Final Exam Cheat Sheet

Commands

awk

Description: Scripting language used for processing and displaying text.

Formula/syntax: `awk + options + {awk command} + file + file to save (optional)`

3 examples:

- Print first column of every line of a file
  - `awk '{print $1}' ~/Documents/Csv/cars.csv`
- Print the first field of a file
  - `awk -F ';' '{print $1}' ~/Downloads/Csv/cars.csv`
- Print first and last field of the /etc/passwd
  - `awk -F: '{print $1," = ", $NF}`
- Start printing a file from a given line (exclude the first 2 lines)
  - `awk 'NR > 2 { print }' /etc/passwd`
- How to change field to upper case
  - `awk -F';' '{print toupper($1)}' ~/Documents/Csv/cars.csv`

cat

Description: Displays the content of a file

Formula/syntax: `cat + option + file(s) to display`

3 examples:

- Display the conten tof a file located in the pwd
  - `cat todo.md`
- Display content of a file with line numbers
  - `cat -n ~/Documents/todo.md`
  - `cat -b ~/Documents/todo.md` -> Excludes empty lines
- Display the content of a file a $ at the end of every line
  - `cat -E ~/Documents/todo.md`

cp

Description: Copies files/directories from a source to a destination

Formula/syntax: `cp + files to copy + destination` -> file `cp -r + directory to copy + destination` -> directory

3 examples:

- Copy the content of a directory to another directory

    - `cp Downloads/wallpapers/* ~/Pictures/` -> will move files not directories inside directories
- Copy multiple files in a single command
    - `sudo cp -r script.sh program.py home.html assets/ /var/www/html/`
- Copy a directory with absolute path
    - `cp -r ~/Downloads/wallpapers ~/Pictures/`

cut

Description: Used to extract a specific section of each line of a file and display it to the screen

Formula/syntax: `cut + option + file(s)`

3 examples:

- Display a list of all the users in your system
    - `cut -d ':' -f1 /etc/passwd`
- Display a list of all te 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: Search text in given file. Works line by line basis.

Formula/syntax: `grep + option + search criteria + file(s)`

3 examples:

- Search any line that contains the word "dracula" in the given file with case insensitivity and line numbers:
    - `grep -i n'dracula' ~/Documents/dracula.txt`
- Search all the lines that do not contain the word 'war'
    - `grep -v 'war' ~/Documents/Books/war-and-peace.txt`
- Invert the search
    - `grep -Ecv "bash|zsh|fish| /etc/passwd`
- Search and display the total number of times a given word appears in a file
    - `grep -wc 'bin/bash' /etc/passwd`

head

Description: displays the top N number of lines of a given file. By default, it prints the first 10.

Formula/syntax: `head + option + file(s)`

3 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: Lists the content of a given directory or the file/directory itself.

Formula/syntax: `ls + option + directory to list`

3 examples:

- List all the files inside a given directory
    - `ls -a ~/Pictures`
- List all the files sorted by file size
    - `ls -S ~/Documents`
- List all the files in a given directory by last modified
    - `ls -t ~/Documents`

man

Description: Manual pages; documentation files that describe Linux shell commands, executable programs, system calls, special files, etc.

Formula/syntax: `man + command name`

3 examples:

- Open the man page of the passwd command
    - `man passwd`
- Show all the available pages of a command
    - `man -a passwd`
- Searches for a man page for a given word or regular expression or phrase
    - `man -k file`

mkdir

Description: Makes single and multiple directories.

Formula/syntax: `mkdir + the name of the directory`

3 examples:

- Create a director in a different directory using relative/absolute path
    - `mkdir wallpapers/ocean`
    - `mkdir ~/wallpapers/forest`
- Create a directory with a space in the name
    - `mkdir wallpaper/'cities usa'`
- Create multiple directories
    - `mkdir wallpapers/cars wallpapers/cities wallpapers/forest`
- 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` and `mv + file/directory to rename + new name`

3 examples:

- Move a file from one directory to another using absolute path
  - `sudo mv ~/Downloads/theme /usr/share/themes`
- Move a file from one directory to another combining absolute and relative path
  - `mv Downloads/english_homework.docx /media/student/flashdrive/
- Move multiple directories/files to a different directory
  - `mv games/ wallpapers rockmusic/ /media/student/flashdrive`
- Rename a file
  - `mv homework.docx cis106homework.docx`
- Move adn rename a file in the same command
  - `mv Downloads/cis106homework.docx Documents/new_cis106homework.docx`

tac

Description: Displays the content of a file in reverse order

Formula/syntax: `tac + option + file(s) to display`

3 examples:

- Display the content of a file located in the pwd
  - `tac todo.md`
- Display the content of a file using absolute path
  - `tac ~/Documents/todo.md`

tail

Description: Displays the last N number of lines of a given file. By default, it prints the last 10 lines.

Formula/syntax: `tail + option + file(`

3 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 files

Formula/syntax: `touch + name of file`

3 examples:

- Create serval files
  - `touch list_of_cars.txt script.py names.csv`
- Create a directory using different paths
  - `touch ~/Downloads/games.txt`
  - `touch Downloads/games2.txt` -> pwd is home directory
- 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`

3 examples:

- Translate one character to another (for example a period with a comma)
  - `cat file.txt | tr '.' ','`
- Translate white space into tabs
  - `cat program.py | tr "[:space:]" '\t'`
- Translate tabs into space
  - `cat file.py | tr -s "[:space:]" ' '`

tree

Description: Lists contents of directories in a tree-like format

Formula/syntax: `tree + name fo directory`

3 examples:

- Display structure of a directory
  - `tree website`

nano

Description: Simple text editor

Formula/syntax: `nano`

3 examples:

- Open a titled nano document
  - `nano program.py`
- Exit nano
  - `^N`
- Save document in nano
  - `^O`
- Open already made document

    - `nano + path/name of document`

Questions

- **How to work with multiple terminals open?** Open one terminal the open another terminal and set them side by side. Or user tillix and split the terminal as needed.
- **How to work with manual pages**
- **How to parse (search) for specific words in the manual page** man 'command' | grep -i 'word' OR man 'command' | options
- **How to redirect output** `(> and |)` Description: Redirect input and output of commands to and from files. File descriptors are used for directing the input and output of commands.

File Descriptor	Abbreviation	Description
0	STDIN	Standard Input
1	STDOUT	Standard Output
2	STDERR	Standard Error

Formula/syntax: `Command output + > + file`

3 examples:

- Save the output of a command to a file
    - `ls -lA ~ > all-files-in-home.txt`
- Save the error generated by a command to a file
    - `ls -lA downloads/ 2> error-of-ls`
- Save error to a file and the success to another
    - `ls -lA downloads/ Pictures > success.txt 2> error.txt`
- Save the error and success to the same file
    - `ls -lA downloads/ Pictures &> alloutput.txt`
- FOR PIPES (|)
    - Description: Allows you to redirect the standard output of a command to the standard input of another

Formula/syntax: `command_1 | command_2 | command_3 | ... | command_N`

3 examples:

- Use grep to look for a string in a particular man page
    - `man ls | grep "human-readable"`

- Display only the ip addresses from the output of the ip command
    - `ip addr | grep -Eo '[[:digit:]]{1,3}.[[:digit:]]{1,3}.[[:digit:]]{1,3}.[[:digit:]]{1,3}'
- Display only the 2nd line in a file
    - `head -2 file.list | tail -1`
- **How to append the output of a command to a file**
    - To overwrite what's inside a file
        - `ls -la > allmyfiles.txt`
    - To add data/save the old data to a file
        - `ls -la >> allmyfiles.txt`
- **How to use wildcards**

Wildcard	Matches	Example
*	0 or multiple characters	`ls *.pdf`
?	1 character	`ls program?.py`
[]	1 character from a given set of characters	`ls document[A-Z].doc`
[!]	The opposite of the given set	`ls new-doc[!0-9].docx

    - For copying and moving multiple files at the same time
        - Move all files inside a directory
            - `mv Pictures/* ~/Backup/`
        - Copy all the files that that have 2 characters between 2 letters
            - `cp Downloads/b??k.pdf Documents/`
        - List all the hidden files that have a 4 letter file extension
            - `ls -A .??*.????`
        - List all the ruby files that do not start with a number
            - `ls -A [!0-9]*.rb`
- **How to use brace expansion**
    - For creating entire directory structures in a single command
        - `mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}/new`
        - `mkdir -p books/{fiction{/Alice_in_wonderland.pdf,/The_Maze_runner.pdf},nonfiction/My_lobotomy}` -> Keep in mind, these are folders

Date format

--time-style=+%D (mm/dd/yyyy)