Linux Commands

1. cd (Change Directory) Command

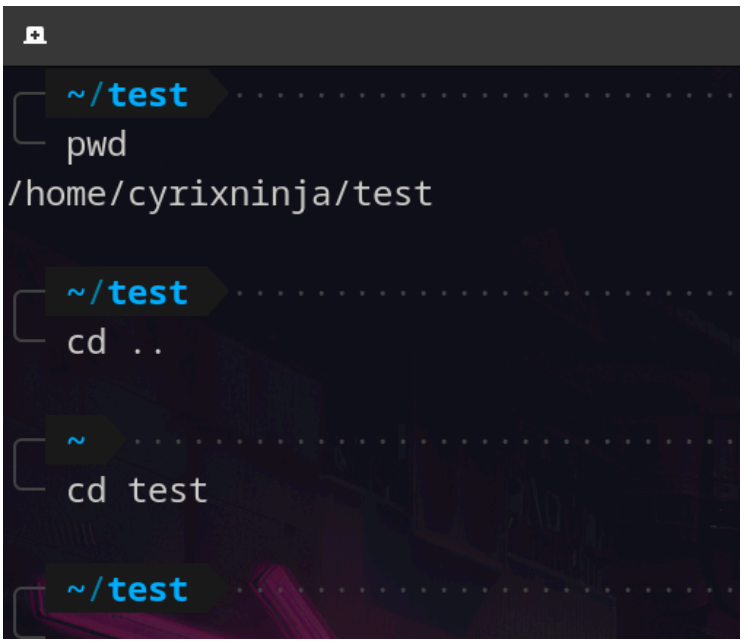
The cd command is used to change the current directory (i.e., the directory in which the user is currently working)

Syntax :

cd [-Options] [Directory]

Example :

Option	Use
cd ..	Change Current directory to parent directory
cd ~	Move to users home directory from anywhere
cd lab_1	Change from current working directory to lab_1
cd ../downloads	If we are currently in /home/username/documents then we would be placed in /home/username/downloads.



2. ls Command

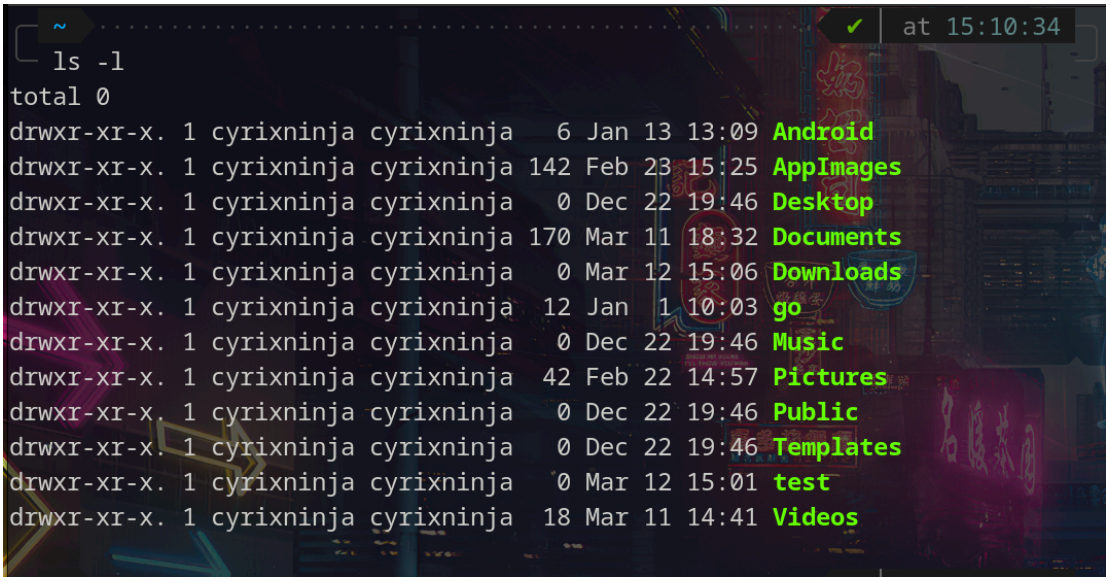
List directory contents.

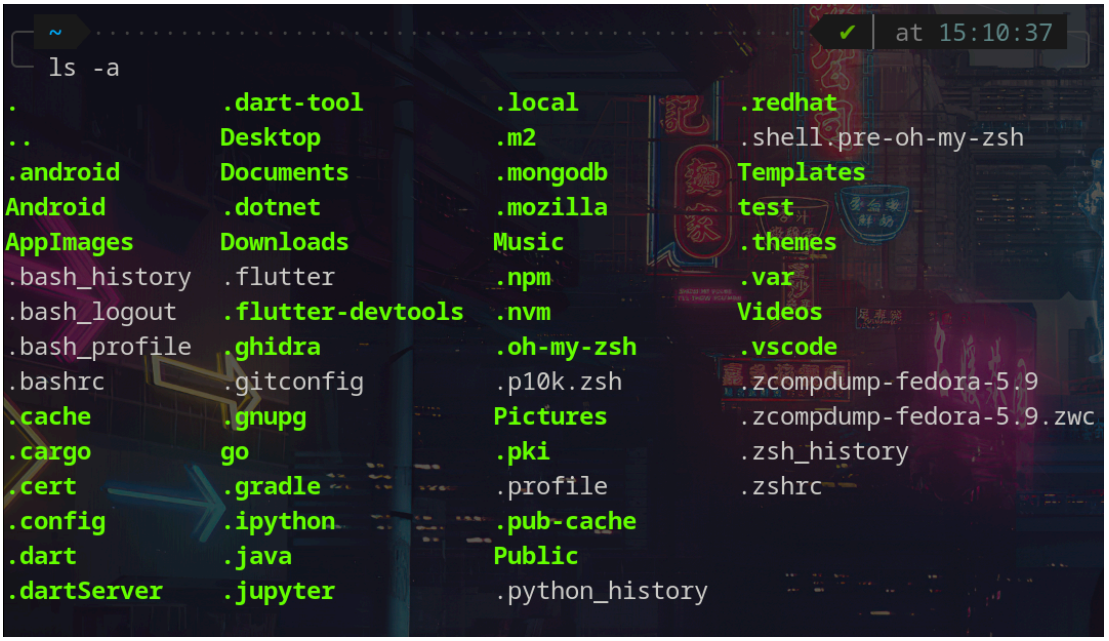
Syntax :

ls [Options] [file|dir]

Example :

Option	Use
ls -l	To show long listing information about the file/directory
ls -a	List all files including hidden file starting with '.'
ls -r	List in reverse order
ls -t	Sort by time & date
ls -s	Sort by file size





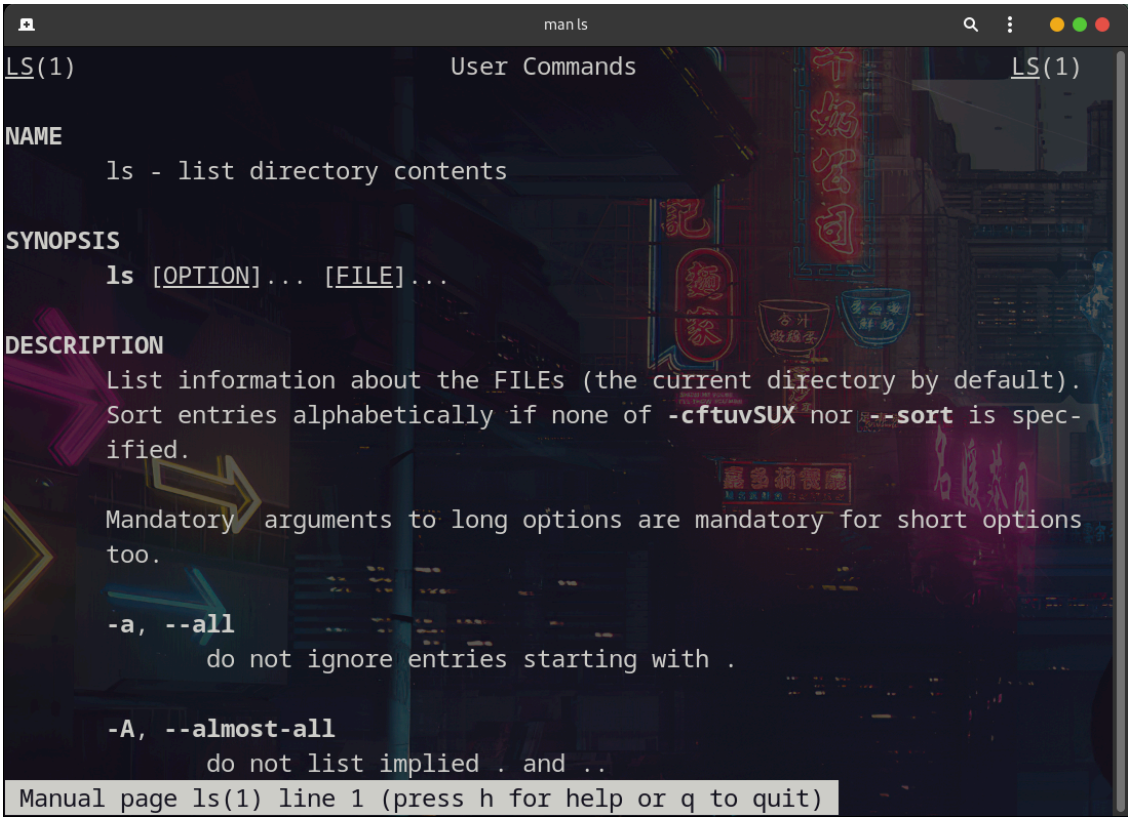
3.man Command

It is the interface used to view the system's reference manuals.

Syntax :

man [command name]

Example:



4.echo Command

Display a line of text/string on standard output or a file.

Syntax :

echo [option] [string]

Example :

Option	**Use**
echo -n	Do not output a trailing newline
echo -e	Enable interpretation of backslash escape sequences

Option	Use
\b	It removes all the spaces in between the text
\n	It creates new line from where it is used
\n	It create horizontal tab spaces

5.cal Command

Displays a simple, formatted calendar in your terminal.

Syntax :

cal [options] [[[day] month] year]

Example :

Option	Use
cal -1	Display single month output
cal -3	Display three months spanning the date.
cal -s	Display Sunday as the first day of the week.
cal -m	Display Monday as the first day of the week.
cal -j	Use day-of-year numbering for all

	calendars. These are also called ordinal days. Ordinal days range from 1 to 366.
cal –y	Display a calendar for the whole year

6.date Command

Print or set the system date and time.

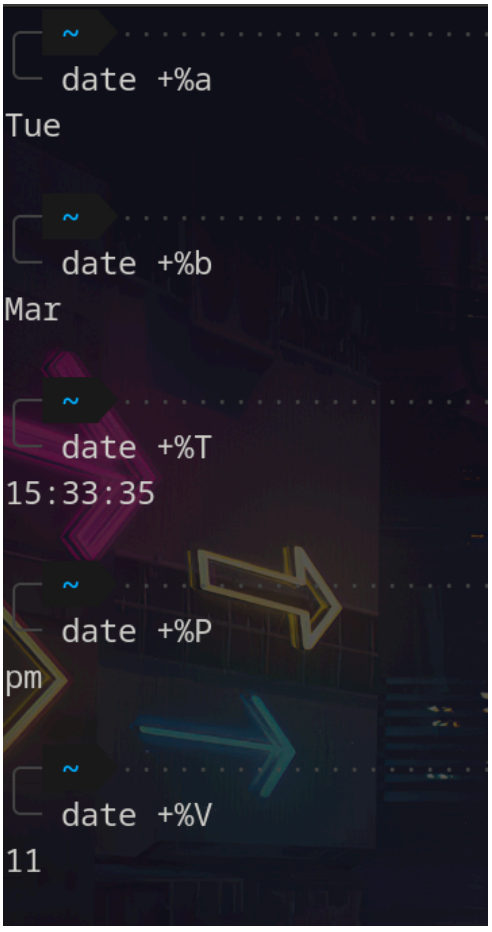
Syntax :

date [OPTION]... [+FORMAT]

Example :

Option	**Use**
date +%a	The abbreviated weekday name (e.g., Sun)
date +%A	The full weekday name (e.g., Sunday)
date +%b	The abbreviated month name (e.g., Jan)
date +%B	Locale's full month name

date +%C	The current century; like %Y, except omit last two digits (e.g., 20)
date +%w	day of week (0..6); 0 is Sunday
date +%d	Display the day of the month
date +%m	Displays the month of year (01 to 12)
date +%y	Displays last two digits of the year(00 to 99)
date +%Y	Display four-digit year.
date +%T	Display the time in 24 hour format as HH:MM:SS
date +%H	Display the hour
date +%M	Display the minute
date +%S	Display the seconds
date +%V	ISO week number, with Monday as first day of week (01..53)
date +%P	locale's equivalent of either AM or PM



7.clear Command

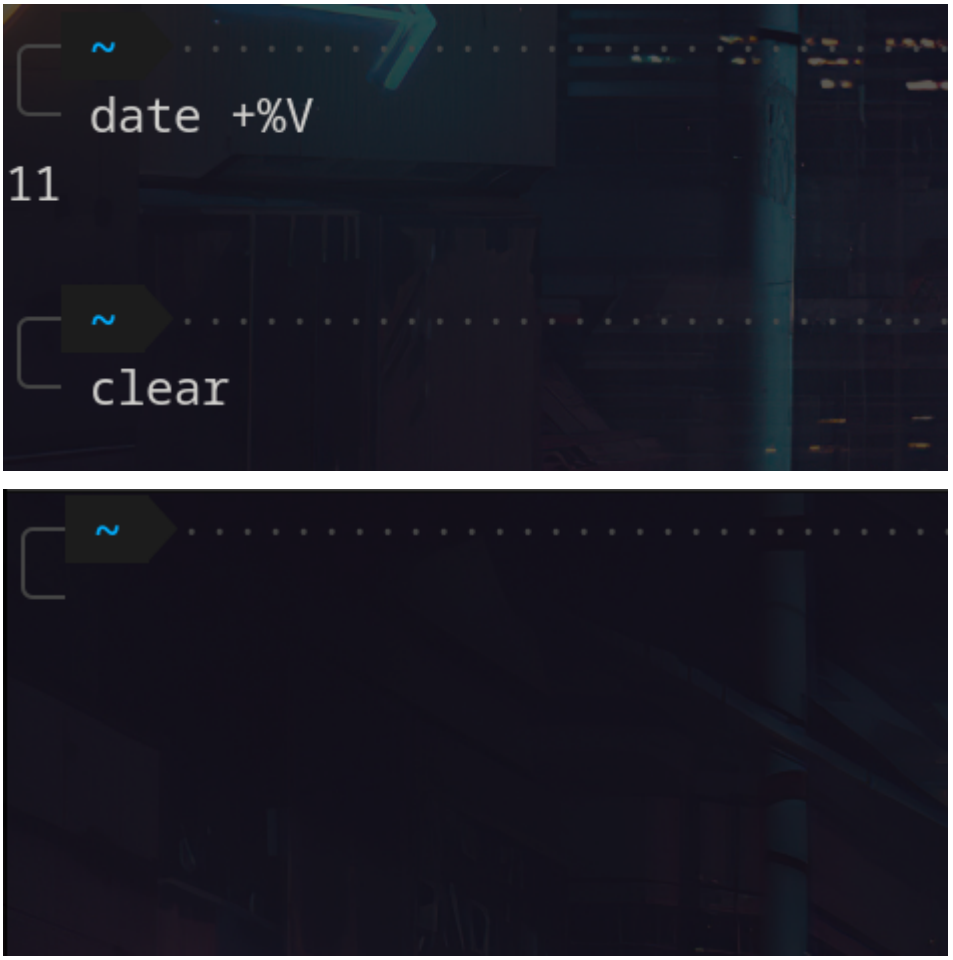
Clear the terminal screen.

If you take a detailed look after running the clear command, you'll find that it doesn't really clear the terminal. The tool just shifts the text upwards, out of the viewable area.

Syntax :

Clear

Example :



8.cat Command

It is used to create, display and concatenate file contents.

Syntax :

cat [OPTION] [FILE]

Example :

Option	**Use**
cat -b	Omits line numbers for blank space in the output
cat -E	Displays a $ (dollar sign) at the end of

	each line
cat -n	Line numbers for all the output lines
cat -s	Suppress repeated empty output lines
cat -T	Displays the tab characters as ^I in the output

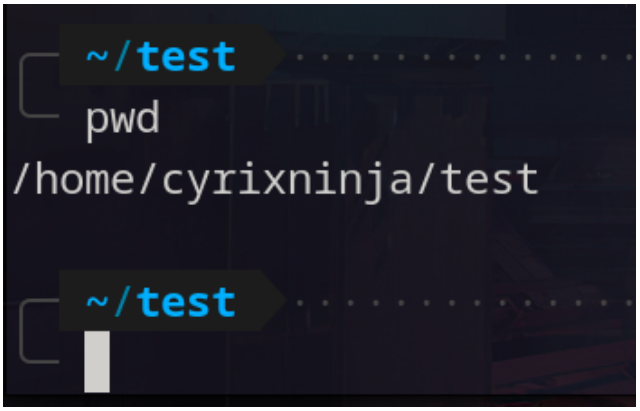
9.pwd (Print working directory) Command

It prints the current working directory name with the complete path starting from root (/).

Syntax :

pwd [-OPTION]

Example :



10.who Command

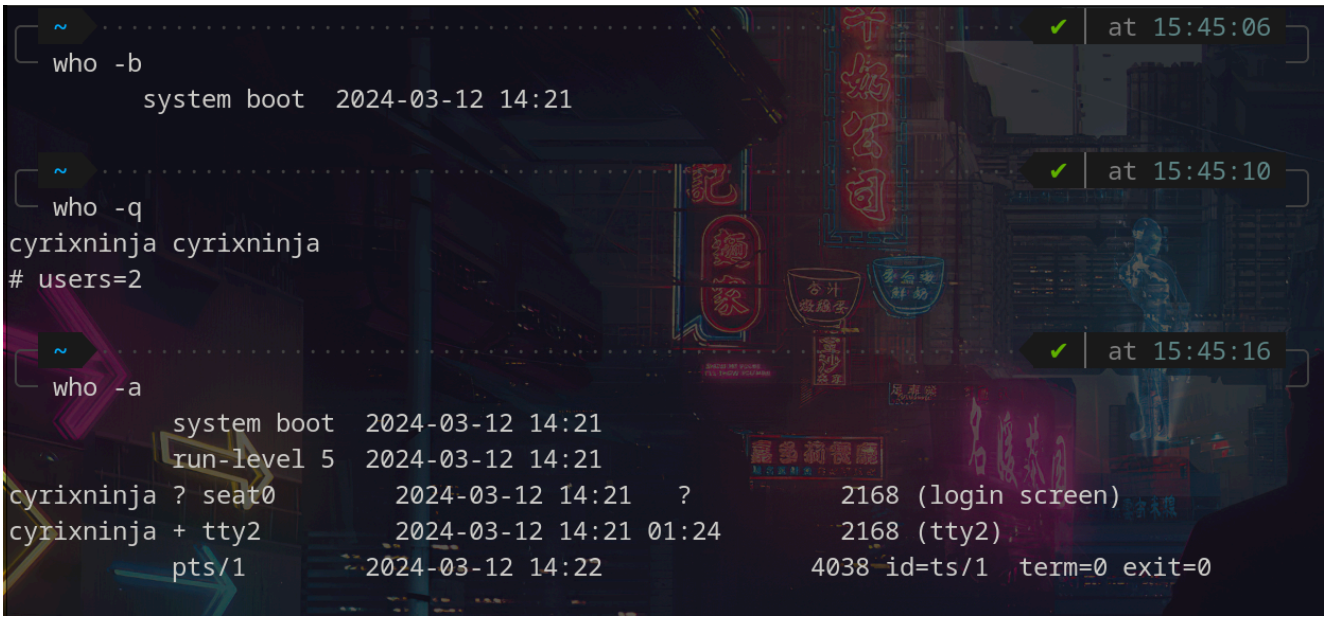
It display the users that are currently logged into your Unix computer system.

Syntax :

who [-options] [filename]

Example :

Option	Use
who -b	Display the time of the last system boot
who -h	Print a line of column headings
who -q	Displays all login names, and a count of all logged-on users
who -a	Display all details of current logged in user



11.whoami Command

This command prints the username associated with the current effective user ID.

Syntax :

whoami [-OPTION]

Example :

Option	Use
whoami --help	Display a help message, and exit
whoami --version	Display version information, and exit

12.uname (unix name) Command

Print information about the current system.

Syntax :

uname [-OPTION]

Example :

Option	Use
uname -s	Print the kernel name
uname -n	Print the network node hostname

uname -v	Print the kernel version
uname -m	Print the machine hardware name
uname -o	Print the operating system

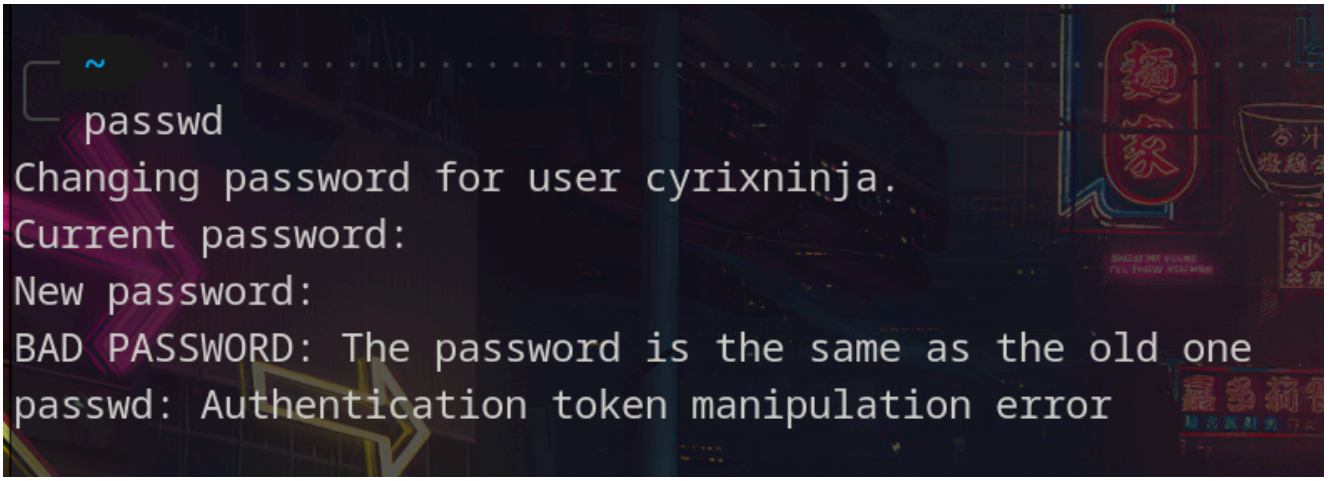
13.passwd Command

The passwd command is used to change the password of a user account.

Syntax :

passwd [-options] [username]

Example :



14.mkdir Command

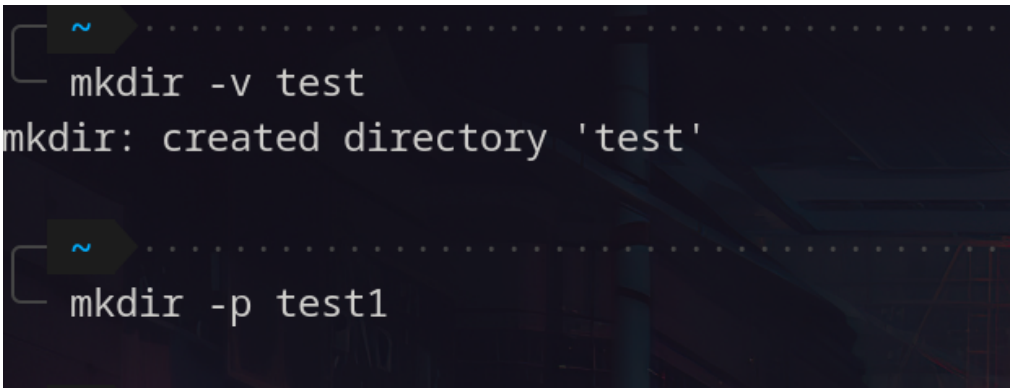
This command is used to make Directories.

Syntax :

mkdir [-OPTION] DIRECTORY

Example :

Option	Use
mkdir -v	Print a message for each created directory
mkdir -p	No error if existing, make parent directories as needed
mkdir -m	To control the permissions of new directories



15.rmdir Command

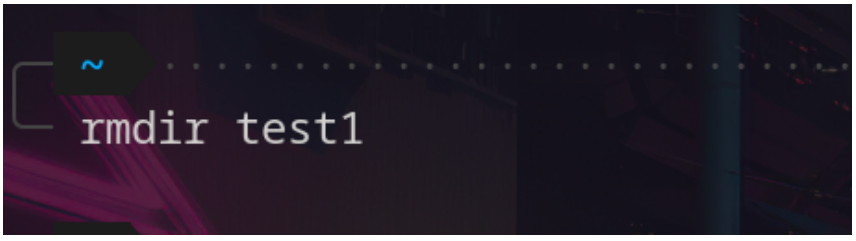
This command removes empty directories from your filesystem.

Syntax :

rmdir [-OPTION] DIRECTORY

Example :

Option	Use
rmdir	Remove directory and its ancestors... e.g., ‘rmdir -p a/b/c’ is similar to ‘rmdir a/b/c a/b a’



16.cp(copy) Command

This command is used to copy files and directories.

Syntax :

cp [option] source destination/directory

Example :

Option	Use
cp -i	Interactive - ask before overwrite
cp -f	Force copy by removing the destination file if needed
cp -n	Do not overwrite an existing file
cp -u	Update - copy when source is newer than destination
cp -s	Make symbolic links instead of copying
cp -R	Copy directories recursively
cp -v	Print informative messages

17.mv(move) Command

mv command is used to move files and directories.

Syntax :

mv [-options] source dest

Example :

Option	Use
mv -i	Interactive prompt before overwrite
mv -f	Force move by overwriting destination file without prompt
mv -n	Never overwrite any existing file
mv -u	Update - move when source is newer than destination
mv -v	Print informative messages

18.rm(remove) Command

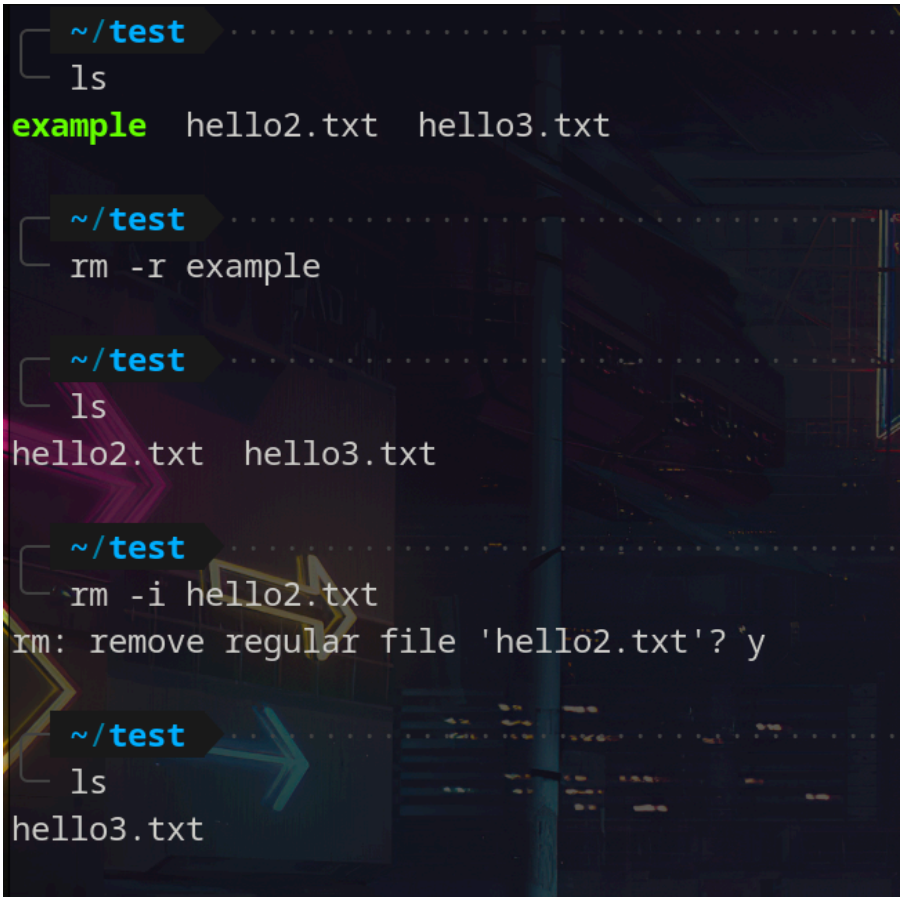
The ‘rm’ command is used to delete files and directories.

Syntax :

rm [-OPTION] Filename

Example :

Option	Use
rm -i	Prompt before every removal
rm -d	Delete a empty directory
rm -r	Remove directories and their contents recursively
rm -f	To remove the file forcefully



19.cut Command

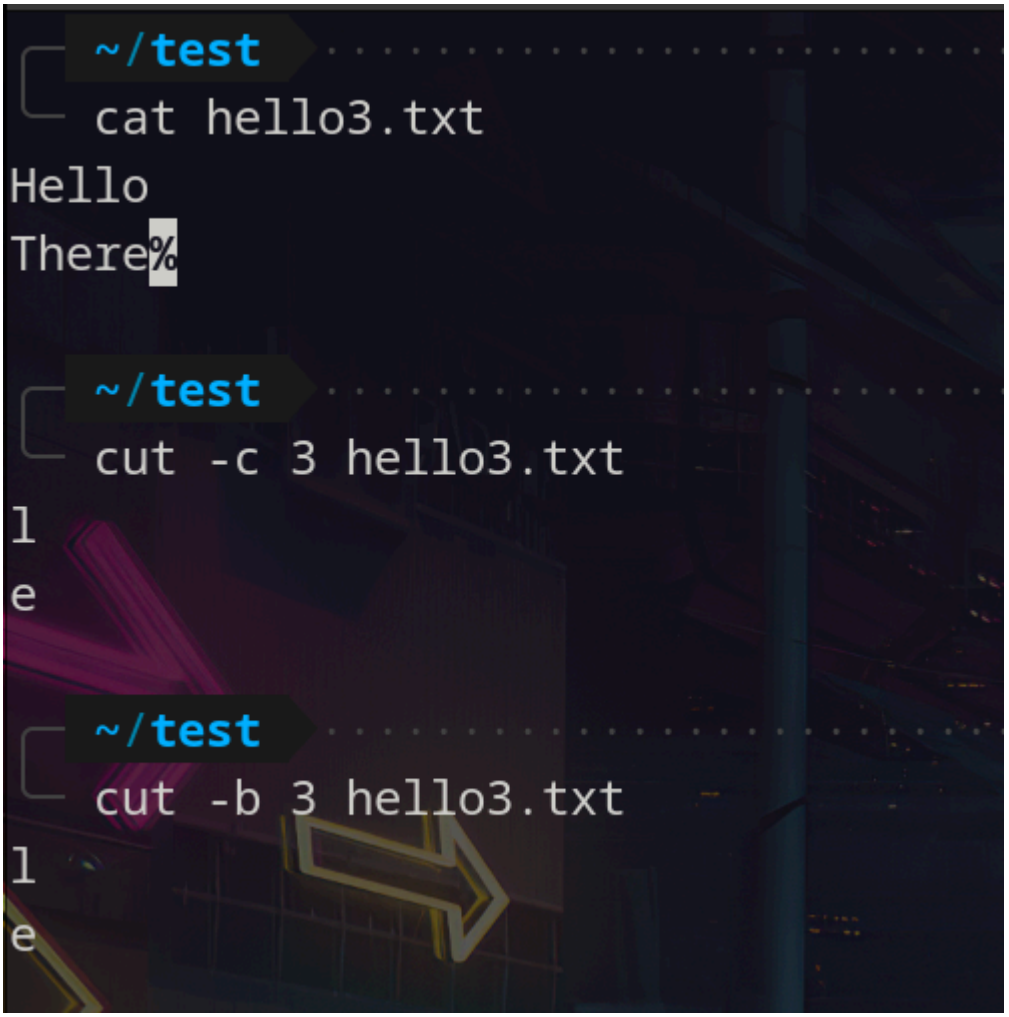
The cut command extracts a given number of characters or columns from a file.

Syntax :

cut [-options] [file]

Example :

Option	Use
cut -c	Select only the characters from each line as specified in LIST
cut -b	Select only the bytes from each line as specified in LIST
cut -f	Cuts the input file using a list of fields. The default field to be used TAB. The default behaviour can be overwritten by use of -d option
cut -c	Specifies a delimiter to be used as a field. Default field is TAB and this option overwrites this default behaviour



20.paste Command

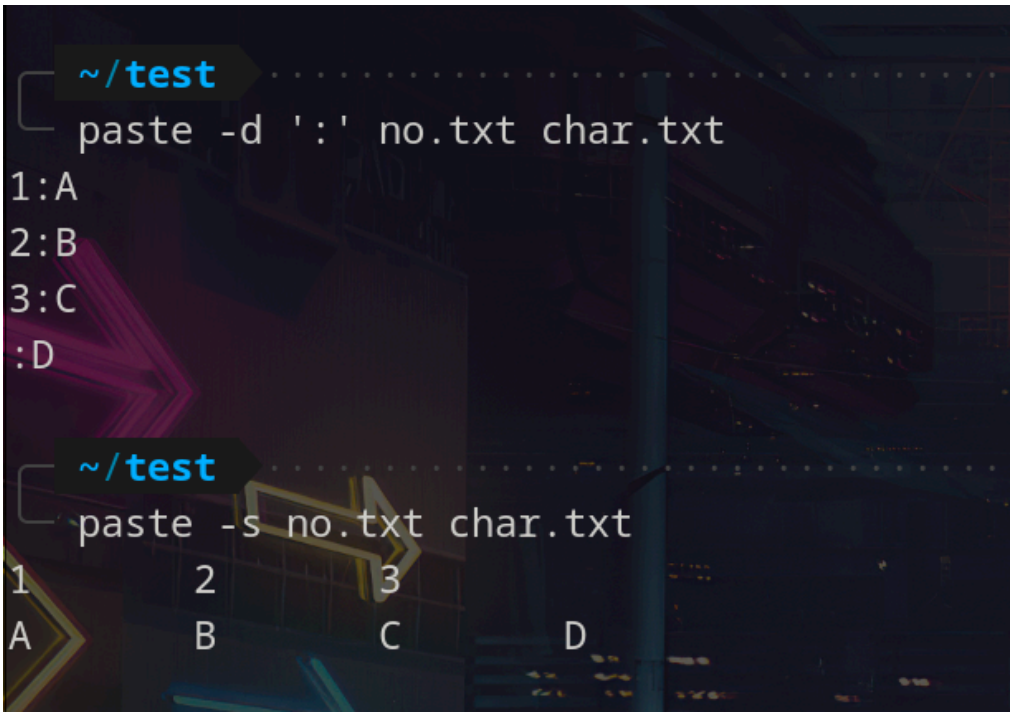
The paste command displays the corresponding lines of multiple files side-by-side.

Syntax :

paste [-options] [file]

Example :

Option	Use
paste -d	Reuse characters from LIST instead of tabs
paste -s	Paste one file at a time instead of in parallel



21. more Command

The more command is a command line utility for viewing the contents of a file or files once screen at a time.

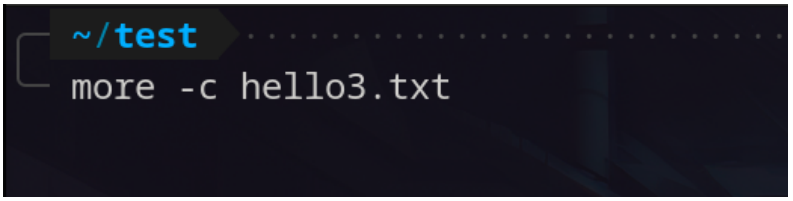
Syntax :

more [-options] [file]

Example :

Option	Use
more -c	Clear screen before displaying
more -number	To Specify how many lines are printed

	in the screen for a given file
more -s	Doesn’t display extra blank lines



22.cmp Command

cmp command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.

If a difference is found, it reports the byte and line number where the first difference is found.

If no differences are found, by default, cmp returns no output.

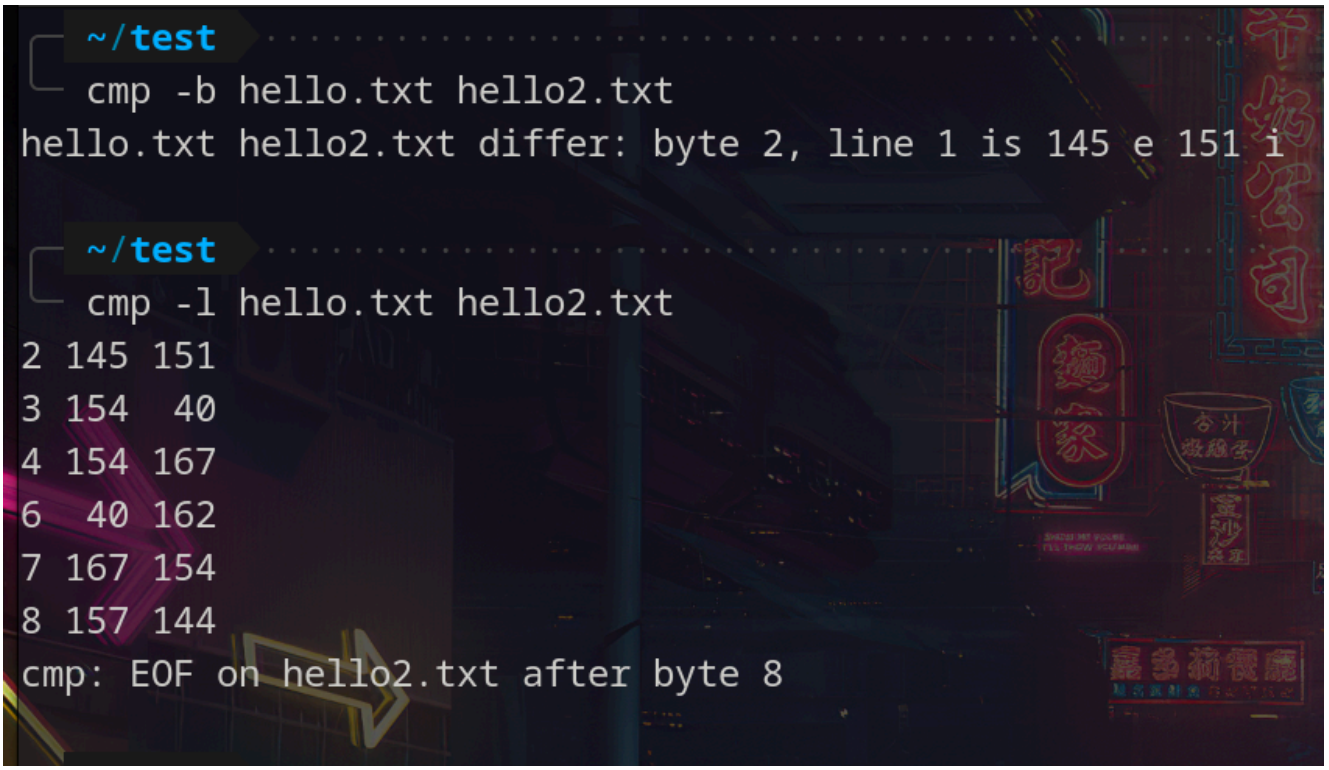
Syntax :

cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]

Example :

Option	**Use**
cmp -b	Print differing bytes
cmp -i	Skip a particular number of initial bytes from both the files
cmp -s	Do not print anything; only return an exit status indicating whether the files differ

cmp -n	Compare at most LIMIT bytes
cmp -l	Print byte position and byte value for all differing bytes



23.comm Command

Compare two sorted files line by line.

Syntax :

comm [OPTION]... FILE1 FILE2

Example :

Option	**Use**
comm -1	Suppress column 1 (lines unique to FILE1)
comm -2	Suppress column 2 (lines unique to FILE2)
comm -3	Suppress column 3 (lines that appear in both files)

24.diff(difference) Command

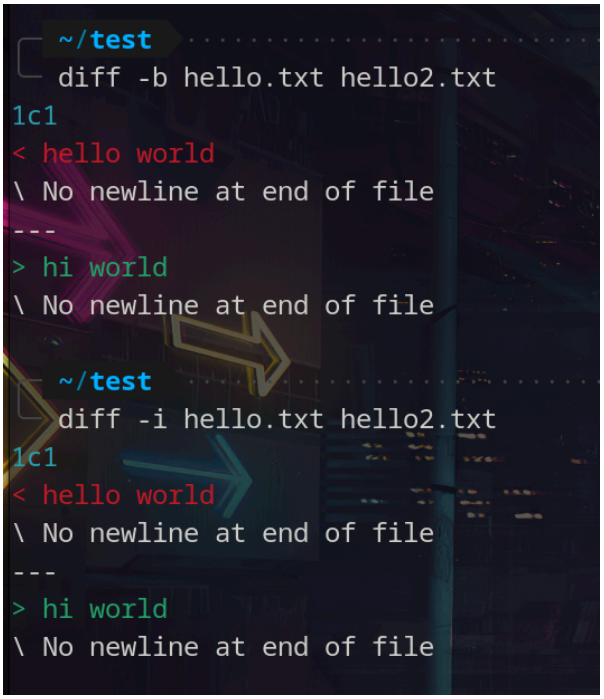
This command is used to display the differences in the files by comparing the files line by line.Diff analyses two files and prints the lines that are different. Essentially, it outputs a set of instructions for how to change one file to make it identical to the second file.

Syntax :

diff [options] File1 File2

Example :

Option	Use
diff -b	Ignores spacing differences
diff -i	Ignores case



25.chmod(change mode) Command

chmod is used to change the permissions of files or directories.

Syntax :

chmod [reference][operator][mode] file...

Example :

Reference	Class	Description
u	owner	file's owner
g	group	users who are members of the file's group
o	others	users who are neither the file's owner nor members of the file's group
a	all	All three of the above

Operator	Description
+	Adds the specified modes to the specified classes
-	Removes the specified modes from the specified classes
=	The modes specified are to be made the exact modes for the specified classes

Permission	Description
r	Permission to read the file
w	Permission to write (or delete) the file
x	Permission to execute the file, or, in the case of a directory, search it

```
~/test
chmod +x hello.sh

~/test
chmod 764 hello.txt
```

```
~/test
ls -l
total 24
-rw-r--r--. 1 cyrixninja cyrixninja  8 Mar 12 16:24 char.txt
-rw-r--r--. 1 cyrixninja cyrixninja  8 Mar 12 16:33 hello2.txt
-rw-r--r--. 1 cyrixninja cyrixninja 11 Mar 12 16:05 hello3.txt
-rwxr-xr-x. 1 cyrixninja cyrixninja 19 Mar 12 16:49 hello.sh
-rwxrw-r--. 1 cyrixninja cyrixninja 11 Mar 12 16:33 hello.txt
-rw-r--r--. 1 cyrixninja cyrixninja  6 Mar 12 16:23 no.txt
```

26.chown(change owner) Command

The chown command changes ownership of files and directories in a Linux filesystem.

Syntax :

chown [OPTIONS] USER[:GROUP] FILE(s)

Example :

```
~/test
sudo chown root example
[sudo] password for cyrixninja:

~/test
ls -l
total 24
-rw-r--r--. 1 cyrixninja cyrixninja  8 Mar 12 16:24 char.txt
drwxr-xr-x. 1 root       cyrixninja  0 Mar 12 16:53 example
-rw-r--r--. 1 cyrixninja cyrixninja  8 Mar 12 16:33 hello2.txt
-rw-r--r--. 1 cyrixninja cyrixninja 11 Mar 12 16:05 hello3.txt
-rwxr-xr-x. 1 cyrixninja cyrixninja 19 Mar 12 16:49 hello.sh
-rwxrw-r--. 1 cyrixninja cyrixninja 11 Mar 12 16:33 hello.txt
-rw-r--r--. 1 cyrixninja cyrixninja  6 Mar 12 16:23 no.txt
```

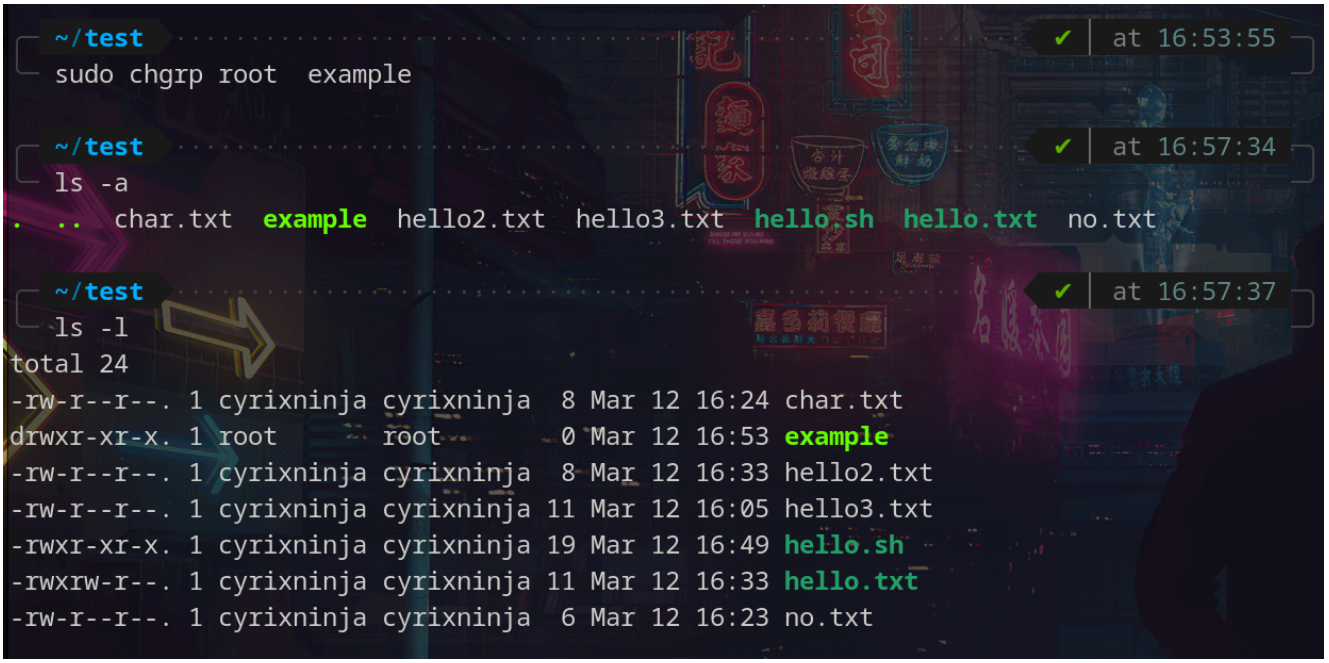
27.chgrp(change group) Command

The chgrp command is used to change group ownership of a file/directory.

Syntax :

chgrp [OPTION]... GROUP FILE/DIR…

Example :



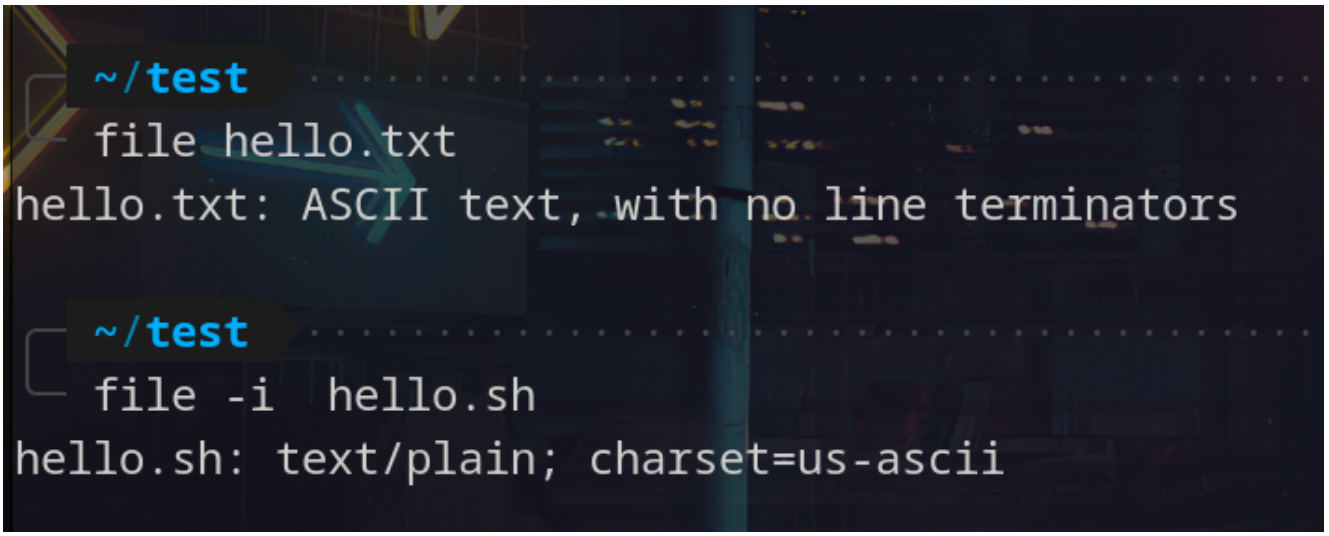
28.file Command

The file command is used to determine a file's type.

Syntax :

file [OPTIONS] file1 file2 ...

Example :



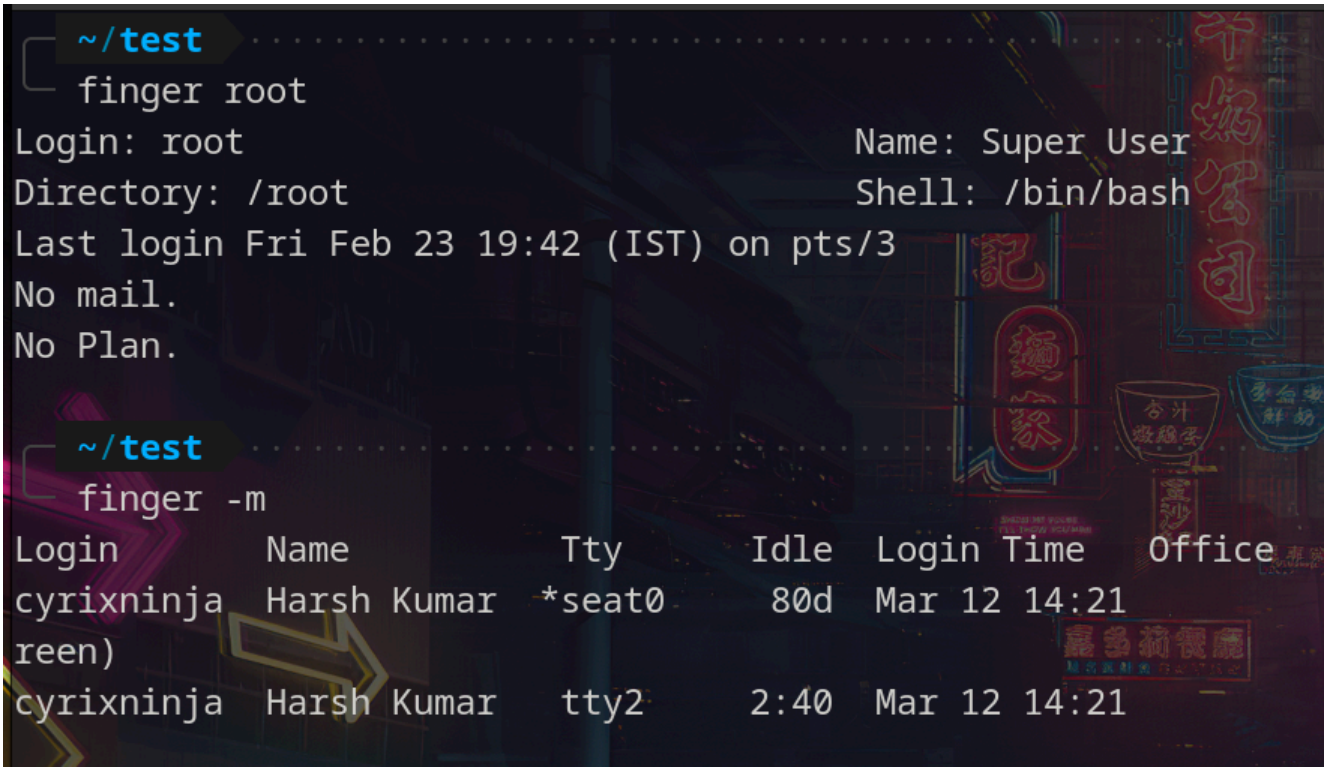
29.finger Command

finger looks up and displays information about system users.

Syntax :

finger [-option] [username]

Example :



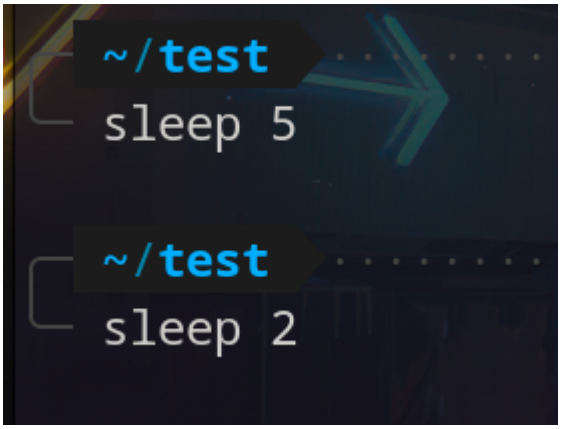
30.sleep Command

The sleep command is used to delay for a specified amount of time.

Syntax :

sleep NUMBER[SUFFIX]...

Example :



31.ps Command

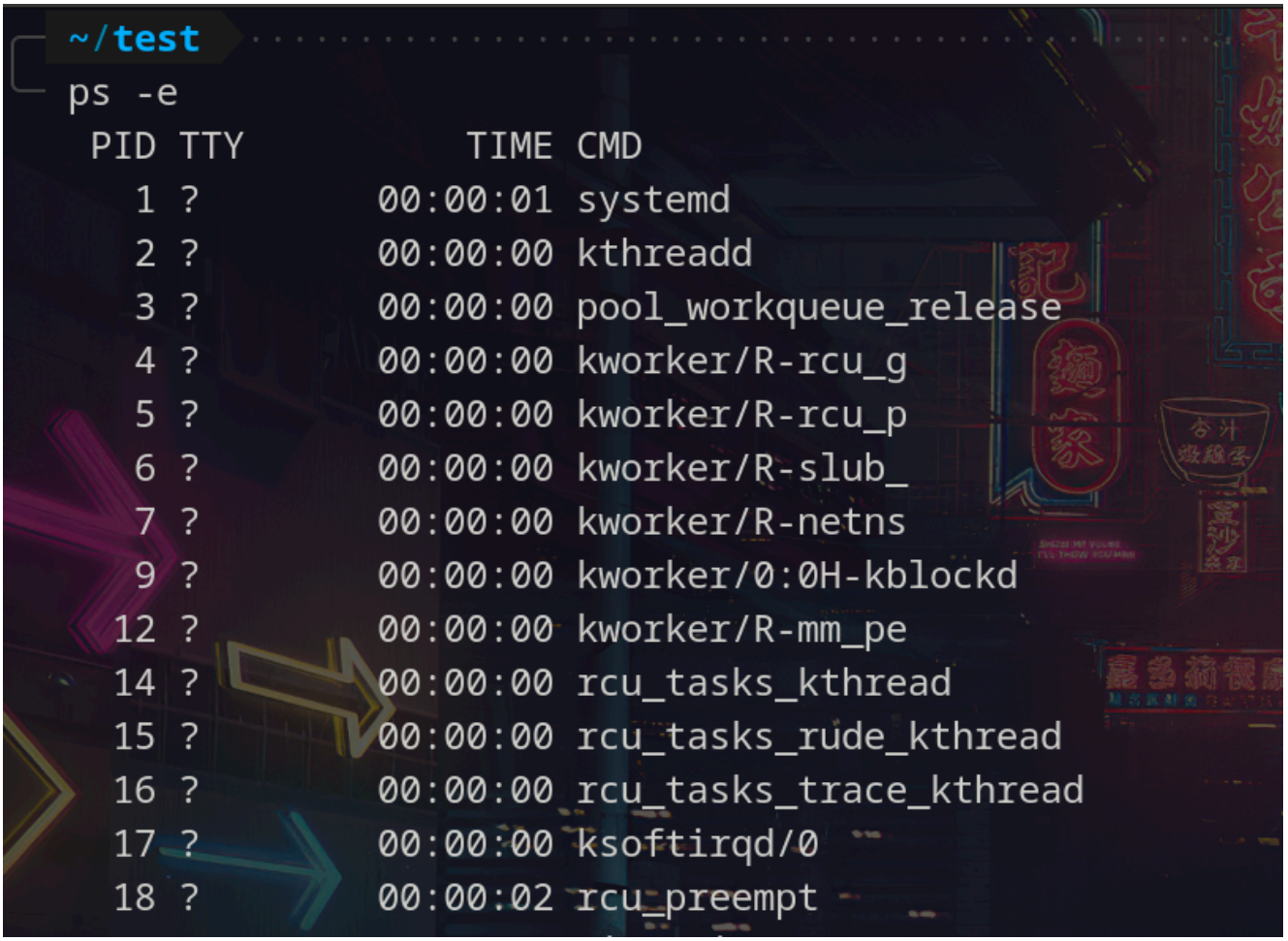
Reports a snapshot of the status of currently running processes.

Syntax :

ps [option]

Example :

Option	Use
ps -e	Display every active process on a Linux system in generic (Unix/Linux) format
ps -x	View all processes owned by you
ps -f	To provide more information on processes
ps -u	Filter processes by its user



```
~/test
ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
cyrixni+   11979   11954  0 15:01 pts/0    00:00:03 zsh
cyrixni+   12044    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12363    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12364    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12366   12044  0 15:01 pts/0    00:00:00 /home/cyrixninja/.cache/gitstatus/git
cyrixni+   37788   11979  0 17:07 pts/0    00:00:00 ps -f
```

32.kill Command

It is used to terminate processes manually.
kill command sends a signal to a process which terminates the process.
If the user doesn't specify any signal which is to be sent along with kill command then default TERM signal is sent that terminates the process..

Syntax :

kill [option] PID

Example:

```
~/test
ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
cyrixni+   11979   11954  0 15:01 pts/0    00:00:03 zsh
cyrixni+   12044    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12363    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12364    2107  0 15:01 pts/0    00:00:00 zsh
cyrixni+   12366   12044  0 15:01 pts/0    00:00:00 /home/cyri
cyrixni+   37788   11979  0 17:07 pts/0    00:00:00 ps -f

~/test
kill 11979
```

```
~/test                                                    at 17:12:55
kill -l
HUP INT QUIT ILL TRAP IOT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TER
M STKFLT CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF WI
NCH POLL PWR SYS
```

33.wc Command

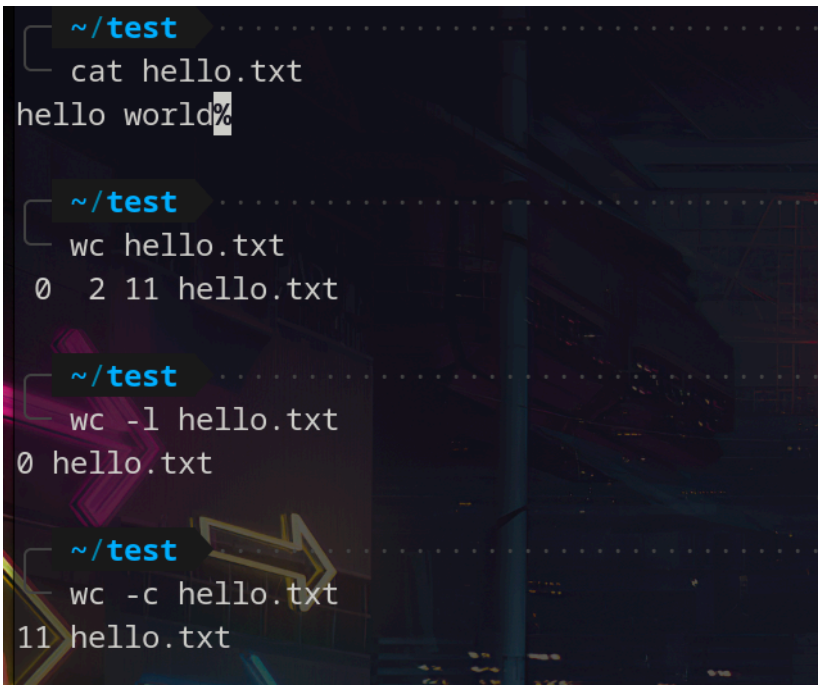
It’s used to find out number of newline count, word count, byte and characters count in a files specified by the file arguments.

Syntax :

wc [options] filenames

Example:

Option	Use
wc -l	Prints the number of lines in a file
wc -w	Prints the number of words in a file
wc -c	Displays the count of bytes in a file
wc -L	Prints only the length of the longest line in a file



34.ln Command

ln creates links between files.

ln creates hard links by default, or symbolic links if the -s (--symbolic) option is specified. When creating hard links, each TARGET must exist.

Syntax :

ln [OPTION]... [-T] TARGET LINK_NAME

Example:

35.nl Command

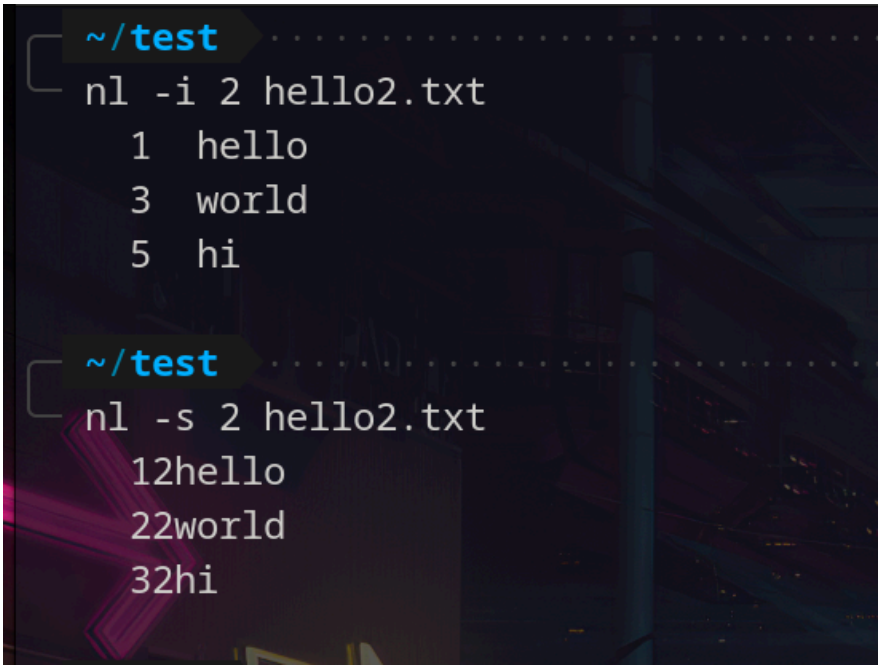
nl command numbers the lines in a file.

Syntax :

nl [OPTION]... [FILE]...

Example :

Option	Use
nl -i	Line number increment at each line
nl -s	Add STRING after (possible) line number
nl -w	Use NUMBER columns for line numbers



36.head Command

head makes it easy to output the first part (10 lines by default) of files.

Syntax :

head [OPTION]... [FILE]...

Example :

Option	**Use**
head -n	Print the first n lines instead of the first 10; with the leading '-', print all but the last n lines of each file
head -c	Print the first n bytes of each file; with a leading '-', print all but the last n bytes of each file
head -q	Never print headers identifying file names

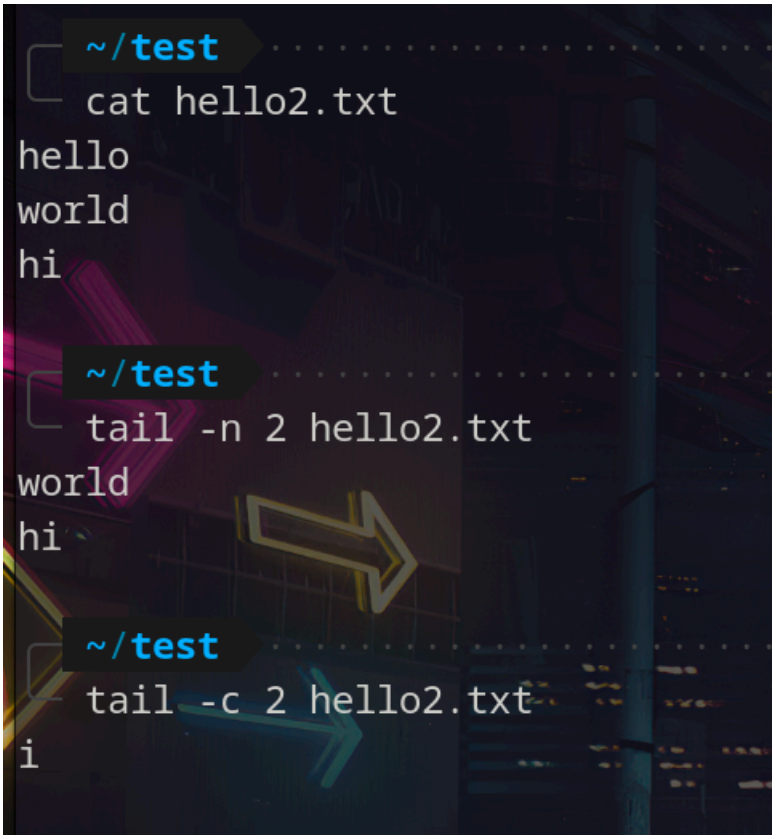
37.tail Command

tail is a command which prints the last few number of lines (10 lines by default) of a certain file, then terminates.

Syntax :
tail [OPTION]... [FILE]...

Example :

Option	**Use**
tail -n	Output the last num lines, instead of the default (10)
tail -c	Output the last num bytes of each file
tail -q	Never output headers



38.sort Command

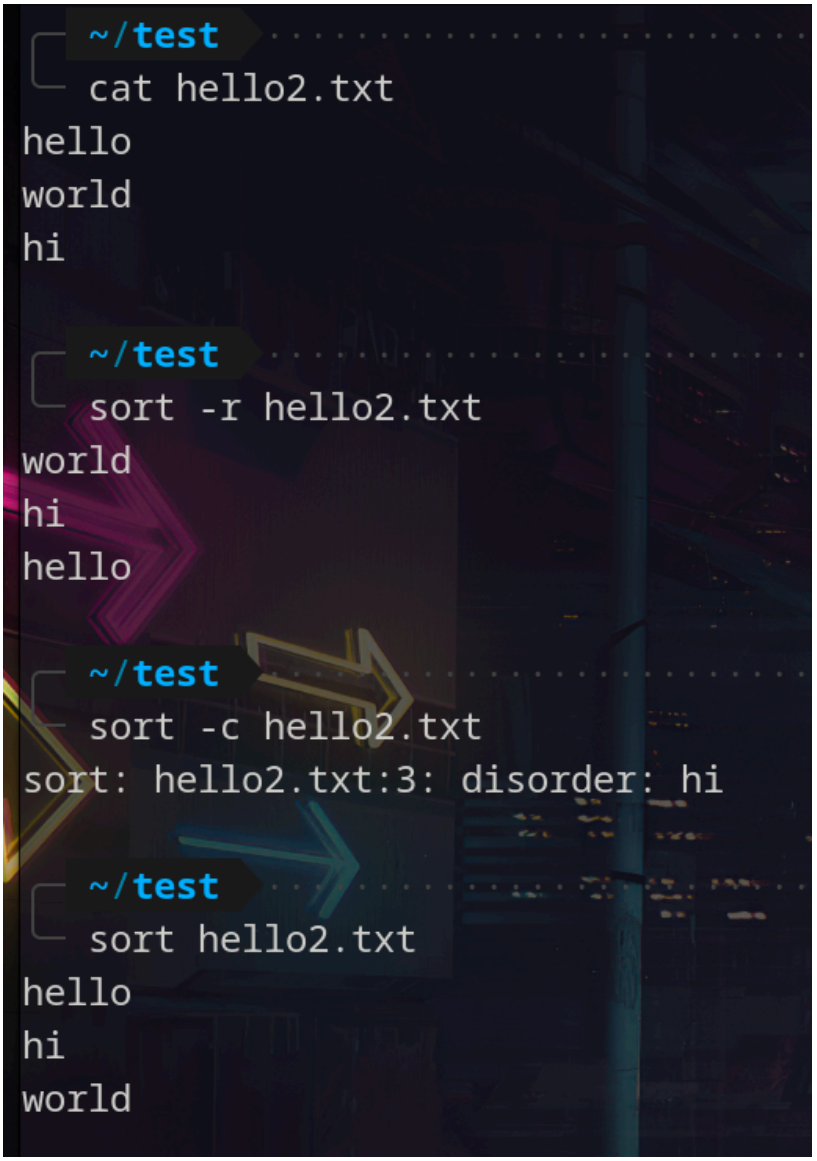
sort command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically.

Syntax :

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

Option	**Use**
sort -c	To check if the file given is already sorted or not
sort -r	Reverse the result of comparisons
sort -n	Compare according to string numerical value
sort -nr	To sort a file with numeric data in reverse order
sort -k	Sorting a table on the basis of any column
sort -b	Ignore leading blanks

Example :



39.find Command

find command searches for files in a directory hierarchy.

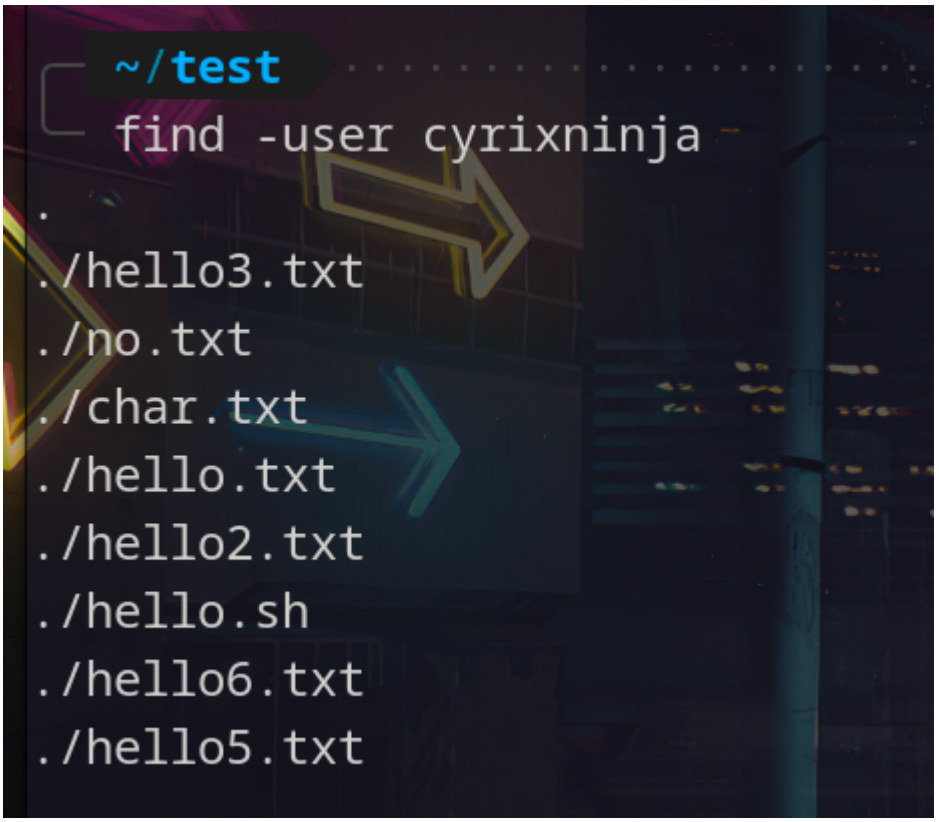
Syntax :

find [option] [path...] [expression]

Option	**Use**
find -name filename	Search for files that are specified by ‘filename’
find -newer filename	Search for files that were modified/created after ‘filename’
find -user name	Search for files owned by user name or ID ‘name’
find -size +N/-N	Search for files of ‘N’ blocks; ‘N’

	followed by ‘c’ can be used to measure size in characters
find -empty	Search for empty files and directories
find -perm octal	Search for the file if permission is ‘octal’

Example :



40.uniq Command

uniq reports or filters out repeated lines in a file.
It can remove duplicates, show a count of occurrences, show only repeated lines, ignore certain characters and compare on specific fields.

Syntax :

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

Option	**Use**
uniq -u	Prints only unique lines
uniq -d	Only print duplicated lines
uniq -D	Print all duplicate lines
uniq -c	Prefix lines with a number representing how many times they occurred
uniq -i	Ignore case when comparing

Example :

41. grep Command

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.

The pattern that is searched in the file is referred to as the regular expression.

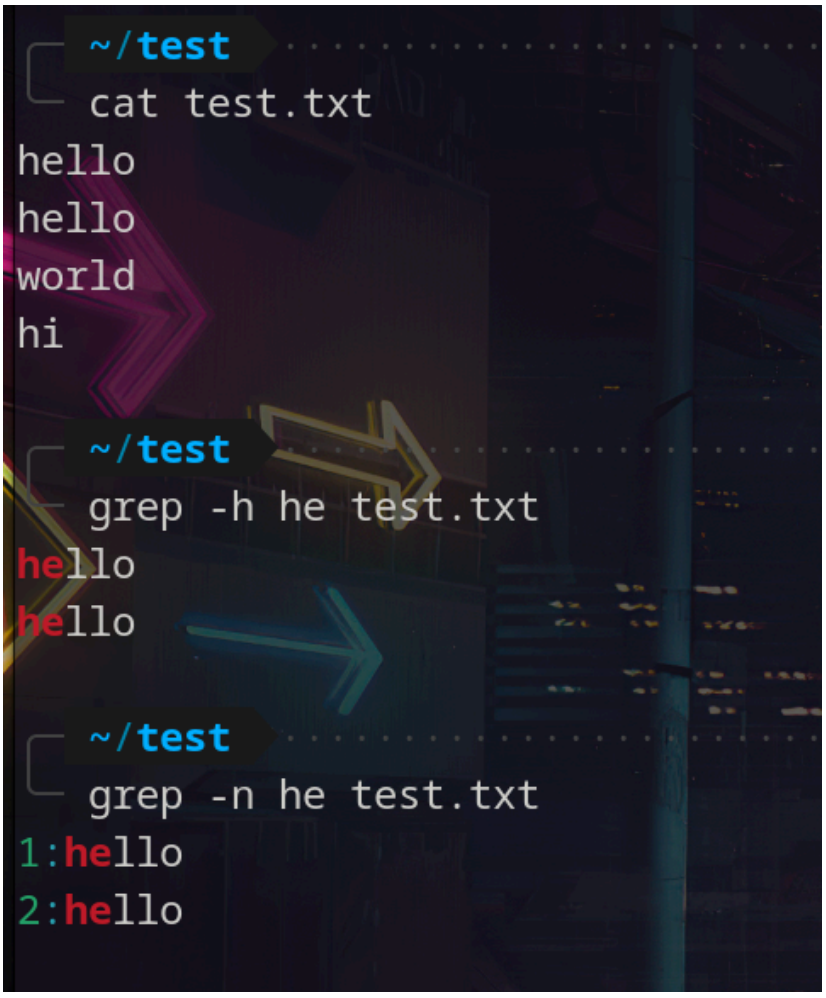
grep stands for globally search for regular expression and print out.

Syntax :

grep [options] pattern [files]

Option	**Use**
grep -c	Prints only a count of the lines that match a pattern
grep -h	Display the matched lines, but do not display the filenames
grep -l	Displays list of a filenames only
grep -i	Ignores, case for matching

Example :



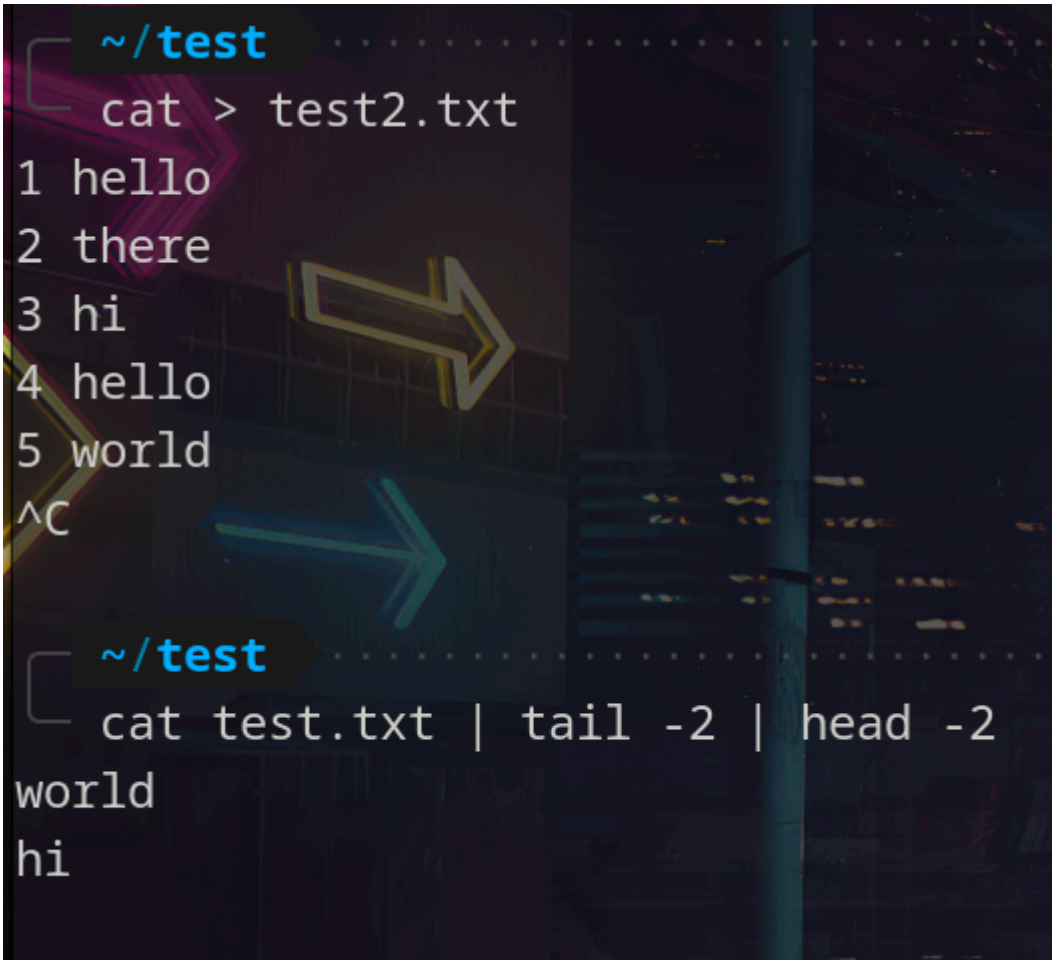
42. pipe (|) Command

It redirects the command STDOUT or standard output into the given next command STDIN or standard input.
In short, the output of each process directly as input to the next one like a pipeline.
The symbol '|' denotes a pipe.
Pipes help you mash-up two or more commands at the same time and run them consecutively.

Syntax :

command_1 | command_2 | command_3 | .... | command_N...

Example :



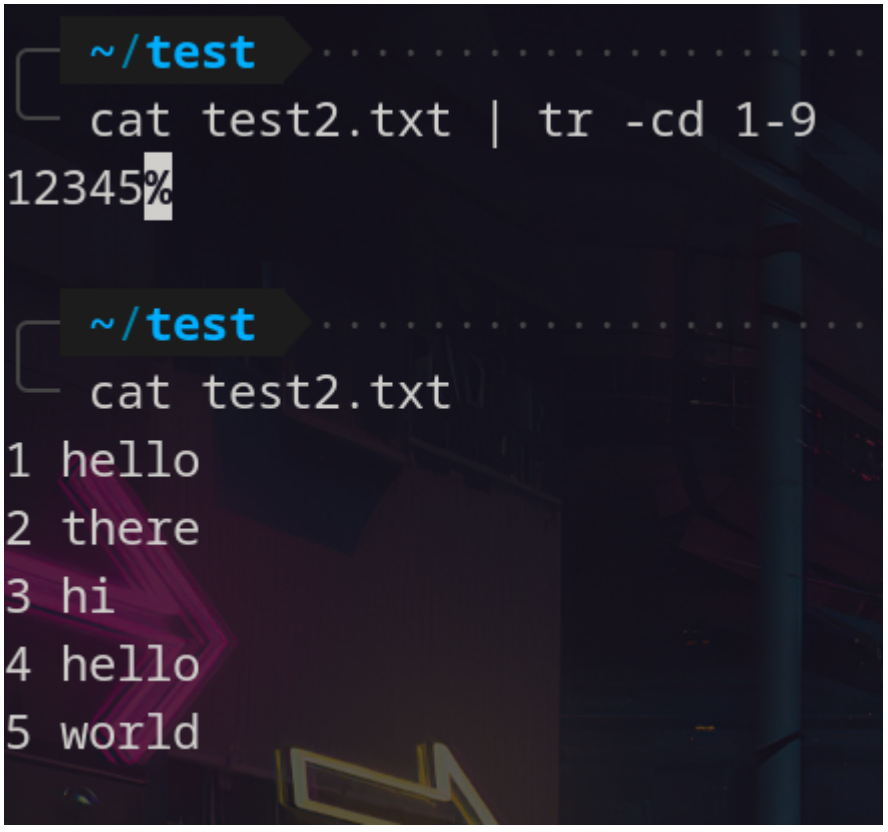
43. tr(translate) Command

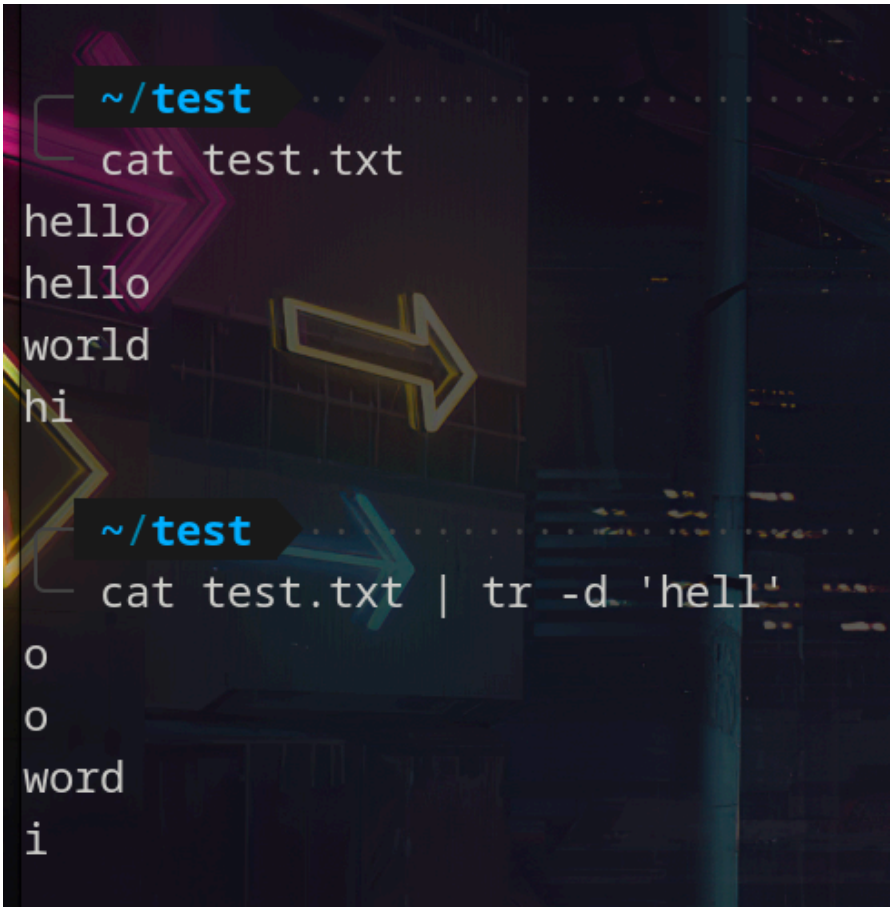
The tr command in UNIX is a command line utility for translating or deleting characters.
It supports a range of transformations including uppercase to lowercase, squeezing repeating characters, deleting specific characters and basic find and replace.
It can be used with UNIX pipes to support more complex translation.
tr stands for translate.

Syntax :
tr [OPTION] SET1 [SET2]
POSIX Character set supported by tr command :
• [:digit:] Only the digits 0 to 9.
• [:alnum:] Any alphanumeric character.
• [:alpha:] Any alpha character A to Z or a to z.
• [:blank:] Space and TAB characters only.
• [:digit:] Hexadecimal notation 0-9, A-F, a-f.
• [:upper:] Any alpha character A to Z.
• [:lower:] Any alpha character a to z..

Option	Use
tr -s	Replaces repeated characters listed in the set1 with single occurrence
tr -d	Delete characters in string1 from the input
tr -c	complements the set of characters in string. i.e., operations apply to characters not in the given set
tr -cd	Remove all characters except digits

Example :





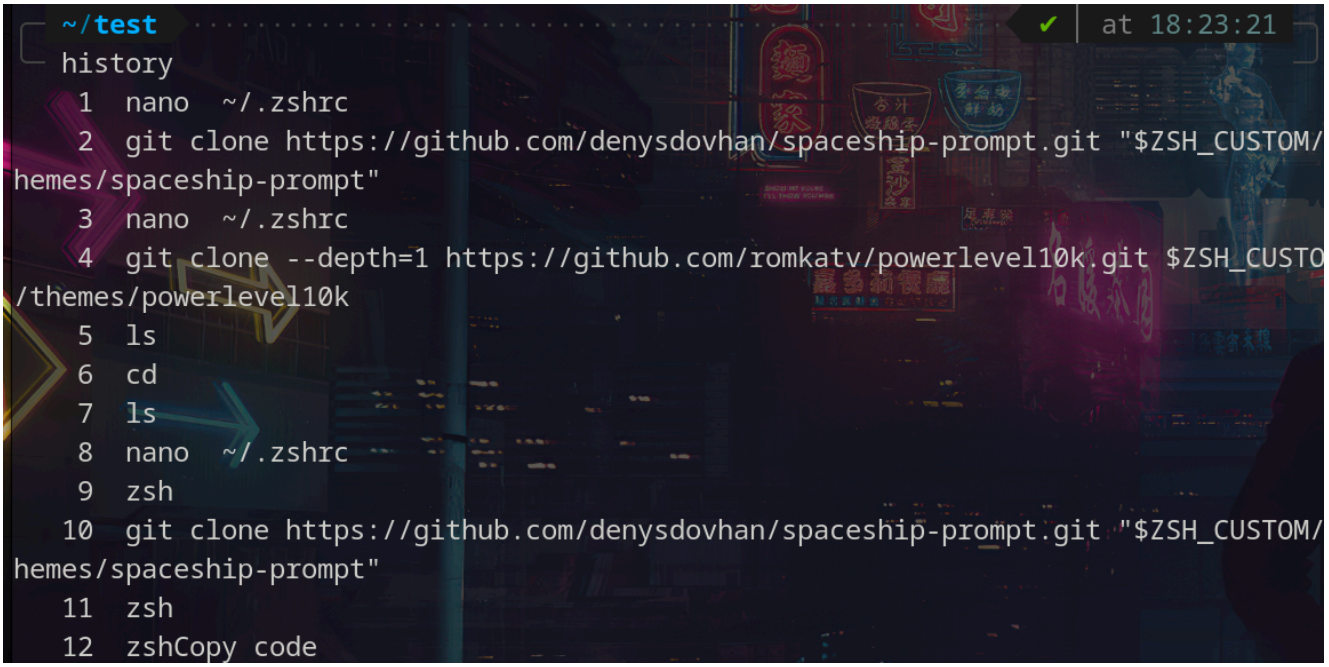
44. history Command

history command is used to view the previously executed command.

Syntax :

history

Example :



45. write Command

write sends a message to another user.

Syntax :

write user [ttyname]

Option	Use
user	The user to write to
tty	The specific terminal to write to, if the user is logged in to more than one session

Example :

46. wall Command

wall send a message to everybody's terminal.

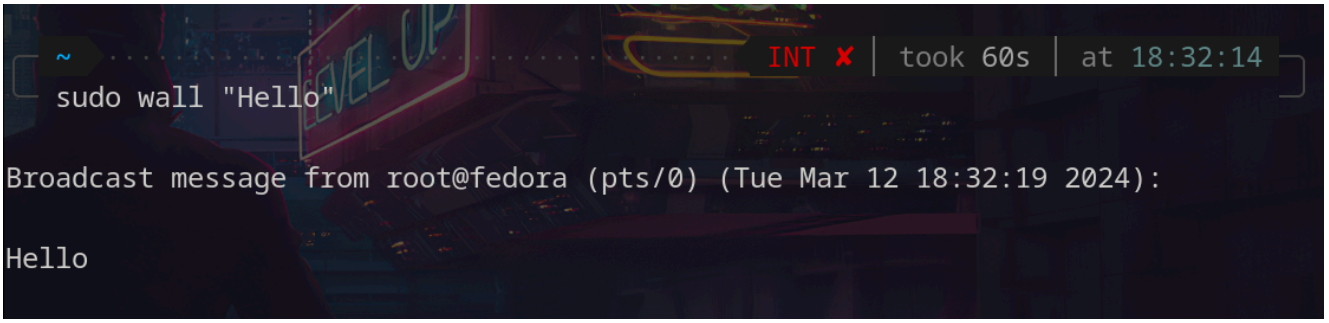
wall sends a message to everybody logged in with their mesg permission set to yes.

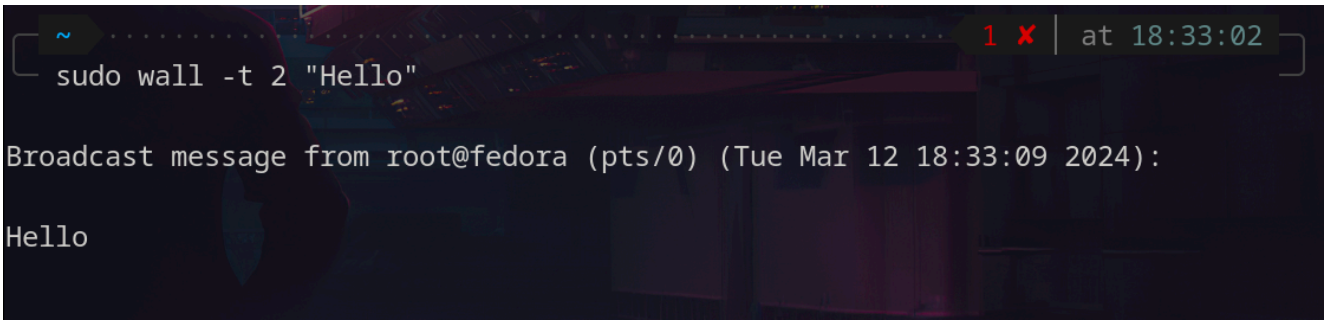
Syntax :

wall [-n] [-t TIMEOUT] [file]

Option	Use
wall -n	--nobanner Suppress banner
wall -t	--timeout TIMEOUT Write timeout to terminals in seconds. TIMEOUT must be positive integer. Default value is 300 seconds, which is a legacy from time when people ran terminals over modem lines.

Example :


~
INT ✘ | took 60s | at 18:32:14
sudo wall "Hello"
Broadcast message from root@fedora (pts/0) (Tue Mar 12 18:32:19 2024):
Hello


~
1 ✘ | at 18:33:02
sudo wall -t 2 "Hello"
Broadcast message from root@fedora (pts/0) (Tue Mar 12 18:33:09 2024):
Hello