
  -r, --rice-partition-order=[#,]#
                        Set [min,]max residual partition order

format options:
      --endian={big|little}          Set byte order for samples
      --channels=#                   Number of channels
      --bps=#                        Number of bits per sample
      --sample-rate=#                Sample rate in Hz
      --sign={signed|unsigned}       Sign of samples
      --input-size=#                 Size of the raw input in bytes
      --force-aiff-format            Force decoding to AIFF format
      --force-raw-format             Treat input or output as raw
                                      samples

negative options:
      --no-adaptive-mid-side
      --no-decode-through-errors
      --no-delete-input-file
      --no-keep-foreign-metadata
      --no-exhaustive-model-search
      --no-lax
      --no-mid-side
      --no-ogg
      --no-padding
      --no-qlp-coeff-prec-search
      --no-replay-gain
      --no-residual-gnuplot
      --no-residual-text
      --no-sector-align
      --no-seektable
      --no-silent
      --no-force
      --no-verify
```

--no-warnings-as-errors

Notes This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Flex

Usage: `flex [-bcdfhilnpstvwBFILTV78+? -C[aefFmr] -ooutput -Pprefix -Sskeleton] [--help --version] [file ...]`

Synopsis Unix-like command

Location Development:bin

Function Flex takes pairs of regular expressions and C code as input and generates a C source file as output.

Inputs
-b generate backing-up information to lex.backup
-c do-nothing POSIX option
-d turn on debug mode in generated scanner
-f generate fast, large scanner
-h produce this help message
-i generate case-insensitive scanner
-l maximal compatibility with original lex
-n do-nothing POSIX option
-p generate performance report to stderr
-s suppress default rule to ECHO unmatched text
-t write generated scanner on stdout instead of lex.yy.c
-v write summary of scanner statistics to f
-w do not generate warnings
-B generate batch scanner (opposite of -I)
-F use alternative fast scanner representation
-I generate interactive scanner (opposite of -B)
-L suppress #line directives in scanner
-T flex should run in trace mode
-V report flex version
-7 generate 7-bit scanner
-8 generate 8-bit scanner
-+ generate C++ scanner class
-? produce this help message
-C specify degree of table compression (default is -Cem):
-Ca trade off larger tables for better memory alignment

-Ce construct equivalence classes
-Cf do not compress scanner tables; use -f representation
-CF do not compress scanner tables; use -F representation
-Cm construct meta-equivalence classes
-Cr use read() instead of stdio for scanner input
-o specify output filename
-P specify scanner prefix other than "yy"
-S specify skeleton file
--help produce this help message
--version report flex version

Flex++

Usage: `flex++ [-bcdfhilnpstvwBFILTV78+? -C[aefFmr] -ooutput -Pprefix -Sskeleton] [--help --version] [file ...]`

Synopsis Unix-like command

Location Development:bin

Function Fast lexical analyzer generator

Inputs
-b generate backing-up information to lex.backup
-c do-nothing POSIX option
-d turn on debug mode in generated scanner
-f generate fast, large scanner
-h produce this help message
-i generate case-insensitive scanner
-l maximal compatibility with original lex
-n do-nothing POSIX option
-p generate performance report to stderr
-s suppress default rule to ECHO unmatched text
-t write generated scanner on stdout instead of lex.yy.cc
-v write summary of scanner statistics to f
-w do not generate warnings
-B generate batch scanner (opposite of -I)
-F use alternative fast scanner representation
-I generate interactive scanner (opposite of -B)
-L suppress #line directives in scanner
-T flex++ should run in trace mode
-V report flex++ version
-7 generate 7-bit scanner
-8 generate 8-bit scanner
-+ generate C++ scanner class
-? produce this help message
-C specify degree of table compression (default is -Cem):
-Ca trade off larger tables for better memory alignment
-Ce construct equivalence classes

-Cf do not compress scanner tables; use -f representation
-CF do not compress scanner tables; use -F representation
-Cm construct meta-equivalence classes
-Cr use read() instead of stdio for scanner input
-o specify output filename
-P specify scanner prefix other than "yy"
-S specify skeleton file
--help produce this help message
--version report flex++ version

Fmt

Usage: `fmt [-DIGITS] [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Reformat each paragraph in the FILE(s), writing to standard output.
If no FILE or if FILE is `-', read standard input.

Inputs Mandatory arguments to long options are mandatory for short options too.

-c, --crown-margin	preserve indentation of first two lines
-p, --prefix=STRING	reformat only lines beginning with STRING, reattaching the prefix to reformatted lines
-s, --split-only	split long lines, but do not refill
-t, --tagged-paragraph	indentation of first line different from second
-u, --uniform-spacing	one space between words, two after sentences
-w, --width=WIDTH	maximum line width (default of 75 columns)
--help	display this help and exit
--version	output version information and exit

With no FILE, or when FILE is -, read standard input.

Fold

Usage: `fold [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Wrap input lines in each FILE (standard input by default), writing to standard output.

Inputs Mandatory arguments to long options are mandatory for short options too.

```
  -b, --bytes             count bytes rather than columns
  -s, --spaces            break at spaces
  -w, --width=WIDTH   use WIDTH columns instead of 80
      --help              display this help and exit
      --version           output version information and exit
```

g++

Usage: `g++ [options] file...`

Synopsis Unix-like command

Location Development:bin

Function Compiles human-readable sources into machine-readable code

Inputs Options:

```
  -pass-exit-codes
       Exit with highest error code from a phase
  --help
       Display this information
  --target-help
       Display target specific command line options
  (Use '-v --help' to display command line options of sub-processes)
  -dumpspecs
       Display all of the built in spec strings
  -dumpversion
       Display the version of the compiler
  -dumpmachine
       Display the compiler's target processor
  -print-search-dirs
       Display the directories in the compiler's search path
  -print-libgcc-file-name
       Display the name of the compiler's companion library
  -print-file-name=<lib>
       Display the full path to library <lib>
  -print-prog-name=<prog>
       Display the full path to compiler component <prog>
  -print-multi-directory
        Display the root directory for versions of libgcc
  -print-multi-lib
        Display the mapping between command line options and
                        multiple library search directories
  -print-multi-os-directory
```

```
    Display the relative path to OS libraries
-Wa,<options>
    Pass comma-separated <options> on to the assembler
-Wp,<options>
    Pass comma-separated <options> on to the preprocessor
-Wl,<options>
    Pass comma-separated <options> on to the linker
-Xassembler <arg>
    Pass <arg> on to the assembler
-Xpreprocessor <arg>
    Pass <arg> on to the preprocessor
-Xlinker <arg>
    Pass <arg> on to the linker
-combine
    Pass multiple source files to compiler at once
-save-temps
    Do not delete intermediate files
-pipe
    Use pipes rather than intermediate files
-time
   Time the execution of each subprocess
-specs=<file>
   Override built-in specs with the contents of <file>
-std=<standard>
   Assume that the input sources are for <standard>
--sysroot=<directory>
    Use <directory> as the root directory for headers
    and libraries
-B <directory>
    Add <directory> to the compiler's search paths
-b <machine>
    Run gcc for target <machine>, if installed
-V <version>
    Run gcc version number <version>, if installed
-v
    Display the programs invoked by the compiler
-###
    Like -v but options quoted and commands not executed
-E
    Preprocess only; do not compile, assemble or link
-S
    Compile only; do not assemble or link
-c
    Compile and assemble, but do not link
-o <file>
    Place the output into <file>
-x <language>
    Specify the language of the following input files
     Permissible languages include: c c++ assembler none
     'none' means revert to the default behavior of
     guessing the language based on the file's extension
```

Options starting with -g, -f, -m, -O, -W, or --param are automatically passed on to the various sub-processes invoked by g++. In order to pass other options on to these processes the -W<letter> options must be used.

Gawk / Gawk-

Usage: `gawk[-] [POSIX or GNU style options] -f progfile [--] file ...`
Or: `gawk[-] [POSIX or GNU style options] [--] 'program' file ...`

Synopsis Unix-like command

Location Development:bin

Function Gawk is a pattern scanning and processing language.
By default it reads standard input and writes standard output.

Inputs Gawk accepts the following options, listed alphabetically.

-F fs --field-separator fs
Use fs for the input field separator (the value of the FS predefined variable).
-v var=val --assign var=val
Assign the value val to the variable var, before execution of the program begins. Such variable values are available to the BEGIN block of an AWK program.
-f program-file --file program-file
Read the AWK program source from the file program-file, instead of from the first command line argument. Multiple -f (or --file) options may be used.
-mf NNN -mr NNN
Set various memory limits to the value NNN. The f flag sets the maximum number of fields, and the r flag sets the maximum record size. These two flags and the -m option are from the Bell Laboratories research version of UNIX awk. They are ignored by gawk, since gawk has no pre-defined limits.
-W compat -W traditional
--compat --traditional
Run in compatibility mode. In compatibility mode, gawk behaves identically to UNIX awk; none of the GNU-specific extensions are recognized. The use of --traditional is preferred over the other forms of this option. See GNU EXTENSIONS, below, for more information.
-W copyleft -W copyright
--copyleft --copyright
Print the short version of the GNU copyright information message on the standard output and exit

successfully.

-W dump-variables[=file] --dump-variables[=file]
Print a sorted list of global variables, their types and final values to file. If no file is provided, gawk uses a file named awkvars.out in the current directory.

Having a list of all the global variables is a good way to look for typographical errors in your programs. You would also use this option if you have a large program with a lot of functions, and you want to be sure that your functions don't inadvertently use global variables that you meant to be local. (This is a particularly easy mistake to make with simple variable names like i, j, and so on.)

-W help --help
-W usage --usage
Print a relatively short summary of the available options on the standard output. (Per the GNU Coding Standards, these options cause an immediate, successful exit.)

-W lint[=fatal] --lint[=fatal]
Provide warnings about constructs that are dubious or non-portable to other AWK implementations. With an optional argument of fatal, lint warnings become fatal errors. This may be drastic, but its use will certainly encourage the development of cleaner AWK programs.

-W lint-old --lint-old
Provide warnings about constructs that are not portable to the original version of Unix awk.

-W gen-po --gen-po
Scan and parse the AWK program, and generate a GNU .po format file on standard output with entries for all localizable strings in the program. The program itself is not executed. See the GNU gettext distribution for more information on .po files.

-W non-decimal-data --non-decimal-data
Recognize octal and hexadecimal values in input data. Use this option with great caution!

-W posix --posix
This turns on compatibility mode, with the following additional restrictions:

* \x escape sequences are not recognized.
* Only space and tab act as field separators when FS is set to a single space, newline does not.
* You cannot continue lines after ? and :.
* The synonym func for the keyword function is not recognized.
*

The operators ** and **= cannot be used in place of ^ and ^=.

*
The fflush() function is not available.

-W profile[=prof_file] --profile[=prof_file]
Send profiling data to prof_file. The default is awkprof.out. When run with gawk, the profile is just a ``pretty printed'' version of the program. When run with pgawk, the profile contains execution counts of each statement in the program in the left margin and function call counts for each user-defined function.

-W re-interval --re-interval
Enable the use of interval expressions in regular expression matching (see Regular Expressions, below). Interval expressions were not traditionally available in the AWK language. The POSIX standard added them, to make awk and egrep consistent with each other. However, their use is likely to break old AWK programs, so gawk only provides them if they are requested with this option, or when --posix is specified.

-W source program-text --source program-text
Use program-text as AWK program source code. This option allows the easy intermixing of library functions (used via the -f and --file options) with source code entered on the command line. It is intended primarily for medium to large AWK programs used in shell scripts.

-W version --version
Print version information for this particular copy of gawk on the standard output. This is useful mainly for knowing if the current copy of gawk on your system is up to date with respect to whatever the Free Software Foundation is distributing. This is also useful when reporting bugs. (Per the GNU Coding Standards, these options cause an immediate, successful exit)

-- Signal the end of options. This is useful to allow further arguments to the AWK program itself to start with a ``-''. This is mainly for consistency with the argument parsing convention used by most other POSIX programs.

Example

```
gawk '{ sum += $1 }; END { print sum }' file

gawk -F: '{ print $1 }' /etc/passwd
```

Note A far more complete description of Gawk is available here: http://linux.about.com/library/cmd/blcmdl1_gawk.htm

Gcc

Usage: `gcc [options] file...`

Synopsis Unix-like command

Location Development:bin

Function Turns human-readable source files into executables

Inputs

-pass-exit-codes	Exit with highest error code from a phase
--help	Display this information
--target-help	Display target specific command line options (Use '-v --help' to display command line options of sub-processes)
-dumpspecs	Display all of the built in spec strings
-dumpversion	Display the version of the compiler
-dumpmachine	Display the compiler's target processor
-print-search-dirs	Display the directories in the compiler's search path
-print-libgcc-file-name	Display the name of the compiler's companion library
-print-file-name=<lib>	Display the full path to library <lib>
-print-prog-name=<prog>	Display the full path to compiler component <prog>
-print-multi-directory	Display the root directory for versions of libgcc
-print-multi-lib	Display the mapping between command line options and multiple library search directories
-print-multi-os-directory	Display the relative path to OS libraries
-Wa,<options>	Pass comma-separated <options> on to the assembler
-Wp,<options>	Pass comma-separated <options> on to the preprocessor
-Wl,<options>	Pass comma-separated <options> on to the linker
-Xassembler <arg>	Pass <arg> on to the assembler
-Xpreprocessor <arg>	Pass <arg> on to the preprocessor
-Xlinker <arg>	Pass <arg> on to the linker
-combine	Pass multiple source files to compiler at once
-save-temps	Do not delete intermediate files

-pipe	Use pipes rather than intermediate files
-time	Time the execution of each subprocess
-specs=<file>	Override built-in specs with the contents of <file>
-std=<standard>	Assume that the input sources are for <standard>
--sysroot=<directory>	Use <directory> as the root directory for headers and libraries
-B <directory>	Add <directory> to the compiler's search paths
-b <machine>	Run gcc for target <machine>, if installed
-V <version>	Run gcc version number <version>, if installed
-v	Display the programs invoked by the compiler
-###	Like -v but options quoted and commands not executed
-E	Preprocess only; do not compile, assemble or link
-S	Compile only; do not assemble or link
-c	Compile and assemble, but do not link
-o <file>	Place the output into <file>
-x <language>	Specify the language of the following input files. Permissible languages include: c c++ assembler none 'none' means revert to the default behavior of guessing the language based on the file's extension

Options starting with -g, -f, -m, -O, -W, or --param are automatically passed on to the various sub-processes invoked by gcc. In order to pass other options on to these processes the -W<letter> options must be used.

Gcov

Usage: `gcov [OPTION]... SOURCEFILE`

Synopsis Unix-like command

Location Development:bin

Function Print code coverage information.

Inputs

-h, --help	Print this help, then exit
-v, --version	Print version number, then exit
-a, --all-blocks	Show information for every basic block
-b, --branch-probabilities	Include branch probabilities in output
-c, --branch-counts	Given counts of branches taken

rather than percentages
-n, --no-output Do not create an output file
-l, --long-file-names Use long output file names for included source files
-f, --function-summaries
Output summaries for each function
-o, --object-directory DIR|FILE
Search for object files in DIR or called FILE
-p, --preserve-paths Preserve all pathname components
-u, --unconditional-branches
Show unconditional branch counts too

Genbrk

Usage: `genbrk [-v] [-options] -r rule-file -o output-file`

Synopsis Unix-like command

Location Development:bin

Function Read in break iteration rules text and write out the binary data

Inputs options:
-h or -? or --help this usage text
-V or --version show a version message
-c or --copyright include a copyright notice
-v or --verbose turn on verbose output
-i or --icudatadir directory for locating any needed intermediate data files,
followed by path, defaults to
-d or --destdir destination directory, followed by the path

Genctd

Usage: `genctd [-v] [-options] -o output-file dictionary-file`

Synopsis Unix-like command

Location Development:bin

Function Read in word list and write out compact trie dictionary

Inputs -h or -? or --help this usage text
-V or --version show a version message
-c or --copyright include a copyright notice
-v or --verbose turn on verbose output

-i or --icudatadir	directory for locating any needed intermediate data files, followed by path, defaults to
-d or --destdir	destination directory, followed by the path

Gprof

Usage:
```
gprof [-[abcDhilLsTvwxyz]] [-[ACeEfFJnNOpPqSQZ][name]] [-I dirs]
[-d[num]] [-k from/to] [-m min-count] [-t table-length]
[--[no-]annotated-source[=name]] [--[no-]exec-counts[=name]]
[--[no-]flat-profile[=name]] [--[no-]graph[=name]]
[--[no-]time=name] [--all-lines] [--brief] [--debug[=level]]
[--function-ordering] [--file-ordering]
[--directory-path=dirs] [--display-unused-functions]
[--file-format=name] [--file-info] [--help] [--line] [--min-
count=n]  [--no-static] [--print-path] [--separate-files]
[--static-call-graph] [--sum] [--table-length=len] [--traditional]
[--version] [--width=n] [--ignore-non-functions]
[--demangle[=STYLE]] [--no-demangle] [--external-symbol-table=name]
[@FILE] [image-file] [profile-file...]
```

Synopsis Unix-like command

Location Development:bin

Function Gprof is the GNU profiler, written by Jay Fenlason.

Inputs OUTPUT OPTIONS:

-A[symspec] --annotated-source[=symspec]
The `-A' option causes gprof to print annotated source code. If symspec is specified, print output only for matching symbols. See The Annotated Source Listing.

-b --brief
If the `-b' option is given, gprof doesn't print the verbose blurbs that try to explain the meaning of all of the fields in the tables. This is useful if you intend to print out the output, or are tired of seeing the blurbs.

-C[symspec] --exec-counts[=symspec]
The `-C' option causes gprof to print a tally of functions and the number of times each was called. If symspec is specified, print tally only for matching symbols.

If the profile data file contains basic-block count records, specifying the `-l' option, along with `-C', will cause basic-block execution counts to be tallied and displayed.

-i --file-info
The `-i' option causes gprof to display summary information about the profile data file(s) and then exit. The number of histogram, call graph, and basic-block count records is displayed.

-I dirs --directory-path=dirs
The `-I' option specifies a list of search directories in which to find source files. Environment variable GPROF_PATH can also be used to convey this information. Used mostly for annotated source output.

-J[symspec] --no-annotated-source[=symspec]
The `-J' option causes gprof not to print annotated source code. If symspec is specified, gprof prints annotated source, but excludes matching symbols.

-L --print-path
Normally, source filenames are printed with the path component suppressed. The `-L' option causes gprof to print the full pathname of source filenames, which is determined from symbolic debugging information in the image file and is relative to the directory in which the compiler was invoked.

-p[symspec] --flat-profile[=symspec]
The `-p' option causes gprof to print a flat profile. If symspec is specified, print flat profile only for matching symbols. See The Flat Profile.

-P[symspec] --no-flat-profile[=symspec]
The `-P' option causes gprof to suppress printing a flat profile. If symspec is specified, gprof prints a flat profile, but excludes matching symbols.

-q[symspec] --graph[=symspec]
The `-q' option causes gprof to print the call graph analysis. If symspec is specified, print call graph only for matching symbols and their children. See The Call Graph.

-Q[symspec] --no-graph[=symspec]
The `-Q' option causes gprof to suppress printing the call graph. If symspec is specified, gprof prints a call graph, but excludes matching symbols.

-t --table-length=num
The `-t' option causes the num most active source lines in each source file to be listed when source annotation is enabled. The default is 10.

-y --separate-files
This option affects annotated source output only. Normally, gprof prints annotated source files to standard-output. If this option

is specified, annotated source for a file named path/filename is generated in the file filename-ann. If the underlying file system would truncate filename-ann so that it overwrites the original filename, gprof generates annotated source in the file filename.ann instead (if the original file name has an extension, that extension is replaced with .ann).

-Z[symspec] --no-exec-counts[=symspec]
The `-Z' option causes gprof not to print a tally of functions and the number of times each was called. If symspec is specified, print tally, but exclude matching symbols.

-r --function-ordering
The `--function-ordering' option causes gprof to print a suggested function ordering for the program based on profiling data. This option suggests an ordering which may improve paging, tlb and cache behavior for the program on systems which support arbitrary ordering of functions in an executable.

The exact details of how to force the linker to place functions in a particular order is system dependent and out of the scope of this manual.

-R map_file --file-ordering map_file
The `--file-ordering' option causes gprof to print a suggested .o link line ordering for the program based on profiling data. This option suggests an ordering which may improve paging, tlb and cache behavior for the program on systems which do not support arbitrary ordering of functions in an executable.

Use of the `-a' argument is highly recommended with this option.

The map_file argument is a pathname to a file which provides function name to object file mappings. The format of the file is similar to the output of the program nm.

```
c-parse.o:00000000 T yyparse
c-parse.o:00000004 C yyerrflag
c-lang.o:00000000 T maybe_objc_method_name
c-lang.o:00000000 T print_lang_statistics
c-lang.o:00000000 T recognize_objc_keyword
c-decl.o:00000000 T print_lang_identifier
c-decl.o:00000000 T print_lang_type
...
```

To create a map_file with gnu nm, type a command like nm --extern-only --defined-only -v --print-file-name program-name.

-T --traditional
The `-T' option causes gprof to print its output in “traditional”

BSD style.

-w width --width=width
  Sets width of output lines to width. Currently only used when printing the function index at the bottom of the call graph.

-x --all-lines
  This option affects annotated source output only. By default, only the lines at the beginning of a basic-block are annotated. If this option is specified, every line in a basic-block is annotated by repeating the annotation for the first line. This behavior is similar to tcov's `-a'.

--demangle[=style] --no-demangle
  These options control whether C++ symbol names should be demangled when printing output. The default is to demangle symbols. The --no-demangle option may be used to turn off demangling. Different compilers have different mangling styles. The optional demangling style argument can be used to choose an appropriate demangling style for your compiler.

ANALYSIS OPTIONS:

-a --no-static
  The `-a' option causes gprof to suppress the printing of statically declared (private) functions. (These are functions whose names are not listed as global, and which are not visible outside the file/function/block where they were defined.) Time spent in these functions, calls to/from them, etc., will all be attributed to the function that was loaded directly before it in the executable file. This option affects both the flat profile and the call graph.

-c --static-call-graph
  The `-c' option causes the call graph of the program to be augmented by a heuristic which examines the text space of the object file and identifies function calls in the binary machine code. Since normal call graph records are only generated when functions are entered, this option identifies children that could have been called, but never were. Calls to functions that were not compiled with profiling enabled are also identified, but only if symbol table entries are present for them. Calls to dynamic library routines are typically not found by this option. Parents or children identified via this heuristic are indicated in the call graph with call counts of `0'.

-D --ignore-non-functions
  The `-D' option causes gprof to ignore symbols which are not known to be functions. This option will give more accurate profile data on systems where it is supported (Solaris and HPUX for example).

-k from/to

The `-k' option allows you to delete from the call graph any arcs from symbols matching symspec from to those matching symspec to.

-l --line
The `-l' option enables line-by-line profiling, which causes histogram hits to be charged to individual source code lines, instead of functions. This feature only works with programs compiled by older versions of the gcc compiler. Newer versions of gcc are designed to work with the gcov tool instead.

If the program was compiled with basic-block counting enabled, this option will also identify how many times each line of code was executed. While line-by-line profiling can help isolate where in a large function a program is spending its time, it also significantly increases the running time of gprof, and magnifies statistical inaccuracies. See Statistical Sampling Error.

-m num --min-count=num
This option affects execution count output only. Symbols that are executed less than num times are suppressed.

-nsymspec --time=symspec
The `-n' option causes gprof, in its call graph analysis, to only propagate times for symbols matching symspec.

-Nsymspec --no-time=symspec
The `-n' option causes gprof, in its call graph analysis, not to propagate times for symbols matching symspec.

-Sfilename --external-symbol-table=filename
The `-S' option causes gprof to read an external symbol table file, such as /proc/kallsyms, rather than read the symbol table from the given object file (the default is a.out). This is useful for profiling kernel modules.

-z --display-unused-functions
If you give the `-z' option, gprof will mention all functions in the flat profile, even those that were never called, and that had no time spent in them. This is useful in conjunction with the `-c' option for discovering which routines were never called.

MISC OPTIONS:

-d[num] --debug[=num]
The `-d num' option specifies debugging options. If num is not specified, enable all debugging. See Debugging gprof.

-h --help
The `-h' option prints command line usage.

-Oname --file-format=name
Selects the format of the profile data files. Recognized formats are `auto' (the default), `bsd', `4.4bsd', `magic', and `prof' (not yet supported).

-s --sum
The `-s' option causes gprof to summarize the information in the profile data files it read in, and write out a profile data file called gmon.sum, which contains all the information from the profile data files that gprof read in. The file gmon.sum may be one of the specified input files; the effect of this is to merge the data in the other input files into gmon.sum.

Eventually you can run gprof again without `-s' to analyze the cumulative data in the file gmon.sum.

-v --version
The `-v' flag causes gprof to print the current version number, and then exit.

Copyright Excerpts taken from
http://sourceware.org/binutils/docs/gprof/
Replicated under the terms of the GNU Free Documentation Licence version 1.3

Grep

Usage: `grep [OPTION]... PATTERN [FILE] ...`

Synopsis Unix-like command

Location Development:bin

Function searches a file for a given pattern

Inputs Regexp selection and interpretation:

-E, --extended-regexp
PATTERN is an extended regular expression
-F, --fixed-strings
PATTERN is a set of newline-separated strings
-G, --basic-regexp
PATTERN is a basic regular expression
-P, --perl-regexp
PATTERN is a Perl regular expression
-e, --regexp=PATTERN
use PATTERN as a regular expression
-f, --file=FILE obtain PATTERN from FILE
-i, --ignore-case

```
                      ignore case distinctions
  -w, --word-regexp
                      force PATTERN to match only whole words
  -x, --line-regexp
                      force PATTERN to match only whole lines
  -z, --null-data
                      a data line ends in 0 byte, not newline

Miscellaneous:

  -s, --no-messages
                      suppress error messages
  -v, --invert-match
                      select non-matching lines
  -V, --version       print version information and exit
      --help          display this help and exit
      --mmap          use memory-mapped input if possible

Output control:

  -m, --max-count=NUM
                      stop after NUM matches
  -b, --byte-offset
                      print the byte offset with output lines
  -n, --line-number
                      print line number with output lines
      --line-buffered
                      flush output on every line
  -H, --with-filename
                      print the filename for each match
  -h, --no-filename
                      suppress the prefixing filename on output
      --label=LABEL
                      print LABEL as filename for standard input
  -o, --only-matching
                      show only the part of a line matching PATTERN
  -q, --quiet, --silent
                      suppress all normal output
      --binary-files=TYPE
                      assume that binary files are TYPE
                      TYPE is 'binary', 'text', or 'without-match'
  -a, --text          equivalent to --binary-files=text
  -I                  equivalent to --binary-files=without-match
  -d, --directories=ACTION
                      how to handle directories
                      ACTION is 'read', 'recurse', or 'skip'
  -D, --devices=ACTION
                      how to handle devices, FIFOs and sockets
                      ACTION is 'read' or 'skip'
  -R, -r, --recursive
                      equivalent to --directories=recurse
```

--include=PATTERN
files that match PATTERN will be examined
--exclude=PATTERN
files that match PATTERN will be skipped.
--exclude-from=FILE
files that match PATTERN in FILE will be skipped.
-L, --files-without-match
only print FILE names containing no match
-l, --files-with-matches
only print FILE names containing matches
-c, --count only print a count of matching lines per FILE
-Z, --null print 0 byte after FILE name

Context control:

-B, --before-context=NUM
print NUM lines of leading context
-A, --after-context=NUM
print NUM lines of trailing context
-C, --context=NUM
print NUM lines of output context
-NUM same as --context=NUM
--color[=WHEN], --colour[=WHEN]
use markers to distinguish the matching string
WHEN may be `always', `never' or `auto'.

`egrep' means `grep -E'. `fgrep' means `grep -F'.
With no FILE, or when FILE is -, read standard input. If less than two FILEs given, assume -h. Exit status is 0 if match, 1 if no match, and 2 if trouble.

Example `grep -i 'hello world' menu.h main.c`

Gzip

Usage: `gzip [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Compress or uncompress FILEs (by default, compress FILES in-place).

Inputs Mandatory arguments to long options are mandatory for short options too.

-c, --stdout write on standard output, keep original files

unchanged
-d, --decompress
decompress
-f, --force force overwrite of output file and compress links
-h, --help give this help
-l, --list list compressed file contents
-L, --license display software license
-n, --no-name do not save or restore the original name and time stamp
-N, --name save or restore the original name and time stamp
-q, --quiet suppress all warnings
-r, --recursive operate recursively on directories
-S, --suffix=SUF use suffix SUF on compressed files
-t, --test test compressed file integrity
-v, --verbose verbose mode
-V, --version display version number
-1, --fast compress faster
-9, --best compress better

With no FILE, or when FILE is -, read standard input.

Head

Usage: `head [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Print the first 10 lines of each FILE to standard output.

Inputs With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.

-c, --bytes=[-]N print the first N bytes of each file; with the leading `-', print all but the last N bytes of each file
-n, --lines=[-]N print the first N lines instead of the first 10; with the leading `-', print all but the last N lines of each file
-q, --quiet, --silent never print headers giving file names
-v, --verbose always print headers giving file names
--help display this help and exit
--version output version information and exit

N may have a multiplier suffix: b 512, k 1024, m 1024*1024.

Hostname

Usage: `hostname [NAME]`
or: `hostname OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print or set the hostname of the current system.

Inputs --help display this help and exit
--version output version information and exit

i386-aros-c++

Usage: `i386-aros-c++ [options] file...`

Notes See c++

i386-aros-g++

Usage: `i386-aros-c++ [options] file...`

Notes See g++

i386-aros-gcc

Usage: `i386-aros-gcc [options] file...`

Notes See gcc

i386-aros-gcc-4.2.4

Usage: `i386-aros-gcc-4.2.4 [options] file...`

Notes See gcc

Kill

Usage: `kill [-s SIGNAL | -SIGNAL] PID...`
`  or:  kill -l [SIGNAL]...`
`  or:  kill -t [SIGNAL]...`

Synopsis Unix-like command

Location Development:bin

Function Send signals to processes, or list signals.

Inputs Mandatory arguments to long options are mandatory for short options too.

-s, --signal=SIGNAL, -SIGNAL
specify the name or number of the signal to be sent
-l, --list list signal names, or convert signal names to/from numbers
-t, --table print a table of signal information
--help display this help and exit
--version output version information and exit

SIGNAL may be a signal name like `HUP', or a signal number like `1', or an exit status of a process terminated by a signal.
PID is an integer; if negative it identifies a process group.

NOTE Kill is not implemented in AROS. Use native BREAK command instead.

Ld

Usage: `ld [options] file...`

Synopsis Unix-like command

Location Development:bin

Function ld is the GNU linker. ld combines a number of object and archive files, relocates their data and ties up symbol references. Usually the last step in compiling a program is to run ld.

Inputs -a KEYWORD Shared library control for HP/UX compatibility
-A ARCH, --architecture ARCH
Set architecture
-b TARGET, --format TARGET

```
                              Specify target for following input files
  -c FILE, --mri-script FILE
                              Read MRI format linker script
  -d, -dc, -dp                Force common symbols to be defined
  -e ADDRESS, --entry ADDRESS
                              Set start address
  -E, --export-dynamic        Export all dynamic symbols
  --no-export-dynamic         Undo the effect of --export-dynamic
  -EB                         Link big-endian objects
  -EL                         Link little-endian objects
  -f SHLIB, --auxiliary SHLIB
                              Auxiliary filter for shared object symbol
                              table
  -F SHLIB, --filter SHLIB
                              Filter for shared object symbol table
  -g                          Ignored
  -G SIZE, --gpsize SIZE
                              Small data size (if no size, same as
                              --shared)
  -h FILENAME, -soname FILENAME
                              Set internal name of shared library
  -I PROGRAM, --dynamic-linker PROGRAM
                              Set PROGRAM as the dynamic linker to
                              use
  -l LIBNAME, --library LIBNAME
                              Search for library LIBNAME
  -L DIRECTORY, --library-path DIRECTORY
                              Add DIRECTORY to library search path
  --sysroot=<DIRECTORY>
                              Override the default sysroot location
  -m EMULATION                Set emulation
  -M, --print-map             Print map file on standard output
  -n, --nmagic                Do not page align data
  -N, --omagic                Do not page align data, do not make text
readonly
  --no-omagic                 Page align data, make text readonly
  -o FILE, --output FILE
                               Set output file name
  -O                          Optimize output file
  -Qy                          Ignored for SVR4 compatibility
  -q, --emit-relocs           Generate relocations in final output
  -r, -i, --relocatable       Generate relocatable output
  -R FILE, --just-symbols FILE
                              Just link symbols (if directory, same as
                              --rpath)
  -s, --strip-all             Strip all symbols
  -S, --strip-debug           Strip debugging symbols
  --strip-discarded           Strip symbols in discarded sections
  --no-strip-discarded        Do not strip symbols in discarded
                              sections
  -t, --trace                 Trace file opens
```

```
  -T FILE, --script FILE      Read linker script
  --default-script FILE, -dT
                              Read default linker script
  -u SYMBOL, --undefined SYMBOL
                              Start with undefined reference to SYMBOL
  --unique [=SECTION]
                              Don't merge input [SECTION | orphan]
                              sections
  -Ur                         Build global constructor/destructor tables
  -v, --version               Print version information
  -V                          Print version and emulation information
  -x, --discard-all           Discard all local symbols
  -X, --discard-locals        Discard temporary local symbols (default)
  --discard-none              Don't discard any local symbols
  -y SYMBOL, --trace-symbol SYMBOL
                              Trace mentions of SYMBOL
  -Y PATH                     Default search path for Solaris
                              compatibility
  -(, --start-group           Start a group
  -), --end-group             End a group
  --accept-unknown-input-arch
                              Accept input files whose architecture
                              cannot be determined
  --no-accept-unknown-input-arch
                              Reject input files whose architecture is
                              unknown
  --as-needed                 Only set DT_NEEDED for following
                              dynamic libs if used
  --no-as-needed              Always set DT_NEEDED for dynamic
                              libraries mentioned on the command line
  -assert KEYWORD             Ignored for SunOS compatibility
  -Bdynamic, -dy, -call_shared
                              Link against shared libraries
  -Bstatic, -dn, -non_shared, -static
                              Do not link against shared libraries
  -Bsymbolic                  Bind global references locally
  -Bsymbolic-functions        Bind global function references locally
  --check-sections            Check section addresses for overlaps
                              (default)
  --no-check-sections         Do not check section addresses for
overlaps
  --copy-dt-needed-entries
                              Copy DT_NEEDED links mentioned inside
                              DSOs that follow
  --no-copy-dt-needed-entries
                              Do not copy DT_NEEDED links mentioned
                              inside DSOs that follow
  --cref                      Output cross reference table
  --defsym SYMBOL=EXPRESSION
                              Define a symbol
  --demangle [=STYLE]
```

```
                        Demangle symbol names [using STYLE]
--embedded-relocs       Generate embedded relocs
--fatal-warnings        Treat warnings as errors
--no-fatal-warnings     Do not treat warnings as errors (default)
-fini SYMBOL            Call SYMBOL at unload-time
--force-exe-suffix      Force generation of file with .exe suffix
--gc-sections           Remove unused sections (on some
                        targets)
--no-gc-sections        Don't remove unused sections (default)
--print-gc-sections     List removed unused sections on stderr
--no-print-gc-sections  Do not list removed unused sections
--hash-size=<NUMBER>
                        Set default hash table size close to
                        <NUMBER>
--help                  Print option help
-init SYMBOL            Call SYMBOL at load-time
-Map FILE               Write a map file
--no-define-common      Do not define Common storage
--no-demangle           Do not demangle symbol names
--no-keep-memor         Use less memory and more disk I/O
--no-undefined          Do not allow unresolved references in
                        object files
--allow-shlib-undefined
                        Allow unresolved references in shared
                        libaries
--no-allow-shlib-undefined
                        Do not allow unresolved references in
                        shared libs
--allow-multiple-definition
                        Allow multiple definitions
--no-undefined-version
                        Disallow undefined version
--default-symver        Create default symbol version
--default-imported-symver
                        Create default symbol version for
                        imported symbols
--no-warn-mismatch      Don't warn about mismatched input files
--no-warn-search-mismatch
                        Don't warn on finding an incompatible
                        library
--no-whole-archive      Turn off --whole-archive
--noinhibit-exec        Create an output file even if errors occur
-nostdlib               Only use library directories specified on
                        the command line
--oformat TARGET        Specify target of output file
-qmagic                 Ignored for Linux compatibility
--reduce-memory-overheads
                        Reduce memory overheads, possibly
                        taking much longer
--relax                 Reduce code size by using target specific
                        optimizations
```

```
  --no-relax                  Do not use relaxation techniques to
                              reduce code size
  --retain-symbols-file FILE
                              Keep only symbols listed in FILE
  -rpath PATH                 Set runtime shared library search path
  -rpath-link PATH            Set link time shared library search path
  -shared, -Bshareable        Create a shared library
  -pie, --pic-executable      Create a position independent executable
  --sort-common [=ascending|descending]
                              Sort common symbols by alignment [in
                              specified order]
  --sort-section name|alignment
                              Sort sections by name or maximum
                              alignment
  --spare-dynamic-tags COUNT
                              How many tags to reserve in .dynamic
                              section
  --split-by-file [=SIZE]     Split output sections every SIZE octets
  --split-by-reloc [=COUNT]
                              Split output sections every COUNT relocs
  --stats                     Print memory usage statistics
  --target-help               Display target specific options
  --task-link SYMBOL          Do task level linking
  --traditional-format        Use same format as native linker
  --section-start SECTION=ADDRESS
                              Set address of named section
  -Tbss ADDRESS               Set address of .bss section
  -Tdata ADDRESS              Set address of .data section
  -Ttext ADDRESS              Set address of .text section
  -Ttext-segment ADDRESS
                              Set address of text segment
  --unresolved-symbols=<method>
                              How to handle unresolved symbols.
                              <method> is:
                              ignore-all, report-all, ignore-in-object-
files,
                              ignore-in-shared-libs
  --verbose [=NUMBER]         Output lots of information during link
  --version-script FILE       Read version information script
  --version-exports-section SYMBOL
                              Take export symbols list from .exports,
                              using SYMBOL as the version.
  --dynamic-list-data         Add data symbols to dynamic list
  --dynamic-list-cpp-new
                              Use C++ operator new/delete dynamic
                              list
  --dynamic-list-cpp-typeinfo
                              Use C++ typeinfo dynamic list
  --dynamic-list FILE         Read dynamic list
  --warn-common               Warn about duplicate common symbols
  --warn-constructors         Warn if global constructors/destructors
```

```
                              are seen
  --warn-multiple-gp          Warn if the multiple GP values are used
  --warn-once                 Warn only once per undefined symbol
  --warn-section-align        Warn if start of section changes due to
                              alignment
  --warn-shared-textrel       Warn if shared object has DT_TEXTREL
  --warn-alternate-em         Warn if an object has alternate ELF
                              machine code
  --warn-unresolved-symbols
                              Report unresolved symbols as warnings
  --error-unresolved-symbols
                              Report unresolved symbols as errors
  --whole-archive             Include all objects from following archives
  --wrap SYMBOL               Use wrapper functions for SYMBOL
  @FILE                       Read options from FILE

ld: supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-
little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex
binary ihex
ld: supported emulations: elf_i386
ld: emulation specific options:
elf_i386:
  --build-id[=STYLE]          Generate build ID note
  --audit=AUDITLIB            Specify a library to use for auditing
  -P AUDITLIB, --depaudit=AUDITLIB
                              Specify a library to use for auditing
                              dependencies
  -Bgroup                     Selects group name lookup rules for DSO
  --disable-new-dtags         Disable new dynamic tags
  --enable-new-dtags          Enable new dynamic tags
  --eh-frame-hdr              Create .eh_frame_hdr section
  --hash-style=STYLE          Set hash style to sysv, gnu or both
  -z combreloc                Merge dynamic relocs into one section
                              and sort
  -z defs                     Report unresolved symbols in object files.
  -z execstack                Mark executable as requiring executable
                              stack
  -z initfirst                Mark DSO to be initialized first at runtime
  -z interpose                Mark object to interpose all DSOs but
                              executable
  -z lazy                     Mark object lazy runtime binding (default)
  -z loadfltr                 Mark object requiring immediate process
  -z muldefs                  Allow multiple definitions
  -z nocombreloc              Don't merge dynamic relocs into one
section
  -z nocopyreloc              Don't create copy relocs
  -z nodefaultlib             Mark object not to use default search
                              paths
  -z nodelete                 Mark DSO non-deletable at runtime
  -z nodlopen                 Mark DSO not available to dlopen
  -z nodump                   Mark DSO not available to dldump
```

-z noexecstack	Mark executable as not requiring executable stack
-z norelro	Don't create RELRO program header
-z now	Mark object non-lazy runtime binding
-z origin	Mark object requiring immediate $ORIGIN processing at runtime
-z relro	Create RELRO program header
-z max-page-size=SIZE	Set maximum page size to SIZE
-z common-page-size=SIZE	Set common page size to SIZE
-z KEYWORD	Ignored for Solaris compatibility

Ld.bfd

Usage: `ld.bfd [options] file...`

Synopsis Unix-like command

Location Development:bin

Function This version of ld uses the general purpose BFD libraries to operate on object files. This allows ld to read, combine, and write object files in many different formats---for example, COFF or "a.out". Different formats may be linked together to produce any available kind of object file.

Inputs See `ld`

@FILE	Read options from FILE

ld.bfd: supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex
ld.bfd: supported emulations: elf_i386

Note A more extensive manual of ld.bfd can be found here: http://www.man-online.org/page/1-ld.bfd/

Link

Usage: `link FILE1 FILE2`
`or: link OPTION`

Synopsis Unix-like command

Location Development:bin

Function Call the link function to create a link named FILE2 to an existing FILE1

Inputs
--help display this help and exit
--version output version information and exit

See also ln

Ln

Usage:
```
ln [OPTION]... [-T] TARGET LINK_NAME   (1st form)
  or:  ln [OPTION]... TARGET                  (2nd form)
  or:  ln [OPTION]... TARGET... DIRECTORY     (3rd form)
  or:  ln [OPTION]... -t DIRECTORY TARGET...  (4th form)
```

Synopsis Unix-like command

Location Development:bin

Function Create hard links by default, symbolic links with --symbolic.
When creating hard links, each TARGET must exist.

In the 1st form, create a link to TARGET with the name LINK_NAME. In the 2nd form, create a link to TARGET in the current directory. In the 3rd and 4th forms, create links to each TARGET in DIRECTORY.

Inputs Mandatory arguments to long options are mandatory for short options too.

--backup[=CONTROL]
make a backup of each existing destination file
-b like --backup but does not accept an argument
-d, -F, --directory
allow the superuser to attempt to hard link directories (note: will probably fail due to system restrictions, even for the superuser)
-f, --force remove existing destination files
-n, --no-dereference
treat destination that is a symlink to a directory as if it were a normal file
-i, --interactive
prompt whether to remove destinations
-s, --symbolic
make symbolic links instead of hard links
-S, --suffix=SUFFIX
override the usual backup suffix

-t, --target-directory=DIRECTORY
specify the DIRECTORY in which to create the links
-T, --no-target-directory
treat LINK_NAME as a normal file
-v, --verbose print name of each linked file
--help display this help and exit
--version output version information and exit

The backup suffix is `~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the --backup option or through the VERSION_CONTROL environment variable. Here are the values:

none, off	never make backups (even if --backup is given)
numbered, t	make numbered backups
existing, nil	numbered if numbered backups exist, simple otherwise
simple, never	always make simple backups

Logname

Usage: `logname OPTIONS`

Synopsis Unix-like command

Location Development:bin

Function Print the current login name.

Inputs --help display this help and exit
--version output version information and exit

Note Not very useful on AROS, since there are no login ability nor any multiuser environment.

Ls

Usage: `ls [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function List information about the FILEs (the current directory by default).

Inputs

Sort entries alphabetically if none of -cftuvSUX nor --sort.
Mandatory arguments to long options are mandatory for short options too.

```
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print octal escapes for nongraphic
                             characters
      --block-size=SIZE
                             use SIZE-byte blocks
  -B, --ignore-backups
                             do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of last
                             modification of file status information)
                             with -l: show ctime and sort by name
                             otherwise: sort by ctime
  -C                         list entries by columns
      --color[=WHEN]
                             control whether color is used to distinguish file
                             types.  WHEN may be `never', `always', or
                             `auto'
  -d, --directory            list directory entries instead of contents,
                             and do not dereference symbolic links
  -D, --dired                generate output designed for Emacs' dired
                             mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append `*'
      --format=WORD
                             across -x, commas -m, horizontal -x, long -l,
                             single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
  -g                         like -l, but do not list owner
      --group-directories-first
                             group directories before files
  -G, --no-group             in a long listing, don't print group names
  -h, --human-readable
                             with -l, print sizes in human readable format
                             (e.g., 1K 234M 2G)
      --si                   likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                             follow symbolic links listed on the command
                             line
      --dereference-command-line-symlink-to-dir
                             follow each command line symbolic link
                             that points to a directory
      --hide=PATTERN
                             do not list implied entries matching shell
                             PATTERN (overridden by -a or -A)
      --indicator-style=WORD
```

```
                       append indicator with style WORD to entry
                       names:
                       none (default), slash (-p),
                       file-type (--file-type), classify (-F)
-i, --inode            print the index number of each file
-I, --ignore=PATTERN
                       do not list implied entries matching shell
                       PATTERN
-k                     like --block-size=1K
-l                     use a long listing format
-L, --dereference
                       when showing file information for a symbolic
                       link, show information for the file the link
                       references rather than for the link itself
-m                     fill width with a comma separated list of
                       entries
-n, --numeric-uid-gid
                       like -l, but list numeric user and group IDs
-N, --literal          print raw entry names (don't treat e.g. control
                       characters specially)
-o                     like -l, but do not list group information
-p, --indicator-style=slash
                       append / indicator to directories
-q, --hide-control-chars
                       print ? instead of non graphic characters
   --show-control-chars
                       show non graphic characters as-is (default
                       unless program is `ls' and output is a
                       terminal)
-Q, --quote-name
                       enclose entry names in double quotes
   --quoting-style=WORD
                       use quoting style WORD for entry names:
                       literal, locale, shell, shell-always, c, escape
-r, --reverse          reverse order while sorting
-R, --recursive        list subdirectories recursively
-s, --size             print the size of each file, in blocks
-S                     sort by file size
   --sort=WORD
                       sort by WORD instead of name: none -U,
                       extension -X, size -S, time -t, version -v
   --time=WORD
                       with -l, show time as WORD instead of
                       modification time: atime -u, access -u, use -u,
                       ctime -c, or status -c; use specified time as
                       sort key if --sort=time
   --time-style=STYLE
                       with -l, show times using style STYLE:
                       full-iso, long-iso, iso, locale, +FORMAT.
                       FORMAT is interpreted like `date'; if FORMAT
                       is FORMAT1<newline>FORMAT2, FORMAT1
```

applies to non-recent files and FORMAT2 to recent files; if STYLE is prefixed with `posix-', STYLE takes effect only outside the POSIX locale

-t sort by modification time

-T, --tabsize=COLS assume tab stops at each COLS instead of 8

-u with -lt: sort by, and show, access time with -l: show access time and sort by name otherwise: sort by access time

-U do not sort; list entries in directory order

-v sort by version

-w, --width=COLS assume screen width instead of current value

-x list entries by lines instead of by columns

-X sort alphabetically by entry extension

-1 list one file per line

--help display this help and exit

--version output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: kB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z, Y. By default, color is not used to distinguish types of files. That is equivalent to using --color=none. Using the --color option without the optional WHEN argument is equivalent to using --color=always. With --color=auto, color codes are output only if standard output is connected to a terminal (tty). The environment variable LS_COLORS can influence the colors, and can be set easily by the dircolors command. Exit status is 0 if OK, 1 if minor problems, 2 if serious trouble.

Note: AROS' shell may not support colors correctly. Color options may not work at all.

M4

Usage: `m4 [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function m4 is a macro processor:it copies its input to the output, expanding macros as it goes.

Inputs Operation modes:

--help display this help and exit

--version output version information and exit

-E, --fatal-warnings

```
                      stop execution after first warning
  -e, --interactive   unbuffer output, ignore interrupts
  -P, --prefix-builtins
                      force a `m4_' prefix to all builtins
  -Q, --quiet, --silent
                      suppress some warnings for builtins

Preprocessor features:
  -D, --define=NAME[=VALUE]
                      enter NAME has having VALUE, or empty
  -I, --include=DIRECTORY
                      append this directory to include path
  -s, --synclines
                      generate `#line NO "FILE"' lines
  -U, --undefine=NAME
                      delete builtin NAME

Limits control:
  -G, --traditional   suppress all GNU extensions
  -H, --hashsize=PRIME
                      set symbol lookup hash table size [509]
  -L, --nesting-limit=NUMBER
                      change artificial nesting limit [1024]

Frozen state files:
  -F, --freeze-state=FILE
                      produce a frozen state on FILE at end
  -R, --reload-state=FILE
                      reload a frozen state from FILE at start

Debugging:
  -d, --debug[=FLAGS]
                      set debug level (no FLAGS implies `aeq')
  -l, --arglength=NUM
                      restrict macro tracing size
  -o, --error-output=FILE
                      redirect debug and trace output
  -t, --trace=NAME
                      trace NAME when it will be defined

FLAGS is any of:
  a   show actual arguments
  c   show before collect, after collect and after call
  e   show expansion
  f   say current input file name
  i   show changes in input files
  l   say current input line number
  p   show results of path searches
  q   quote values as necessary, with a or e flag
  t   trace for all macro calls, not only traceon'ed
  V   shorthand for all of the other flags
```

x add a unique macro call id, useful with c flag

If defined, the environment variable `M4PATH' is a colon-separated list of directories included after any specified by `-I'.
If no FILE or if FILE is `-', standard input is read.
Exit status is 0 for success, 1 for failure, or whatever value was passed to the m4exit macro.

Notes A complete, updated manual for `m4` is available here: http://www.cs.utah.edu/dept/old/texinfo/m4/m4.html

Make

Usage: `make [OPTIONS] [TARGET] ...`

Synopsis Unix-like command

Location Development:bin

Function Automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them.

Inputs Options:

```
  -b, -m                    Ignored for compatibility.
  -B, --always-make
                            Unconditionally make all targets.
  -C DIRECTORY, --directory=DIRECTORY
                            Change to DIRECTORY before doing anything.
  -d                        Print lots of debugging information.
  --debug[=FLAGS]
                            Print various types of debugging information.
  -e, --environment-overrides
                            Environment variables override makefiles.
  -f FILE, --file=FILE, --makefile=FILE
                            Read FILE as a makefile.
  -h, --help                Print this message and exit.
  -i, --ignore-errors
                            Ignore errors from commands.
  -I DIRECTORY, --include-dir=DIRECTORY
                            Search DIRECTORY for included makefiles.
  -j [N], --jobs[=N]
                            Allow N jobs at once; infinite jobs with no arg.
  -k, --keep-going          Keep going when some targets can't be
                            made.
  -l [N], --load-average[=N], --max-load[=N]
                            Don't start multiple jobs unless load is below
                            N.
  -L, --check-symlink-times
                            Use the latest mtime between symlinks and
```

target.
-n, --just-print, --dry-run, --recon
Don't actually run any commands; just print them.
-o FILE, --old-file=FILE, --assume-old=FILE
Consider FILE to be very old and don't remake it.
-p, --print-data-base
Print make's internal database.
-q, --question Run no commands; exit status says if up to date.
-r, --no-builtin-rules
Disable the built-in implicit rules.
-R, --no-builtin-variables
Disable the built-in variable settings.
-s, --silent, --quiet
Don't echo commands.
-S, --no-keep-going, --stop
Turns off -k.
-t, --touch Touch targets instead of remaking them.
-v, --version Print the version number of make and exit.
-w, --print-directory
Print the current directory.
--no-print-directory
Turn off -w, even if it was turned on implicitly.
-W FILE, --what-if=FILE, --new-file=FILE, --assume-new=FILE
Consider FILE to be infinitely new.
--warn-undefined-variables
Warn when an undefined variable is referenced.

Md5sum

Usage: `md5sum [OPTION] [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Print or check MD5 (128-bit) checksums.

Inputs With no FILE, or when FILE is -, read standard input.

-b, --binary read in binary mode
-c, --check read MD5 sums from the FILEs and check them
-t, --text read in text mode (default)

The following two options are useful only when verifying

checksums:

--status	don't output anything, status code shows success
-w, --warn	warn about improperly formatted checksum lines
--help	display this help and exit
--version	output version information and exit

The sums are computed as described in RFC 1321. When checking, the input should be a former output of this program. The default mode is to print a line with checksum, a character indicating type (`*' for binary, ` ' for text), and name for each FILE.

Metaflac

Usage: `metaflac [options] [operations] FLACfile [FLACfile ...]`

Synopsis Unix-like command

Location Development:bin

Function Command-line FLAC metadata editor version 1.2.1

Inputs Use metaflac to list, add, remove, or edit metadata in one or more FLAC files. You may perform one major operation, or many shorthand operations at a time.

Options:

--preserve-modtime
Preserve the original modification time in spite of edits

--with-filename
Prefix each output line with the FLAC file name (the default if more than one FLAC file is specified)

--no-filename
Do not prefix each output line with the FLAC file name (the default if only one FLAC file is specified)

--no-utf8-convert
Do not convert tags from UTF-8 to local charset, or vice versa. This is useful for scripts, and setting tags in situations where the locale is wrong.

--dont-use-padding
By default metaflac tries to use padding where possible to avoid rewriting the entire file if the metadata size changes. Use this option to tell metaflac to not take advantage of padding this way.

Shorthand operations:

--show-md5sum
Show the MD5 signature from the STREAMINFO block.

--show-min-blocksize
Show the minimum block size from the STREAMINFO block.

--show-max-blocksize
Show the maximum block size from the STREAMINFO block.

--show-min-framesize
Show the minimum frame size from the STREAMINFO block.

--show-max-framesize
Show the maximum frame size from the STREAMINFO block.

--show-sample-rate
Show the sample rate from the STREAMINFO block.

--show-channels
Show the number of channels from the STREAMINFO block.

--show-bps
Show the # of bits per sample from the STREAMINFO block.

--show-total-samples
Show the total # of samples from the STREAMINFO block.

--show-vendor-tag

Show the vendor string from the VORBIS_COMMENT block.

--show-tag=NAME
Show all tags where the the field name matches 'NAME'.

--remove-tag=NAME
Remove all tags whose field name is 'NAME'.

--remove-first-tag=NAME
Remove first tag whose field name is 'NAME'.

--remove-all-tags
Remove all tags, leaving only the vendor string.

--set-tag=FIELD
Add a tag. The FIELD must comply with the Vorbis comment spec, of the form "NAME=VALUE". If there is currently no tag block, one will be created.

--set-tag-from-file=FIELD
Like --set-tag, except the VALUE is a filename whose contents will be read verbatim to set the tag value. Unless --no-utf8-convert is specified, the contents will be converted to UTF-8 from the local charset. This can be used to store a cuesheet in a tag (e.g. --set-tag-from-file="CUESHEET=image.cue"). Do not try to store binary data in tag fields! Use APPLICATION blocks for that.

--import-tags-from=FILE
Import tags from a file. Use '-' for stdin. Each line should be of the form NAME=VALUE. Multi-line comments are currently not supported. Specify --remove-all-tags and/or --no-utf8-convert before --import-tags-from if necessary. If FILE is '-' (stdin), only one FLAC file may be specified.

--export-tags-to=FILE Export tags to a file. Use '-' for stdout. Each line will be of the form NAME=VALUE. Specify --no-utf8-convert if necessary.

--import-cuesheet-from=FILE
Import a cuesheet from a file. Use '-' for stdin. Only one FLAC file may be specified. A seekpoint will be added for each index point in the cuesheet to the SEEKTABLE unless --no-cued-seekpoints is specified.

--export-cuesheet-to=FILE
Export CUESHEET block to a cuesheet file, suitable for use by CD authoring software. Use '-' for stdout. Only one FLAC file may be specified on the command line.

--import-picture-from=FILENAME|SPECIFICATION
Import a picture and store it in a PICTURE block. Either a filename for the picture file or a more complete specification form can be used. The SPECIFICATION is a string whose parts are separated by | characters. Some parts may be left empty to invoke default values. FILENAME is just shorthand for "||||FILENAME". The format of SPECIFICATION is: [TYPE]|[MIME-TYPE]|[DESCRIPTION]|[WIDTHxHEIGHTxDEPTH[/COLORS]]|FILE
TYPE is optional; it is a number from one of:
0: Other
1: 32x32 pixels 'file icon' (PNG only)
2: Other file icon

3: Cover (front)
4: Cover (back)
5: Leaflet page
6: Media (e.g. label side of CD)
7: Lead artist/lead performer/soloist
8: Artist/performer
9: Conductor
10: Band/Orchestra
11: Composer
12: Lyricist/text writer
13: Recording Location
14: During recording
15: During performance
16: Movie/video screen capture
17: A bright coloured fish
18: Illustration
19: Band/artist logotype
20: Publisher/Studio logotype
The default is 3 (front cover). There may only be one picture each of type 1 and 2 in a file.
MIME-TYPE is optional; if left blank, it will be detected from the file. For best compatibility with players, use pictures with MIME type image/jpeg or image/png. The MIME type can also be --> to mean that FILE is actually a URL to an image, though this use is discouraged.
DESCRIPTION is optional; the default is an empty string
The next part specfies the resolution and color information. If the MIME-TYPE is image/jpeg, image/png, or image/gif, you can usually leave this empty and they can be detected from the file. Otherwise, you must specify the width in pixels, height in pixels, and color depth in bits-
per-pixel. If the image has indexed colors you should also specify the number of colors used. FILE is the path to the picture file to be imported, or the URL if MIME type is -->
--export-picture-to=FILE Export PICTURE block to a file. Use '-' for stdout.
Only one FLAC file may be specified. The first PICTURE block will be exported unless --export-picture-to is preceded by a --block-number=# option to specify the exact metadata block to extract. Note that the block number is the one shown by --list.

--add-replay-gain
Calculates the title and album gains/peaks of the given FLAC files as if all the files were part of one album, then stores them in the VORBIS_COMMENT block. The tags are the same as those used by vorbisgain. Existing ReplayGain tags will be replaced. If only one FLAC file is given, the album and title gains will be the same. Since this operation requires two passes, it is always executed last, after all other operations have been completed and written to disk. All FLAC files specified must have the same resolution, sample rate, and number of channels. The sample rate must be one of 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz.

--remove-replay-gain

Removes the ReplayGain tags.

--add-seekpoint={#|X|#x|#s}

Add seek points to a SEEKTABLE block
: a specific sample number for a seek point
X : a placeholder point (always goes at the end of the SEEKTABLE)
#x : # evenly spaced seekpoints, the first being at sample 0
#s : a seekpoint every # seconds; # does not have to be a whole number
If no SEEKTABLE block exists, one will be created. If one already exists, points will be added to the existing table, and any duplicates will be turned into placeholder points. You may use many --add-seekpoint options; the resulting SEEKTABLE will be the unique-ified union of all such values. Example: --add-seekpoint=100x --add-seekpoint=3.5s will add 100 evenly spaced seekpoints and a seekpoint every 3.5 seconds.

--add-padding=length

Add a padding block of the given length (in bytes). The overall length of the new block will be 4 + length; the extra 4 bytes is for the metadata block header.

Major operations:

--version

Show the metaflac version number.

--list

List the contents of one or more metadata blocks to stdout. By default, all metadata blocks are listed in text format. Use the following options to change this behavior:

--block-number=#[,#[...]]
An optional comma-separated list of block numbers to display. The first block, the STREAMINFO block, is block 0.

--block-type=type[,type[...]]
--except-block-type=type[,type[...]]
An optional comma-separated list of block types to be included or ignored with this option. Use only one of --block-type or --except-block-type.
The valid block types are: STREAMINFO, PADDING, APPLICATION, SEEKTABLE, VORBIS_COMMENT. You may narrow down the types of APPLICATION blocks displayed as follows:

APPLICATION:abcd
The APPLICATION block(s) whose textual representation of the 4-byte ID is "abcd"
APPLICATION:0xXXXXXXXX
The APPLICATION block(s) whose hexadecimal big-endian representation of the 4-byte ID is "0xXXXXXXXX". For the example "abcd" above the hexadecimal equivalalent is 0x61626364

NOTE: if both --block-number and --[except-]block-type are specified, the result is the logical AND of both arguments.

--application-data-format=hexdump|text
If the application block you are displaying contains binary data but your --data-format=text, you can display a hex dump of the application data contents instead using --application-data-format=hexdump

--remove
Remove one or more metadata blocks from the metadata. Unless --dont-use-padding is specified, the blocks will be replaced with padding. You may not remove the STREAMINFO block.

--block-number=#[,#[...]]
--block-type=type[,type[...]]
--except-block-type=type[,type[...]]
See --list above for usage.

NOTE: if both --block-number and --[except-]block-type are specified, the result is the logical AND of both arguments.

--remove-all
Remove all metadata blocks (except the STREAMINFO block) from the metadata. Unless --dont-use-padding is specified, the blocks will be replaced with padding.

--merge-padding
Merge adjacent PADDING blocks into single blocks.

--sort-padding
Move all PADDING blocks to the end of the metadata and merge them into a single block.

Mkdir

Usage: `mkdir DIRECTORY`

Synopsis	Unix-like command
Location	Development:bin
Function	Creates a directory inside the current path
Inputs	DIRECTORY name of directory to create.
Note	Newly created directories won't have a icon (.info) file associated.

Mkfifo

Usage: `mkfifo [OPTION] NAME...`

Synopsis Unix-like command

Location Development:bin

Function Create named pipes (FIFOs) with the given NAMEs

Inputs Mandatory arguments to long options are mandatory for short options too.

```
  -m, --mode=MODE
                    set file permission bits to MODE, not a=rw - umask
      --help        display this help and exit
      --version     output version information and exit
```

Mknod

Usage: `mknod [OPTION]... NAME TYPE [MAJOR MINOR]`

Synopsis Unix-like command

Location Development:bin

Function Create the special file NAME of the given TYPE.

Inputs Mandatory arguments to long options are mandatory for short options too.

```
  -m, --mode=MODE
                    set file permission bits to MODE, not a=rw - umask
      --help        display this help and exit
      --version     output version information and exit
```

Both MAJOR and MINOR must be specified when TYPE is b, c, or u, and they must be omitted when TYPE is p. If MAJOR or MINOR begins with 0x or 0X, it is interpreted as hexadecimal; otherwise, if it begins with 0, as octal; otherwise, as decimal. TYPE may be:

```
  b             create a block (buffered) special file
  c, u          create a character (unbuffered) special file
  p             create a FIFO
```

Mv

Usage: `mv [OPTION]... [-T] SOURCE DEST`
`or: mv [OPTION]... SOURCE... DIRECTORY`
`or: mv [OPTION]... -t DIRECTORY SOURCE...`

Synopsis Unix-like command

Location Development:bin

Function Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Inputs Mandatory arguments to long options are mandatory for short options too.

```
    --backup[=CONTROL]
                  make a backup of each existing destination file
-b                like --backup but does not accept an argument
-f, --force       do not prompt before overwriting
-i, --interactive
                  prompt before overwrite
    --strip-trailing-slashes
                  remove any trailing slashes from each SOURCE
                  argument
-S, --suffix=SUFFIX
                  override the usual backup suffix
-t, --target-directory=DIRECTORY
                  move all SOURCE arguments into DIRECTORY
-T, --no-target-directory
                  treat DEST as a normal file
-u, --update      move only when the SOURCE file is newer
                  than the destination file or when the
                  destination file is missing
-v, --verbose     explain what is being done
    --help        display this help and exit
    --version     output version information and exit
```

The backup suffix is `~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the --backup option or through the VERSION_CONTROL environment variable. Here are the values:

```
none, off         never make backups (even if --backup is given)
numbered, t       make numbered backups
existing, nil     numbered if numbered backups exist, simple
                  otherwise
simple, never     always make simple backups
```

Nl

Usage: `nl [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Write each FILE to standard output, with line numbers added.
With no FILE, or when FILE is -, read standard input.

Inputs Mandatory arguments to long options are mandatory for short options too.

```
  -b, --body-numbering=STYLE
                      use STYLE for numbering body lines
  -d, --section-delimiter=CC
                      use CC for separating logical pages
  -f, --footer-numbering=STYLE
                      use STYLE for numbering footer lines
  -h, --header-numbering=STYLE
                      use STYLE for numbering header lines
  -i, --page-increment=NUMBER
                      line number increment at each line
  -l, --join-blank-lines=NUMBER
                      group of NUMBER empty lines counted as one
  -n, --number-format=FORMAT
                      insert line numbers according to FORMAT
  -p, --no-renumber
                      do not reset line numbers at logical pages
  -s, --number-separator=STRING
                      add STRING after (possible) line number
  -v, --first-page=NUMBER
                      first line number on each logical page
  -w, --number-width=NUMBER
                      use NUMBER columns for line numbers
      --help          display this help and exit
      --version       output version information and exit
```

By default, selects -v1 -i1 -l1 -sTAB -w6 -nrn -hn -bt -fn. CC are two delimiter characters for separating logical pages, a missing second character implies :. Type \\ for \. STYLE is one of:

```
  a                   number all lines
  t                   number only nonempty lines
  n                   number no lines
  pBRE                number only lines that contain a match for the
                      basic regular expression, BRE
```

FORMAT is one of:

```
  ln                  left justified, no leading zeros
  rn                  right justified, no leading zeros
  rz                  right justified, leading zeros
```

Nm

Usage: `nm [option(s)] [file(s)]`

Synopsis Unix-like command

Location Development:bin

Function List symbols in [file(s)] (a.out by default).

Inputs The options are:

```
 -a, --debug-syms
                    Display debugger-only symbols
 -A, --print-file-name
                    Print name of the input file before every symbol
 -B                 Same as --format=bsd
 -C, --demangle[=STYLE]
                    Decode low-level symbol names into user-level
                    names The STYLE, if specified, can be `auto' (the
                    default), `gnu', `lucid', `arm', `hp', `edg', `gnu-
                    v3', `java' or `gnat'
    --no-demangle
                    Do not demangle low-level symbol names
 -D, --dynamic
                    Display dynamic symbols instead of normal
                    symbols
    --defined-only
                    Display only defined symbols
 -e                 (ignored)
 -f, --format=FORMAT
                    Use the output format FORMAT.  FORMAT can be
                    `bsd', `sysv' or `posix'.  The default is `bsd'
 -g, --extern-only
                    Display only external symbols
 -l, --line-numbers
                    Use debugging information to find a filename and
                    line number for each symbol
 -n, --numeric-sort
                    Sort symbols numerically by address
 -o                 Same as -A
 -p, --no-sort      Do not sort the symbols
 -P, --portability
                    Same as --format=posix
 -r, --reverse-sort
                    Reverse the sense of the sort
 -S, --print-size   Print size of defined symbols
 -s, --print-armap
                    Include index for symbols from archive members
    --size-sort     Sort symbols by size
    --special-syms
```

Include special symbols in the output
--synthetic Display synthetic symbols as well
-t, --radix=RADIX
Use RADIX for printing symbol values
--target=BFDNAME
Specify the target object format as BFDNAME
-u, --undefined-only
Display only undefined symbols
-X 32_64 (ignored)
@FILE Read options from FILE
-h, --help Display this information
-V, --version Display this program's version number

Notes Supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex.

Nohup

Usage: `nohup COMMAND [ARG]...`
`or: nohup OPTION`

Synopsis Unix-like command

Location Development:bin

Function Run COMMAND, ignoring hangup signals.

Inputs Accepted OPTIONS are:

--help display this help and exit
--version output version information and exit

Objcopy

Usage: `objcopy [option(s)] in-file [out-file]`

Synopsis Unix-like command

Location Development:bin

Function Copies a binary file, possibly transforming it in the process

Inputs The options are:
-I --input-target <bfdname>
Assume input file is in format <bfdname>

```
-O --output-target <bfdname>
                Create an output file in format <bfdname>
-B --binary-architecture <arch>
                Set output arch, when input is arch-less
-F --target <bfdname>
                Set both input and output format to <bfdname>
  --debugging
                Convert debugging information, if possible
-p --preserve-dates
                Copy modified/access timestamps to the output
-j --only-section <name>
                Only copy section <name> into the output
  --add-gnu-debuglink=<file>
                Add section .gnu_debuglink linking to <file>
-R --remove-section <name>
                Remove section <name> from the output
-S --strip-all
                Remove all symbol and relocation information
-g --strip-debug
                Remove all debugging symbols & sections
  --strip-unneeded
                Remove all symbols not needed by relocations
-N --strip-symbol <name>
                Do not copy symbol <name>
  --strip-unneeded-symbol <name>
                Do not copy symbol <name> unless needed by
                relocations
  --only-keep-debug
                Strip everything but the debug information
  --extract-symbol
                Remove section contents but keep symbols
-K --keep-symbol <name>
                Do not strip symbol <name>
  --keep-file-symbols
                Do not strip file symbol(s)
  --localize-hidden
                Turn all ELF hidden symbols into locals
-L --localize-symbol <name>
                Force symbol <name> to be marked as a local
  --globalize-symbol <name>
                Force symbol <name> to be marked as a global
-G --keep-global-symbol <name>
                Localize all symbols except <name>
-W --weaken-symbol <name>
                Force symbol <name> to be marked as a weak
  --weaken
                Force all global symbols to be marked as weak
-w --wildcard
                Permit wildcard in symbol comparison
-x --discard-all
                Remove all non-global symbols
```

```
-X --discard-locals
                Remove any compiler-generated symbols
-i --interleave [<number>]
                Only copy N out of every <number> bytes
  --interleave-width <number>
                Set N for --interleave
-b --byte <num>
                Select byte <num> in every interleaved block
  --gap-fill <val>
                Fill gaps between sections with <val>
  --pad-to <addr>
                Pad the last section up to address <addr>
  --set-start <addr>
                Set the start address to <addr>
 {--change-start|--adjust-start} <incr>
                Add <incr> to the start address
 {--change-addresses|--adjust-vma} <incr>
                Add <incr> to LMA, VMA and start addresses
 {--change-section-address|--adjust-section-vma}
 <name>{=|+|-}<val>
                Change LMA and VMA of section <name> by <val>
  --change-section-lma <name>{=|+|-}<val>
                Change the LMA of section <name> by <val>
  --change-section-vma <name>{=|+|-}<val>
                Change the VMA of section <name> by <val>
 {--[no-]change-warnings|--[no-]adjust-warnings}
                Warn if a named section does not exist
  --set-section-flags <name>=<flags>
                Set section <name>'s properties to <flags>
  --add-section <name>=<file>
                Add section <name> found in <file> to output
  --rename-section <old>=<new>[,<flags>]
                Rename section <old> to <new>
  --long-section-names {enable|disable|keep}
                 Handle long section names in Coff objects.
  --change-leading-char
                Force output format's leading character style
  --remove-leading-char
                Remove leading character from global symbols
  --reverse-bytes=<num>
                Reverse <num> bytes at a time, in output sections
                with content
  --redefine-sym <old>=<new>
                Redefine symbol name <old> to <new>
  --redefine-syms <file>
                --redefine-sym for all symbol pairs listed in <file>
  --srec-len <number>
                Restrict the length of generated Srecords
  --srec-forceS3
                Restrict the type of generated Srecords to S3
  --strip-symbols <file>
```

```
                -N for all symbols listed in <file>
  --strip-unneeded-symbols <file>
                 --strip-unneeded-symbol for all symbols listed
                 in <file>
  --keep-symbols <file>
                -K for all symbols listed in <file>
  --localize-symbols <file>
                -L for all symbols listed in <file>
  --globalize-symbols <file>
                --globalize-symbol for all in <file>
  --keep-global-symbols <file>
                -G for all symbols listed in <file>
  --weaken-symbols <file>
                -W for all symbols listed in <file>
  --alt-machine-code <index>
                Use the target's <index>'th alternative machine
  --writable-text
                Mark the output text as writable
  --readonly-text
                Make the output text write protected
  --pure
                Mark the output file as demand paged
  --impure
                Mark the output file as impure
  --prefix-symbols <prefix>
                Add <prefix> to start of every symbol name
  --prefix-sections <prefix>
                Add <prefix> to start of every section name
  --prefix-alloc-sections <prefix>
                Add <prefix> to start of every allocatable
                section name
  --file-alignment <num>
                Set PE file alignment to <num>
  --heap <reserve>[,<commit>]
                Set PE reserve/commit heap to <reserve>/<commit>
  --image-base <address>
                Set PE image base to <address>
  --section-alignment <num>
                Set PE section alignment to <num>
  --stack <reserve>[,<commit>]
                Set PE reserve/commit stack to <reserve>/<commit>
  --subsystem <name>[:<version>]
                Set PE subsystem to <name> [& <version>]
  --compress-debug-sections
                Compress DWARF debug sections using zlib
  --decompress-debug-sections
                Decompress DWARF debug sections using zlib
-v --verbose
                List all object files modified
@<file>      Read options from <file>
-V --version
```

Display this program's version number
-h --help Display this output
--info List object formats & architectures supported

objcopy: supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Objdump

Usage: `objdump <option(s)> <file(s)>`

Synopsis Unix-like command

Location Development:bin

Function Display information from object <file(s)>.

Inputs At least one of the following switches must be given:
-a, --archive-headers
Display archive header information
-f, --file-headers Display the contents of the overall file header
-p, --private-headers
Display object format specific file header contents
-h, --[section-]headers
Display the contents of the section headers
-x, --all-headers Display the contents of all headers
-d, --disassemble Display assembler contents of executable sections
-D, --disassemble-all
Display assembler contents of all sections
-S, --source Intermix source code with disassembly
-s, --full-contents Display the full contents of all sections requested
-g, --debugging Display debug information in object file
-e, --debugging-tags
Display debug information using ctags style
-G, --stabs Display (in raw form) any STABS info in the file
-W[lLiaprmfFsoRt] or – dwarf[=rawline,=decodedline,=info,
=abbrev,=pubnames,=aranges,=macro,=frames,
=frames-interp,=str,=loc,=Ranges,=pubtypes,
=gdb_index,=trace_info,=trace_abbrev,=trace_aranges]
Display DWARF info in the file
-t, --syms Display the contents of the symbol table(s)
-T, --dynamic-syms
Display the contents of the dynamic symbol table

```
-r, --reloc             Display the relocation entries in the file
-R, --dynamic-reloc
                        Display the dynamic relocation entries in the
                        file
@<file>                 Read options from <file>
-v, --version           Display this program's version number
-i, --info              List object formats and architectures
                        supported
-H, --help              Display this information

The following switches are optional:
-b, --target=BFDNAME
                        Specify the target object format as BFDNAME
-m, --architecture=MACHINE
                        Specify the target architecture as MACHINE
-j, --section=NAME
                        Only display information for section NAME
-M, --disassembler-options=OPT
                        Pass text OPT on to the disassembler
-EB --endian=big
                        Assume big endian format when
                        disassembling
-EL --endian=little
                        Assume little endian format when
                        disassembling
    --file-start-context
                        Include context from start of file (with -S)
-I, --include=DIR
                        Add DIR to search list for source files
-l, --line-numbers
                        Include line numbers and filenames in output
-F, --file-offsets      Include file offsets when displaying
                        information
-C, --demangle[=STYLE]
                        Decode mangled/processed symbol names
                        The STYLE, if specified, can be `auto', `gnu',
                        `lucid', `arm', `hp', `edg', `gnu-v3', `java'
                        or `gnat'
-w, --wide              Format output for more than 80 columns
-z, --disassemble-zeroes
                        Do not skip blocks of zeroes when
                        disassembling
    --start-address=ADDR
                        Only process data whose address is >= ADDR
    --stop-address=ADDR
                        Only process data whose address is <= ADDR
    --prefix-addresses
                        Print complete address alongside disassembly
    --[no-]show-raw-insn
                        Display hex alongside symbolic disassembly
    --insn-width=WIDTH
```

Display WIDTH bytes on a signle line for -d
--adjust-vma=OFFSET
Add OFFSET to all displayed section addresses
--special-syms
Include special symbols in symbol dumps
--prefix=PREFIX
Add PREFIX to absolute paths for -S
--prefix-strip=LEVEL
Strip initial directory names for -S

The following i386/x86-64 specific disassembler options are supported for use
with the -M switch (multiple options should be separated by commas):

x86-64	Disassemble in 64bit mode
i386	Disassemble in 32bit mode
i8086	Disassemble in 16bit mode
att	Display instruction in AT&T syntax
intel	Display instruction in Intel syntax
att-mnemonic	Display instruction in AT&T mnemonic
intel-mnemonic	Display instruction in Intel mnemonic
addr64	Assume 64bit address size
addr32	Assume 32bit address size
addr16	Assume 16bit address size
data32	Assume 32bit data size
data16	Assume 16bit data size
suffix	Always display instruction suffix in AT&T syntax

Notes supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

supported architectures: i386 i386:x86-64 i8086 i386:intel i386:x86-64:intel

Od

Usage:
```
od [OPTION]... [FILE]...
  or:  od [-abcdfilosx]... [FILE] [[+]OFFSET[.][b]]
  or:  od --traditional [OPTION]... [FILE] [[+]OFFSET[.][b]
       [+][LABEL][.][b]]
```

Synopsis Unix-like command

Location Development:bin

Function Write an unambiguous representation, octal bytes by default,

of FILE to standard output. With more than one FILE argument, concatenate them in the listed order to form the input. With no FILE, or when FILE is -, read standard input.

Inputs

All arguments to long options are mandatory for short options.

-A, --address-radix=RADIX
decide how file offsets are printed
-j, --skip-bytes=BYTES
skip BYTES input bytes first
-N, --read-bytes=BYTES
limit dump to BYTES input bytes
-S, --strings[=BYTES]
output strings of at least BYTES graphic chars
-t, --format=TYPE
select output format or formats
-v, --output-duplicates
do not use * to mark line suppression
-w, --width[=BYTES]
output BYTES bytes per output line
--traditional
accept arguments in traditional form
--help display this help and exit
--version output version information and exit

Traditional format specifications may be intermixed; they accumulate:

-a same as -t a
select named characters, ignoring high-order bit
-b same as -t o1
select octal bytes
-c same as -t c
select ASCII characters or backslash escapes
-d same as -t u2
select unsigned decimal 2-byte units
-f same as -t fF
select floats
-i same as -t dI
select decimal ints
-l same as -t dL
select decimal longs
-o same as -t o2
select octal 2-byte units
-s same as -t d2
select decimal 2-byte units
-x same as -t x2
select hexadecimal 2-byte units

If first and second call formats both apply, the second format is assumed if the last operand begins with + or (if there are 2

operands) a digit. An OFFSET operand means -j OFFSET. LABEL is the pseudo-address at first byte printed, incremented when dump is progressing. For OFFSET and LABEL, a 0x or 0X prefix indicates hexadecimal, suffixes may be . for octal and b for multiply by 512.

TYPE is made up of one or more of these specifications:

a	named character, ignoring high-order bit
c	ASCII character or backslash escape
d[SIZE]	signed decimal, SIZE bytes per integer
f[SIZE]	floating point, SIZE bytes per integer
o[SIZE]	octal, SIZE bytes per integer
u[SIZE]	unsigned decimal, SIZE bytes per integer
x[SIZE]	hexadecimal, SIZE bytes per integer

SIZE is a number. For TYPE in doux, SIZE may also be C for sizeof(char), S for sizeof(short), I for sizeof(int) or L for sizeof(long). If TYPE is f, SIZE may also be F for sizeof(float), D for sizeof(double) or L for sizeof(long double).

RADIX is d for decimal, o for octal, x for hexadecimal or n for none. BYTES is hexadecimal with 0x or 0X prefix, it is multiplied by 512 with b suffix, by 1024 with k and by 1048576 with m. Adding a z suffix to any type adds a display of printable characters to the end of each line of output. --string without a number implies 3. --width without a number implies 32. By default, od uses -A o -t d2 -w16.

OpenSSL

Usage:
```
openssl command [ command_opts ] [ command_args ]
  or:  openssl [ list-standard-commands | list-message-digest-commands
       | list-cipher-commands | list-cipher-algorithms |
       list-message-digest-algorithms | list-public-key-algorithms]
  or:  openssl no- XXX [ arbitrary options ]
```

Synopsis Unix-like command

Location Development:bin

Function OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for

o Creation and management of private keys, public keys and

parameters
- o Public key cryptographic operations
- o Creation of X.509 certificates, CSRs and CRLs
- o Calculation of Message Digests
- o Encryption and Decryption with Ciphers
- o SSL/TLS Client and Server Tests
- o Handling of S/MIME signed or encrypted mail
- o Time Stamp requests, generation and verification

Inputs

The openssl program provides a rich variety of commands (command in the SYNOPSIS above), each of which often has a wealth of options and arguments (command_opts and command_args in the SYNOPSIS).

The pseudo-commands list-standard-commands, list-message-digest-commands, and list-cipher-commands output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present openssl utility.

The pseudo-commands list-cipher-algorithms and list-message-digest-algorithms list all cipher and message digest names, one entry per line. Aliases are listed as:

from => to

The pseudo-command list-public-key-algorithms lists all supported public key algorithms.

The pseudo-command no- XXX tests whether a command of the specified name is available. If no command named XXX exists, it returns 0 (success) and prints no- XXX ; otherwise it returns 1 and prints XXX . In both cases, the output goes to stdout and nothing is printed to stderr. Additional command line arguments are always ignored. Since for each cipher there is a command of the same name, this provides an easy way for shell scripts to test for the availability of ciphers in the openssl program. (no- XXX is not able to detect pseudo-commands such as quit, list-...-commands, or no-XXX itself.)

STANDARD COMMANDS

asn1parse	Parse an ASN .1 sequence.
ca	Certificate Authority (CA) Management.
ciphers	Cipher Suite Description Determination.
cms	CMS (Cryptographic Message Syntax) utility
crl	Certificate Revocation List (CRL) Management.
crl2pkcs7	CRL to PKCS#7 Conversion.
dgst	Message Digest Calculation.
dh	Diffie-Hellman Parameter Management. Obsoleted by dhparam.

dhparam	Generation and Management of Diffie-Hellman Parameters. Superseded by genpkey and pkeyparam
dsa	DSA Data Management.
dsaparam	DSA Parameter Generation and Management. Superseded by genpkey and pkeyparam
ec	EC (Elliptic curve) key processing
ecparam	EC parameter manipulation and generation
enc	Encoding with Ciphers.
engine	Engine (loadble module) information and manipulation.
errstr	Error Number to Error String Conversion.
gendh	Generation of Diffie-Hellman Parameters. Obsoleted by dhparam.
gendsa	Generation of DSA Private Key from Parameters. Superseded by genpkey and pkey
genpkey	Generation of Private Key or Parameters.
genrsa	Generation of RSA Private Key. Superceded by genpkey.
nseq	Create or examine a netscape certificate sequence
ocsp	Online Certificate Status Protocol utility.
passwd	Generation of hashed passwords.
pkcs12	PKCS#12 Data Management.
pkcs7	PKCS#7 Data Management.
pkey	Public and private key management.
pkeyparam	Public key algorithm parameter management.
pkeyutl	Public key algorithm cryptographic operation utility.
rand	Generate pseudo-random bytes.
req	PKCS#10 X.509 Certificate Signing Request (CSR) Management.
rsa	RSA key management.
rsautl	RSA utility for signing, verification, encryption, and decryption. Superseded by pkeyutl
s_client	This implements a generic SSL/TLS client which can establish a transparent connection to a remote server speaking SSL/TLS . It's intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL ssl library.
s_server	This implements a generic SSL/TLS server which accepts connections from remote clients speaking SSL/TLS . It's intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL ssl library. It provides both an own command line oriented protocol for testing SSL functions and a simple HTTP response facility to emulate an SSL/TLS-aware webserver.
s_time	SSL Connection Timer.
sess_id	SSL Session Data Management.
smime	S/MIME mail processing.
speed	Algorithm Speed Measurement.

spkac SPKAC printing and generating utility
ts Time Stamping Authority tool (client/server)
verify X.509 Certificate Verification.
version OpenSSL Version Information.
x509 X.509 Certificate Data Management.

MESSAGE DIGEST COMMANDS

md2 MD2 Digest
md5 MD5 Digest
mdc2 MDC2 Digest
rmd160 RMD-160 Digest
sha SHA Digest
sha1 SHA-1 Digest
sha224 SHA-224 Digest
sha256 SHA-256 Digest
sha384 SHA-384 Digest
sha512 SHA-512 Digest

ENCODING AND CIPHER COMMANDS

base64 Base64 Encoding
bf bf-cbc bf-cfb bf-ecb bf-ofb
Blowfish Cipher
cast cast-cbc
CAST Cipher
cast5-cbc cast5-cfb cast5-ecb cast5-ofb
CAST5 Cipher
des des-cbc des-cfb des-ecb des-ede des-ede-cbc des-ede-cfb des-ede-ofb des-ofb
DES Cipher
des3 desx des-ede3 des-ede3-cbc des-ede3-cfb des-ede3-ofb
Triple-DES Cipher
idea idea-cbc idea-cfb idea-ecb idea-ofb
IDEA Cipher
rc2 rc2-cbc rc2-cfb rc2-ecb rc2-ofb
RC2 Cipher
rc4 RC4 Cipher
rc5 rc5-cbc rc5-cfb rc5-ecb rc5-ofb
RC5 Cipher

Pass Phrase Arguments

Several commands accept password arguments, typically using -passin and -passout for input and output passwords respectively. These allow the password to be obtained from a variety of sources. Both of these options take a single argument whose format is described below. If no password argument is given and a password is required then the user is prompted to enter one: this will typically be read from the current terminal with echoing turned off.

pass:password
the actual password is password. Since the password is visible to utilities (like 'ps' under Unix) this form should only be used where security is not important.

env:var obtain the password from the environment variable var. Since the environment of other processes is visible on certain platforms (e.g. ps under certain Unix OSes) this option should be used with caution.

file:pathname
the first line of pathname is the password. If the same pathname argument is supplied to -passin and -passout arguments then the first line will be used for the input password and the next line for the output password. pathname need not refer to a regular

file: it could for example refer to a device or named pipe.

fd:number read the password from the file descriptor number. This can be used to send the data via a pipe for example.

Stdin read the password from standard input.

Notes OpenSSL manuals and informations here:
http://linux.die.net/man/1/openssl
http://cims.nyu.edu/systems/software/desc/OpenSSL.html

Paste

Usage: `paste [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Write lines consisting of the sequentially corresponding lines from each FILE, separated by TABs, to standard output.
With no FILE, or when FILE is -, read standard input.

Inputs Mandatory arguments to long options are mandatory for short options too.

-d, --delimiters=LIST
reuse characters from LIST instead of TABs
-s, --serial paste one file at a time instead of in parallel
--help display this help and exit
--version output version information and exit

Patch

Usage: `patch [OPTION]... [ORIGFILE [PATCHFILE]]`

Synopsis Unix-like command

Location Development:bin

Function apply a diff file to an original one.

Inputs Input options:

```
-p NUM  --strip=NUM
                Strip NUM leading components from file names.
-F LINES  --fuzz LINES
                Set the fuzz factor to LINES for inexact matching.
-l  --ignore-whitespace
                Ignore white space changes between patch and
                input.
-c  --context   Interpret the patch as a context difference.
-e  --ed        Interpret the patch as an ed script.
-n  --normal    Interpret the patch as a normal difference.
-u  --unified   Interpret the patch as a unified difference.
-N  --forward   Ignore patches that appear to be reversed or
                already applied.
-R  --reverse   Assume patches were created with old and new
                files swapped.
-i PATCHFILE  --input=PATCHFILE
                Read patch from PATCHFILE instead of stdin.
```

Output options:

```
-o FILE  --output=FILE
                Output patched files to FILE.
-r FILE  --reject-file=FILE
                Output rejects to FILE.
-D NAME  --ifdef=NAME
                Make merged if-then-else output using NAME.
-E  --remove-empty-files
                Remove output files that are empty after
                patching.
-Z  --set-utc   Set times of patched files, assuming diff uses UTC
                (GMT).
-T  --set-time  Likewise, assuming local time.
--quoting-style=WORD
                output file names using quoting style WORD.
                Valid WORDs are: literal, shell, shell-always, c,
                escape. Default is taken from QUOTING_STYLE
```

env variable, or 'shell' if unset.

Backup and version control options:

-b --backup Back up the original contents of each file.
--backup-if-mismatch
Back up if the patch does not match exactly.
--no-backup-if-mismatch
Back up mismatches only if otherwise requested.
-V STYLE --version-control=STYLE
Use STYLE version control. STYLE is either 'simple', 'numbered', or 'existing'.
-B PREFIX --prefix=PREFIX
Prepend PREFIX to backup file names.
-Y PREFIX --basename-prefix=PREFIX
Prepend PREFIX to backup file basenames.
-z SUFFIX --suffix=SUFFIX
Append SUFFIX to backup file names.
-g NUM --get=NUM
Get files from RCS etc. if positive; ask if negative.

Miscellaneous options:

-t --batch Ask no questions; skip bad-Prereq patches; assume reversed.
-f --force Like -t, but ignore bad-Prereq patches, and assume unreversed.
-s --quiet --silent
Work silently unless an error occurs.
--verbose Output extra information about the work being done.
--dry-run Do not actually change any files; just print what would happen.
--posix Conform to the POSIX standard.
-d DIR --directory=DIR
Change the working directory to DIR first.
--binary Read and write data in binary mode (no effect on this platform).
-v --version Output version info.
--help Output this help.

Pathchk

Usage: `pathchk [OPTION]... NAME...`

Synopsis Unix-like command

Location Development:bin

Function Diagnose unportable constructs in NAME.

Inputs

-p	check for most POSIX systems
-P	check for empty names and leading "-"
--portability	check for all POSIX systems (equivalent to -p -P)
--help	display this help and exit
--version	output version information and exit

Pcregrep

Usage: `pcregrep [OPTION]... [PATTERN] [FILE1 FILE2 ...]`

Synopsis Unix-like command

Location Development:bin

Function Search for PATTERN in each FILE or standard input.
PATTERN must be present if -f is not used.

Inputs Options:

--help	display this help and exit
-c, --count	print only a count of matching lines per FILE
-h, --no-filename	suppress the prefixing filename on output
-i, --ignore-case	ignore case distinctions
-l, --files-with-matches	print only FILE names containing matches
-n, --line-number	print line number with output lines
-r, --recursive	recursively scan sub-directories
-s, --no-messages	suppress error messages
-u, --utf-8	use UTF-8 mode
-V, --version	print version information and exit
-v, --invert-match	select non-matching lines
-x, --line-regex	force PATTERN to match only whole lines
-x, --line-regexp	force PATTERN to match only whole lines
-f<filename> or --file=<filename>	Read patterns from <filename> instead of using a command line option. Trailing white space is removed; blanks lines are ignored. There is a maximum of 100 patterns.

With no FILE, read standard input. If fewer than two FILEs given,

assume -h. Exit status is 0 if any matches, 1 if no matches, and 2 if trouble.

Example `pcregrep -i 'hello.*world' menu.h main.c`

Pcretest

Usage: `pcretest [-d] [-i] [-o <n>] [-p] [-s] [-t] [<input> [<output>]]`

Synopsis Unix-like command

Location Development:bin

Function *** REPLACE ME ***

Inputs Usage:
-C show PCRE compile-time options and exit
-d debug: show compiled code; implies -i
-i show information about compiled pattern
-o <n> set size of offsets vector to <n>
-p use POSIX interface
-s output store information
-t time compilation and execution

Perl

Usage: `perl [switches] [--] [programfile] [arguments]`

Synopsis Unix-like command

Location Development:bin

Function Executes a program written in the perl language

Inputs -0[octal] specify record separator (\0, if no argument)
-a autosplit mode with -n or -p (splits $_ into @F)
-C enable native wide character system interfaces
-c check syntax only (runs BEGIN and CHECK blocks)
-d[:debugger]
run program under debugger
-D[number/list]
set debugging flags (argument is a bit mask or alphabets)

-e 'command'
one line of program (several -e's allowed, omit programfile)
-F/pattern/ split() pattern for -a switch (//'s are optional)
-i[extension]
edit <> files in place (makes backup if extension supplied)
-Idirectory specify @INC/#include directory (several -I's allowed)
-l[octal] enable line ending processing, specifies line terminator
-[mM][-]module
execute `use/no module...' before executing program
-n assume 'while (<>) { ... }' loop around program
-p assume loop like -n but print line also, like sed
-P run program through C preprocessor before compilation
-s enable rudimentary parsing for switches after programfile
-S look for programfile using PATH environment variable
-T enable tainting checks
-u dump core after parsing program
-U allow unsafe operations
-v print version, subversion (includes VERY IMPORTANT perl info)
-V[:variable]
print configuration summary (or a single Config.pm variable)
-w enable many useful warnings (RECOMMENDED)
-W enable all warnings
-X disable all warnings
-x[directory]
strip off text before #!perl line and perhaps cd to directory

Pgawk / Pgawk-

Usage: `pgawk [POSIX or GNU style options] -f progfile [--] file ...`
or: `pgawk [POSIX or GNU style options] [--] 'program' file ...`

Notes See: awk / gawk

PngToPnm

Usage: `pngtopnm [-verbose] [-alpha | -mix] [-background color] [-gamma value] [-text file] [-time] [pngfile]`

Synopsis Unix-like command

Location Development:bin

Function Reads a Portable Network Graphics as input. Produces a portable anymap as output. The type of the output file depends on the input file - if it's black & white, a pbm file is written, else if it's grayscale a pgm file, else a ppm file.

Inputs

-verbose
Display the format of the input file and the type of the output file. If the chunks are part of the png-file, the alpha, transparency and gamma-values will be indicated.

-alpha
Output the alpha channel or transparency mask of the image. The result is either a pbm file or pgm file, depending on whether different levels of transparency appear.

-mix
Compose the image with the transparency or alpha mask against a the background. When a background chunk is available that color is taken, else black will do.

-background color
If no background color chunck is present in the png-file, or when another color is required this parameter can be used to set the background color of images. This is especially useful for alpha-channel images or those with transparency chunks. The format, to specify the color in, is either (in the case of orange) "1.0,0.5,0.0", where the values are floats between zero and one, or with the syntax "#RGB", "#RRGGBB" or "#RRRRGGGGBBBB" where R, G and B are hexa-decimal numbers.

-gamma value
Converts the image to a new display-gamma value. When a gAMA chunk is present in the png-file, the image-gamma value will be used. When not, the image-gamma is considered to be 1.0. Based on the image-gamma and the display-gamma given with this option the colors written to the pnm-file will be adjusted.
Because the gamma's of uncompensated monitors are around 2.6, which results in an image-gamma of 0.45, some typical situations are: when the image-gamma is 0.45 (use -verbose to check) and the picture is too light, your system is gamma-corrected, so convert with "-gamma 1.0". When no gAMA chunk is present or the image-gamma is 1.0, use 2.2 to make the picture lighter and 0.45 to make the picture darker.

-text file	Writes the tEXt and zTXt chunks to a file, in a format as described in the pnmtopng man-page. These chunks contain text comments or annotations.
-time	Prints the tIME chunk to stderr.

Note All flags can be abbreviated to their shortest unique prefix.

Man page at:
http://linux.about.com/library/cmd/blcmdl1_pngtopnm.htm

PpmToIlbm

Usage: `ppmtoilbm [-maxplanes|-mp N] [-fixplanes|-fp N] [-ham6|-ham8] [-dcbits|-dcplanesrgb] [-normal|-hamif|-hamforce|-24if|-24force| -dcif|-dcforce|-cmaponly] [-ecs|-aga] [-compress|-nocompress] [-cmethod type] [-mapppmfile] [-savemem] [ppmfile]`

Synopsis Unix-like command

Location Development:bin

Function Reads a portable pixmap as input. Produces an ILBM file as output.

Inputs Options marked with (*) can be prefixed with a "no", e.g. "-nohamif". All options can be abbreviated to their shortest unique prefix.

-maxplanes | -mp n
(default 5, minimum 1, maximum 16) Maximum planes to write in a normal ILBM. If the pixmap does not fit into <n> planes, ppmtoilbm writes a HAM file (if -hamif is used), a 24bit file (if -24if is used) or a direct color file (if -dcif is used) or aborts with an error.

-fixplanes | -fp n
(min 1, max 16) If a normal ILBM is written, it will have exactly <n> planes.

-hambits | -hamplanes n
(default 6, min 3, max 16) Select number of planes for HAM picture. The current Amiga hardware supports 6 and 8 planes, so for now you should only use this values.

-normal (default)
Turns off -hamif/-24if/-dcif, -hamforce/-24force/-dcforce and -cmaponly. Also sets compression type to byterun1.

-hamif (*)
-24if (*)
-dcif (*)
Write a HAM/24bit/direct color file if the pixmap does not fit into <maxplanes> planes.

-hamforce (*)
-24force (*)
-dcforce (*)
Write a HAM/24bit/direct color file.

-dcbits | -dcplanes r g b
(default 5, min 1, max 16). Select number of bits for red, green & blue in a direct color ILBM.

-ecs (default)
Shortcut for: -hamplanes 6 -maxplanes 5

-aga Shortcut for: -hamplanes 8 -maxplanes 8

-ham6 Shortcut for: -hamplanes 6 -hamforce

-ham8 Shortcut for: -hamplanes 8 -hamforce

-compress (*) (default)

-cmethod none|byterun1
Compress the BODY chunk. The default compression method is byterun1. Compression requires building the ILBM image in memory;

turning compression off allows stream-writing of the image, but the resulting file will usually be 30% to 50% larger. Another alternative is the -savemem option, this will keep memory requirements for compression at a minimum, but is very slow.

-map ppmfile — Write a normal ILBM using the colors in <ppmfile> as the colormap. The colormap file also determines the number of planes, a -maxplanes or -fixplanes option is ignored.

-cmaponly — Write a colormap file: only BMHD and CMAP chunks, no BODY chunk, nPlanes = 0.

-savemem — See the -compress option.

Notes

Supported ILBM types are:
o Normal ILBMs with 1-16 planes.
o Amiga HAM with 3-16 planes.
o 24 bit.
o Color map (BMHD + CMAP chunk only, nPlanes = 0).

Unofficial direct color.
1-16 planes for each color component.

Chunks written:
BMHD, CMAP, CAMG (only for HAM), BODY (not for colormap files) unofficial DCOL chunk for direct color ILBM

Pr

Usage: `pr [OPTION]... [FILE]...`

Synopsis — Unix-like command

Location — Development:bin

Function — Paginate or columnate FILE(s) for printing.

Inputs — Mandatory arguments to long options are mandatory for short options too.

+FIRST_PAGE[:LAST_PAGE], --pages=FIRST_PAGE[:LAST_PAGE]
begin [stop] printing with page FIRST_[LAST_]PAGE

-COLUMN, --columns=COLUMN
output COLUMN columns and print columns down, unless -a is used. Balance number of lines in the columns on each page.

-a, --across
print columns across rather than down, used together with -COLUMN

-c, --show-control-chars

```
                use hat notation (^G) and octal backslash notation
  -d, --double-space
                double space the output
  -D, --date-format=FORMAT
                use FORMAT for the header date
  -e[CHAR[WIDTH]], --expand-tabs[=CHAR[WIDTH]]
                expand input CHARs (TABs) to tab WIDTH (8)
  -F, -f, --form-feed
                use form feeds instead of newlines to separate pages
                (by a 3-line page header with -F or a 5-line header
                and trailer without -F)
  -h HEADER, --header=HEADER
                use a centered HEADER instead of filename in page
                header, -h "" prints a blank line, don't use -h""
  -i[CHAR[WIDTH]], --output-tabs[=CHAR[WIDTH]]
                replace spaces with CHARs (TABs) to tab WIDTH (8)
  -J, --join-lines
                merge full lines, turns off -W line truncation, no
                column alignment, --sep-string[=STRING] sets
                separators
  -l PAGE_LENGTH, --length=PAGE_LENGTH
                set the page length to PAGE_LENGTH (66) lines
                (default number of lines of text 56, and with -F 63)
  -m, --merge
                print all files in parallel, one in each column,
                truncate lines, but join lines of full length with -J
  -n[SEP[DIGITS]], --number-lines[=SEP[DIGITS]]
                number lines, use DIGITS (5) digits, then SEP (TAB),
                default counting starts with 1st line of input file
  -N NUMBER, --first-line-number=NUMBER
                start counting with NUMBER at 1st line of first
                page printed (see +FIRST_PAGE)
  -o MARGIN, --indent=MARGIN
                offset each line with MARGIN (zero) spaces, do not
                affect -w or -W, MARGIN will be added to
PAGE_WIDTH
  -r, --no-file-warnings
                omit warning when a file cannot be opened
  -s[CHAR],--separator[=CHAR]
                separate columns by a single character, default for
                CHAR is the <TAB> character without -w and 'no
                char' with -w -s[CHAR] turns off line truncation of all
                3 column options (-COLUMN|-a -COLUMN|-m) except
                -w is set
  -SSTRING, --sep-string[=STRING]
                separate columns by STRING,
                without -S: Default separator <TAB> with -J and
                <space> otherwise (same as -S" "), no effect on
                column options
  -t, --omit-header
                omit page headers and trailers
```

-T, --omit-pagination
omit page headers and trailers, eliminate any pagination by form feeds set in input files
-v, --show-nonprinting
use octal backslash notation
-w PAGE_WIDTH, --width=PAGE_WIDTH
set page width to PAGE_WIDTH (72) characters for multiple text-column output only, -s[char] turns off (72)
-W PAGE_WIDTH, --page-width=PAGE_WIDTH
set page width to PAGE_WIDTH (72) characters always, truncate lines, except -J option is set, no i nterference with -S or -s
--help display this help and exit
--version output version information and exit

-T implied by -l nn when nn <= 10 or <= 3 with -F. With no FILE, or when FILE is -, read standard input.

Printenv

Usage: `printenv [VARIABLE]...`
or: `printenv OPTION`

Synopsis Unix-like command

Location Development:bin

Function Show the value of a specified variable

Inputs If no environment VARIABLE specified, print them all.

--help display this help and exit
--version output version information and exit

Printf

Usage: `printf FORMAT [ARGUMENT]...`
or: `printf OPTION`

Synopsis Unix-like command

Location Development:bin

Function Print ARGUMENT(s) according to FORMAT.

Inputs

--help display this help and exit
--version output version information and exit

FORMAT controls the output as in C printf. Interpreted sequences are:

\" double quote
\NNN character with octal value NNN (1 to 3 digits)
\\ backslash
\a alert (BEL)
\b backspace
\c produce no further output
\f form feed
\n new line
\r carriage return
\t horizontal tab
\v vertical tab
\xHH byte with hexadecimal value HH (1 to 2 digits)
\uHHHH Unicode (ISO/IEC 10646) character with hex value HHHH (4 digits)
\UHHHHHHHH
Unicode character with hex value HHHHHHHH (8 digits)
%% a single %
%b ARGUMENT as a string with `\' escapes interpreted, except that octal escapes are of the form \0 or \0NNN

and all C format specifications ending with one of diouxXfeEgGcs, with ARGUMENTs converted to proper type first. Variable widths are handled.

Ptx

Usage: `ptx [OPTION]... [INPUT]... (without -G)`
`or: ptx -G [OPTION]... [INPUT [OUTPUT]]`

Synopsis Unix-like command

Location Development:bin

Function Output a permuted index, including context, of the words in the input files.

Inputs Mandatory arguments to long options are mandatory for short options too.

-A, --auto-reference
output automatically generated references
-G, --traditional

```
                behave more like System V `ptx'
  -F, --flag-truncation=STRING
                use STRING for flagging line truncations
  -M, --macro-name=STRING
                macro name to use instead of `xx'
  -O, --format=roff
                generate output as roff directives
  -R, --right-side-refs
                put references at right, not counted in -w
  -S, --sentence-regexp=REGEXP
                for end of lines or end of sentences
  -T, --format=tex
                generate output as TeX directives
  -W, --word-regexp=REGEXP
                use REGEXP to match each keyword
  -b, --break-file=FILE
                word break characters in this FILE
  -f, --ignore-case
                fold lower case to upper case for sorting
  -g, --gap-size=NUMBER
                gap size in columns between output fields
  -i, --ignore-file=FILE
                read ignore word list from FILE
  -o, --only-file=FILE
                read only word list from this FILE
  -r, --references
                first field of each line is a reference
  -t, --typeset-mode
                - not implemented -
 -w, --width=NUMBER
                output width in columns, reference excluded
      --help    display this help and exit
      --version output version information and exit

With no FILE or if FILE is -, read Standard Input.  `-F /' by default.
```

Pwd

Usage: `pwd [OPTION]`

Synopsis	Unix-like command
Location	Development:bin
Function	Print the full filename of the current working directory.
Inputs	--help display this help and exit --version output version information and exit
Note	The equivalent command in AROS shell is just 'cd' without arguments.

Python

Usage: `python [option] ... [-c cmd | -m mod | file | -] [arg] ...`

Synopsis Unix-like command

Location Development:bin

Function Python language compiler and interpreter

Inputs
-c cmd : program passed in as string (terminates option list)
-d : debug output from parser (also PYTHONDEBUG=x)
-E : ignore environment variables (such as PYTHONPATH)
-h : print this help message and exit (also --help)
-i : inspect interactively after running script, (also PYTHONINSPECT=x)
and force prompts, even if stdin does not appear to be a terminal
-m mod : run library module as a script (terminates option list)
-O : optimize generated bytecode (a tad; also PYTHONOPTIMIZE=x)
-OO : remove doc-strings in addition to the -O optimizations
-Q arg : division options: -Qold (default), -Qwarn, -Qwarnall, -Qnew
-S : don't imply 'import site' on initialization
-t : issue warnings about inconsistent tab usage (-tt: issue errors)
-u : unbuffered binary stdout and stderr (also PYTHONUNBUFFERED=x)
see man page for details on internal buffering relating to '-u'
-v : verbose (trace import statements) (also PYTHONVERBOSE=x)
-V : print the Python version number and exit (also --version)
-W arg : warning control (arg is action:message:category:module:lineno)
-x : skip first line of source, allowing use of non-Unix forms of #!cmd
file : program read from script file
- : program read from stdin (default; interactive mode if a tty)
arg ...: arguments passed to program in sys.argv[1:]

Other variables:
PYTHONSTARTUP: file executed on interactive startup (no default)
PYTHONPATH : ':'-separated list of directories prefixed to the default module search path. The result is sys.path.
PYTHONHOME : alternate <prefix> directory (or <prefix>:<exec_prefix>).
The default module search path uses <prefix>/pythonX.X.
PYTHONCASEOK : ignore case in 'import' statements (Windows).

Ranlib

Usage: `ranlib [OPTIONS] archive`

Synopsis Unix-like command

Location Development:bin

Function ranlib generates an index to the contents of an archive and stores it in the archive. The index lists each symbol defined by a member of an archive that is a relocatable object file.
You may use `nm -s` or `nm --print-armap` to list this index.
An archive with such an index speeds up linking to the library and allows routines in the library to call each other without regard to their placement in the archive.
The GNU ranlib program is another form of GNU ar; running ranlib is completely equivalent to executing `ar -s`

Inputs OPTIONS

-v, --version
Show the version number of ranlib.

-t Update the timestamp of the symbol map of an archive.

@file
Read command-line options from file. The options read are inserted in place of the original @file option. If file does not exist, or cannot be read, then the option will be treated literally, and not removed.

Options in file are separated by whitespace. A whitespace character may be included in an option by surrounding the entire option in either single or double quotes. Any character (including a backslash) may be included by prefixing the character to be included with a backslash. The file may itself contain additional @file options; any such options will be processed recursively.

See also `Ar`

Readelf

Usage: `readelf <option(s)> elf-file(s)`

Synopsis Unix-like command

Location Development:bin

Function Display information about the contents of ELF format files

Inputs Options are:

-a --all Equivalent to: -h -l -S -s -r -d -V -A -I
-h --file-header
Display the ELF file header
-l --program-headers
Display the program headers
--segments

```
                An alias for --program-headers
-S --section-headers
                Display the sections' header
  --sections
                An alias for --section-headers
-g --section-groups
                Display the section groups
-t --section-details
                Display the section details
-e --headers
                Equivalent to: -h -l -S
-s --syms       Display the symbol table
  --symbols     An alias for --syms
--dyn-syms      Display the dynamic symbol table
-n --notes      Display the core notes (if present)
-r --relocs     Display the relocations (if present)
-u --unwind     Display the unwind info (if present)
-d --dynamic
                Display the dynamic section (if present)
-V --version-info
                Display the version sections (if present)
-A --arch-specific
                Display architecture specific information (if any).
-c --archive-index
                Display the symbol/file index in an archive
-D --use-dynamic
                Use the dynamic section info when displaying
                symbols
-x --hex-dump=<number|name>
                Dump the contents of section <number|name>
                as bytes
-p --string-dump=<number|name>
                Dump the contents of section <number|name>
                as strings
-R --relocated-dump=<number|name>
                Dump the contents of section <number|name>
                as relocated bytes
-w[lLiaprmfFsoRt] or
--debug-dump[=rawline,=decodedline,=info,=abbrev,
                  =pubnames,=aranges,=macro,=frames,
                  =frames-interp,=str,=loc,=Ranges,=pubtypes,
                  =gdb_index,=trace_info,=trace_abbrev,
                  =trace_aranges]
                 Display the contents of DWARF2 debug sections
-I --histogram
                Display histogram of bucket list lengths
-W --wide       Allow output width to exceed 80 characters
@<file>         Read options from <file>
-H --help       Display this information
-v --version    Display the version number of readelf
```

Readlink

Usage: `readlink [OPTION]... FILE`

Synopsis Unix-like command

Location Development:bin

Function Display value of a symbolic link on standard output.

Inputs
-f, --canonicalize
canonicalize by following every symlink in every component of the given name recursively; all but the last component must exist
-e, --canonicalize-existing
canonicalize by following every symlink in every component of the given name recursively, all components must exist
-m, --canonicalize-missing
canonicalize by following every symlink in every component of the given name recursively, without requirements on components existence
-n, --no-newline
do not output the trailing newline
-q, --quiet, -s, --silent
suppress most error messages
-v, --verbose
report error messages
--help display this help and exit
--version
output version information and exit

Rm

Usage: `rm [OPTION]... FILE...`

Synopsis Unix-like command

Location Development:bin

Function Remove (unlink) the FILE(s).

Inputs
-f, --force ignore nonexistent files, never prompt
-i prompt before every removal
-I prompt once before removing more than three files, or when removing recursively. Less intrusive than -i,

while still giving protection against most mistakes
--interactive[=WHEN]
prompt according to WHEN: never, once (-I), or always (-i). Without WHEN, prompt always
--one-file-system
when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument
--no-preserve-root
do not treat `/' specially
--preserve-root
do not remove `/' (default)
-r, -R, --recursive
remove directories and their contents recursively
-v, --verbose
explain what is being done
--help display this help and exit
--version
output version information and exit

By default, rm does not remove directories. Use the --recursive (-r or -R) option to remove each listed directory, too, along with all of its contents.

Example To remove a file whose name starts with a `-', for example `-foo', use one of these commands:

```
rm -- -foo
rm ./-foo
```

Rmdir

Usage: `rmdir [OPTION]... DIRECTORY...`

Synopsis Unix-like command

Location Development:bin

Function Remove the DIRECTORY(ies), if they are empty.

Inputs
--ignore-fail-on-non-empty
ignore each failure that is solely because a directory is non-empty
-p, --parents
Remove DIRECTORY and its ancestors. E.g., `rmdir -p a/b/c' is similar to `rmdir a/b/c a/b a'.
-v, --verbose
output a diagnostic for every directory processed

--help display this help and exit
--version
output version information and exit

Sdiff

Usage: `sdiff [OPTION]... FILE1 FILE2`

Synopsis Unix-like command

Location Development:bin

Function Side-by-side merge of file differences.

Inputs -o FILE --output=FILE
Operate interactively, sending output to FILE.
-i --ignore-case
Consider upper- and lower-case to be the same.
-E --ignore-tab-expansion
Ignore changes due to tab expansion.
-b --ignore-space-change
Ignore changes in the amount of white space.
-W --ignore-all-space
Ignore all white space.
-B --ignore-blank-lines
Ignore changes whose lines are all blank.
-I RE --ignore-matching-lines=RE
Ignore changes whose lines all match RE.
--strip-trailing-cr
Strip trailing carriage return on input.
-a --text
Treat all files as text.
-w NUM --width=NUM
Output at most NUM (default 130) print columns.
-l --left-column
Output only the left column of common lines.
-s --suppress-common-lines
Do not output common lines.
-t --expand-tabs
Expand tabs to spaces in output.
--tabsize=NUM
Tab stops are every NUM (default 8) print columns.
-d --minimal
Try hard to find a smaller set of changes.
-H --speed-large-files
Assume large files and many scattered small changes.
--diff-program=PROGRAM
Use PROGRAM to compare files.
-v --version

Output version info.

--help
Output this help.

If a FILE is `-', read standard input.

Sed

Usage: `sed [OPTION]... {script-only-if-no-other-script} [input-file]...`

Synopsis Unix-like command

Location Development:bin

Function *** REPLACE ME ***

Inputs
-n, --quiet, --silent
suppress automatic printing of pattern space
-e script, --expression=script
add the script to the commands to be executed
-f script-file, --file=script-file
add the contents of script-file to the commands to be executed
-i[SUFFIX], --in-place[=SUFFIX]
edit files in place (makes backup if extension supplied)
-l N, --line-length=N
specify the desired line-wrap length for the `l' command
--posix
disable all GNU extensions.
-r, --regexp-extended
use extended regular expressions in the script.
-s, --separate
consider files as separate rather than as a single continuous long stream.
-u, --unbuffered
load minimal amounts of data from the input files and flush the output buffers more often
--help display this help and exit
--version output version information and exit

If no -e, --expression, -f, or --file option is given, then the first non-option argument is taken as the sed script to interpret. All remaining arguments are names of input files; if no input files are specified, then the standard input is read.

Seq

Usage: `seq [OPTION]... LAST`
or: `seq [OPTION]... FIRST LAST`
or: `seq [OPTION]... FIRST INCREMENT LAST`

Synopsis Unix-like command

Location Development:bin

Function Print numbers from FIRST to LAST, in steps of INCREMENT.

Inputs
-f, --format=FORMAT
use printf style floating-point FORMAT
-s, --separator=STRING
use STRING to separate numbers (default: \n)
-w, --equal-width equalize width by padding with leading zeroes
--help display this help and exit
--version output version information and exit

Usage If FIRST or INCREMENT is omitted, it defaults to 1. That is, an omitted INCREMENT defaults to 1 even when LAST is smaller than FIRST. FIRST, INCREMENT, and LAST are interpreted as floating point values. INCREMENT is usually positive if FIRST is smaller than LAST, and INCREMENT is usually negative if FIRST is greater than LAST. FORMAT must be suitable for printing one argument of type `double'; it defaults to %.PRECf if FIRST, INCREMENT, and LAST are all fixed point decimal numbers with maximum precision PREC, and to %g otherwise.

Sha256sum (and similar ones)

Usage: `sha256sum [OPTION] [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Print or check SHA256 (256-bit) checksums.

Inputs With no FILE, or when FILE is -, read standard input.

-b, --binary read in binary mode
-c, --check read SHA256 sums from the FILEs and check them

-t, --text read in text mode (default)

The following two options are useful only when verifying checksums:

--status don't output anything, status code shows success
-w, --warn warn about improperly formatted checksum lines

--help display this help and exit
--version output version information and exit

The sums are computed as described in FIPS-180-2. When checking, the input should be a former output of this program. The default mode is to print a line with checksum, a character indicating type (`*' for binary, ` ' for text), and name for each FILE.

Note There are also other commands similar to this one, printing or checking different types of checksums:

sha1sum → SHA160 (160 bits) described in FIPS-180-1
sha224sum → SHA224 (224 bits) described in RFC 3874
sha384sum → SHA384 (384 bits) described in FIPS-180-2
sha512sum → SHA512 (512 bits) described in FIPS-180-2

they all share the same syntax and options.

Shred

Usage: `shred [OPTIONS] FILE [...]`

Synopsis Unix-like command

Location Development:bin

Function Overwrite the specified FILE(s) repeatedly, in order to make it harder for even very expensive hardware probing to recover the data.

CAUTION **It doesn't work correctly with AROS filesystems. Trying to use shred on an AROS file will likely make the file completely unusable, but still present on drive.**

Inputs Mandatory arguments to long options are mandatory for short options too.
-f, --force change permissions to allow writing if necessary
-n, --iterations=N
Overwrite N times instead of the default (25)

--random-source=FILE
get random bytes from FILE (default /dev/urandom)
-s, --size=N
shred this many bytes (suffixes like K, M, G accepted)
-u, --remove
truncate and remove file after overwriting
-v, --verbose
show progress
-x, --exact
do not round file sizes up to the next full block;
this is the default for non-regular files
-z, --zero add a final overwrite with zeros to hide shredding
--help display this help and exit
--version
output version information and exit

If FILE is -, shred standard output.

Warning Delete FILE(s) if --remove (-u) is specified. The default is not to remove the files because it is common to operate on device files like /dev/hda, and those files usually should not be removed. When operating on regular
files, most people use the --remove option.

CAUTION: Note that shred relies on a very important assumption: that the file system overwrites data in place. This is the traditional way to do things, but many modern file system designs do not satisfy this assumption. The following are examples of file systems on which shred is not effective, or is not guaranteed to be effective in all file system modes:

* log-structured or journaled file systems, such as those supplied with AIX and Solaris (and JFS, ReiserFS, XFS, Ext3, etc.)

* file systems that write redundant data and carry on even if some writes fail, such as RAID-based file systems

* file systems that make snapshots, such as Network Appliance's NFS server

* file systems that cache in temporary locations, such as NFS version 3 clients

* compressed file systems

In the case of ext3 file systems, the above disclaimer applies (and shred is thus of limited effectiveness) only in data=journal mode, which journals file data in addition to just metadata. In both the data=ordered (default) and data=writeback modes, shred works as usual. Ext3 journaling modes can be changed by adding the data=something option
to the mount options for a particular file system in the /etc/fstab file, as documented in the mount man page (man mount).

In addition, file system backups and remote mirrors may contain copies

of the file that cannot be removed, and that will allow a shredded file to be recovered later.

Shuf

Usage: `shuf [OPTION]... [FILE]`
`or: shuf -e [OPTION]... [ARG]...`
`or: shuf -i LO-HI [OPTION]...`

Synopsis Unix-like command

Location Development:bin

Function Write a random permutation of the input lines to standard output.

Inputs Mandatory arguments to long options are mandatory for short options too.

-e, --echo treat each ARG as an input line
-i, --input-range=LO-HI
treat each number LO through HI as an input line
-n, --head-lines=LINES
output at most LINES lines
-o, --output=FILE
write result to FILE instead of standard output
--random-source=FILE
get random bytes from FILE (default /dev/urandom)
-z, --zero-terminated
end lines with 0 byte, not newline
--help display this help and exit
--version output version information and exit

With no FILE, or when FILE is -, read standard input.

Size

Usage: `size [option(s)] [file(s)]`

Synopsis Unix-like command

Location Development:bin

Function Displays the sizes of sections inside binary files
If no input file(s) are specified, a.out is assumed

Inputs The options are:

-A|-B --format={sysv|berkeley} Select output style (default is berkeley)
-o|-d|-x --radix={8|10|16} Display numbers in octal, decimal or hex
-t --totals Display the total sizes (Berkeley only)
--common Display total size for *COM* syms
--target=<bfdname>
Set the binary file format
@<file> Read options from <file>
-h --help Display this information
-v --version Display the program's version

Supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Sleep

Usage: `sleep NUMBER[SUFFIX]...`
or: `sleep OPTION`

Synopsis: Unix-like command

Location: Development:bin

Function: Pause for NUMBER seconds. SUFFIX may be `s' for seconds (the default), `m' for minutes, `h' for hours or `d' for days. Unlike most implementations that require NUMBER be an integer, here NUMBER may be an arbitrary floating point number. Given two or more arguments, pause for the amount of time specified by the sum of their values.

Inputs:
--help display this help and exit
--version output version information and exit

Split

Usage: `split [OPTION] [INPUT [PREFIX]]`

Synopsis: Unix-like command

Location: Development:bin

Function: Output fixed-size pieces of INPUT to PREFIXaa, PREFIXab, ...; default size is 1000 lines, and default PREFIX is `x'. With no

INPUT, or when INPUT is -, read standard input.

Inputs

-a, --suffix-length=N	use suffixes of length N (default 2)
-b, --bytes=SIZE	put SIZE bytes per output file
-C, --line-bytes=SIZE	put at most SIZE bytes of lines per output file
-d, --numeric-suffixes	use numeric suffixes instead of alphabetic
-l, --lines=NUMBER	put NUMBER lines per output file
--verbose	print a diagnostic to standard error just before each output file is opened
--help	display this help and exit
--version	output version information and exit

SIZE may have a multiplier suffix: b for 512, k for 1K, m for 1 Meg.

Stat

Usage: `stat [OPTION] FILE...`

Synopsis Unix-like command

Location Development:bin

Function Display file or file system status.

Inputs

-L, --dereference	follow links
-f, --file-system	display file system status instead of file status
-c --format=FORMAT	use the specified FORMAT instead of the default; output a newline after each use of FORMAT
--printf=FORMAT	like --format, but interpret backslash escapes, and do not output a mandatory trailing newline. If you want a newline, include \n in FORMAT.
-t, --terse	print the information in terse form
--help	display this help and exit
--version	output version information and exit

The valid format sequences for files (without --file-system):

%a	Access rights in octal
%A	Access rights in human readable form
%b	Number of blocks allocated (see %B)
%B	The size in bytes of each block reported by %b
%d	Device number in decimal
%D	Device number in hex
%f	Raw mode in hex
%F	File type
%g	Group ID of owner
%G	Group name of owner
%h	Number of hard links
%i	Inode number
%n	File name
%N	Quoted file name with dereference if symbolic link

%o I/O block size
%s Total size, in bytes
%t Major device type in hex
%T Minor device type in hex
%u User ID of owner
%U User name of owner
%x Time of last access
%X Time of last access as seconds since Epoch
%y Time of last modification
%Y Time of last modification as seconds since Epoch
%z Time of last change
%Z Time of last change as seconds since Epoch

Valid format sequences for file systems:

%a Free blocks available to non-superuser
%b Total data blocks in file system
%c Total file nodes in file system
%d Free file nodes in file system
%f Free blocks in file system
%i File System ID in hex
%l Maximum length of filenames
%n File name
%s Block size (for faster transfers)
%S Fundamental block size (for block counts)
%t Type in hex
%T Type in human readable form

Notes Your shell may have its own version of stat, which usually supersedes the version described here. Please refer to your shell's documentation for details about the options it supports.

Strings

Usage: `strings [option(s)] [file(s)]`

Synopsis Unix-like command

Location Development:bin

Function Display printable strings in [file(s)] (stdin by default)

Inputs -a - --all Scan the entire file, not just the data section
-f --print-file-name
Print the name of the file before each string
-n --bytes=[number]
Locate & print any NUL-terminated sequence of at
-<number> least [number] characters (default 4).
-t --radix={o,d,x}
Print the location of the string in base 8, 10 or 16

-o An alias for --radix=o
-T --target=<BFDNAME>
Specify the binary file format
-e --encoding={s,S,b,l,B,L}
Select character size and endianness:
s = 7-bit, S = 8-bit, {b,l} = 16-bit, {B,L} = 32-bit
@<file> Read options from <file>
-h --help Display this information
-v -V --version Print the program's version number

Supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Strip

Usage: `strip <option(s)> in-file(s)`

Synopsis Unix-like command

Location Development:bin

Function Removes symbols and sections from files

Inputs The options are:
-I –input-target=<bfdname>
Assume input file is in format <bfdname>
-O --output-target=<bfdname>
Create an output file in format <bfdname>
-F --target=<bfdname>
Set both input and output format to <bfdname>
-p --preserve-dates
Copy modified/access timestamps to the output
-R --remove-section=<name>
Remove section <name> from the output
-s --strip-all Remove all symbol and relocation information
-g -S -d --strip-debug
Remove all debugging symbols & sections
--strip-unneeded
Remove all symbols not needed by relocations
--only-keep-debug
Strip everything but the debug information
-N --strip-symbol=<name>
Do not copy symbol <name>
-K --keep-symbol=<name>
Do not strip symbol <name>
--keep-file-symbols
Do not strip file symbol(s)
-w --wildcard Permit wildcard in symbol comparison
-x --discard-all Remove all non-global symbols

-X --discard-locals
Remove any compiler-generated symbols
-v --verbose List all object files modified
-V --version Display this program's version number
-h --help Display this output
--info List object formats & architectures supported
-o <file> Place stripped output into <file>

Supported targets: elf32-i386 elf64-x86-64 elf32-powerpc elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Stty

Usage: `stty [-F DEVICE] [--file=DEVICE] [SETTING]...`
or: `stty [-F DEVICE] [--file=DEVICE] [-a|--all]`
or: `stty [-F DEVICE] [--file=DEVICE] [-g|--save]`

Synopsis: Unix-like command

Location: Development:bin

Function: Print or change terminal characteristics.

Inputs:
-a, --all print all current settings in human-readable form
-g, --save print all current settings in a stty-readable form
-F, --file=DEVICE
open and use the specified DEVICE instead of stdin
--help display this help and exit
--version output version information and exit

Optional - before SETTING indicates negation. An * marks non-POSIX settings. The underlying system defines which settings are available.

Special characters:
* dsusp CHAR CHAR will send a terminal stop signal once input flushed
eof CHAR CHAR will send an end of file (terminate the input)
eol CHAR CHAR will end the line
* eol2 CHAR alternate CHAR for ending the line
erase CHAR CHAR will erase the last character typed
intr CHAR CHAR will send an interrupt signal
kill CHAR CHAR will erase the current line
* lnext CHAR CHAR will enter the next character quoted
quit CHAR CHAR will send a quit signal
* rprnt CHAR CHAR will redraw the current line
start CHAR CHAR will restart the output after stopping it
stop CHAR CHAR will stop the output
susp CHAR CHAR will send a terminal stop signal
* swtch CHAR CHAR will switch to a different shell layer
* werase CHAR CHAR will erase the last word typed

Special settings:

```
  N             set the input and output speeds to N bauds
* cols N        tell the kernel that the terminal has N columns
* columns N     same as cols N
  ispeed N      set the input speed to N
* line N        use line discipline N
  min N         with -icanon, set N characters minimum for a completed
                read
  ospeed N      set the output speed to N
* rows N        tell the kernel that the terminal has N rows
* size          print the number of rows and columns according to the
                kernel
  speed         print the terminal speed
  time N        with -icanon, set read timeout of N tenths of a second

Control settings:
  [-]clocal     disable modem control signals
  [-]cread      allow input to be received
* [-]crtscts    enable RTS/CTS handshaking
  csN           set character size to N bits, N in [5..8]
  [-]cstopb     use two stop bits per character (one with `-')
  [-]hup        send a hangup signal when the last process closes the tty
  [-]hupcl      same as [-]hup
  [-]parenb     generate parity bit in output and expect parity bit in input
  [-]parodd     set odd parity (even with `-')

Input settings:
  [-]brkint     breaks cause an interrupt signal
  [-]icrnl      translate carriage return to newline
  [-]ignbrk     ignore break characters
  [-]igncr      ignore carriage return
  [-]ignpar     ignore characters with parity errors
* [-]imaxbel    beep and do not flush a full input buffer on a character
  [-]inlcr      translate newline to carriage return
  [-]inpck      enable input parity checking
  [-]istrip     clear high (8th) bit of input characters
* [-]iutf8      assume input characters are UTF-8 encoded
* [-]iuclc      translate uppercase characters to lowercase
* [-]ixany      let any character restart output, not only start character
  [-]ixoff      enable sending of start/stop characters
  [-]ixon       enable XON/XOFF flow control
  [-]parmrk     mark parity errors (with a 255-0-character sequence)
  [-]tandem     same as [-]ixoff

Output settings:
* bsN           backspace delay style, N in [0..1]
* crN           carriage return delay style, N in [0..3]
* ffN           form feed delay style, N in [0..1]
* nlN           newline delay style, N in [0..1]
* [-]ocrnl      translate carriage return to newline
* [-]ofdel      use delete characters for fill instead of null characters
* [-]ofill      use fill (padding) characters instead of timing for delays
* [-]olcuc      translate lowercase characters to uppercase
* [-]onlcr      translate newline to carriage return-newline
* [-]onlret     newline performs a carriage return
* [-]onocr      do not print carriage returns in the first column
  [-]opost      postprocess output
```

```
 * tabN          horizontal tab delay style, N in [0..3]
 * tabs          same as tab0
 * -tabs         same as tab3
 * vtN           vertical tab delay style, N in [0..1]

Local settings:
   [-]crterase   echo erase characters as backspace-space-backspace
 * crtkill       kill all line by obeying the echoprt and echoe settings
 * -crtkill      kill all line by obeying the echoctl and echok settings
 * [-]ctlecho    echo control characters in hat notation (`^c')
   [-]echo       echo input characters
 * [-]echoctl    same as [-]ctlecho
   [-]echoe      same as [-]crterase
   [-]echok      echo a newline after a kill character
 * [-]echoke     same as [-]crtkill
   [-]echonl     echo newline even if not echoing other characters
 * [-]echoprt    echo erased characters backward, between `\' and '/'
   [-]icanon     enable erase, kill, werase, and rprnt special characters
   [-]iexten     enable non-POSIX special characters
   [-]isig       enable interrupt, quit, and suspend special characters
   [-]noflsh     disable flushing after interrupt and quit special characters
 * [-]prterase   same as [-]echoprt
 * [-]tostop     stop background jobs that try to write to the terminal
 * [-]xcase      with icanon, escape with `\' for uppercase characters

Combination settings:
 * [-]LCASE     same as [-]lcase
   cbreak       same as -icanon
   -cbreak      same as icanon
   cooked       same as brkint ignpar istrip icrnl ixon opost isig
                icanon, eof and eol characters to their default values
   -cooked      same as raw
   crt          same as echoe echoctl echoke
   dec          same as echoe echoctl echoke -ixany intr ^c erase 0177
                kill ^u
 * [-]decctlq   same as [-]ixany
   ek           erase and kill characters to their default values
   evenp        same as parenb -parodd cs7
   -evenp       same as -parenb cs8
 * [-]lcase     same as xcase iuclc olcuc
   litout       same as -parenb -istrip -opost cs8
   -litout      same as parenb istrip opost cs7
   nl           same as -icrnl -onlcr
   -nl          same as icrnl -inlcr -igncr onlcr -ocrnl -onlret
   oddp         same as parenb parodd cs7
   -oddp        same as -parenb cs8
   [-]parity    same as [-]evenp
   pass8        same as -parenb -istrip cs8
   -pass8       same as parenb istrip cs7
   raw          same as -ignbrk -brkint -ignpar -parmrk -inpck -istrip
                -inlcr -igncr -icrnl  -ixon  -ixoff  -iuclc  -ixany
                -imaxbel -opost -isig -icanon -xcase min 1 time 0
   -raw         same as cooked
   sane         same as cread -ignbrk brkint -inlcr -igncr icrnl -iutf8
                -ixoff -iuclc -ixany imaxbel opost -olcuc -ocrnl onlcr
                -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
```

isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke, all special characters to their default values.

Handle the tty line connected to standard input. Without arguments, prints baud rate, line discipline, and deviations from stty sane. In settings, CHAR is taken literally, or coded as in ^c, 0x37, 0177 or 127; special values ^- or undef used to disable special characters.

Sum

Usage: `sum [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Print checksum and block counts for each FILE.

Inputs
-r defeat -s, use BSD sum algorithm, use 1K blocks
-s, --sysv use System V sum algorithm, use 512 bytes blocks
--help display this help and exit
--version output version information and exit

With no FILE, or when FILE is -, read standard input.

Sync

Usage: `sync [OPTION]`

Synopsis Unix-like command

Location Development:bin

Function Force changed blocks to disk, update the super block.

Inputs
--help display this help and exit
--version output version information and exit

Tac

Usage: `tac [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Write each FILE to standard output, last line first.

Inputs With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.

-b, --before	attach the separator before instead of after
-r, --regex	interpret the separator as a regular expression
-s, --separator=STRING	use STRING as the separator instead of newline
--help	display this help and exit
--version	output version information and exit

Tee

Usage: `tee [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Copy standard input to each FILE, and also to standard output.

Inputs

-a, --append	append to the given FILEs, do not overwrite
-i, --ignore-interrupts	ignore interrupt signals
--help	display this help and exit
--version	output version information and exit

Notes If a FILE is -, copy again to standard output.

Test

Usage: `test EXPRESSION`
or: `test`
or: `[ EXPRESSION ]`
or: `[ ]`
or: `[ OPTION`

Synopsis Unix-like command

Location Development:bin

Function Exit with the status determined by EXPRESSION.

Inputs --help display this help and exit
--version output version information and exit

An omitted EXPRESSION defaults to false. Otherwise,
EXPRESSION is true or false and sets exit status. It is one of:

(EXPRESSION)
EXPRESSION is true
! EXPRESSION
EXPRESSION is false
EXPRESSION1 -a EXPRESSION2
both EXPRESSION1 and EXPRESSION2 are true
EXPRESSION1 -o EXPRESSION2
either EXPRESSION1 or EXPRESSION2 is true

-n STRING
the length of STRING is nonzero
STRING
equivalent to -n STRING
-z STRING
the length of STRING is zero
STRING1 = STRING2
the strings are equal
STRING1 != STRING2
the strings are not equal

INTEGER1 -eq INTEGER2
INTEGER1 is equal to INTEGER2
INTEGER1 -ge INTEGER2
INTEGER1 is greater than or equal to INTEGER2
INTEGER1 -gt INTEGER2
INTEGER1 is greater than INTEGER2
INTEGER1 -le INTEGER2
INTEGER1 is less than or equal to INTEGER2
INTEGER1 -lt INTEGER2
INTEGER1 is less than INTEGER2
INTEGER1 -ne INTEGER2
INTEGER1 is not equal to INTEGER2

FILE1 -ef FILE2
FILE1 and FILE2 have the same device and inode numbers
FILE1 -nt FILE2
FILE1 is newer (modification date) than FILE2
FILE1 -ot FILE2
FILE1 is older than FILE2

-b FILE
FILE exists and is block special
-c FILE

FILE exists and is character special
-d FILE
FILE exists and is a directory
-e FILE
FILE exists
-f FILE
FILE exists and is a regular file
-g FILE
FILE exists and is set-group-ID
-G FILE
FILE exists and is owned by the effective group ID
-h FILE
FILE exists and is a symbolic link (same as -L)
-k FILE
FILE exists and has its sticky bit set
-L FILE
FILE exists and is a symbolic link (same as -h)
-O FILE
FILE exists and is owned by the effective user ID
-p FILE
FILE exists and is a named pipe
-r FILE
FILE exists and read permission is granted
-s FILE
FILE exists and has a size greater than zero
-S FILE
FILE exists and is a socket
-t FD
file descriptor FD is opened on a terminal
-u FILE
FILE exists and its set-user-ID bit is set
-w FILE
FILE exists and write permission is granted
-x FILE
FILE exists and execute (or search) permission is granted

Except for -h and -L, all FILE-related tests dereference symbolic links. Beware that parentheses need to be escaped (e.g., by backslashes) for shells. INTEGER may also be -l STRING, which evaluates to the length of STRING.

Tr

Usage: `tr [OPTION]... SET1 [SET2]`

Synopsis Unix-like command

Location Development:bin

Function Translate, squeeze, and/or delete characters from standard input, writing to standard output.

Inputs

-c, -C, --complement	first complement SET1
-d, --delete	delete characters in SET1, do not translate
-s, --squeeze-repeats	replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character
-t, --truncate-set1	first truncate SET1 to length of SET2
--help	display this help and exit
--version	output version information and exit

SETs are specified as strings of characters. Most represent themselves. Interpreted sequences are:

\NNN	character with octal value NNN (1 to 3 octal digits)
\\	backslash
\a	audible BEL
\b	backspace
\f	form feed
\n	new line
\r	return
\t	horizontal tab
\v	vertical tab
CHAR1-CHAR2	all characters from CHAR1 to CHAR2 in ascending order
[CHAR*]	in SET2, copies of CHAR until length of SET1
[CHAR*REPEAT]	REPEAT copies of CHAR, REPEAT octal if starting with 0
[:alnum:]	all letters and digits
[:alpha:]	all letters
[:blank:]	all horizontal whitespace
[:cntrl:]	all control characters
[:digit:]	all digits
[:graph:]	all printable characters, not including space
[:lower:]	all lower case letters
[:print:]	all printable characters, including space
[:punct:]	all punctuation characters
[:space:]	all horizontal or vertical whitespace
[:upper:]	all upper case letters
[:xdigit:]	all hexadecimal digits
[=CHAR=]	all characters which are equivalent to CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear. -t may be used only when translating. SET2 is extended to length of SET1 by repeating its last character as necessary. Excess characters of SET2 are ignored. Only [:lower:] and [:upper:] are guaranteed to expand in ascending order; used in SET2 while translating, they may only be used in pairs to specify case conversion. -s uses SET1 if not

translating nor deleting; else squeezing uses SET2 and occurs after translation or deletion.

True

Usage: `true [ignored command line arguments]`
or: `true OPTION`

Synopsis Unix-like command

Location Development:bin

Function Exit with a status code indicating success.

Inputs
--help display this help and exit
--version output version information and exit

Tsort

Usage: `tsort [OPTION] [FILE]`

Synopsis Unix-like command

Location Development:bin

Function Write totally ordered list consistent with the partial ordering in FILE.

Inputs With no FILE, or when FILE is -, read standard input.

--help display this help and exit
--version output version information and exit

Example `1.sys> list | grep Device`

Searches for the "Device" word in the output of List command. If exists, it show only the occurring line

Uname

Usage: `uname [OPTION]...`

Synopsis Unix-like command

Location Development:bin

Function Print certain system information.

Inputs With no OPTION, same as -s.

-a, --all	print all information, in the following order, except omit -p and -i if unknown:
-s, --kernel-name	print the kernel name
-n, --nodename	print the network node hostname
-r, --kernel-release	print the kernel release
-v, --kernel-version	print the kernel version
-m, --machine	print the machine hardware name
-p, --processor	print the processor type or "unknown"
-i, --hardware-platform	print the hardware platform or "unknown"
-o, --operating-system	print the operating system
--help	display this help and exit
--version	output version information and exit

Unlink

Usage: `unlink FILE`
`or: unlink OPTION`

Synopsis Unix-like command

Location Development:bin

Function Call the unlink function to remove the specified FILE

Options

--help	display this help and exit
--version	output version information and exit

Unexpand

Usage: `unexpand [OPTION]... [FILE]...`

Synopsis Unix-like command

Location Development:bin

Function Convert blanks in each FILE to tabs, writing to standard output.
With no FILE, or when FILE is -, read standard input.

Inputs Mandatory arguments to long options are mandatory for short options too.

-a, --all	convert all blanks, instead of just initial blanks
--first-only	convert only leading sequences of blanks (overrides -a)
-t, --tabs=N	have tabs N characters apart instead of 8 (enables -a)
-t, --tabs=LIST	use comma separated LIST of tab positions (enables -a)
--help	display this help and exit
--version	output version information and exit

Wc

Usage: `wc [OPTION]... [FILE]...`
or: `wc [OPTION]... --files0-from=F`

Synopsis Unix-like command

Location Development:bin

Function Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. With no FILE, or when FILE is -, read standard input.

Inputs

-c, --bytes	print the byte counts
-m, --chars	print the character counts
-l, --lines	print the newline counts
--files0-from=F	read input from the files specified by NUL-terminated names in file F
-L, --max-line-length	print the length of the longest line
-w, --words	print the word counts
--help	display this help and exit
--version	output version information and exit

Xargs

Usage:
```
xargs [-0prtx] [--interactive] [--null] [-d|--delimiter=delim]
      [-E eof-str] [-e[eof-str]]  [--eof[=eof-str]]
      [-L max-lines] [-l[max-lines]] [--max-lines[=max-lines]]
      [-I replace-str] [-i[replace-str]] [--replace[=replace-str]]
      [-n max-args] [--max-args=max-args]
      [-s max-chars] [--max-chars=max-chars]
      [-P max-procs]  [--max-procs=max-procs] [--show-limits]
      [--verbose] [--exit] [--no-run-if-empty] [--arg-file=file]
      [--version] [--help] [command [initial-arguments]]
```

Synopsis Unix-like command

Location Development:bin

Function xargs reads items from the standard input, delimited by blanks (which can be

protected with double or single quotes or a backslash) or newlines, and executes the command (default is /bin/echo) one or more times with any initial-arguments followed by items read from standard input. Blank lines on the standard input are ignored.

Because Unix filenames can contain blanks and newlines, this default ehaviour is often problematic; filenames containing blanks and/or newlines are incorrectly processed by xargs. In these situations it is better to use the `-0' option, which prevents such problems. When using this option you will need to ensure that the program which produces the input for xargs also uses a null character as a separator. If that program is GNU find for example, the `-print0' option does this for you.

If any invocation of the command exits with a status of 255, xargs will stop immediately without reading any further input. An error message is issued on stderr when this happens.

Options

--arg-file=file, -a file
Read items from file instead of standard input. If you use this option, stdin remains unchanged when commands are run. Otherwise, stdin is redirected from /dev/null.

--null, -0
Input items are terminated by a null character instead of by whitespace, and the quotes and backslash are not special (every character is taken literally). Disables the end of file string, which is treated like any other argument. Useful when input items might contain white space, quote marks, or backslashes. The GNU find -print0 option produces input suitable for this mode.

--delimiter=delim, -d delim
Input items are terminated by the specified character. Quotes and backslash are not special; every character in the input is taken literally. Disables the end-of-file string, which is treated like any other argument. This can be used when the input consists of simply newline-separated items, although it is almost always better to design your program to use `--null' where this is possible. The specified delimiter may be a single character, a C-style character escape such as \n, or an octal or hexadecimal escape code. Octal and hexadecimal escape codes are understood as for the printf command. Multibyte characters are not supported.

-Eeof-str
Set the end of file string to eof-str. If the end of file string occurs as a line of input, the rest of the input is ignored. If neither -E nor -e is used, no end of file string is used.

--eof[=eof-str], -e[eof-str]
This option is a synonym for the `-E' option. Use `-E' instead, because it is POSIX compliant while this option is not. If eof-str is omitted, there is no end of file string. If neither -E nor -e is used, no end of file string is used.

--help Print a summary of the options to xargs and exit.

-I replace-str
Replace occurrences of replace-str in the initial-arguments with names read from standard input. Also, unquoted blanks do not terminate input items; instead the separator is the newline character. Implies -x and -L 1.

--replace[=replace-str], -i[replace-str]
This option is a synonym for -Ireplace-str if replace-str is specified, and for -I{} otherwise. This option is deprecated; use -I instead.

-L max-lines
Use at most max-lines nonblank input lines per command line. Trailing blanks cause an input line to be logically continued on the next input line. Implies -x.

--max-lines[=max-lines], -l[max-lines]
Synonym for the -L option. Unlike -L, the max-lines argument is optional. If max-args is not specified, it defaults to one. The -l option is deprecated since the POSIX standard specifies -L instead.

--max-args=max-args, -n max-args
Use at most max-args arguments per command line. Fewer than max-args arguments will be used if the size (see the -s option) is exceeded, unless the -x option is given, in which case xargs will exit.

--interactive, -p
Prompt the user about whether to run each command line and read a line from the terminal. Only run the command line if the response starts with `y' or `Y'. Implies -t.

--no-run-if-empty, -r
If the standard input does not contain any nonblanks, do not run the command. Normally, the command is run once even if there is no input. This option is a GNU extension.

--max-chars=max-chars, -s max-chars
Use at most max-chars characters per command line, including the command and initial-arguments and the terminating nulls at the ends of the argument strings. The default is 131072 characters, not including the size of the environment variables (which are provided for separately so that it doesn't matter if your environment variables take up more than 131072 bytes). The operating system places limits on the values that you can usefully specify, and if you exceed these a warning message is printed and the value actually used is set to the appropriate upper or lower limit.

--verbose, -t
Print the command line on the standard error output before executing it.

--version

Print the version number of xargs and exit.

--show-limits
Display the limits on the command-line length which are imposed by the operating system, xargs' choice of buffer size and the -s option. Pipe the input from /dev/null (and perhaps specify --no-run-if-empty) if you don't want xargs to do anything.

--exit, -x
Exit if the size (see the -s option) is exceeded.

--max-procs=max-procs, -P max-procs
Run up to max-procs processes at a time; the default is 1. If max-procs is 0, xargs will run as many processes as possible at a time. Use the -n option with -P; otherwise chances are that only one exec will be done.

Examples

```
find /tmp -name core -type f -print | xargs /bin/rm -f
```

Find files named core in or below the directory /tmp and delete them. Note that this will work incorrectly if there are any filenames containing newlines or spaces.

```
find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f
```

Find files named core in or below the directory /tmp and delete them, processing filenames in such a way that file or directory names containing spaces or newlines are correctly handled.

```
cut -d: -f1 < /etc/passwd | sort | xargs echo
```

Generates a compact listing of all the users on the system.

Exit Status

xargs exits with the following status:
0 if it succeeds
123 if any invocation of the command exited with status 1-125
124 if the command exited with status 255
125 if the command is killed by a signal
126 if the command cannot be run
127 if the command is not found
1 if some other error occurred.

Exit codes greater than 128 are used by the shell to indicate that a program died due to a fatal signal.

Xmlcatalog

Usage: `xmlcatalog [options] catalogfile entities...`

Synopsis Unix-like command

Location Development:bin

Function Parse the catalog file and query it for the entities

Inputs
--sgml handle SGML Super catalogs for --add and --del
--shell run a shell allowing interactive queries
--create create a new catalog
--add 'type' 'orig' 'replace'
add an XML entry
--add 'entry'
add an SGML entry
--del 'values'
remove values
--noout avoid dumping the result on stdout
used with --add or --del, it saves the catalog changes
and with --sgml it automatically updates the super
catalog
--no-super-update
do not update the SGML super catalog
-v –verbose
provide debug informations

Xmllint

Usage: `xmllint [options] XMLfiles ...`

Synopsis Unix-like command

Location Development:bin

Function Parse the XML files and output the result of the parsing

Inputs
--version : display the version of the XML library used
--debug : dump the nodes content when using --stream
--copy : used to test the internal copy implementation
--recover : output what was parsable on broken XML documents
--huge : remove any internal arbitrary parser limits
--noent : substitute entity references by their value
--noout : don't output the result tree
--path 'paths': provide a set of paths for resources
--load-trace : print trace of all external entites loaded
--nonet : refuse to fetch DTDs or entities over network
--nocompact : do not generate compact text nodes
--htmlout : output results as HTML
--nowrap : do not put HTML doc wrapper
--valid : validate the document in addition to std well-formed check
--postvalid : do a posteriori validation, i.e after parsing
--dtdvalid URL : do a posteriori validation against a given DTD
--dtdvalidfpi FPI : same but name the DTD with a Public Identifier
--timing : print some timings
--output file or -o file: save to a given file

--repeat : repeat 100 times, for timing or profiling
--insert : ad-hoc test for valid insertions
--compress : turn on gzip compression of output
--html : use the HTML parser
--xmlout : force to use the XML serializer when using --html
--nodefdtd : do not default HTML doctype
--push : use the push mode of the parser
--maxmem nbbytes : limits memory allocation to nbbytes bytes
--nowarning : do not emit warnings from parser/validator
--noblanks : drop (ignorable?) blanks spaces
--nocdata : replace cdata section with text nodes
--format : reformat/reindent the input
--encode encoding : output in the given encoding
--dropdtd : remove the DOCTYPE of the input docs
--pretty STYLE : pretty-print in a particular style
0 Do not pretty print
1 Format the XML content, as --format
2 Add whitespace inside tags, preserving content
--c14n : save in W3C canonical format v1.0 (with comments)
--c14n11 : save in W3C canonical format v1.1 (with comments)
--exc-c14n : save in W3C exclusive canonical format (with comments)
--nsclean : remove redundant namespace declarations
--testIO : test user I/O support
--catalogs : use SGML catalogs from $SGML_CATALOG_FILES otherwise XML Catalogs starting from file:///etc/xml/catalog are activated by default
--nocatalogs: deactivate all catalogs
--auto : generate a small doc on the fly
--xinclude : do XInclude processing
--noxincludenode : same but do not generate XInclude nodes
--nofixup-base-uris : do not fixup xml:base uris
--loaddtd : fetch external DTD
--dtdattr : loaddtd + populate the tree with inherited attributes
--stream : use the streaming interface to process very large files
--walker : create a reader and walk though the resulting doc
--pattern pattern_value : test the pattern support
--chkregister : verify the node registration code
--relaxng schema : do RelaxNG validation against the schema
--schema schema : do validation against the WXS schema
--schematron schema : do validation against a schematron
--sax1: use the old SAX1 interfaces for processing
--sax: do not build a tree but work just at the SAX level
--oldxml10: use XML-1.0 parsing rules before the 5th edition
--xpath expr: evaluate the XPath expression, inply --noout

Yes

Usage: `yes [STRING]...`

or: `yes OPTION`

Synopsis Unix-like command

Location Development:bin

Function Repeatedly output a line with all specified STRING(s), or `y'.

Inputs --help display this help and exit
--version output version information and exit