

Question 1

awk

- Description:
 - It is used to display a text of a file or input with a condition. It doesn't filter the text line by line.
- Formula:
 - `awk + option + {command} + file`
- Examples:
 - `awk -F ':' {'print $1,2'} bible.txt`
 - `awk 'NR > 75 {print}' cereal.csv`

cat

- Description:
 - Print the content a file.
- Formula:
 - `cat + option + file`
- Examples:
 - `cat myfile.txt`
 - `cat -n myfile.txt`
 - `cat -E myfile.txt`

cp

- Description:
 - This command is used to copy or change the name of a file and directories.
- Formula: `cp + file + destination`
- Examples:
 - `cp ~/Documents/Csv/cars.csv ~/Downloads`
 - `cp ~/Documents/Csv/cars.csv ~/Documents/Csv/listofcars.csv`
 - `cp -r ~/Documents/Csv ~/Desktop`

cut

- Description:
 - This command is used to display a specific section of a file.
- Formula: `cut + option + file`

- Examples:

- `cut -d " " -f1 ~/Document/Csv/cars.csv`
- `cut -d ";" -f1,7 --output-delimiter ' = ' ~/Documents/Csv/cars.csv`
- `cut -b 1-5 ~/Documents/Csv/cars.csv`

grep

- Description:

- This command is used to search specific words in a file. It does work line by line finding the match of the condition requested.

- Formula:

- `grep + option + search criteria + file`

- Examples:

- `grep "Honda" cars.csv`
- `grep -i "honda" card.csv`
- `grep -ni "honda" cars.csv`

head

- Description:

- This command will print the top number of lines of a file. By default will print the first 10 lines.

- Formula:

- `head + option + file`

- Examples:

- `head -5 cars.csv`
- `head cars.csv`
- `head -1 cars.csv`

ls

- Description:

- This command will list files and directories inside of a directories.

- Formula:

- `ls + option + directory to list`

- Examples:

- `ls -a ~/Downloads`
- `ls -lR ~/Documents`

- `ls -lXs ~/Downloads/`

man

- Description:
 - This command is used to see manuals of other commands in order to find options with instructions. It is a quick reference of what each option does.

- Formula:

- `man + option + command`

- Examples:

- `man -a ls`
- `man -f cp`
- `man -k "inxi"`

mkdir

- Description:
 - This command is used to create one or multiple directories at the same time.

- Formula:

- `mkdir + new directory name or path + new directory name`

- Examples:

- `mkdir ~/Pictures/3D`
- `mkdir -p movies/old\ movies`
- `mkdir -p movies/{old,new}/{3D,2D}`

mv

- Description:
 - This command is used to move or rename files and directories.

- Formula:

- `mv + source + destination`
- `mv + file/directory to rename + new name`

- Examples:

- `mv ~/Documents/Csv/cars.csv ~/Downloads/`
- `mv cars.csv listofcars.csv`
- `sudo mv ~/Document/Csv/cars.csv /usr/share/`

tac

- Description:
 - This command works similar to `cat` but in an opposite way. It does display the content of a file in reverse order.

- Formula:

- `tac + option + file`

- Examples:

- `tac ~/Documents/Csv/cars.csv`
- `tac -n ~/Documents/Csv/cars.csv`
- `tac -E cars.csv`

tail

- Description:

- This command works similar to `head` but in from the bottom of the file instead from the top. It will display the last 10 lines of the file by default.

- Formula:

- `tail + option + file`

- Examples:

- `tail -10 cars.csv`
- `tail ~/Documents/Csv/card.csv`
- `tail -5 cars.csv | head -10`

touch

- Description:

- This command is executed to create files.

- Formula:

- `touch + name of new file`

- Examples:

- `touch ~/Documents/sample.txt`
- `touch "new listing.txt"`
- `touch programm.py newcars.csv mylist.txt`

tr

- Description:

- This command will substitute or change characters for a requested display.

- Formula:

- `standard output | tr + option + set + set`

- Examples:

- `cat cars.csv | tr ";" "="`
- `tac -E cars.csv | tr "[:spaces:]" '\t'`
- `cat -n cars.csv | tr ":" " "`

tree

- Description:

- It is a recursive directory listing that list all the directories within all their content including other directories and files.
- Formula:
 - `tree + option + directory`
- Examples:
 - `tree -a ~/Documents`
 - `tree -X ~/Documents`
 - `tree -d ~/Downloads`

Questions 2

How to work with multiple terminals open?

On the top left of the command line window there is a button to open an additional command line tab.

How to work with manual pages?

Input `man + option + command to be search` and the command will display the manual of that command searched with all option, instructions, and description. In the terminal we can scroll up and down with the mouse or use the arrows up and down to navigate through the manual itself.

How to parse (search) for specific words in the manual page

By using `man -k + key word`. This command will find all the manual related to the key word.

How to redirect output (> and |)

Using the `>` key will redirect the output to a file. With the syntax `command output + > + file` will forward the output to a file and using `command output + >> + file` will keep the old data and add the new one. Also using some standards will make the redirection to be specific for example using `command output + 2> + file` will save the error messages to the file, `command output + 1> + file` will save the success of the command, `command output + &> + file` will save the error and the success to the same file, `command output + > + file + 2> + file` will save the success to a file and the error to another file.

The `|` will command redirect standard output with some command to clarify the search and look. For example `cat welcome.txt | grep "hello"` will redirect the output and only display the lines containing "hello".

How to append the output of a command to a file

By using `command output + >> + file` this will append the file without deleting the data already inside of the file.

How to use wildcards

By using the `*` wildcard we can simply replace characters of the file name or extension in order to include all the file with the requested input. For example, `rm -r image*.png` this will remove/delete all the png file named "image" and having anything after.

By using the `?` wildcard we can replace characters with it to have the specific files we need. For example, `cp house???.???` this will copy all the files named "house" with 2 characters after and specifically 3 character extension.

By using the `[]` wildcard will match a single character in the range. For example, `ls file[0-9].txt ~/Downloads` this will display all the text files named "files" and one number character after its name.

For copying and moving multiple files at the same time

For copying multiple files in a single command we can use `cp python.py wallpaper.png index.html ~/Desktop/website/` or also we can use wildcards such as `*` or `?`. While moving files at the same time does not need an option for example `mv python.py wallpaper.png index.html ~/Downloads/`.

How to use brace expansion

The brace expansion can be used as an option to create multiple files and directories at one time or remove file and directories too. For example, `mkdir -p website/{design,www}/{active,innactive}`.

For creating entire directory structures in a single command

We can create multiple directories using the `-p` option with the `mkdir` command, adding brace expansion to the directories being created. For example, `mkdir -p movies/{3D,2D}/{old,new}`