

Last updated on: 2020/01/30 03:28

Help and Documentation

Help on external commands

External commands are binary executables or scripts generally found in `/bin`, `/sbin`, `/usr/bin` ou `/usr/sbin` :

```
SLES12SP1:~ # type ifconfig
ifconfig is /sbin/ifconfig
```

Help for an external command can generally be obtained by using the **-help** option with the command in question :

```
SLES12SP1:~ # du --help | more
Usage: du [OPTION]... [FILE]...
  or:  du [OPTION]... --files0-from=F
Summarize disk usage of each FILE, recursively for directories.
```

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

```
-0, --null          end each output line with 0 byte rather than newline
-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 scale sizes by SIZE before printing them; e.g.,
                  '-BM' prints sizes in units of 1,048,576 bytes;
                  see SIZE format below
-b, --bytes        equivalent to '--apparent-size --block-size=1'
-c, --total        produce a grand total
-D, --dereference-args dereference only symlinks that are listed on the
```

```
command line
-d, --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
--files0-from=F      summarize disk usage of the
--More--
```

Use with certain commands, the **-help** option is not valid:

```
SLES12SP1:~ # type --help
-bash: type: --: invalid option
type: usage: type [-afptP] name [name ...]
```

Help on built-in commands

Commands such as **type**, **cd** or **umask** are internal to the shell:

```
SLES12SP1:~ # type type
type is a shell builtin
```

One of the internal Bash commands is **help**. Used with no argument, this command shows a list of all the internal commands:

```
SLES12SP1:~ # help | more
GNU bash, version 4.2.47(1)-release (x86_64-suse-linux-gnu)
These shell commands are defined internally. Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.
```

A star (*) next to a name means that the command is disabled.

```

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name[=value] ... ]
bg [job_spec ...]
bind [-lpvsPVS] [-m keymap] [-f filename] [-q >
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...) COMMANDS>
cd [-L|[-P [-e]]] [dir]
command [-pVv] command [arg ...]
--More--
history [-c] [-d offset] [n] or history -anrw>
if COMMANDS; then COMMANDS; [ elif COMMANDS; >
jobs [-lnprs] [jobspec ...] or jobs -x comman>
kill [-s sigspec | -n signum | -sigspec] pid >
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-n count] [-O origin] [-s count] [-t>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [-i text] [>
readarray [-n count] [-O origin] [-s count] [>
readonly [-aAf] [name[=value] ...] or readonl>

```

To get help concerning a given internal command, help is invoked with the given command as an argument:

```

SLES12SP1:~ # help type
type: type [-afptP] name [name ...]
  Display information about command type.
  For each NAME, indicate how it would be interpreted if used as a
  command name.
  Options:
  -a    display all locations containing an executable named NAME;
        includes aliases, builtins, and functions, if and only if
        the '-p' option is not also used
  -f    suppress shell function lookup
  -P    force a PATH search for each NAME, even if it is an alias,
        builtin, or function, and returns the name of the disk file
        that would be executed
  -p    returns either the name of the disk file that would be executed,
        or nothing if `type -t NAME' would not return `file'.

```

```
-t      output a single word which is one of `alias', `keyword',  
`function', `builtin', `file' or `', if NAME is an alias, shell  
reserved word, shell function, shell builtin, disk file, or not  
found, respectively
```

Arguments:

NAME Command name to be interpreted.

Exit Status:

Returns success if all of the NAMEs are found; fails if any are not found.

typeset: typeset [-aAfFgilrtux] [-p] name[=value] ...

Set variable values and attributes.

Obsolete. See `help declare'.

The man command

The **man** command is used to consult the manual of the command passed as an argument. For example **man help** under SLES 12 produces the following output:

BASH_BUILTINS(1)

General Commands Manual

BASH_BUILTINS(1)

NAME

bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, compopt, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, false, fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see bash(1)

BASH BUILTIN COMMANDS

Unless otherwise noted, each builtin command documented in this section as accepting options preceded by - accepts -- to signify the end of the options. The :, true, false, and test builtins do not accept options and do not treat -- specially. The exit, logout, break, continue, let, and shift builtins accept and process arguments

beginning with - without requiring --. Other builtins that accept arguments but are not specified as accepting options interpret arguments beginning with - as invalid options and require -- to prevent this interpretation.

: [arguments]

No effect; the command does nothing beyond expanding arguments and performing any specified redirections. A zero exit code is returned.

. filename [arguments]

Manual page help(1) line 1 (press h for help or q to quit)

Each manual page can contain several sections:

Section	Contents
NAME	This section is required and has a standardized format consisting of a comma-separated list of program or function names, followed by a dash, followed by a short description of the functionality the program (or function, or file) is supposed to provide. By using the makewhatis command, the name sections are inputted into the whatis database files.
SYNOPSIS	This section gives a short overview on available program options. For functions this section lists corresponding include files and the prototype so the programmer knows the type and number of arguments as well as the return type.
DESCRIPTION	This section describes how to use the command or function and what each argument does.
OPTIONS	This section gives a description of how each option affects program behaviour.
FILES	This section lists files the program or function uses.
EXAMPLE	This section gives examples of how to use the command or function.
ENVIRONMENT	This section lists all environment variables that affect the program or function and explains how they do so.
CONFORMING TO	This section lists the eventual standards that the command or function conforms to.
BUGS / TO DO	This section describes the limitations and known inconveniences of the command or function.
EXIT STATUS / RETURN VALUE	This section lists the exit status codes and their meaning.
SEE ALSO	This section provides a list of related man pages in alphabetical order.

Navigation within a manual is accomplished using the following keys:

Key	Function
Space Bar	Move forward one screen

Key	Function
↵ Enter	Move down one line
↑	Move up one line
↓	Move down one line
Page ↑	Move up (backwards) one half screen
Page ↓	Move down (forward) one half screen
Home	Move to the beginning of the manual
End	Move to the end of the manual
/	Search for the string that follows the / key. The n key then searches for the next occurrence whilst the N searches for the previous occurrence
Q	Quit the manual

A complete Linux manual is comprised of up to 9 sections:

Section	Contents
1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in /dev)
5	File formats and conventions eg /etc/passwd
6	Games
7	Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8	System administration commands (usually only for root)
9	Kernel routines [Non standard]

The available section for a specific command are given by the output of the **whereis** command:

```
SLES12SP1:~ # whereis passwd
passwd: /usr/bin/passwd /etc/passwd /etc/passwd.YaST2save /usr/share/man/man1/passwd.1ssl.gz
/usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.gz
```

To consult a specific section of a manual, the section number must be supplied as an argument to the man command:

```
$ man 5 passwd [Enter]
```

The **-k** option of the man command searches for the string supplied as an argument within the list of available manuals:

```
SLES12SP1:~ # man -k passwd
chpasswd (8) - update passwords in batch mode
Crypt::SmbHash (3pm) - Perl-only implementation of lanman and nt md4 hash functions, for use in...
fgetpwent_r (3) - get passwd file entry reentrantly
getpwent_r (3) - get passwd file entry reentrantly
gpasswd (1) - administer /etc/group
grub2-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
ldappasswd (1) - change the password of an LDAP entry
lppasswd (1) - add, change, or delete digest passwords.
opiepasswd (1) - Change or set a user's password for the OPIE authentication system.
pam_localuser (8) - require users to be listed in /etc/passwd
passwd (1) - change user password
passwd (1ssl) - compute password hashes
passwd (5) - password file
passwd2des (3) - RFS password encryption
saslpasswd2 (8) - set a user's sasl password
smbpasswd (5) - The Samba encrypted password file
smbpasswd (8) - change a user's SMB password
vncpasswd (1) - change the VNC password
yppasswd (1) - change your password in the NIS database
```

The output is a list of the available manuals complete with their description. The number between parentheses indicates the manual section number.

Command Line Switches

The switches associated with the **man** command are:

```
SLES12SP1:~ # man --help
```

Usage: man [OPTION...] [SECTION] PAGE...

-C, --config-file=FILE use this user configuration file
-d, --debug emit debugging messages
-D, --default reset all options to their default values
--warnings[=WARNINGS] enable warnings from groff

Main modes of operation:

-f, --whatis equivalent to whatis
-k, --apropos equivalent to apropos
-K, --global-apropos search for text in all pages
-l, --local-file interpret PAGE argument(s) as local filename(s)
-w, --where, --path, --location
print physical location of man page(s)
-W, --where-cat, --location-cat
print physical location of cat file(s)

-c, --catman used by catman to reformat out of date cat pages
-R, --recode=ENCODING output source page encoded in ENCODING

Finding manual pages:

-L, --locale=LOCALE define the locale for this particular man search
-m, --systems=SYSTEM use manual pages from other systems
-M, --manpath=PATH set search path for manual pages to PATH

-S, -s, --sections=LIST use colon separated section list

-e, --extension=EXTENSION limit search to extension type EXTENSION

-i, --ignore-case look for pages case-insensitively (default)
-I, --match-case look for pages case-sensitively

--regex show all pages matching regex
--wildcard show all pages matching wildcard

```
--names-only      make --regex and --wildcard match page names only,
                  not descriptions

-a, --all         find all matching manual pages
-u, --update      force a cache consistency check

--no-subpages     don't try subpages, e.g. 'man foo bar' => 'man
                  foo-bar'
```

Controlling formatted output:

```
-P, --pager=PAGER  use program PAGER to display output
-r, --prompt=STRING provide the `less' pager with a prompt
```

```
-7, --ascii       display ASCII translation of certain latin1 chars
-E, --encoding=ENCODING use selected output encoding
--no-hyphenation, --nh turn off hyphenation
--no-justification, --nj turn off justification
-p, --preprocessor=STRING STRING indicates which preprocessors to run:
                        e - [n]eqn, p - pic, t - tbl,
```

```
g - grap, r - refer, v - vgrind
```

```
-t, --troff        use groff to format pages
-T, --troff-device[=DEVICE] use groff with selected device
```

```
-H, --html[=BROWSER] use or BROWSER to display HTML output
-X, --gxditview[=RESOLUTION] use groff and display through gxditview
(X11):
-X = -TX75, -X100 = -TX100, -X100-12 = -TX100-12
-Z, --ditroff     use groff and force it to produce ditroff
```

```
-, --help        give this help list
--usage          give a short usage message
-V, --version     print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

Report bugs to cjwatson@debian.org.

The apropos command

apropos searches the **whatis** database for instances of a keyword passed to the command as an argument. With no further options, the output is identical to **man -k**:

```
SLES12SP1:~ # apropos passwd
chpasswd (8)          - update passwords in batch mode
Crypt::SmbHash (3pm) - Perl-only implementation of lanman and nt md4 hash functions, for use in Samba style
smbpasswd entries
fgetpwent_r (3)      - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
gpasswd (1)          - administer /etc/group
grub2-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
ldappasswd (1)       - change the password of an LDAP entry
lppasswd (1)         - add, change, or delete digest passwords.
opiepasswd (1)       - Change or set a user's password for the OPIE authentication system.
pam_localuser (8)    - require users to be listed in /etc/passwd
passwd (1)           - change user password
passwd (1ssl)        - compute password hashes
passwd (5)           - password file
passwd2des (3)       - RFS password encryption
saslpasswd2 (8)      - set a user's sasl password
smbpasswd (5)        - The Samba encrypted password file
smbpasswd (8)        - change a user's SMB password
vncpasswd (1)        - change the VNC password
yppasswd (1)         - change your password in the NIS database
```

Command Line Switches

The switches associated with the **apropos** command are:

```
SLES12SP1:~ # apropos --help
Usage: apropos [OPTION...] KEYWORD...

-d, --debug           emit debugging messages
-v, --verbose        print verbose warning messages
-e, --exact          search each keyword for exact match
-r, --regex          interpret each keyword as a regex
-w, --wildcard       the keyword(s) contain wildcards
-a, --and            require all keywords to match
-l, --long           do not trim output to terminal width
-C, --config-file=FILE use this user configuration file
-L, --locale=LOCALE  define the locale for this search
-m, --systems=SYSTEM use manual pages from other systems
-M, --manpath=PATH   set search path for manual pages to PATH
-s, --sections=LIST, --section=LIST
                    search only these sections (colon-separated)
-?, --help          give this help list
    --usage         give a short usage message
-V, --version       print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

The `--regex` option is enabled by default.

Report bugs to cjwatson@debian.org.

The mandb and whatis commands under SLES 12

In SLES 12, the database is maintained by root by using the `/bin/mandb` or `/usr/bin/mandb` command.

The use of **mandb** is very simple:

```
SLES12SP1:~ # mandb
mandb: warning: $MANPATH set, ignoring /etc/manpath.config
Purging old database entries in /usr/share/man/ca...
Processing manual pages under /usr/share/man/ca...
Purging old database entries in /usr/share/man/cs...
Processing manual pages under /usr/share/man/cs...
Purging old database entries in /usr/share/man/da...
Processing manual pages under /usr/share/man/da...
Purging old database entries in /usr/share/man/de...
Processing manual pages under /usr/share/man/de...
Purging old database entries in /usr/share/man/es...
Processing manual pages under /usr/share/man/es...
...
10 man subdirectories contained newer manual pages.
0 manual pages were added.
0 stray cats were added.
0 old database entries were purged.
```

Now the **whatis** command can be used to check which manual sections are available for a specific command:

```
SLES12SP1:~ # whatis passwd
passwd (1ssl)      - compute password hashes
passwd (1)         - change user password
passwd (5)         - password file
```

Command Line Switches

The switches associated with the **mandb** command are:

```
SLES12SP1:~ # mandb --help
Usage: mandb [OPTION...] [MANPATH]

-c, --create           create dbs from scratch, rather than updating
-C, --config-file=FILE use this user configuration file
-d, --debug           emit debugging messages
-f, --filename=FILENAME update just the entry for this filename
-p, --no-purge        don't purge obsolete entries from the dbs
-q, --quiet           work quietly, except for 'bogus' warning
-s, --no-straycats    don't look for or add stray cats to the dbs
-t, --test            check manual pages for correctness
-u, --user-db         produce user databases only
-?, --help            give this help list
    --usage           give a short usage message
-V, --version         print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

Report bugs to cjwatson@debian.org.

The switches associated with the **whatis** command are:

```
SLES12SP1:~ # whatis --help
Usage: whatis [OPTION...] KEYWORD...

-d, --debug           emit debugging messages
-v, --verbose         print verbose warning messages
-r, --regex           interpret each keyword as a regex
```

```
-w, --wildcard      the keyword(s) contain wildcards
-l, --long         do not trim output to terminal width
-C, --config-file=FILE use this user configuration file
-L, --locale=LOCALE define the locale for this search
-m, --systems=SYSTEM use manual pages from other systems
-M, --manpath=PATH  set search path for manual pages to PATH
-s, --sections=LIST, --section=LIST
                   search only these sections (colon-separated)
-?, --help         give this help list
  --usage          give a short usage message
-V, --version      print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

Report bugs to cjwatson@debian.org.

The info command

In addition to the man system, information regarding programs and commands can also be found using the **info system**. Background information, tutorials, and detailed information on using the program in question are sometimes lacking in man pages and as a result the info system was created to resolve that issue.

Within the info system, multiple individual pages of information make up the entire set of info pages for each specific application. These pages of information are referred to as nodes. You can page through nodes one at a time, or you can jump to specific pages through the use of hypertext links.

To help node navigation, info pages all display a header across the top of the page. This header includes information such as the current node, next and previous nodes, and information regarding the parent node. Navigation is accomplished using the following keys :

Key	Function
n	Next node.
p	Previous node.

Key	Function
u	Parent node.
Space	Scroll down one page at a time.
Del	Scroll up one page at a time.
b	Return to the beginning of the current node.
Tab ↵	Select next hypertext link.
m <link text>	Performs a direct jump to the specified subnode. Pressing Tab displays all available subnodes.
↵ Enter	Follow current hypertext link. Hypertext links are designated by an asterisk * at the beginning of the link, and a colon : at the end of the link.
q	Quit the info system.

To access the top node of the info system, use the following command :

```
SLES12SP1:~ # info
...
File: dir,      Node: Top      This is the top of the INFO tree

This (the Directory node) gives a menu of major topics.
Typing "q" exits, "?" lists all Info commands, "d" returns here,
"h" gives a primer for first-timers,
"mEmacs<Return>" visits the Emacs manual, etc.

In Emacs, you can click mouse button 2 on a menu item or cross reference
to select it.

* Menu:

Archiving
* Cpio: (cpio).          Copy-in-copy-out archiver to tape or disk.
* Shar utilities: (sharutils). Shell archiver, uuencode/uudecode.
* Tar: (tar).           Making tape (or disk) archives.

Basics
```

```
* Bash: (bash).           The GNU Bourne-Again SHell.
* Common options: (coreutils)Common options.
* Coreutils: (coreutils).   Core GNU (file, text, shell) utilities.
* Date input formats: (coreutils)Date input formats.
-----Info: (dir)Top, 321 lines --Top-----
Welcome to Info version 4.13. Type h for help, m for menu item.
```

Command Line Switches

The switches associated with the **info** command are:

```
SLES12SP1:~ # info --help
Usage: info [OPTION]... [MENU-ITEM...]

Read documentation in Info format.

Options:
  -k, --apropos=STRING      look up STRING in all indices of all manuals.
  -d, --directory=DIR      add DIR to INFOPATH.
  --dribble=FILENAME       remember user keystrokes in FILENAME.
  -f, --file=FILENAME      specify Info file to visit.
  -h, --help               display this help and exit.
  --index-search=STRING    go to node pointed by index entry STRING.
  -n, --node=NODENAME      specify nodes in first visited Info file.
  -o, --output=FILENAME    output selected nodes to FILENAME.
  -R, --raw-escapes        output "raw" ANSI escapes (default).
  --no-raw-escapes        output escapes as literal text.
  --restore=FILENAME       read initial keystrokes from FILENAME.
  -O, --show-options, --usage go to command-line options node.
  --subnodes               recursively output menu items.
  --vi-keys                use vi-like and less-like key bindings.
  --version                display version information and exit.
  -w, --where, --location  print physical location of Info file.
```

The first non-option argument, if present, is the menu entry to start from; it is searched for in all `dir' files along INFOPATH. If it is not present, info merges all `dir' files and shows the result. Any remaining arguments are treated as the names of menu items relative to the initial node visited.

For a summary of key bindings, type h within Info.

Examples:

```
info          show top-level dir menu
info info     show the general manual for Info readers
info info-std show the manual specific to this Info program
info emacs    start at emacs node from top-level dir
info emacs buffers start at buffers node within emacs manual
info --show-options emacs start at node with emacs' command line options
info --subnodes -o out.txt emacs dump entire manual to out.txt
info -f ./foo.info show file ./foo.info, not searching dir
```

Email bug reports to bug-texinfo@gnu.org,
general questions and discussion to help-texinfo@gnu.org.
Texinfo home page: <http://www.gnu.org/software/texinfo/>

<html>

Copyright © 2004-2018 Hugh Norris.

</html>
