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 :

```
root@ubuntu1604:~# 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 :

```
root@ubuntu1604:~# du --help | more
Usage: du [OPTION]... [FILE]...
  or:  du [OPTION]... --files0-from=F
Summarize disk usage of the set of FILEs, recursively for directories.

Mandatory arguments to long options are mandatory for short options too.
  -0, --null            end each output line with NUL, not 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:

```
root@ubuntu1604:~# type --help
-su: 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:

```
root@ubuntu1604:~# 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:

```
root@ubuntu1604:~# help | more
GNU bash, version 4.3.42(1)-release (x86_64-pc-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 [&]                                    history [-c] [-d offset] [n] or history ->
 (( expression ))                                if COMMANDS; then COMMANDS; [ elif COMMAN>
 . filename [arguments]                          jobs [-lnprs] [jobspec ...] or jobs -x co>
 :                                               kill [-s sigspec | -n signum | -sigspec] >
 [ arg... ]                                      let arg [arg ...]
 [[ expression ]]                                local [option] name[=value] ...
 alias [-p] [name[=value] ... ]                  logout [n]
 bg [job_spec ...]                               mapfile [-n count] [-O origin] [-s count]>
 bind [-lpsvPSVX] [-m keymap] [-f filename]>     popd [-n] [+N | -N]
 break [n]                                       printf [-v var] format [arguments]
 builtin [shell-builtin [arg ...]]               pushd [-n] [+N | -N | dir]
 caller [expr]                                   pwd [-LP]
 case WORD in [PATTERN [| PATTERN]...) COMM>     read [-ers] [-a array] [-d delim] [-i tex>
 cd [-L|[-P [-e]] [-@]] [dir]                    readarray [-n count] [-O origin] [-s coun>
 command [-pVv] command [arg ...]                readonly [-aAf] [name[=value] ...] or rea>
--More--
```

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

```
root@ubuntu1604:~# 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.
```

The man command

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

```
PASSWD(1)                                                                User Commands
PASSWD(1)

NAME
       passwd - change user password

SYNOPSIS
       passwd [options] [LOGIN]

DESCRIPTION
       The passwd command changes passwords for user accounts. A normal user may only change the password for
his/her own account, while the superuser may change
       the password for any account.  passwd also changes the account or associated password validity period.

   Password Changes
       The user is first prompted for his/her old password, if one is present. This password is then encrypted
and compared against the stored password. The user
       has only one chance to enter the correct password. The superuser is permitted to bypass this step so that
forgotten passwords may be changed.
```

```
       After the password has been entered, password aging information is checked to see if the user is permitted
to change the password at this time. If not,
       passwd refuses to change the password and exits.

       The user is then prompted twice for a replacement password. The second entry is compared against the first
and both are required to match in order for the
       password to be changed.

       Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to 8
characters including one or more characters from
       each of the following sets:

       ·   lower case alphabetics

       ·   digits 0 thru 9

       ·   punctuation marks

       Care must be taken not to include the system default erase or kill characters.  passwd will reject any
password which is not suitably complex.

   Hints for user passwords
       The security of a password depends upon the strength of the encryption algorithm and the size of the key
space. The legacy UNIX System encryption method is
       based on the NBS DES algorithm. More recent methods are now recommended (see ENCRYPT_METHOD). The size of
the key space depends upon the randomness of the
       password which is selected.

       Compromises in password security normally result from careless password selection or handling. For this
reason, you should not select a password which
       appears in a dictionary or which must be written down. The password should also not be a proper name, your
license number, birth date, or street address.
       Any of these may be used as guesses to violate system security.
```

```
       You can find advices on how to choose a strong password on http://en.wikipedia.org/wiki/Password_strength

OPTIONS
 Manual page passwd(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
↵ 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

Key	Function
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:

```
root@ubuntu1604:~# whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man5/passwd.5.gz /usr/share/man/man1/passwd.1.gz
/usr/share/man/man1/passwd.1ssl.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:

```
root@ubuntu1604:~# man -k passwd
```

```
chgpasswd (8)         - update group passwords in batch mode
chpasswd (8)          - update passwords in batch mode
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)        - get passwd file entry reentrantly
gpasswd (1)           - administer /etc/group and /etc/gshadow
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
pam_localuser (8)     - require users to be listed in /etc/passwd
passwd (1)            - change user password
passwd (1ssl)         - compute password hashes
passwd (5)            - the password file
passwd2des (3)        - RFS password encryption
update-passwd (8)     - safely update /etc/passwd, /etc/shadow and /etc/group
```

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:

```
root@ubuntu1604:~# 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 www-browser 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**:

```
root@ubuntu1604:~# apropos passwd
```

```
chgpasswd (8)         - update group passwords in batch mode
chpasswd (8)          - update passwords in batch mode
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)        - get passwd file entry reentrantly
gpasswd (1)           - administer /etc/group and /etc/gshadow
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
pam_localuser (8)     - require users to be listed in /etc/passwd
passwd (1)            - change user password
passwd (1ssl)         - compute password hashes
passwd (5)            - the password file
passwd2des (3)        - RFS password encryption
update-passwd (8)     - safely update /etc/passwd, /etc/shadow and /etc/group
```

Command Line Switches

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

```
root@ubuntu1604:~# 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 Ubuntu 16.04

In Ubuntu 16.04 the database is maintained by root by using the **/bin/mandb** or **/usr/bin/mandb** command.

Once again, the use of **mandb** is very simple:

```
root@ubuntu1604:~# mandb
Purging old database entries in /usr/share/man...
Processing manual pages under /usr/share/man...
Purging old database entries in /usr/share/man/lv...
Processing manual pages under /usr/share/man/lv...
Purging old database entries in /usr/share/man/pt_BR...
Processing manual pages under /usr/share/man/pt_BR...
Purging old database entries in /usr/share/man/tr...
Processing manual pages under /usr/share/man/tr...
Purging old database entries in /usr/share/man/ca...
Processing manual pages under /usr/share/man/ca...
...
0 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:

```
root@ubuntu1604:~# whatis passwd
passwd (1)           - change user password
passwd (1ssl)        - compute password hashes
passwd (5)           - the password file
```

Command Line Switches

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

```
root@ubuntu1604:~# 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:

```
root@ubuntu1604:~# 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.
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 :

```
root@ubuntu1604:~# info
...
File: dir,      Node: Top,      This is the top of the INFO tree.

This is the Info main menu (aka directory node).
A few useful Info commands:

  'q' quits;
  '?' lists all Info commands;
  'h' starts the Info tutorial;
  'mTexinfo RET' visits the Texinfo manual, etc.
```

```
* Menu:

Basics
* Common options: (coreutils)Common options.
* Coreutils: (coreutils).       Core GNU (file, text, shell) utilities.
* Date input formats: (coreutils)Date input formats.
* Ed: (ed).                     The GNU line editor
* File permissions: (coreutils)File permissions.
                                Access modes.
* Finding files: (find).        Operating on files matching certain criteria.

C++ libraries
-----Info: (dir)Top, 254 lines --Top-----------------------------------------------------
Welcome to Info version 6.1.  Type H for help, h for tutorial.
```

Command Line Switches

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

```
root@ubuntu1604:~# info --help
Usage: info [OPTION]... [MENU-ITEM...]

Read documentation in Info format.

Options:
  -a, --all                   use all matching manuals.
  -k, --apropos=STRING        look up STRING in all indices of all manuals.
  -d, --directory=DIR         add DIR to INFOPATH.
      --dribble=FILE          remember user keystrokes in FILENAME.
  -f, --file=MANUAL           specify Info manual 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=FILE            output selected nodes to FILE.
  -R, --raw-escapes            output "raw" ANSI escapes (default).
      --no-raw-escapes         output escapes as literal text.
      --restore=FILE           read initial keystrokes from FILE.
  -O, --show-options, --usage  go to command-line options node.
      --strict-node-location   (for debugging) use Info file pointers as-is.
      --subnodes               recursively output menu items.
  -v, --variable VAR=VALUE     assign VALUE to Info variable VAR.
      --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.
  -x, --debug=NUMBER           set debugging level (-1 for all).


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-stnd             show the manual specific to this Info program
  info emacs                 start at emacs node from top-level dir
  info emacs buffers         select buffers menu entry in emacs manual
  info emacs -n Files        start at Files node within emacs manual
  info '(emacs)Files'        alternative way to start at Files node
  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>

</html>