

date

Description

It shows the current time in a given format.

Syntax

```
date + options
```

Example

- Using the date command to display the last modification time of a file:
 - `date -r installTextFiles.sh'`
 - This command will show the date in the format month/day/year:
 - `date +%D`
- This command will show the date in the format Sun, 15 May 2022 15:33:18 -0400:
 - `date -R`

uname

Description

It gives information about your system.

Syntax

```
uname + options
```

Example

- It shows all your system information:
 - `uname -a`
- It shows the kernel name:
 - `uname -s`
- It shows the Os:
 - `uname -o`

du

Description

It shows an estimate file space usage.

Syntax

du + options

Example

- it shows the size of all files ,not just directories:
 - `du -a final`
- it shows the estimate file space usage in a human readable way:
 - `du -h final`
- it shows a grand totaol of the usage space:
 - `du -c final`

free

Description

It shows the amount of free and used memory in the system.

Syntax

free+ options

Example

- Display the amount of memory in bytes in your whole OS:
 - `free -b`
- Display the amount of memory in a human readable way in the Os:
 - `free -h`
- Display the amount of memory in a specific directory :
 - `free -h final`

echo

Description

It displays/shows a line of text.

Syntax

echo+ options

Example

- Display a simple text on the screen:
 - `echo "hello world"`
- Display a text on the screen using allowing the backslash escapes:
 - `echo -e "Hello \n\n world!"`
- Do not display the trailing newline:
 - `echo -n "hello"`

apt

Description

It is a high-level command for the package management system.

Syntax

```
apt+ options
```

Example

- To update your system to the latest version:
 - `sudo apt upgrade`
- To install software:
 - `sudo apt package_name`
- To remove packages:
 - `sudo apt remove package_name`

pwd

Description

It displays your current location

Syntax

```
pwd + options
```

Example

- Display the current working directory:
 - `pwd`

cd

Description

It is used to changed your present working directory.

Syntax

```
cd + options
```

Example

- It will change your current working directory to your home directory:

- `cd`
- To change your present directory to the root:
 - `cd /`
- To change your present directory to the Downloads directory:
 - `cd ~/Downloads`

ls

Description

It used to list the content of a directory.

Syntax

```
ls+ options
```

Example

- It displays all entries including:
 - `ls -a`
- Long list sorted:
 - `ls -lc`
- Long list with the following format -rw-rw-r-- 1 katy katy 0 2022-03-29 18:53:26.528871787 -0400 2tutorial.txt:
 - `ls --full-time`

tree

Description

List contents in a tree-like format

Syntax

```
tree + options
```

Example

- print the size of each file in a more human readable way:
 - `tree -h final`
- print the date of last modification time:
 - `tree -D final`
- sort by last modification time:
 - `tree -t final`

man

Description

It is the system's manual pager.

Syntax

```
man+ options
```

Example

- Manual page of the ls command:
 - `man ls`
- Manual page of pwd:
 - `man pwd`
- Manual page of tree:
 - `man tree`

mkdir

Description

Create directory or multiple directories.

Syntax

```
mkdir+ options
```

Example

- create one directory with a single quote in the name:
 - `mkdir final/"final's 2022"`
- create a directory with a parent directory at the same time:
 - `mkdir -p final/linux`
- create multiple directories:
 - `mkdir final/linux final/networking final/math`

touch

Description

create files

Syntax

```
touch + options
```

Example

- create file with space in its name:
 - `"final 2022.txt`
- create a file with absolute path:
 - `touch ~/final/math.txt`
- create several files:
 - `touch math.txt bio.txt`

rm

Description

remove files

Syntax

`rm + options`

Example

- remove all files inside a directory:
 - `rm -I final/notes/*`
- remove empty directory:
 - `rmdir final/project`
- Remove non-empty directory:
 - `rm -r Downloads/picture`

cp

Description

copies files or directories from one place to other

Syntax

`cp + options`

Example

- copy files:
 - `cp +files to copy + destination`
 - `cp ~/Downloads/kodi.png ~/Desktop/kodilogo.png`
- copy directories:
 - `cp -r + directory to copy + destination`
 - `cp -r ~/lab5-docs ~/Desktop`
- copy the content of a directory to another:

- `cp -r ~/lab5-docs/* ~/lab5content`

mv

Description

move and rename directories and files

Syntax

`mv + source +destination`

Example

- move and rename a directory:
 - `mv ~/lab5content ~/Desktop/contentLab5`
- moving multiple files:
 - `mv ~/2tutorial.txt ~/ip2.txt ~/users.txt ~/Desktop`
- moving and renaming files:
 - `mv ~/2tutorial.txt ~/Desktop/tutorial2.txt`

stat

Description

It shows file or file system status. With this command you can see the files size,inode,block,etc.

Syntax

`stat+ options`

Example

- It shows time of last access-human readable.Ex, 2022-04-24 00:11:10.984475151 -0400:
 - `stat --format='%x' ip2.txt`
- It shows only the inode number of a file:
 - `stat --format='%i' ip2.txt`
- User name owner:
 - `stat --format='%U' ip2.txt`

Wildcard (*)

Description

It matches anything.

Syntax

```
ls *.options
```

Example

- List all file with the txt extension:
 - `ls *.txt`
- list txt and pdf file:
 - `ls *.txt *.pdf`
- List anything that beginngs with the vocal u:
 - `ls u*`

Wildcard(?)

Description

it matches one character.

Syntax

```
command ?
```

Example

- List all the hidden files in the current directory:
 - `ls ./.*?*`
- list all files thst have 3 letter as an extension and ended up with g:
 - `ls *.*?g`
- list all file that starts with s with a 3 letter extension an end in g:
 - `ls s*.*?g`

Wildcard[]

Description

It matches a single character in a range.

Syntax

```
command []
```

Example

- list all the file that start with the letter f,g,s,t:
 - `ls [f-gs-t]*`

- list all the files that do not starts with f,g,s:
 - `ls [!fgst]*`
- List all the files that has at least one number:
 - `ls *[0-9]*`

Brace expansion

Description

is a mechanism by which arbitrary strings may be generated.

Syntax

`command {}`

Example

- create a whole directory in a single command:
 - `mkdir -p fall/{math,spanish,bio}/{notes,book,exam}`
- create n number of files:
 - `touch fall/{math,spanish,bio}/{notes,book,exam}/text{1..5}.txt`
- remove multiple files in a single directory:
 - `rm fall/{math,spanish,bio}/{notes,book,exam}/text{1..5}.txt`

cat

Description

It displays the content of a file

Syntax

`cat + options + filename`

Example

- Display the content suppressing repeating empty line to a single empty line:
 - `cat -s dracula.txt`
- line number enclosing empty spaces:
 - `cat -b dracula.txt`
- display a dolla sign at the end of each line:
 - `cat -E dracula.txt`

head

Description

it shows the top number of lines in a file

Syntax

```
cmd + options +filename
```

Example

- print the first 5 lines in a file:
 - `head -5 dracula.txt`
- print the first 20 lines in a file:
 - `head -20 dracula.txt`
- print the first 1 lines in a file:
 - `head -1 dracula.txt`

tail

Description

It shows the last n number of lines in a file.

Syntax

```
tail + options +file
```

Example

- show on the screen the last 4 line:
 - `tail -4 dracula.txt`
- show on the screen the last 10 line:
 - `tail -10 dracula.txt`
- show on the screen the last 20 line:
 - `tail -20 dracula.txt`

cut

Description

It is used to cut specific part of each line in a file.

Syntax

```
cut + options +filename
```

Example

- display all the users in your system:
 - `cut -d ':' -f1 /etc/passwd`

- Display a long list except the permission :
 - `ls -l | cut -d ' ' --complement -s -f1`
- cut and change the delimiter:
 - `cut -d ':' -f2,6 --output-delimiter= '->' /etc/passwd`

tr

Description

It is used to changed an ouput delimiter from the standard output

Syntax

`standard output + tr +options+set+set`

Example

- change the delimiter from ';' to an empty space:
 - `cat cars.csv | tr ';' ' '`
- translate white space into tabs
 - `cat dracula.txt | tr "[:space:]" '\t'`
- translate tab into space:
 - `cat bible.txt | tr -s "[:space:]" ' '`

paste

Description

It merges the content of a file horizontally in columns.

Syntax

`paste + options +file`

Example

- merge the content of 2 files:
 - `paste homework.txt final.txt`
- merger the content of 2 file changing the output delimiter:
 - `paste -d ":" todolist.txt userbinbash.txt`
- merge the content of three files:
 - `paste -d ":" todolist.txt userbinbash.txt users.txt`

WC

Description

It is used for printing the number of lines,characters and bytes in a file.

Syntax

```
wc + options +filename
```

Example

- Print how many word are in a file:
 - `wc -w users.txt`
- print the number of lines in a file:
 - `wc -l users.txt`
- print the number of characters in a file:
 - `wc -m users.txt`

grep

Description

It is used to search a text in a file.

Syntax

```
grep + options + search criteria + file
```

Example

- Search for all line that do not contain the word "dracula" in the file:
 - `grep -v 'dracula' ~/Documents/dracula.txt`
- search for only the given word by itself:
 - `grep -w 'dracula' dracula.txt`
- print the line number that matcched the criteria regardless the case:
 - `grep -in 'dracula' dracula.txt`
- Search for all the lines that starts with the string dracula
 - `grep -ni '^dracula' dracula.txt`
- Search for all the lines that ends with the string dracula
 - `grep -n 'dracula$' dracula.txt`
- search for more than one word per line
 - `grep -Ewn 'horror|love|scarre' dracula.txt`

Output redirection

Description

it is used to redirect input/output of command to a file.

Syntax

```
command+ > + file
```

Example

- save the error generated by a command to a file:
 - `ls -lA desktop/ 2> error-of-ls`
- save the error and success at the same file:
 - `ls -lA downloads/ Pictures >sucess.txt 2> error.txt`
- sends error to the black hole:
 - `ls -lA downloads/ 2> /dev/null`

Appending output to a file

Description

it is used to add more to a file instead of overwriting its content

Syntax

```
command+ >> + file
```

Example

- it saves the output of this command keeping the old data:
 - `ls -la >> allmyfiles.txt`

redirect standard output

Description

The pipe is used to redirect the standard outout of a command to the standard input of another.

Syntax

```
command| command2| comandn
```

Example

- To look for all the option a command
 - `man ls | grep "^[:space:]*[:punct:]"`
- To look for an specific string in the man page
 - `man ls | grep "human-readable"`
- Display only the 2nd line in a file
 - `head -2 text1.txt | tail -1`

vim

Description

is a command-line text editor

Syntax

```
vim +file name"
```

Example

- To open a file in vim
 - `vim + filename`
- To insert text 1- vim file-name 2-press esc + letter i
- to save and close the file
 - press esc + `:wqa!`
- to save the file as a new file
 - press esc + `:w new.txt`
- to edit
 - press the letter r to replace one single character
 - then save it `wqa!`

tar

Description

creates archives with multiple files and directories into a single file.

Syntax

```
tar + options + archive name + files to add to archive
```

Example

- To create an archive
 - `tar -cf example.tar file1 file2 file3`
- To extract archive in a different directory
 - `tar -xf example.tar --directory~/Downloads`
- To delete specific files of an archive
 - `tar --delete -f example.tar file3`

XZ

Description

compress files

Syntax

```
xz + options + file name
```

Example

- compress multiple files
 - `xz file.txt file2.txt`
- compress a single file and keep original file
 - `xz -k file.txt`
- decompress a file
 - `xz -d file.txt.xz`

chmod

Description

it is used to change permissions on files and directories. u(user),g(group),o(other),a(all).

Syntax

```
chmod +category + operator +permission
```

Example

- assigned the permission to user to read,write,execute
 - `chmod u=rwx script.sh`
- remove the permission of the other group to execute
 - `chmod o-x script.sh`
- add the permission to write for the group
 - `chmod g+w file.txt.xz`