

Final Assignment

Question 1

awk

- Description: Scripting language for processing and displaying text
- Formula/Syntax: `awk + options + {awk command} + file + file to save (optional)`
- Examples:
 - Print the first column of every line of a file.
 - `awk '{print $1}' ~/Documents/Csv/cars.csv`
 - Prints the last field of the `/etc/passwd` file.
 - `awk '{print $NF}' /etc/passwd`

cat

- Description: Displays content of a file
- Formula/Syntax: `cat + option + file(s) to display`
- Examples:
 - Display content of a file located in the `pwd`
 - `cat todo.lst`
 - Display content of a file with line numbers
 - `cat -n ~/Documents/todo.md`

cp

- Description: copies files/directories from a source to a destination.
- Formula/Syntax: `cp + files to copy + destination`
- Examples:
 - To copy a file
 - `cp ~/Downloads/games.zip ~/Games/`
 - To copy multiple files in a single command:
 - `sudo cp -r home.html page1.html style.css ~/hw/`

cut

- Description: Used to extract a specific field of a file and display it to the screen.
- Formula/Syntax: `cut + option + file(s)`
- Examples:
 - Display a list of all the users in your system
 - `cut -d ':' -f1 /etc/passwd`
 - Display a list of all the users in your system with their login shell
 - `cut -d ':' -f1,7 /etc/passwd\`
 - Cut a file using a delimiter but changing the delimiter in the output.
 - `cut -d ':' -f1,7 --output-delimiter=' ' /etc/passwd`

grep

- Description: Used to search text in given file. (basically ctrl+f)
- Formula/Syntax: `grep + option + search criteria + file(s)`
- Examples:
 - Search any line that contains the word 'dracula' regardless of the case
 - `grep -i 'dracula' ~/Documents/Books/dracula.txt`
 - Search for all the lines that do not contain the word 'war'
 - `grep -v 'war' ~/Documents/Books/war-and-peace.txt`

head

- Description: Displays the top N number of lines of a given files. By default, first 10 lines.
- Formula/Syntax: `head + option + file(s)`
- Examples:
 - Display the first 10 lines of a file
 - `head ~/Documents/Book/dracula.txt`
 - Display the first 5 lines of a file
 - `head -5 ~/Documents/Book/dracula.txt`

ls

- Description: Displays all the files inside a given directory.
- Formula/Syntax: `ls + option + directory`
- Examples:
 - List all the files in current directory including hidden files.
 - `ls -a`
 - List all the files sorted by extension, last modified, and recursively in current directory.
 - `ls -XtR`

man

- Description: Displays a manual of a given command.
- Formula/Syntax: `man + option + command`
- Examples:
 - Display a manual of the ls command.
 - `man ls`
 - Display a short description of the ls command.
 - `man -f ls`

mkdir

- Description: Creates single or multiple directories.
- Formula/Syntax: `mkdir + name of directory`
- Examples:
 - To create directory with a space in the name
 - `mkdir wallpapers/new\ cars`
 - `mkdir wallpapers/'new cars'`
 - To create a directory with a parent directory at the same time
 - `mkdir -p wallpapers_others/movies`

mv

- Description: moves and renames directories.
- Formula/Syntax: mv + source + destination
- Examples:
 - To move multiple directories/files to a different directory.
 - `mv games/ wallpapers/ rockmusic/ /media/student/flashdrive`
 - To rename a file
 - `mv homework.docx cis106homework.docx`
 - To move and rename a file in the same command
 - `mv Downloads/cis106homework.docx Documents/new_cis106homework.docx`

tac

- Description: Used for displaying the content of a file in reverse order.
- Formula/Syntax: tac + option + file(s) to display
- Examples:
 - Display content of a file located in pwd
 - `tac todo.md`
 - Display content of a file with an attached separator before instead of after
 - `tac -b todo.md`

tail

- Description: Displays the last N number of lines of a given files. By default, last 10 lines.
- Formula/Syntax: tail + option + file(s)
- Examples:
 - Display the last 10 lines of a file
 - `tail ~/Documents/Book/dracula.txt`
 - Display the last 5 lines of a file
 - `tail -5 ~/Documents/Book/dracula.txt`

touch

- Description: Creates a file.
- Formula/Syntax: touch + option + file name and extension
- Examples:
 - To create several files
 - `touch list_of_cars.txt script.py names.csv`
 - To create a file with a space in its name
 - `touch "list of foods.txt"`

tr

- Description: Used for translating or deleting characters from standard output.
- Formula/Syntax: standard output | tr + option + set + set
- Examples:
 - Translate one character to another
 - `cat file.txt | tr '.' ','`

- Translate white space into tabs
 - `cat program | tr "[:space:]" '\t'`
- Translate tabs into space.
 - `cat program | tr -s "[:space:]" ' '`

tree

- Description: List contents of directories in a tree-like format.
- Formula/Syntax: `tree` + option
- Examples:
 - Lists all contents of directories as well as hidden files
 - `tree -a`
 - Lists all contents of directories with full path prefix for each sub-directory and file.
 - `tree -f`

Question 2

How to work with multiple terminals open?

- Click either of the two buttons with the + to open another terminal vertically or horizontally.

How to work with manual pages?

- Use the `man` command provided with a command you need a manual for. Navigate with the arrow keys. Pressing H will display a page on different ways to move, search, jump, etc. Pressing Q will return you to the previous page or command line.

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

- Use `/` to search forward or `?` to search backward.

How to redirect output (> and |)

- To save output of command to a file: `ls -lA > all-files-in-home.txt`
- Display only the options of any command from its man page: `man ls | grep "^[[:space:]]* [[:punct:]]"`
- Do not display errors. Send errors to `/dev/null/` (black hole): `ls -lA Downloads/ 2> /dev/null`

How to append the output of a command to a file

- Append means to add but keep old data: `ls -la >> allmyfiles.lst`

How to use wildcards

- '*' - Asterisk matches anything and nothing and matches any number of characters
 - `log*.log` - selects all files that start with log(any character(s)) and ends with the .log extension.
- '?' - Question mark wild card allow you to match precisely one character.
 - `*.???` - selects all files with 3 letter extension
 - `ls ./.*??*` - list all hidden files in the current directory.

- `[]` - Bracket wild card allows you to match a single character in a range. Can also use POSIX or Character Classes.
 - Match all files whose name begins with a letter from a-p or start with letters s or c: `ls [a-psc]*`

For copying and moving multiple files at the same time:

- `cp *.txt ~/Documents/textfiles | mv ~/Documents/textfiles/*.txt ~/Downloads`

How to use brace expansion

- Using the `{}` allows you to generate arbitrary string to use with commands.

For creating entire directory structures in a single command:

- `mkdir -p music/{jazz,rock}/{mp3files,videos,raw}/new{1..3}`