# Lesson 4. Homework

## Hidden files:

1. How to create a hidden file?
   1. .dot file
2. What Linux command can you use to see hidden files in the current dir?
3. ls -a

## CLI structure:

1. How to see all possible options and arguments a command can accept?
   1. **command [-option(s)] [argument(s)]**
   2. Provide an example of a command that expects two arguments as SRC (source) and DST(destination)?
      1. **cp Src\_file Dest\_file**

## File Permissions:

1. Translate the symbolic to numeric representation:
2. - (r-x) (r-x) (r--) - 4 5 5 -> 5 5 4
3. -rwx rwx r-- - > 774 -- > 477
4. -r-------- 777
5. -rw------x - > 444
6. -rw-rw-rw- 666

2. How many bits are needed to represent basic permission (Pick the right answer)

1. 8
2. 10
3. 9
4. 6
5. 3

3. What chmod parameters you will use to set the following permissions for /tmp/file:

chmod [OPTIONS] [ugoa…][-+=]perms…[,…] FILE...

1. -r-xr-xr--
   1. $ chmod u=r,gr=xw,o=xr/tmp/file
2. -rwxrwxr--
   1. $ chmod u=rwx,g=rwx,o=r/tmp/file
3. -r--------
   1. $ chmod u=r,g=0,o=0/tmp/file
4. -rw------x
   1. $ chmod u=rw,g=0,o=x/tmp/file
5. -rw-rw-rw-
   1. $ chmod u=r+w,g=r+wo=r+w/tmp/file

4. What chmod parameters you will use to update permissions for /tmp/file:

1. Add read to all - chmod a+r /tmp/file
   1. chmod a+r/tmp/file -r--r--r--
2. Remove write from the group and other -
   1. chmod g-w,o-w/tmp/file
3. Add execute to all
   1. chmod a+x/temp/file
4. Add read and execute to the owner and other -
   1. chmod uo+rx/tmp/file

## Kernel Space:

1. What is the main purpose of Kernel space?
   1. A kernel is the foundational layer of an **operating system** (OS). It functions at a basic level, communicating with hardware and managing resources, such as RAM and the CPU. Since a kernel handles many fundamental processes, it must be loaded at the beginning of the boot sequence when a computer starts up.
2. Why is process memory isolation important?
   1. The concept of **process isolation** helps to improve operating system security by providing different privilege levels to certain programs and restricting the **memory** those programs can use. ... It can prevent a program from attempting to access key areas of the operating system and modify or otherwise change them.

## Processes:

1. What is a descriptor in a Linux system? Provide some examples.
   1. A file descriptor is a number that uniquely identifies an open file in a computer's operating system. It describes a data resource, and how that resource may be accessed. Like a socket or kernel
2. Name all 3 pipes each process has?
   1. Execute, read, write
3. What process states do you know?
   1. Running: The currently executing process.
   2. Waiting/Blocked: Process waiting for some event such as completion of I/O operation, waiting for other processes, synchronization signal, etc.
   3. Ready: A process that is waiting to be executed.
   4. New: The process that is just being created. The Program Control Block is already being made but the program is not yet loaded in the main memory. The program remains in the new state until the long term scheduler moves the process to the ready state (main memory).
   5. Terminated/Exit: A process that is finished or aborted due to some reason.
4. Which command will you use to see all processes in the system?
   1. ps
5. Which command and what options will you use to see the owner of processes in the system?
   1. ps -aux
6. Which command will you use to see top processes in the system continuously?
   1. ps -axjf
7. Which command will you use to kill processes in the system?
   1. Ctrl +k

## Streams (Pipes):

1. Provide an example of a command line in Linux to read from STDIN? Read var\_name
   1. **read [STDIN] /var/name**
2. Provide an example
   1. ls > var\_name.txt
3. of a command line in Linux to redirect STDOUT to a blackhole pseudo device? Ls 1 > /dev/null
4. Provide an example of a command line in Linux to redirect STDOUT to a blackhole pseudo device and STDERR to error.log?
   1. 1> /dev/null 2> error.log

./error.sh 1> /dev/null 2> error.log

# Optional:

On a virtual instance in GCP:

1. Develop a command line that will be looking for files in /etc folder and subfolders write filename and the owner to
   1. Find /var/name/etc\_list -type f --write chmod -R 755 {} \;
   2. chmod -R ---/name/ect/hosts -root in format - `<filename> - <owner>`, Example: `/etc/hosts - root`
2. Develop Linux command line to start desired (for example 10) Processes.
   1. ‘sleep 5000’
3. Develop a command line that kills (use: `kill -9 <id>`) all `sleep` processes by IDs (do not use pkill or killall commands).
4. How to make a file in Linux immutable?

chattr