1) NAME: pwd - print name of current/working directory

SYNOPSIS: pwd [OPTION]...

DESCRIPTION: Print the full filename of the current working directory.

-L, --logical: use PWD from environment, even if it contains symlinks

-P, --physical : avoid all symlinks

2) NAME: whoami - print effective userid

SYNOPSIS: whoami [OPTION]...

DESCRIPTION: Print the user name associated with the current effective user ID.

Same as id -un

3)NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

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

Sort entries alphabetically if none of -cftuvSUX nor --sort is speci‐

fied.

4) NAME

mkdir - make directories

SYNOPSIS

mkdir [OPTION]... DIRECTORY...

DESCRIPTION

Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options

too.

-m, --mode=MODE

set file mode (as in chmod), not a=rwx - umask

-p, --parents

no error if existing, make parent directories as needed

-v, --verbose

print a message for each created directory

-Z set SELinux security context of each created directory to the

default type

5) NAME

rmdir - remove empty directories

SYNOPSIS

rmdir [OPTION]... DIRECTORY...

DESCRIPTION

Remove the DIRECTORY(ies), if they are empty.

--ignore-fail-on-non-empty

ignore each failure that is solely because a directory

is non-empty

-p, --parents

remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is

similar to 'rmdir a/b/c a/b a'

-v, --verbose

output a diagnostic for every directory processed

6) NAME

mv - move (rename) files

SYNOPSIS

mv [OPTION]... [-T] SOURCE DEST

mv [OPTION]... SOURCE... DIRECTORY

mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION

Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options

too.

--backup[=CONTROL]

make a backup of each existing destination file

-b like --backup but does not accept an argument

-f, --force

do not prompt before overwriting

-i, --interactive

prompt before overwrite

7)NAME

touch - change file timestamps

SYNOPSIS

touch [OPTION]... FILE...

DESCRIPTION

Update the access and modification times of each FILE to the current

time.

A FILE argument that does not exist is created empty, unless -c or -h

is supplied.

A FILE argument string of - is handled specially and causes touch to

change the times of the file associated with standard output.

8)NAME

stat - display file or file system status

SYNOPSIS

stat [OPTION]... FILE...

DESCRIPTION

Display file or file system status

9)NAME

uname - print system information

SYNOPSIS

uname [OPTION]...

DESCRIPTION

Print certain system information. With no OPTION, same as -s.

-a, --all

print all information, in the following order, except omit -p

and -i if unknown:

-s, --kernel-name

print the kernel name

-n, --nodename

print the network node hostname

10) NAME

grep, egrep, fgrep, rgrep - print lines that match patterns

SYNOPSIS

grep [OPTION...] PATTERNS [FILE...]

grep [OPTION...] -e PATTERNS ... [FILE...]

grep [OPTION...] -f PATTERN\_FILE ... [FILE...]

DESCRIPTION

grep searches for PATTERNS in each FILE. PATTERNS is one or more

patterns separated by newline characters, and grep prints each line

that matches a pattern. Typically PATTERNS should be quoted when grep

is used in a shell command.

A FILE of “-” stands for standard input. If no FILE is given,

recursive searches examine the working directory, and nonrecursive

searches read standard input.

In addition, the variant programs egrep, fgrep and rgrep are the same

as grep -E, grep -F, and grep -r, respectively. These variants are

deprecated, but are provided for backward compatibility.

11)NAME

cat - concatenate files and print on the standard output

SYNOPSIS

cat [OPTION]... [FILE]...

DESCRIPTION

Concatenate FILE(s) to standard output.

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

-A, --show-all

equivalent to -vET

-b, --number-nonblank

number nonempty output lines, overrides -n

-e equivalent to -vE

12)NAME

cut - remove sections from each line of files

SYNOPSIS

cut OPTION... [FILE]...

DESCRIPTION

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

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

13)NAME

paste - merge lines of files

SYNOPSIS

paste [OPTION]... [FILE]...

DESCRIPTION

Write lines consisting of the sequentially corresponding lines from

each FILE, separated by TABs, to standard output.

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

14) NAME

sort - sort lines of text files

SYNOPSIS

sort [OPTION]... [FILE]...

sort [OPTION]... --files0-from=F

DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

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

15) NAME

uniq - report or omit repeated lines

SYNOPSIS

uniq [OPTION]... [INPUT [OUTPUT]]

DESCRIPTION

Filter adjacent matching lines from INPUT (or standard input), writing

to OUTPUT (or standard output).

With no options, matching lines are merged to the first occurrence.

16)NAME

tr - translate or delete characters

SYNOPSIS

tr [OPTION]... SET1 [SET2]

DESCRIPTION

Translate, squeeze, and/or delete characters from standard input,

writing to standard output.

17)NAME

sed - stream editor for filtering and transforming text

SYNOPSIS

sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION

Sed is a stream editor. A stream editor is used to perform basic text

transformations on an input stream (a file or input from a pipeline).

While in some ways similar to an editor which permits scripted edits

(such as ed), sed works by making only one pass over the input(s), and

is consequently more efficient. But it is sed's ability to filter

text in a pipeline which particularly distinguishes it from other

types of editors.

18) NAME

awk - pattern scanning and text processing language

SYNOPSIS

mawk [-W option] [-F value] [-v var=value] [--] 'program text' [file

...]

mawk [-W option] [-F value] [-v var=value] [-f program-file] [--]

[file ...]

DESCRIPTION

mawk is an interpreter for the AWK Programming Language. The AWK lan‐

guage is useful for manipulation of data files, text retrieval and

processing, and for prototyping and experimenting with algorithms.

mawk is a new awk meaning it implements the AWK language as defined in

Aho, Kernighan and Weinberger, The AWK Programming Language, Addison-

Wesley Publishing, 1988 (hereafter referred to as the AWK book.) mawk

conforms to the POSIX 1003.2 (draft 11.3) definition of the AWK lan‐

guage which contains a few features not described in the AWK book, and

mawk provides a small number of extensions

19) NAME

find - search for files in a directory hierarchy

SYNOPSIS

find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [ex‐

pression]

DESCRIPTION

This manual page documents the GNU version of find. GNU find searches

the directory tree rooted at each given starting-point by evaluating

the given expression from left to right, according to the rules of

precedence (see section OPERATORS), until the outcome is known (the

left hand side is false for and operations, true for or), at which

point find moves on to the next file name. If no starting-point is

specified, `.' is assumed.

20)NAME

bc - An arbitrary precision calculator language

SYNTAX

bc [ -hlwsqv ] [long-options] [ file ... ]

DESCRIPTION

bc is a language that supports arbitrary precision numbers with inter‐

active execution of statements. There are some similarities in the

syntax to the C programming language. A standard math library is

available by command line option. If requested, the math library is

defined before processing any files. bc starts by processing code

from all the files listed on the command line in the order listed.

After all files have been processed, bc reads from the standard input.

All code is executed as it is read. (If a file contains a command to

halt the processor, bc will never read from the standard input.)

21) NAME

xargs - build and execute command lines from standard input

SYNOPSIS

xargs [options] [command [initial-arguments]]

DESCRIPTION

This manual page documents the GNU version of xargs. xargs reads

items from the standard input, delimited by blanks (which can be pro‐

tected with double or single quotes or a backslash) or newlines, and

executes the command (default is /bin/echo) one or more times with any

initial-arguments followed by items read from standard input. Blank

lines on the standard input are ignored.

22) NAME

ps - report a snapshot of the current processes.

SYNOPSIS

ps [options]

DESCRIPTION

ps displays information about a selection of the active processes. If

you want a repetitive update of the selection and the displayed

information, use top(1) instead.

This version of ps accepts several kinds of options:

1 UNIX options, which may be grouped and must be preceded by a dash.

2 BSD options, which may be grouped and must not be used with a

dash.

3 GNU long options, which are preceded by two dashes.

23)NAME

nice - run a program with modified scheduling priority

SYNOPSIS

nice [OPTION] [COMMAND [ARG]...]

DESCRIPTION

Run COMMAND with an adjusted niceness, which affects process schedul‐

ing. With no COMMAND, print the current niceness. Niceness values

range from -20 (most favorable to the process) to 19 (least favorable

to the process)

24)NAME

renice - alter priority of running processes

SYNOPSIS

renice [-n] priority [-g|-p|-u] identifier...

DESCRIPTION

renice alters the scheduling priority of one or more running pro‐

cesses. The first argument is the priority value to be used. The

other arguments are interpreted as process IDs (by default), process

group IDs, user IDs, or user names. renice'ing a process group causes

all processes in the process group to have their scheduling priority

altered. renice'ing a user causes all processes owned by the user to

have their scheduling priority altered.

25)NAME

kill - send a signal to a process

SYNOPSIS

kill [options] <pid> [...]

DESCRIPTION

The default signal for kill is TERM. Use -l or -L to list available

signals. Particularly useful signals include HUP, INT, KILL, STOP,

CONT, and 0. Alternate signals may be specified in three ways: -9,

-SIGKILL or -KILL. Negative PID values may be used to choose whole

process groups; see the PGID column in ps command output. A PID of -1

is special; it indicates all processes except the kill process itself

and init

26)NAME

chown - change file owner and group

SYNOPSIS

chown [OPTION]... [OWNER][:[GROUP]] FILE...

chown [OPTION]... --reference=RFILE FILE...

DESCRIPTION

This manual page documents the GNU version of chown. chown changes

the user and/or group ownership of each given file. If only an owner

(a user name or numeric user ID) is given, that user is made the owner

of each given file, and the files' group is not changed. If the owner

is followed by a colon and a group name (or numeric group ID), with no

spaces between them, the group ownership of the files is changed as

well. If a colon but no group name follows the user name, that user

is made the owner of the files and the group of the files is changed

to that user's login group. If the colon and group are given, but the

owner is omitted, only the group of the files is changed; in this

case, chown performs the same function as chgrp. If only a colon is

given, or if the entire operand is empty, neither the owner nor the

group is changed

27)NAME

chmod - change file mode bits

SYNOPSIS

chmod [OPTION]... MODE[,MODE]... FILE...

chmod [OPTION]... OCTAL-MODE FILE...

chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION

This manual page documents the GNU version of chmod. chmod changes

the file mode bits of each given file according to mode, which can be

either a symbolic representation of changes to make, or an octal num‐

ber representing the bit pattern for the new mode bits.

The format of a symbolic mode is [ugoa...][[-+=][perms...]...], where

perms is either zero or more letters from the set rwxXst, or a single

letter from the set ugo. Multiple symbolic modes can be given, sepa‐

rated by commas

28)NAME

chroot - run command or interactive shell with special root directory

SYNOPSIS

chroot [OPTION] NEWROOT [COMMAND [ARG]...]

chroot OPTION

DESCRIPTION

Run COMMAND with root directory set to NEWROOT.

--groups=G\_LIST

specify supplementary groups as g1,g2,..,gN

--userspec=USER:GROUP

specify user and group (ID or name) to use

--skip-chdir

do not change working directory to '/'

29) NAME

chgrp - change group ownership

SYNOPSIS

chgrp [OPTION]... GROUP FILE...

chgrp [OPTION]... --reference=RFILE FILE...

DESCRIPTION

Change the group of each FILE to GROUP. With --reference, change the

group of each FILE to that of RFILE.

-c, --changes

like verbose but report only when a change is made

-f, --silent, --quiet

suppress most error messages

-v, --verbose

output a diagnostic for every file processed