

Version : **2024.01**

Last update : 2024/11/25 13:04

RH12403 - Help and Documentation

Contents

- **RH12403 - Help and Documentation**
 - Contents
 - LAB #1 - Help with Commands
 - 1.1 - Help on External Commands
 - 1.2 - Help on Built-In Commands
 - LAB #2 - The man command
 - 2.1 - The mandb and whatis commands
 - 2.2 - The apropos command
 - LAB #3 - The info command

LAB #1 - Help with Commands

1.1 - Help on External Commands

External shell commands are executable binaries or scripts, usually located in `/bin`, `/sbin`, `/usr/bin` or `/usr/sbin` :

```
[trainee@redhat9 ~]$ su -  
Password: fenestros  
[root@redhat9 ~]# type ifconfig  
ifconfig is /usr/sbin/ifconfig
```

The help of an external shell command can be viewed in most cases by passing the **-help** parameter as an argument to the command in question:

```
[root@redhat9 ~]# 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)

1.2 - Help on Built-In Commands

Built-In or internal shell commands are commands such as **type**, **cd** or **umask**. To check the type of command, use the **type** command:

```
[root@redhat9 ~]# type type
type is a shell builtin
```

The shell has the **help** command. Used alone, this command provides the list of built-in commands:

```
[root@redhat9 ~]# help
GNU bash, version 5.1.8(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 -anr>
(( expression )) if COMMANDS; then COMMANDS; [ elif COMMANDS;>
. filename [arguments] jobs [-lnprs] [jobspec ...] or jobs -x comma>
: kill [-s sigspec | -n signum | -sigspec] pid>
[ 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] [->
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]...) COMMAND> read [-ers] [-a array] [-d delim] [-i text] >
```

```

cd [-L|[-P [-e]] [-@]] [dir] readarray [-d delim] [-n count] [-O origin] >
command [-pVv] command [arg ...] readonly [-aAf] [name[=value] ...] or readon>
compgen [-abcdefghijklmnop] [-o option] [-A actio> return [n]
complete [-abcdefghijklmnop] [-pr] [-DEI] [-o opt> select NAME [in WORDS ... ;] do COMMANDS; do>
compopt [-o|+o option] [-DEI] [name ...] set [-abefhkmnptuvxBCHP] [-o option-name] [->
continue [n] shift [n]
coproc [NAME] command [redirections] shopt [-pqsu] [-o] [optname ...]
declare [-aAfFgiIlnrtux] [-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 [argument ...]]> true
exit [n] type [-afptP] name [name ...]
export [-fn] [name[=value] ...] or export -p typeset [-aAfFgiIlnrtux] [-p] name[=value] .>
false ulimit [-SHabcdefghijklmnopqrstuvxPT] [limit]
fc [-e ename] [-lnr] [first] [last] or fc -s > 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; don> until COMMANDS; do COMMANDS; done
function name { COMMANDS ; } or name () { COM> variables - Names and meanings of some shell>
getopts optstring name [arg ...] wait [-fn] [-p var] [id ...]
hash [-lr] [-p pathname] [-dt] [name ...] while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...] { COMMANDS ; }

```

Help on a specific command can be obtained by passing the command concerned as an argument to the **help** command:

```

[root@redhat9 ~]# 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.

LAB #2 - The man command

The man command provides access to the manual for the command passed as an argument. For example **man passwd** :

```
[root@redhat9 ~]# man passwd
```

```
PASSWD(1) User utilities PASSWD(1)
```

NAME

passwd - update user's authentication tokens

SYNOPSIS

```
passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n minddays] [-x maxdays] [-w warndays] [-i inactivedays] [-S] [--  
stdin] [-?] [--usage] [username]
```

DESCRIPTION

The `passwd` utility is used to update user's authentication token(s).

This task is achieved through calls to the Linux-PAM and Libuser API. Essentially, it initializes itself as a “`passwd`” service with Linux-PAM and utilizes configured password modules to authenticate and then update a user's password.

A simple entry in the global Linux-PAM configuration file for this service would be:

```
#
# passwd service entry that does strength checking of
# a proposed password before updating it.
#
passwd password requisite pam_cracklib.so retry=3
passwd password required pam_unix.so use_authtok
#
```

Note, other module types are not required for this application to function correctly.

OPTIONS

`-k, --keep-tokens`

The option `-k` is used to indicate that the update should only be for expired authentication tokens (passwords); the user wishes to keep their non-expired tokens as before.

`-l, --lock`

This option is used to lock the password of specified account and it is available to root only. The locking is performed by rendering the encrypted password into an invalid string (by prefixing the encrypted string with an `!`). Note that the account is not fully locked - the user can still log in by other means of authentication such as the ssh public key authentication. Use `chage -E 0 user` command instead for full account locking.

`--stdin`

This option is used to indicate that `passwd` should read the new password from standard input, which can be a pipe.

-u, --unlock

This is the reverse of the -l option - it will unlock the account password by removing the ! prefix. This option is available to root only. By default passwd will refuse to create a passwordless account (it will not unlock an account that has only "!" as a password). The force option -f will override this protection.

-d, --delete

This is a quick way to delete a password for an account. It will set the named account passwordless. Available to root only.

Note that if the password was locked, this implicitly removes the password lock as well.

-e, --expire

This is a quick way to expire a password for an account. The user will be forced to change the password during the next login attempt. Available to root only.

-f, --force

Force the specified operation.

-n, --minimum DAYS

This will set the minimum password lifetime, in days, if the user's account supports password lifetimes. Available to root only.

-x, --maximum DAYS

This will set the maximum password lifetime, in days, if the user's account supports password lifetimes. Available to root only.

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

A manual page can contain several sections:

Section	Contents
NAME	Name and role of the command
SYNOPSIS	Command syntax, parameters and arguments
DESCRIPTION	Instructions for use and main arguments

Section	Contents
OPTIONS	Detailed descriptions of each parameter
EXAMPLES	Examples of command use
ENVIRONMENT VARIABLES	Shell environment operation
STANDARDS / CONFORMING TO	Possible standards to which the command conforms
BUGS/TO DO	Known bugs, if any
EXIT STATUS/RETURN VALUE	Error codes and their meanings
SEE ALSO	Commands related to those in the current manual

The manual page is navigated by using certain 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 lowercase key then searches for the next occurrence whilst the N uppercase key searches for the previous occurrence
Q	Quit the manual

A complete manual is made up of several 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

Section	Contents
6	Games and Gadgets
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 various available sections can be viewed using the **whereis** command:

```
[root@redhat9 ~]# 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.1.gz
```

To view a specific section, specify its number:

```
[root@redhat9 ~]# man 5 passwd
passwd(5) File Formats 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 dis-

abled in applications, or configurable (for example using the “nullok” or “nonull” arguments to `pam_unix(8)`).

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 themselves 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 fields 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 example, `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 \$IDENTcard. Not elegant.”

directory This is the user's home directory: the initial directory where the user is placed after logging in. The value in this field is used to set the HOME environment variable.

shell This is the program to run at login (if empty, use /bin/sh). If set to a nonexistent executable, the user will be unable to login through login(1). The value in this field is used to set the SHELL environment variable.

FILES

/etc/passwd

NOTES

If you want to create user groups, there must be an entry in /etc/group, or no group will exist.

If the encrypted password is set to an asterisk (*), the user will be unable to login using login(1), but may still login using rlogin(1), run existing processes and initiate new ones through rsh(1), cron(8), at(1), or mail filters, etc. Trying to lock an account by simply changing the shell field yields the same result and additionally allows the use of su(1).

SEE ALSO

Manual page passwd(5) line 1 (press h for help or q to quit)

2.1 - The mandb and whatis commands

Each manual page contains a brief description. These descriptions and the name of the manual are stored in the **whatis** database.

This database can be maintained manually by root by invoking the **/usr/bin/mandb** executable:

```
[root@redhat9 ~]# mandb
Purging old database entries in /usr/share/man/overrides...
Processing manual pages under /usr/share/man/overrides...
```

```
Purging old database entries in /usr/share/man...
Processing manual pages under /usr/share/man...
Purging old database entries in /usr/share/man/overrides...
Processing manual pages under /usr/share/man/overrides...
Purging old database entries in /usr/share/man/ru...
Processing manual pages under /usr/share/man/ru...
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/...
Processing manual pages under /usr/share/man/de...
Purging old database entries in /usr/share/man/hu...
Processing manual pages under /usr/share/man/hu...
Purging old database entries in /usr/share/man/id...
Processing manual pages under /usr/share/man/id...
Purging old database entries in /usr/share/man/it...
Processing manual pages under /usr/share/man/it...
Purging old database entries in /usr/share/man/ja...
Processing manual pages under /usr/share/man/ja...
Purging old database entries in /usr/share/man/ko...
Processing manual pages under /usr/share/man/ko...
Purging old database entries in /usr/share/man/pl...
Processing manual pages under /usr/share/man/pl...
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/sv...
Processing manual pages under /usr/share/man/sv...
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/zh_CN...
Processing manual pages under /usr/share/man/zh_CN...
```

```
Purging old database entries in /usr/share/man/zh_TW...
Processing manual pages under /usr/share/man/zh_TW...
Purging old database entries in /usr/share/man/uk...
Processing manual pages under /usr/share/man/uk...
Purging old database entries in /usr/share/man/es...
Processing manual pages under /usr/share/man/es...
Purging old database entries in /usr/share/man/sk...
Processing manual pages under /usr/share/man/sk...
Purging old database entries in /usr/share/man/nl...
Processing manual pages under /usr/share/man/nl...
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/pt...
Processing manual pages under /usr/share/man/pt...
Purging old database entries in /usr/share/man/sr...
Processing manual pages under /usr/share/man/sr...
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.
24 old database entries were purged.
```

The **whatis** command can now be used to identify the manual sections available for a given command:

```
[root@redhat9 ~]# whatis passwd
passwd (5) - password file
passwd (1) - update user's authentication tokens
passwd (1ossl) - OpenSSL application commands
```

2.2 - The apropos command

The **apropos** command searches the **whatis** database for the character string passed as an argument to the command. Without options, the output is

identical to the **man -k** command:

```
[root@redhat9 ~]# apropos passwd
chgpaswd (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
gpaswd (1) - administer /etc/group and /etc/gshadow
grub2-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
lpaswd (1) - Change group or user password
openssl-paswd (1ossl) - compute password hashes
pam_localuser (8) - require users to be listed in /etc/passwd
passwd (1) - update user's authentication tokens
passwd (1ossl) - OpenSSL application commands
passwd (5) - password file
passwd2des (3) - RFS password encryption
pwhistory_helper (8) - Helper binary that transfers password hashes from passwd or shadow to opaswd
smbpasswd (5) - The Samba encrypted password file
```

The result is a list of commands followed by a brief description of them.



Important - Note that the numbers in brackets indicate the available sections.

LAB #3 - The info Command

In addition to the manual system, information about executables can be found in the **info** system. Detailed information, examples and tutorials may be missing from the manual system. For this reason, the **info** system was created.

In the **info** system, multiple pages of information about an executable, called nodes, are grouped together. Navigation between nodes is simple, using

a system of hypertext links.

To facilitate navigation, each page contains a header that includes information on the current node, the parent node, the previous node and the next node. To navigate between nodes, use 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 first node, use the following command:

```
[root@redhat9 ~]# 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

- * Gzip: (gzip). General (de)compression of files (lzw).

Editors

- * nano: (nano). Small and friendly text editor.

Encryption

- * Nettle: (nettle). A low-level cryptographic library.

GNU Gettext Utilities

- * autopoint: (gettext)autopoint Invocation.
Copy gettext infrastructure.
- * envsubst: (gettext)envsubst Invocation.
Expand environment variables.
- * gettextize: (gettext)gettextize Invocation.
Prepare a package for gettext.
- * gettext: (gettext). GNU gettext utilities.
- * ISO3166: (gettext)Country Codes.
ISO 3166 country codes.

```
* ISO639: (gettext)Language Codes.  
           ISO 639 language codes.  
* msgattrib: (gettext)msgattrib Invocation.  
           Select part of a PO file.  
* msgcat: (gettext)msgcat Invocation.  
           Combine several PO files.  
* msgcmp: (gettext)msgcmp Invocation.  
           Compare a PO file and template.  
* msgcomm: (gettext)msgcomm Invocation.  
           Match two PO files.  
* msgconv: (gettext)msgconv Invocation.  
-----Info: (dir)Top, 318 lines --Top-----  
-----  
Welcome to Info version 6.7.  Type H for help, h for tutorial.
```