Version : **2023.01**

Updated : 2023/05/10 16:19

LCE403 - Help and Documentation

Contents

- **LCE403 - Help and Documentation**
  - Contents
  - Help on External Commands
  - Help on Built-In Commands
  - The man Command
  - The mandb and whatis Commands
  - The apropos Command
  - The info Command

Help on External Commands

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

```
[root@centos8 ~]# type ifconfig
ifconfig is /usr/sbin/ifconfig
```

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

```
[root@centos8 ~]# ifconfig --help
Usage:
  ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
```

```
[del <address>[/<prefixlen>]]
[[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
[netmask <address>]  [dstaddr <address>]  [tunnel <address>]
[outfill <NN>] [keepalive <NN>]
[hw <HW> <address>]  [mtu <NN>]
[[-]trailers]  [[-]arp]  [[-]allmulti]
[multicast]  [[-]promisc]
[mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
[txqueuelen <NN>]
[[-]dynamic]
[up|down] ...


<HW>=Hardware Type.
List of possible hardware types:
  loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
  slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adaptive Serial Line IP)
  ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
  netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
  ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
  arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
  sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
  irda (IrLAP) x25 (generic X.25) infiniband (InfiniBand)
  eui64 (Generic EUI-64)
<AF>=Address family. Default: inet
List of possible address families:
  unix (UNIX Domain) inet (DARPA Internet) inet6 (IPv6)
  ax25 (AMPR AX.25) netrom (AMPR NET/ROM) rose (AMPR ROSE)
  ipx (Novell IPX) ddp (Appletalk DDP) ash (Ash)
  x25 (CCITT X.25)
```

Help on Built-In Commands

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

```
[root@centos8 ~]# 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@centos8 ~]# help
GNU bash, version 4.4.19(1)-release (x86_64-redhat-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 -anrw [filename] or history -ps arg [arg...]
 (( expression ))                                                      if COMMANDS;
then COMMANDS; [ elif COMMANDS; then COMMANDS; ]... [ else COMMANDS; ] fi
 . filename [arguments]                                                jobs [-lnprs]
[jobspec ...] or jobs -x command [args]
 :                                                                     kill [-s
sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
 [ arg... ]                                                            let arg [arg
...]
 [[ expression ]]                                                      local [option]
name[=value] ...
 alias [-p] [name[=value] ... ]                                        logout [n]
 bg [job_spec ...]                                                     mapfile [-d
delim] [-n count] [-O origin] [-s count] [-t] [-u fd] [-C callback] [-c quantum] >
 bind [-lpsvPSVX] [-m keymap] [-f filename] [-q name] [-u name] [-r keyseq] [-x keyseq:shell-co>  popd [-n] [+N |
-N]
 break [n]                                                             printf [-v var]
```

```
format [arguments]                                                                          pushd [-n] [+N
 builtin [shell-builtin [arg ...]]
| -N | dir]
 caller [expr]                                                                              pwd [-LP]
 case WORD in [PATTERN [| PATTERN]...) COMMANDS ;;]... esac                                  read [-ers] [-a
array] [-d delim] [-i text] [-n nchars] [-N nchars] [-p prompt] [-t timeout] >
 cd [-L|[-P [-e]] [-@]] [dir]                                                               readarray [-n
count] [-O origin] [-s count] [-t] [-u fd] [-C callback] [-c quantum] [array]
 command [-pVv] command [arg ...]                                                           readonly [-aAf]
[name[=value] ...] or readonly -p
 compgen [-abcdefgjksuv] [-o option] [-A action] [-G globpat] [-W wordlist]  [-F function] [-C >   return [n]
 complete [-abcdefgjksuv] [-pr] [-DE] [-o option] [-A action] [-G globpat] [-W wordlist]  [-F f>   select NAME [in
WORDS ... ;] do COMMANDS; done
 compopt [-o|+o option] [-DE] [name ...]                                                    set [-
abefhkmnptuvxBCHP] [-o option-name] [--] [arg ...]
 continue [n]                                                                               shift [n]
 coproc [NAME] command [redirections]                                                       shopt [-pqsu]
[-o] [optname ...]
 declare [-aAfFgilnrtux] [-p] [name[=value] ...]                                            source filename
[arguments]
 dirs [-clpv] [+N] [-N]                                                                     suspend [-f]
 disown [-h] [-ar] [jobspec ... | pid ...]                                                  test [expr]
 echo [-neE] [arg ...]                                                                      time [-p]
pipeline
 enable [-a] [-dnps] [-f filename] [name ...]                                               times
 eval [arg ...]                                                                             trap [-lp]
[[arg] signal_spec ...]
 exec [-cl] [-a name] [command [arguments ...]] [redirection ...]                           true
 exit [n]                                                                                   type [-afptP]
name [name ...]
 export [-fn] [name[=value] ...] or export -p                                              typeset [-
aAfFgilnrtux] [-p] name[=value] ...
 false                                                                                      ulimit [-
SHabcdefiklmnpqrstuvxPT] [limit]
```

```
 fc [-e ename] [-lnr] [first] [last] or fc -s [pat=rep] [command]                        umask [-p] [-S]
[mode]
 fg [job_spec]                                                                           unalias [-a]
name [name ...]
 for NAME [in WORDS ... ] ; do COMMANDS; done                                            unset [-f] [-v]
[-n] [name ...]
 for (( exp1; exp2; exp3 )); do COMMANDS; done                                           until COMMANDS;
do COMMANDS; done
 function name { COMMANDS ; } or name () { COMMANDS ; }                                   variables -
Names and meanings of some shell variables
 getopts optstring name [arg]                                                            wait [-n] [id
...]
 hash [-lr] [-p pathname] [-dt] [name ...]                                               while COMMANDS;
do COMMANDS; done
 help [-dms] [pattern ...]                                                               { COMMANDS ; }
```

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

```
[root@centos8 ~]# 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:

```
[root@centos8 ~]# man passwd
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, return, break, continue,
let, and shift builtins accept and process arguments beginning with -  with-
       out  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. The return status is zero.

       . filename [arguments]
       source filename [arguments]
              Read and execute commands from filename in the current shell environment and return the exit status
of the last command executed from filename. If filename does not contain a
              slash, filenames in PATH are used to find the directory containing filename. The file searched
for in PATH need not be executable. When bash is not in posix mode, the cur-
              rent directory is searched if no file is found in PATH. If the sourcepath option to the shopt
builtin command is turned off, the PATH is not searched. If any arguments are
              supplied, they become the positional parameters when filename is executed. Otherwise the
positional parameters are unchanged. If the -T option is enabled, source inherits
              any trap on DEBUG; if it is not, any DEBUG trap string is saved and restored around the call to
source, and source unsets the DEBUG trap while it executes. If -T is not set,
              and the sourced file changes the DEBUG trap, the new value is retained when source completes. The
return status is the status of the last command exited within the script (0
              if no commands are executed), and false if filename is not found or cannot be read.

       alias [-p] [name[=value] ...]
              Alias with no arguments or with the -p option prints the list of aliases in the form alias
name=value on standard output. When arguments are supplied, an alias is defined for
              each name whose value is given. A trailing space in value causes the next word to be checked for
alias substitution when the alias is expanded. For each name in the argument
              list for which no value is supplied, the name and value of the alias is printed. Alias returns
true unless a name is given for which no alias has been defined.

       bg [jobspec ...]
              Resume each suspended job jobspec in the background, as if it had been started with &. If jobspec
is not present, the shell's notion of the current job is used.   bg jobspec
              returns 0 unless run when job control is disabled or, when run with job control enabled, any
specified jobspec was not found or was started without job control.

       bind [-m keymap] [-lpsvPSVX]

```
      bind [-m keymap] [-q function] [-u function] [-r keyseq]
      bind [-m keymap] -f filename
      bind [-m keymap] -x keyseq:shell-command
      bind [-m keymap] keyseq:function-name
      bind [-m keymap] keyseq:readline-command
          Display  current readline key and function bindings, bind a key sequence to a readline function or
macro, or set a readline variable.  Each non-option argument is a command as
          it would appear in .inputrc, but each binding or command must be passed as a separate argument;
e.g., '"\C-x\C-r": re-read-init-file'.  Options, if supplied, have the  follow-
 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
↵ 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:

```
[root@centos8 ~]# whereis passwd
passwd: /usr/bin/passwd /etc/passwd /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:

```
[root@centos8 ~]# man 5 passwd
PASSWD(5)                                                    Linux Programmer's Manual
PASSWD(5)


NAME
        passwd - password file


DESCRIPTION
        The /etc/passwd file is a text file that describes user login accounts for the system.  It should have
read permission allowed for all users (many utilities, like ls(1) use it to map
        user IDs to usernames), but write access only for the superuser.


        In the good old days there was no great problem with this general read permission.  Everybody could read
the encrypted passwords, but the hardware was too slow to crack a well-chosen
        password, and moreover the basic assumption used to be that of a friendly user-community.  These days many
people run some version of the shadow password suite, where /etc/passwd has
        an 'x' character in the password field, and the encrypted passwords are in /etc/shadow, which is readable
by the superuser only.


        If the encrypted password, whether in /etc/passwd or in /etc/shadow, is an empty string, login is allowed
without even asking for a password.  Note that  this  functionality  may  be
        intentionally disabled in applications, or configurable (for example using the "nullok" or "nonull"
arguments to pam_unix.so).


        If the encrypted password in /etc/passwd is "*NP*" (without the quotes), the shadow record should be
obtained from an NIS+ server.


        Regardless  of whether shadow passwords are used, many system administrators use an asterisk (*) in the
encrypted password field to make sure that this user can not authenticate him-
        or herself using a password.  (But see NOTES below.)


        If you create a new login, first put an asterisk (*) in the password field, then use passwd(1) to set it.


        Each line of the file describes a single user, and contains seven colon-separated fields:
```

```
          name:password:UID:GID:GECOS:directory:shell

     The field are as follows:

     name          This is the user's login name.  It should not contain capital letters.

     password      This is either the encrypted user password, an asterisk (*), or the letter 'x'.  (See
pwconv(8) for an explanation of 'x'.)

     UID           The privileged root login account (superuser) has the user ID 0.

     GID           This is the numeric primary group ID for this user.  (Additional groups for the user are
defined in the system group file; see group(5)).

     GECOS         This field (sometimes called the "comment field") is optional and used only for informational
purposes.  Usually, it contains the full username.  Some programs (for exam-
               ple, finger(1)) display information from this field.

               GECOS  stands for "General Electric Comprehensive Operating System", which was renamed to GCOS
when GE's large systems division was sold to Honeywell.  Dennis Ritchie has
               reported: "Sometimes we sent printer output or batch jobs to the GCOS machine.  The gcos field
in the password file was a place to stash the information for  the  $IDENT-
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

The mandb and whatis Commands

Each manual page has a short description available within it. These descriptions as well as the name of the manual are stored in the **whatis** database.

Under RHEL/CentOS 8, the database is maintained by root by using the **/usr/bin/mandb** command.

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

```
[root@centos8 ~]# mandb
Processing manual pages under /usr/share/man/overrides...
Updating index cache for path `/usr/share/man/overrides/man3'. Wait...done.
Checking for stray cats under /usr/share/man/overrides...
Checking for stray cats under /var/cache/man/overrides...
...
Processing manual pages under /usr/share/man/pt...
Purging old database entries in /usr/local/share/man...
Processing manual pages under /usr/local/share/man...
0 man subdirectories contained newer manual pages.
0 manual pages were added.
0 stray cats were added.
17 old database entries were purged.
```

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

```
[root@centos8 ~]# whatis passwd
openssl-passwd (1ssl) - compute password hashes
passwd (1)           - update user's authentication tokens
passwd (5)           - password file
```

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@centos8 ~]# 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
```

```
grub2-mkpasswd-pbkdf2 (1) - Generate a PBKDF2 password hash.
lpasswd (1)          - Change group or user password
openssl-passwd (1ssl) - compute password hashes
pam_localuser (8)    - require users to be listed in /etc/passwd
passwd (1)           - update user's authentication tokens
passwd (5)           - password file
passwd2des (3)       - RFS password encryption
pwhistory_helper (8) - Helper binary that transfers password hashes from passwd or shadow to opasswd
smbpasswd (5)        - The Samba encrypted password file
sslpasswd (1ssl)     - compute password hashes
```

> **Important** - Note that the numbers between parenthesis indicate the available sections.

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.

Key	Function
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@centos8 ~]# 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;
  'H' lists all Info commands;
  'h' starts the Info tutorial;
  'mTexinfo RET' visits the Texinfo manual, etc.


* Menu:

Archiving
* Cpio: (cpio).               Copy-in-copy-out archiver to tape or disk.
* 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.
* Ed: (ed).                      The GNU line editor
* File permissions: (coreutils)File permissions.
                                 Access modes.
* Finding files: (find).         Operating on files matching certain criteria.
* time: (time).                  GNU time Utility


Compression
-----Info: (dir)Top, 307 lines --Top------------------------------------------------------------------
-
Welcome to Info version 6.5.  Type H for help, h for tutorial.
```