# Contents

# Introduction

In this project,  I will explore Linux, primarily focusing on Ubuntu but also exploring other distributions. I'll be cautious about potential challenges with virtualization, documenting the results. Using practical examples and visuals, I will cover fundamental Linux commands and concepts, culminating in some hands-on shell scripting.

# Theory

## Kommandon och symboler

1.

a) ls Lists the contents of a directory. For example ls -l /home/user/ would list all the files and directories in /home/user/ in long format, showing permissions, ownership, size, and modification date.[i]

b) mkdir creates a new directory. mkdir /home/Desktop/new_folder would create a new directory named new_folder in the /home/Desktop/ directory. [ii]

c) mv moves a file or directory, or renames it. Take mv old_name.txt new_name.txt as an example would rename a file from old_name.txt to new_name.txt. If new_name.txt is a directory, old_name.txt would be moved into that directory. [iii]

d) which shows the full path of shell commands. which ls would show the full path of the ls command, typically something like /bin/ls.[iv]

e) cat concatenates and displays the content of files. For instance, cat file.txt would display the content of file.txt to the terminal.[v]

f) chmod changes the file mode bits (permissions) of a file or directory. chmod 755 script.sh would set the permissions of script.sh to be readable and executable by everyone, but only writable by the owner.[vi]

2.

the "|"[vii] symbol is used as a pipe, which allows the output of one command to be used as the input for another. The ">" symbol is used for output redirection, which redirects the standard output of a command to a file, overwriting its contents. The "<" symbol is used for input redirection, which redirects the contents of a file to be used as the input for a command.[viii]

## Grundläggande systemdata

a) uname displays system information. uname -a would give all the system information, including kernel name, kernel release, kernel version, machine, processor, hardware platform, operating system, and the hostname.[ix]

b) free displays the total amount of free and used physical and swap memory in the system, as well as the buffers and caches used by the kernel. Used, free -m would display the memory information in megabytes.[x]

c) df reports the amount of disk space used and available on file systems. How its used, df -h would show the amount of disk space used and available on all mounted file systems with human-readable formats (e.g., in GBs or MBs).[xi]

d) vmstat reports information about processes, memory, paging, block IO, traps, and CPU activity. Usage, vmstat -s would give you a table of various statistics, including memory usage.[xii]

e) /proc File System is A virtual file system that provides detailed information about different aspects of the operating system, including hardware and current running processes. Used to Read different files under /proc gives real-time system information. For instance, cat /proc/meminfo provides detailed memory information, cat /proc/swaps shows swap usage, and cat /proc/cpuinfo provides detailed CPU information.[xiii]

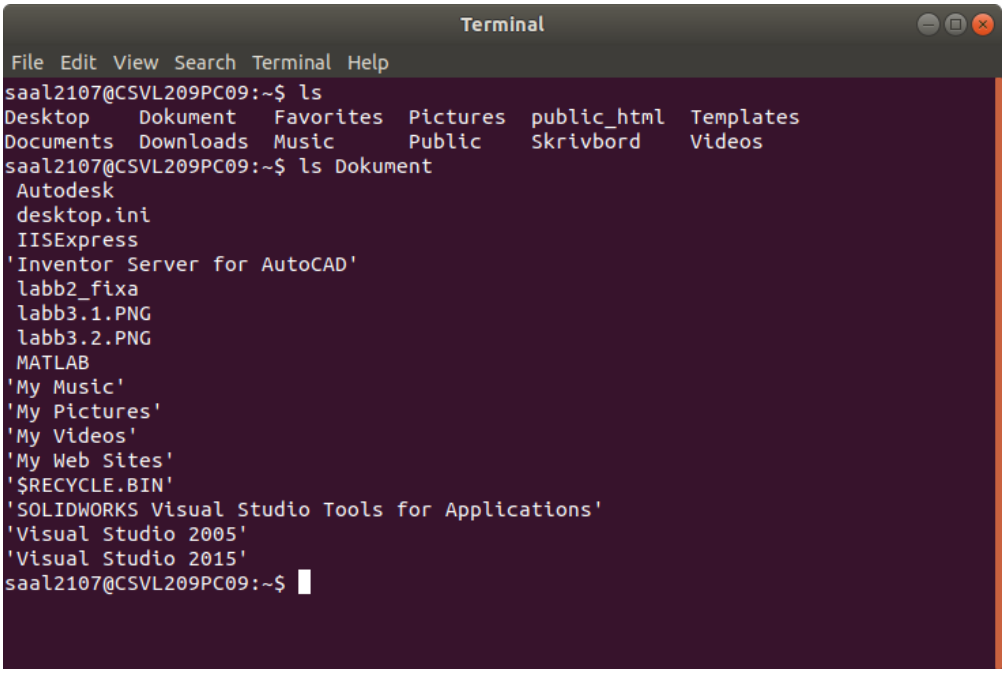
## Prestanda och processer

In the process lifecycle, a new process enters the running or runnable state where it utilizes CPU resources to execute its tasks. If preempted, it enters a run queue, transitioning to a runnable state until its turn for execution. These states are collectively denoted by the character R. During execution, a process may encounter I/O operations, prompting it to enter a sleeping state. This state is further classified into interruptible (S) and uninterruptible (D) sleeping states. While the D state waits for resources, such as network responses, the S state responds to signals and resource availability, like waiting for user input. Processes can also be stopped (T) using signals, with SIGSTOP forcing a stop and SIGTSTP optionally ignorable. A process in the zombie state (Z) has completed execution but remains in limbo until the parent process clears it from the process table by reading its exit value. These states collectively define the dynamic behavior of processes within the operating system.

# Results

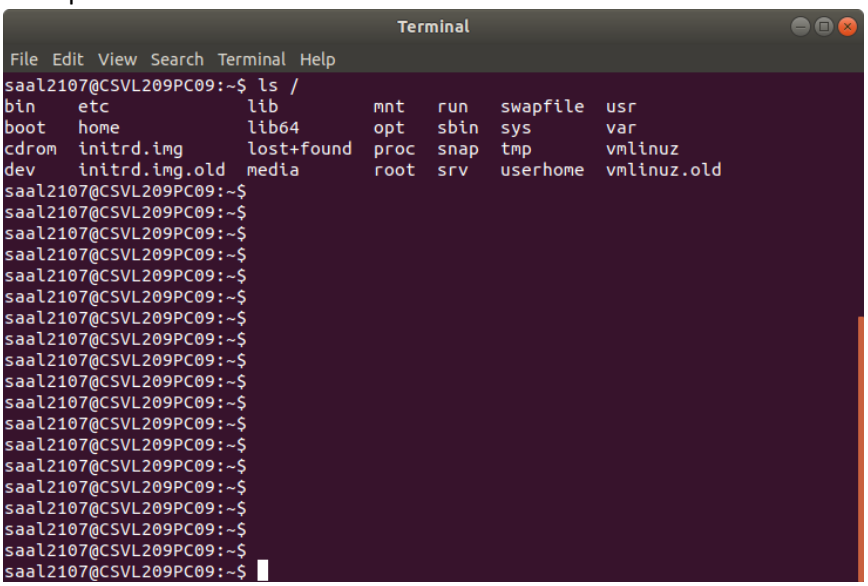
## Kommandon och symboler

1. A) LS: When it comes t[1]o displaying files or directories in Linux and other Unix-based operating system.
   Image:



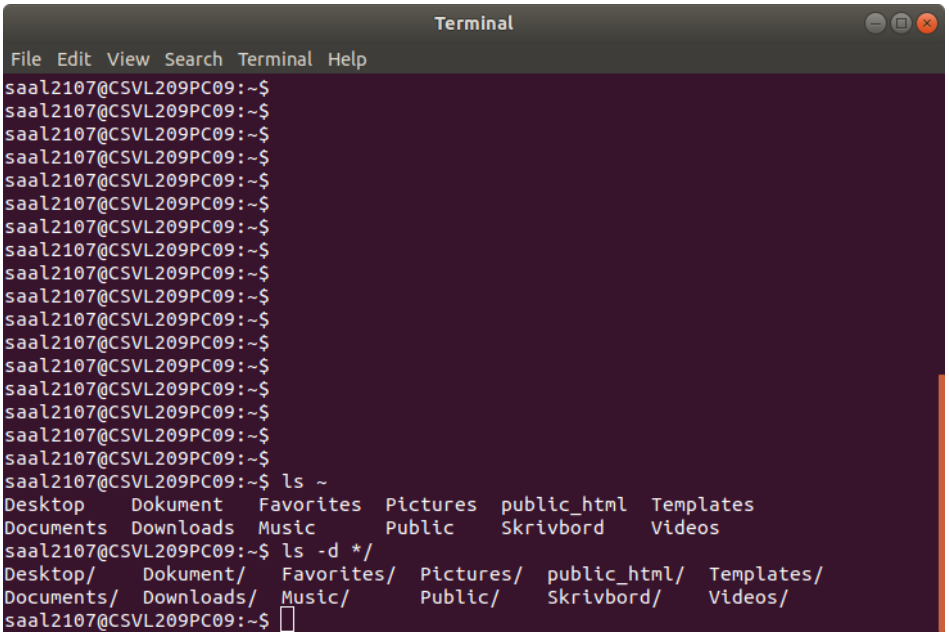

   LS can be used with flags that display files and directories in the terminal differently, or how the command works.

   - List files in a specific directory (ls [directory])
     Example:
   - (Ls /) to show the files in the root directory
     Example:

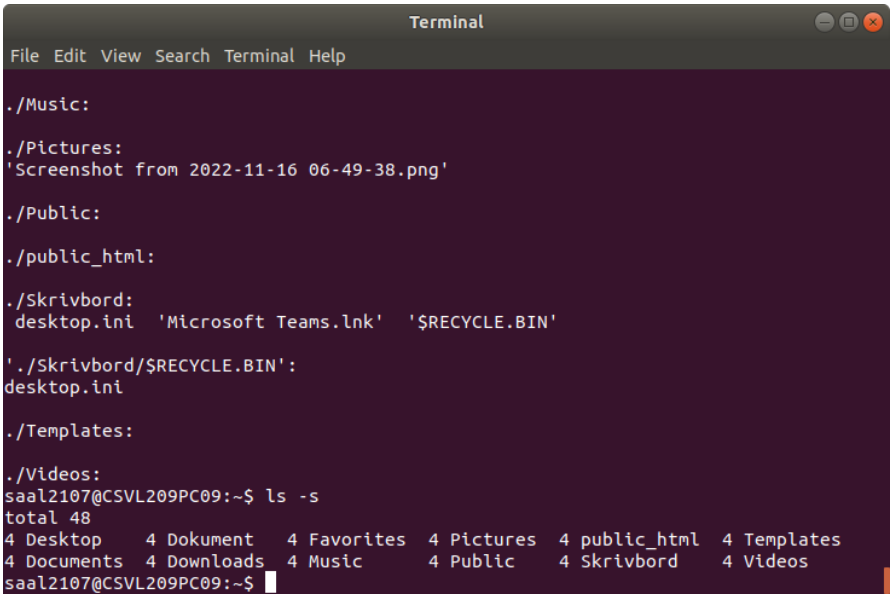



   - Difference between listing files in the home directory and only directories is command which is (ls ~) and (ls -d */) respectively.

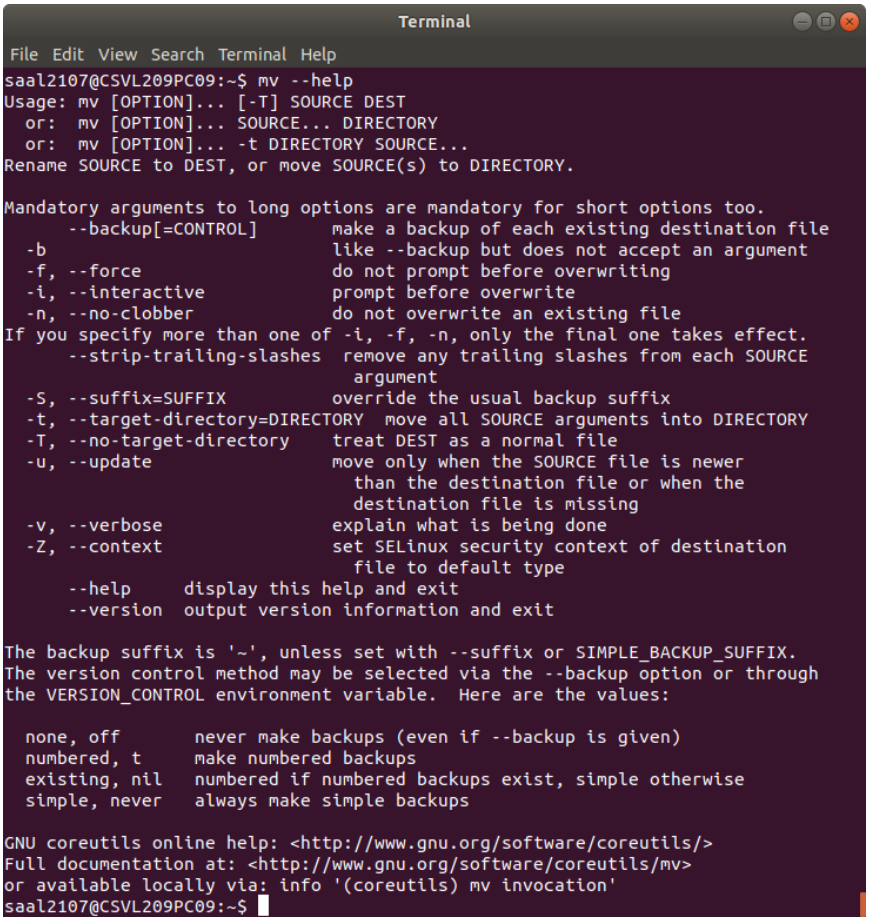
Example:




- To display all files, directories and their subdirectories, use the command (ls -R)
  Example: is difficult to show
- Listing files with their corresponding sizes by using (ls -s)
  Example:



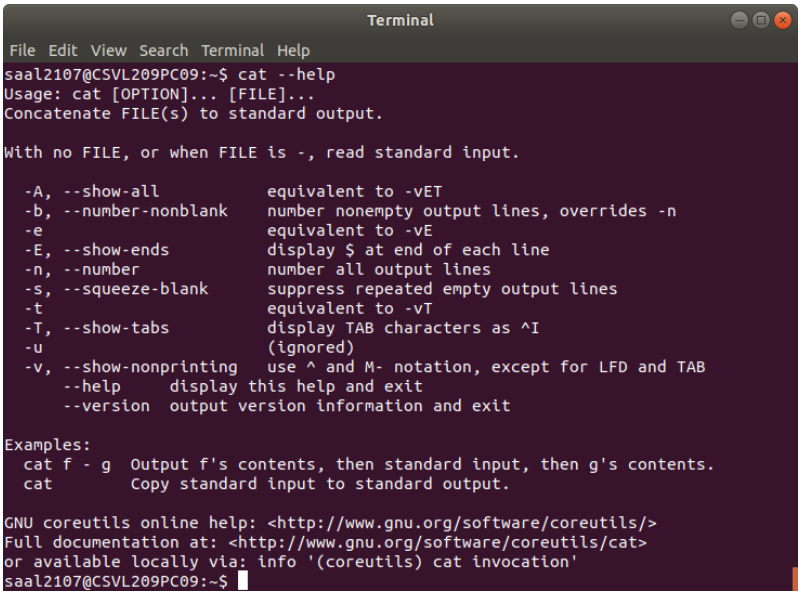

B) MV



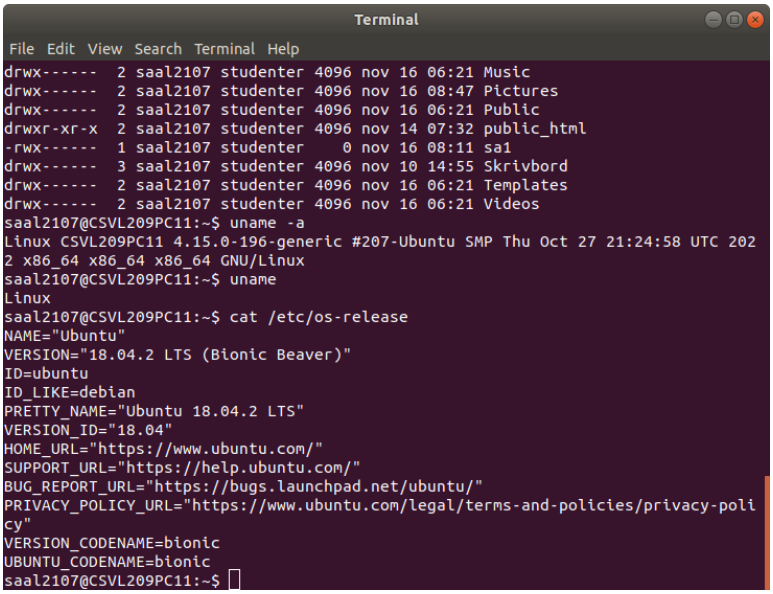

- Moving one or multiple files from one Directory to another, using the command (mv [file_name(s)] [Directory])

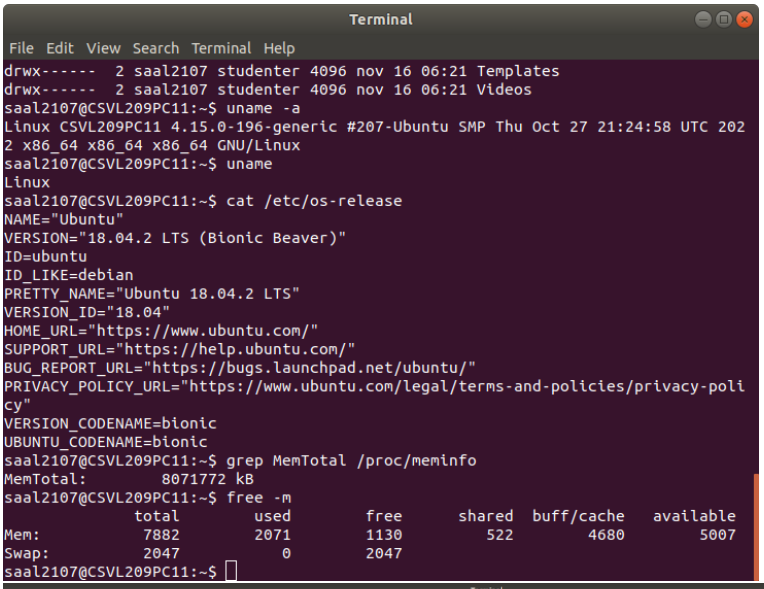
- Rename files and directories by typing (mv [File1/Directory1] [File2/Directory2])
  Where file1/ directory1 is the name of the first file and file2/Directory2 is the new name for it.
- When moving file(s) to a directory where the same files' name(s) exists then the files will overwrite the file(s) in the destinated directory. Therefore, using some commands to either avoid or at least ask before overwriting which is shown as follows:
  - -i/ -n/ -u/ -f/ -b [file] [Directory]
  - -i prompt before overwriting
  - -f doesn't prompt before overwriting
  - -n don't overwrite an existing file

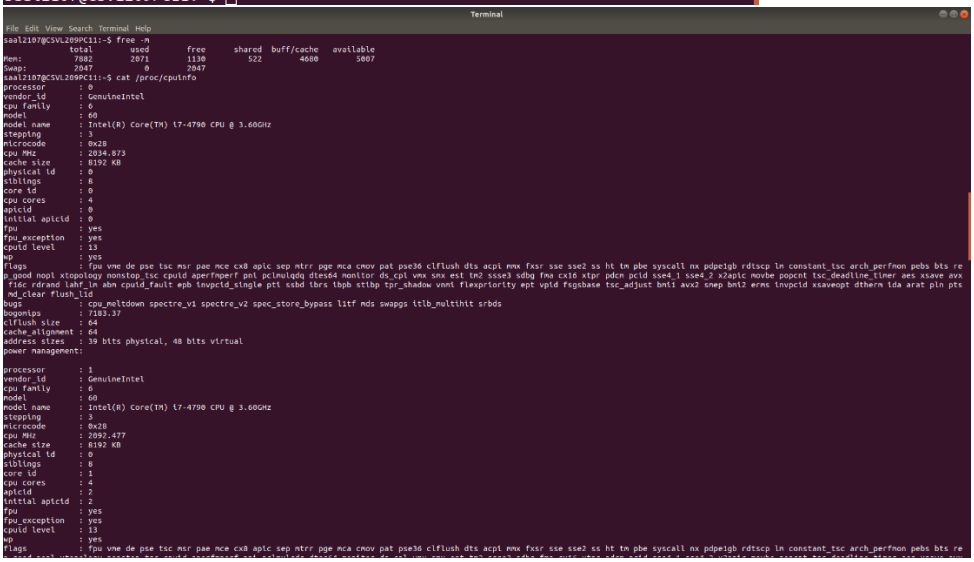
C) CAT



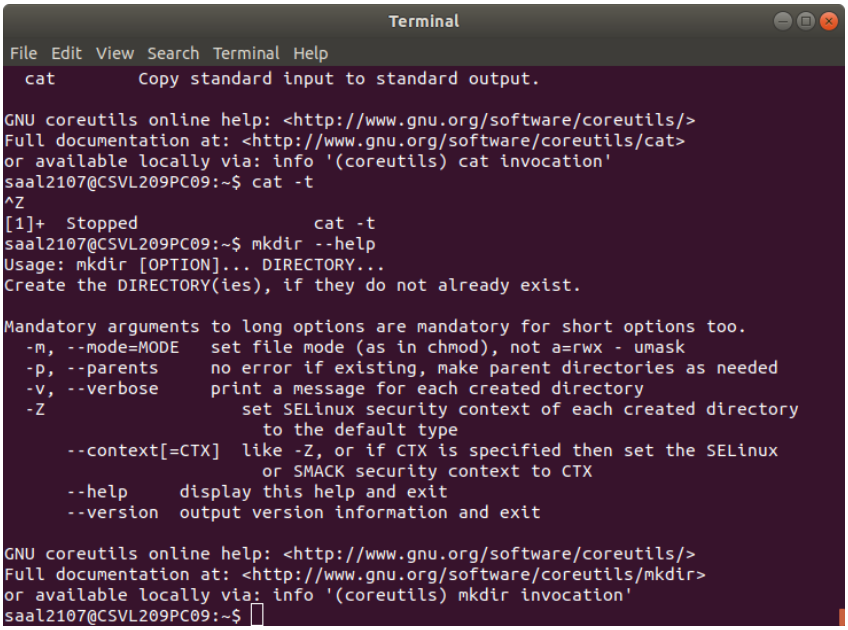

- View: file, multiple files and contents of a file with their corresponding line, command used as follows (cat [file]), (cat [file1] [file2]) and (cat -n [file]) respectively
- To supress repeated empty lines in output command use (cat -s [file]) and if trying to append content of one file to end of another file use command (cat [file1] >> [file2])
- User has also the ability to display contents in reverse or highlight the end of the line by typing the following command (tac [file]) or (cat -E [file])
  Example:





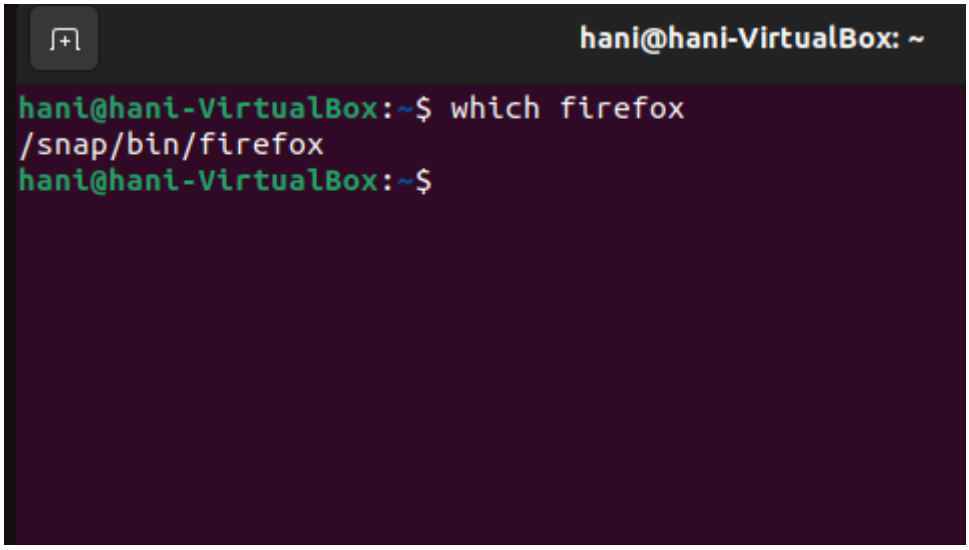




D) mkdir allows the user to create directory or multiple directories and set the permissions for them at the same time




- To display a message after creating every directory with the name of the directory use the command (mkdir -v [directory])
  Example:
- In order to set the files mode of the created directory which specifies the access given for the users. The command used is (mkdir -m a=rwx [Directory]) where r, w, x stand for Read, write, execute respectively.
  Example:

E) “Which” is the command used to determine the location in the PATH environment variable then list the full path of the command specified.




F) “chmod” allows users to set permissions on file(s), it could be by permitting, preventing a file from reading, modifying or executing.

The permission statement is set to a group of people, who are they?

- U: user or owner
- G: Group, members of group the file belongs to

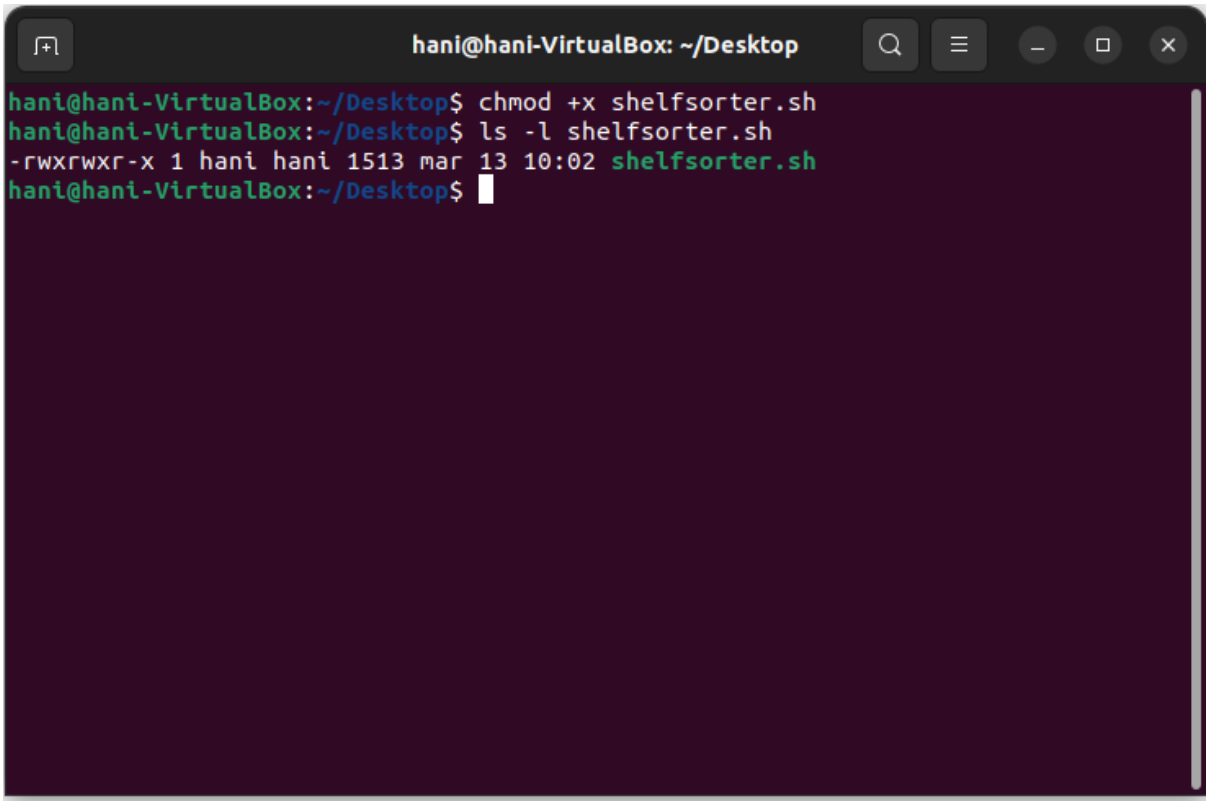
- O: Other, not u or g
- A: All, the previous groups.

What are the permissions set?

- -: removes the permission
- +: to grant the permission
- =: to set the permission

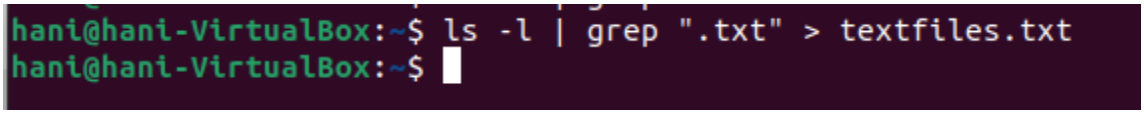
Which permission statements?

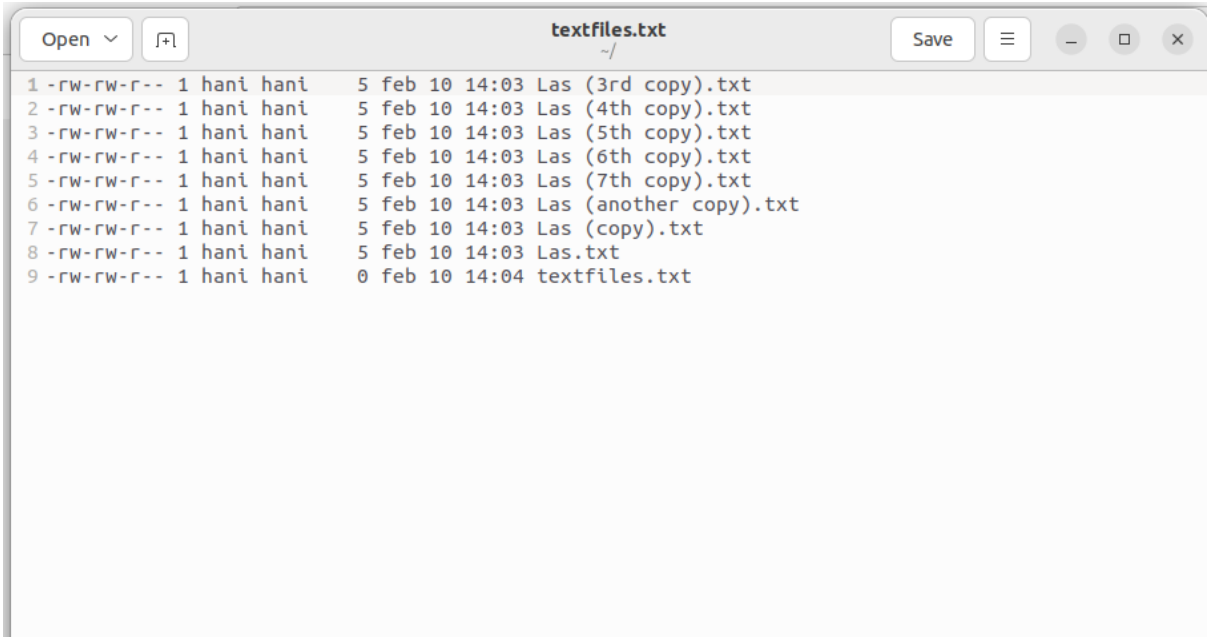
- R: read
- W: write
- X: execute




2.

Example:



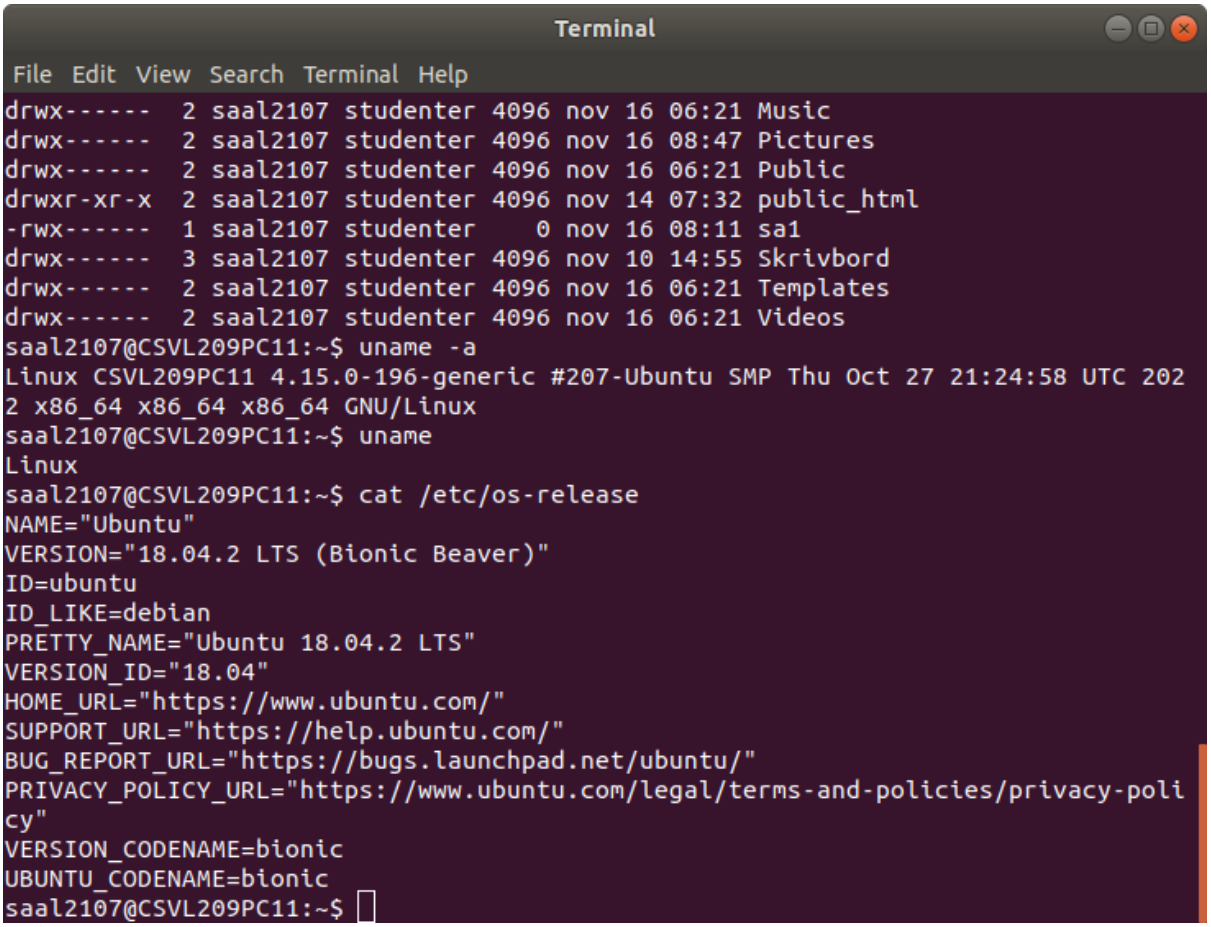



This line performs the following tasks:

ls -l lists the contents of the current directory with detailed information (permissions, owner, size, etc.).
The output of the ls -l command is then passed as input to grep.
grep ".txt" filters the output of ls -l to show only the lines that contain the ".txt" extension.
The output of the grep command is then redirected using the > operator to a file named "textfiles.txt".
In this way, the line performs a simple chore of listing all the files in the current directory with the ".txt" extension and saving the result to a file named "textfiles.txt".

## Grundläggande systemdata

3. Uname, df, free and vmstat
   a) Uname -a: to state operating system, kernel version and hardware.
      Example:

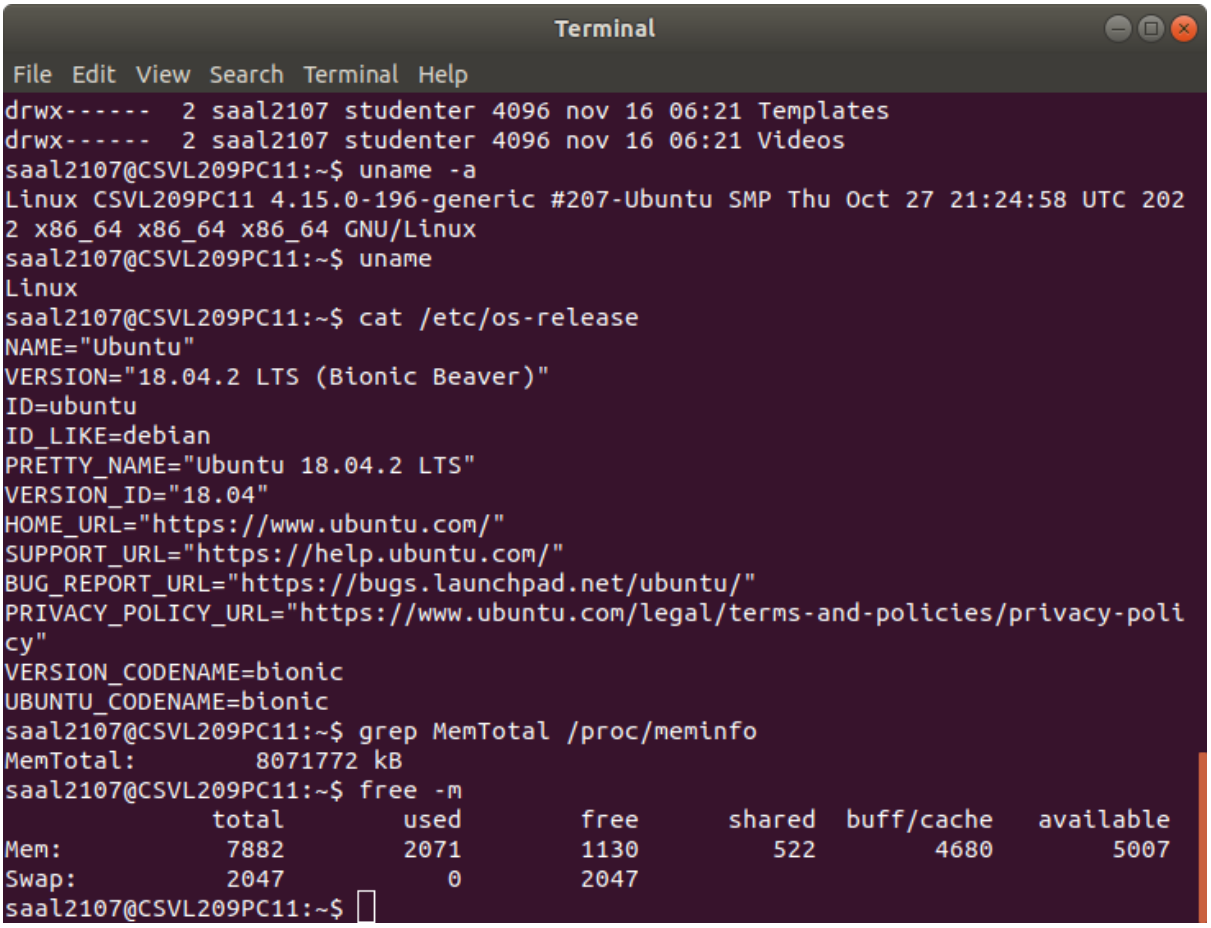



The image shows the operating system used is Linux and version used is “18.04.2 LTS (Bionic Beaver)”

Another way to that is using the command (cat /etc/os-release)

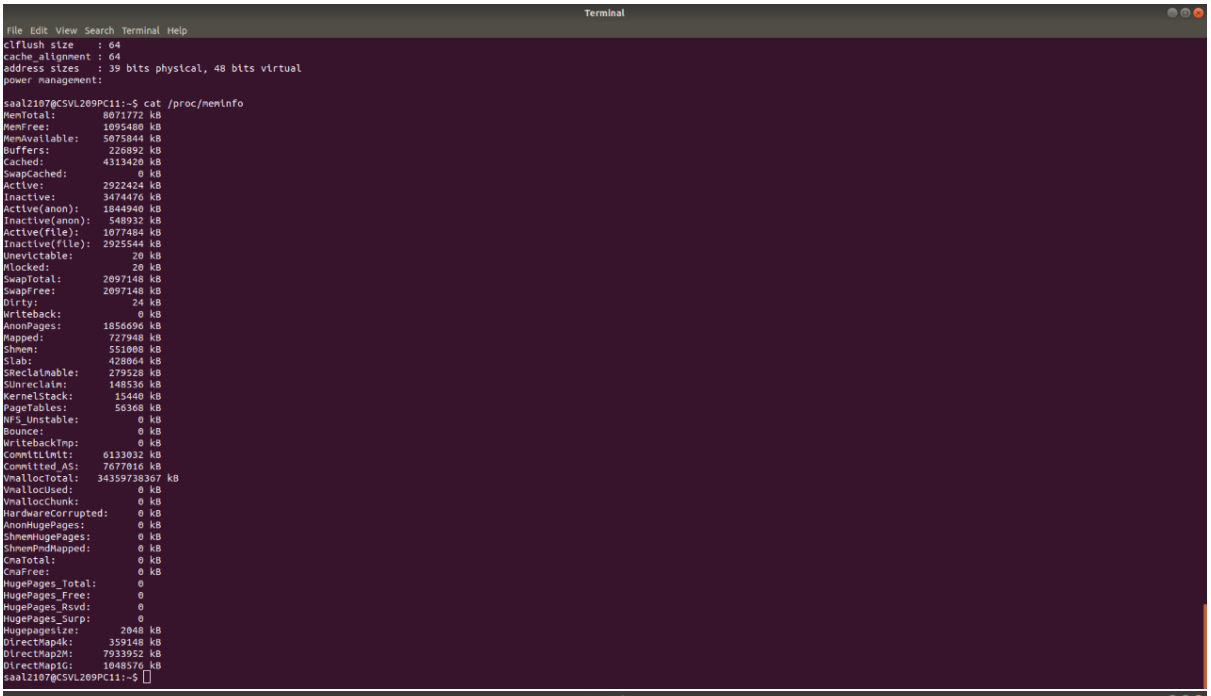
b) Free -m displays the physical and cache memory as shown below:

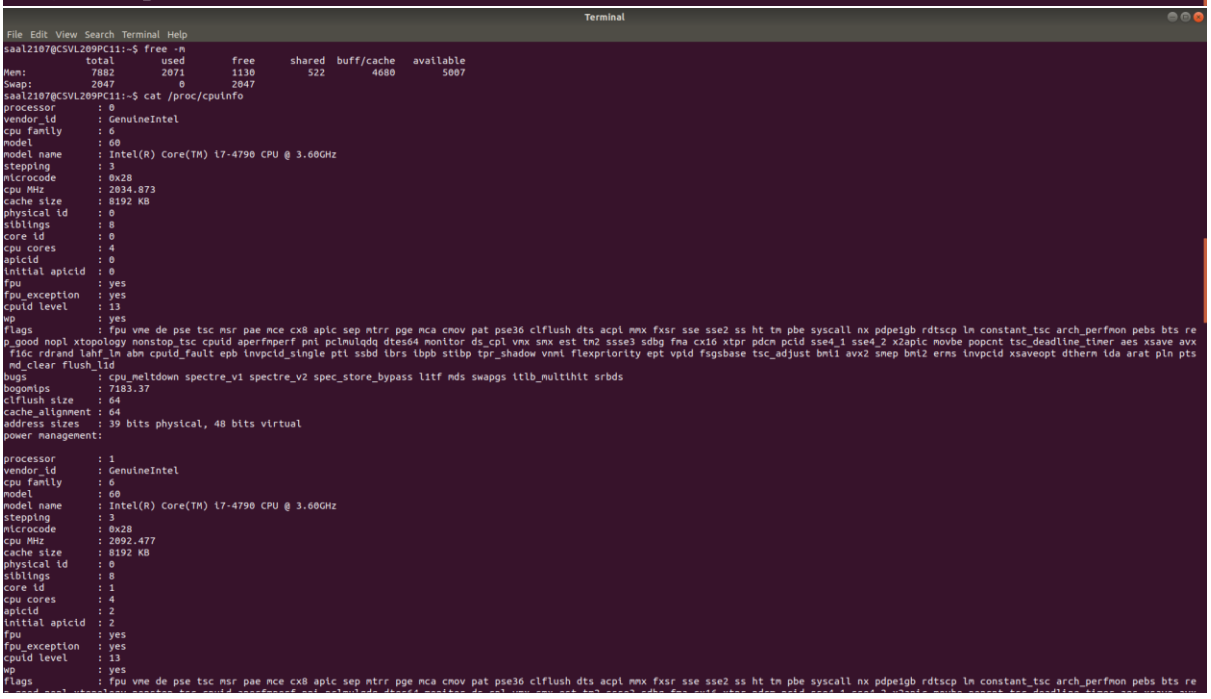
Which means that the total physical memory is 7882 GB



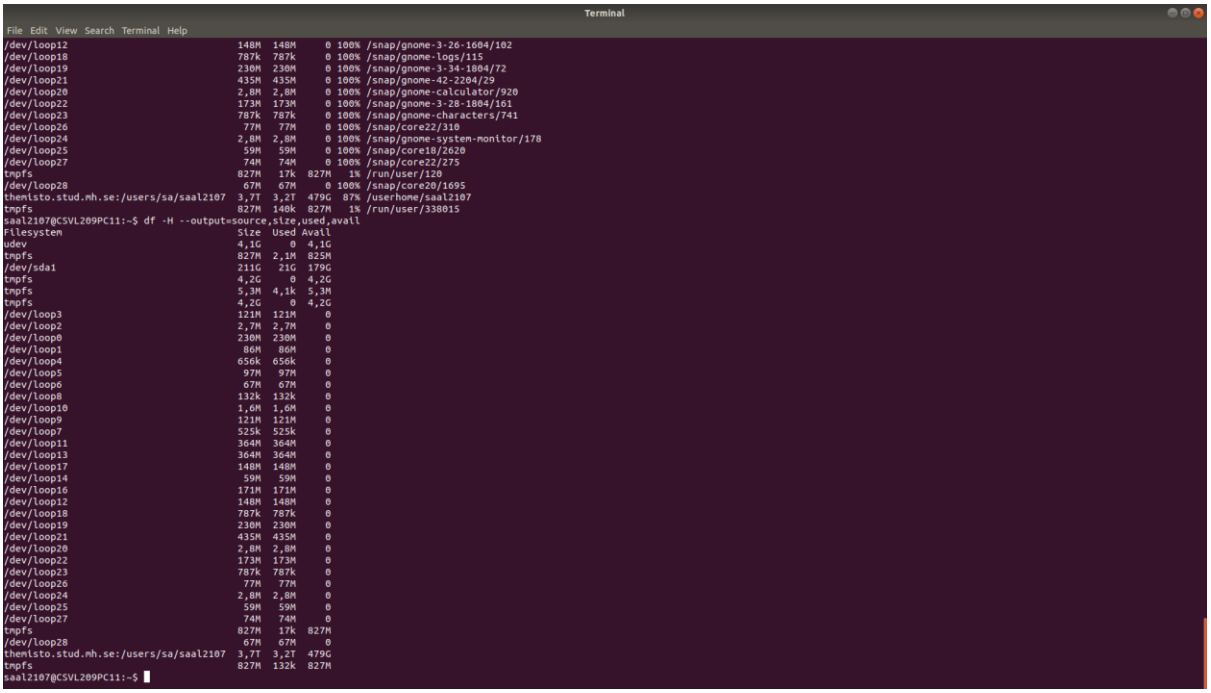

c) Using the previous command to show the swap memory which is 2047 (or inactive memory)

d) To check the primary memory available (cat /proc/meminfo) USE (FREE)
The value of primary memory available for new processes is MemAvailable which is listed as 5075844 kB, which takes into account MemFree, and the memory that can be reclaimed from buffers and cached resources.

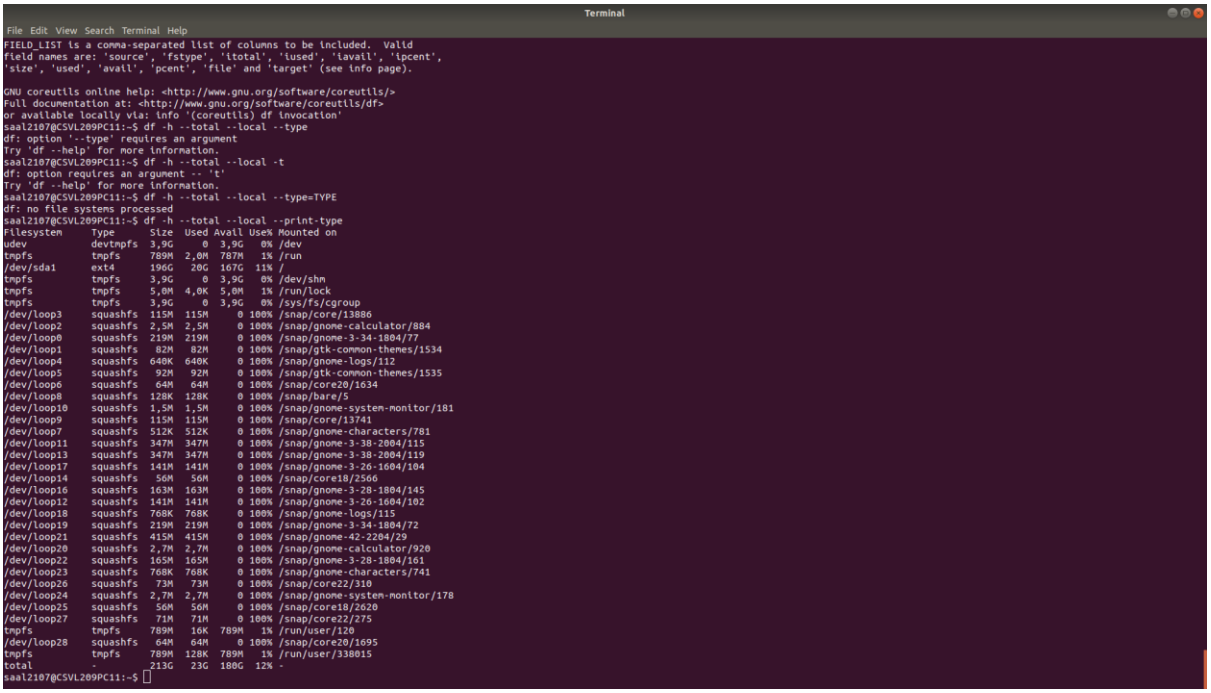



e) The command used to identify the capacity of every file system is (df -h --total) which shows the total capacity size: 213G and the used size: 23G




f) The type of file systems used in the largest local size is: ext4 which is found using the command (df -h –total –local –print-type)




g) Using the "which" command for each command reveals that they are stored in the /usr/bin/ directory.

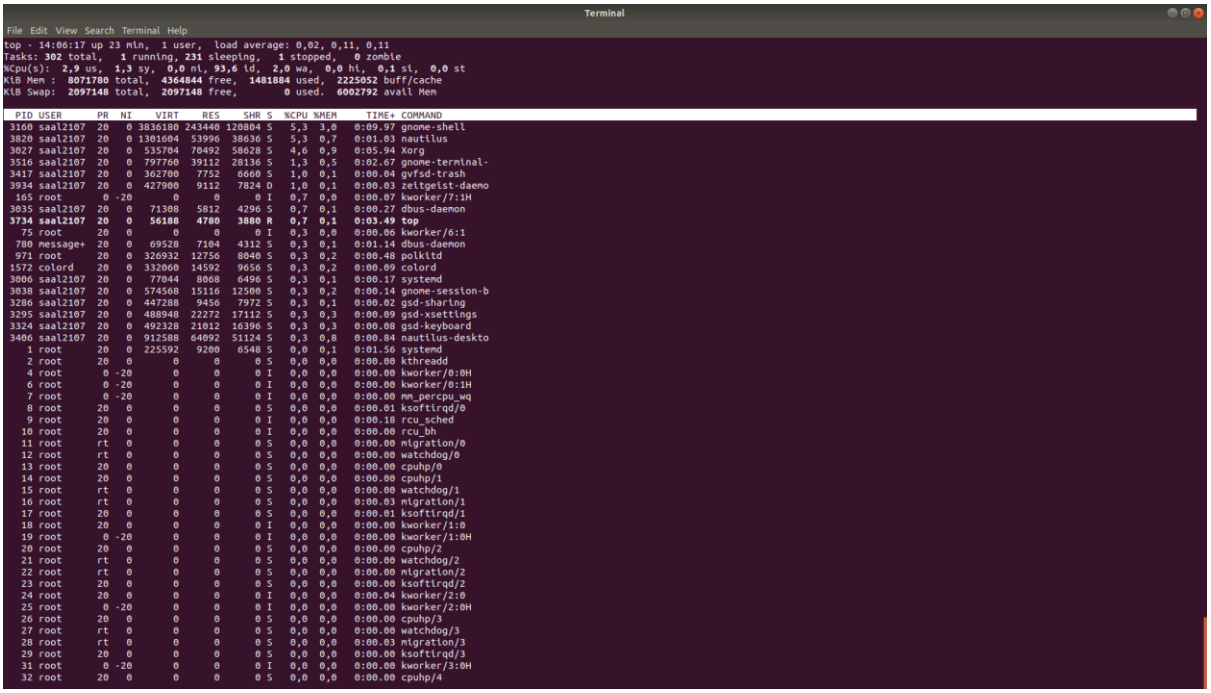
h) To retrieve information about the processor, the command "cat /proc/cpuinfo" is used. This command displays details such as the processor model, which in this case is identified as "Intel® Core™ i7-4790 CPU @ 3.60GHZ". Additionally, it provides information on the cache size, reported as 8192 KB, and the clock frequency (measures the number of cycles your CPU executes per second, measured in GHz) at which the CPU operates, listed as 2194.921 MHz.

## Prestanda och processer

a) The process states in total are:

- Running or Runnable (R)
- Uninterruptible Sleep (D)
- Interruptible Sleep (S)
- Stopped (T)
- Zombie (Z)

The process states found are S, R and I, where “I” stands for “idle” in the new state that was introduced in latest versions of Linux kernel. The Idle state replaces “Task_interruptible”



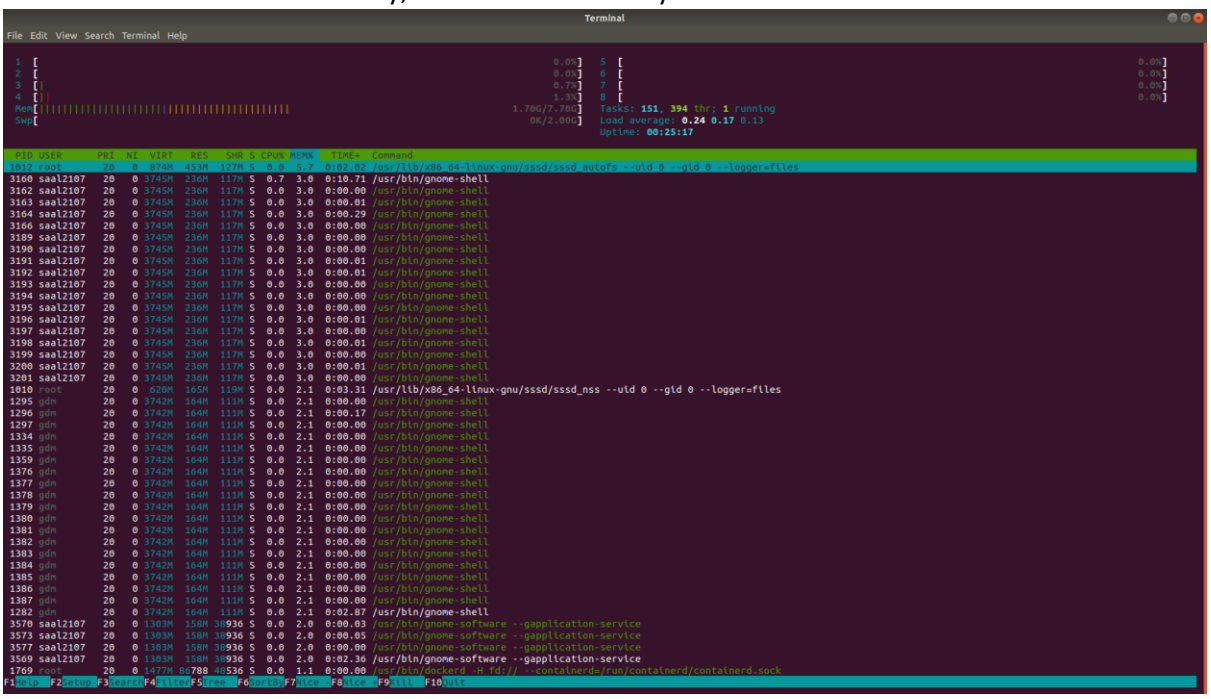

The images shows a total of 302 processors, where 1 is in running state, 231 sleeping, 1 stopped and 0 zombies

b) Running (R): The process is either running on a CPU or waiting to run.
Sleeping (S): The process is waiting for an event or for a resource.
Uninterruptible Sleep (D): The process is waiting for I/O and cannot be interrupted.
Zombie (Z): The process has completed execution but still has an entry in the process table.
Stopped (T): The process is stopped, often by a signal.[xiv]

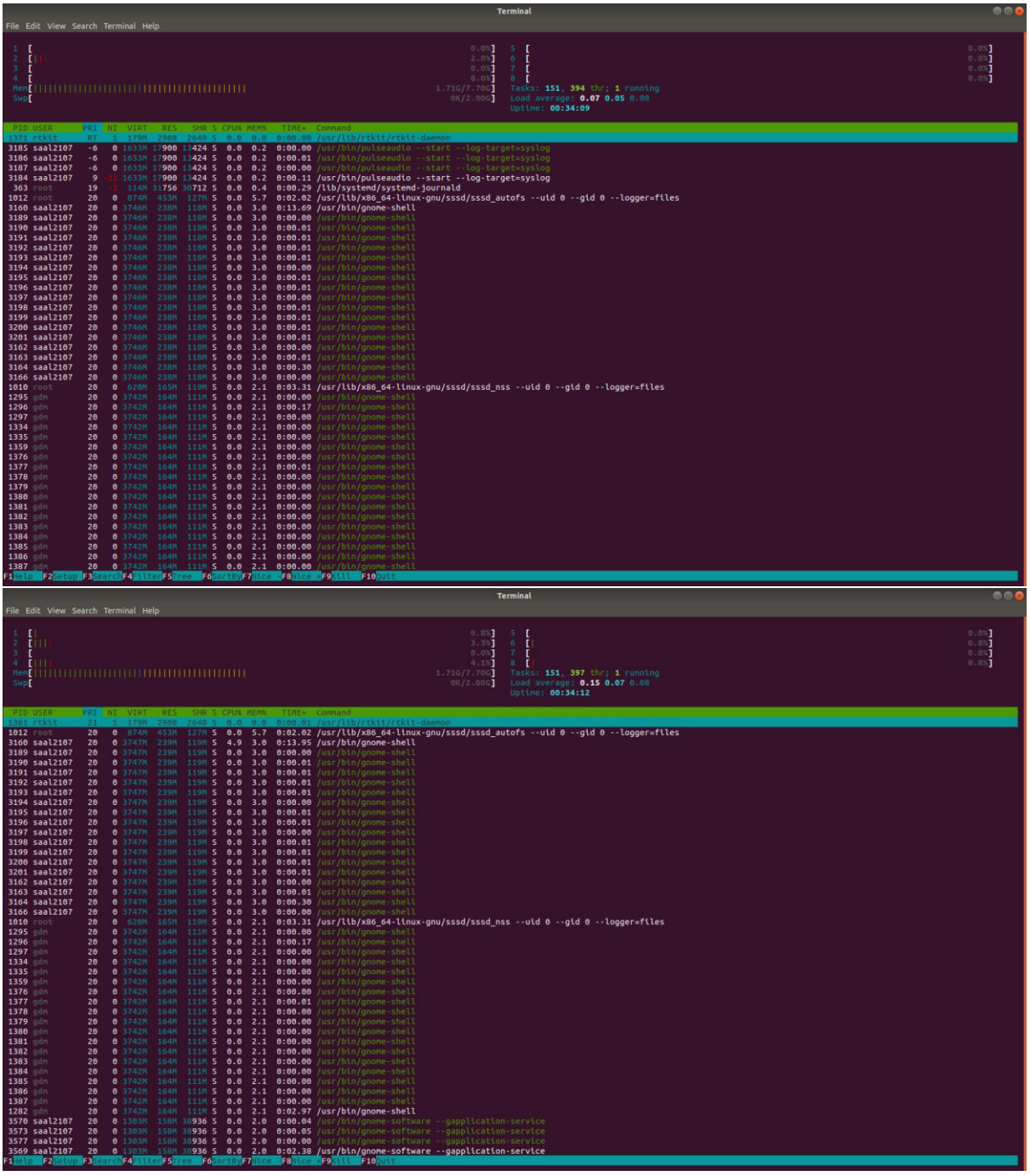
PID (Process ID) represents the unique identifier of a process. USER denotes the username of the user who initiated the process. PR, short for "Priority," displays scheduling priorities from the kernel's perspective. It is calculated by adding +20 to the process's "niceness" value (maps to a kernel call of the same name). A value indicating "rt" signifies that the process is running in real-time. NI shows the "niceness" value of

the process. Niceness is how the Linux system measures priority. A process with a **high** Niceness has **low** priority, while processes with **negative** Niceness are **highest** priority. Processes inherit priority from a parent process. **RES** indicates the amount of **physical** memory a process is using. **SHR** displays the amount of **memory shared** with other processes. **S** displays the processor **state** in the form of a letter, where **D** is **uninterruptible** sleep, I is idle, R is running, S is sleeping (waiting for user input), T is stopped by a control signal, t is stopped by a debugger during a trace, and finally, Z denotes a zombie process that wants to terminate but needs to be killed by its parent. %CPU represents the percentage of CPU time used, while %MEM signifies the same for physical memory. TIME+ denotes the total CPU time used by a process since it started. Lastly, COMMAND represents the name of the process.

c) 1012 root uses the most memory, uses 5.7% of memory




d) The lowest value for PRI is "pipew" at 20. Otherwise, the highest priority is at -11. The maximum value is 19 for "gnome".




e) PR (Priority): The kernel's internal scheduling priority.
NI (Nice): A user-space concept to adjust the priority of a process.

f) f) Theoretical Framework: The nice value influences a process's scheduling precedence. A greater nice value (toward the positive) correlates with a diminished scheduling priority (manifesting as an elevated PR value), whereas a lesser nice value (toward the negative) confers enhanced priority (evidenced by a reduced PR value). It should be noted that a PRI-NI differential of 20 emerges solely in instances where the PRI exhibits a negative magnitude, exclusive of real-time processes.

g) In this instance, the Process ID (PID) for Firefox is 2216. The nice value of this process was successfully altered to 10 with the command renice 10 2216. While attempts to raise the niceness to 100 failed, reflecting the system-enforced maximum of 19, it's also critical to note that once the nice value of a process is increased within the 0 to 19 range, it cannot be decreased again in this session, indicating a one-way adjustment limitation. Similarly, attempts to set the nice value to -19 reached the lower limit, and my permissions restrict setting a niceness lower than 10. Therefore, it's crucial to ensure that the desired niceness level is set correctly on the first attempt, as further decreases are not permitted.

h) Pressing [Ctrl + Z] suspends the process, transitioning it into the background and **halting** its execution, which results in an uninterruptible stop state for the application—indicated by the 'T' status in htop. If we issue the `**fg**` command, this suspension is interrupted, and the process **resumes**, bringing it back into the foreground for user interaction.

i) When closing the terminal, it also closes Firefox because the terminal acts as a parent process to Firefox; terminating the terminal effectively kills its child processes. This occurs as closing the terminal issues a SIGHUP signal to the foreground process group, leading to the termination of associated processes if they have not been disowned or moved to the background.

## References


[1] "ls(1) - Linux man page," man7.org. [Online]. Available: https://man7.org/linux/man-pages/man1/ls.1.html [Accessed: Mar. 17, 2024].

[2] "mkdir(1) - Linux man page," man7.org. [Online]. Available: https://man7.org/linux/man-pages/man1/mkdir.1.html . [Accessed: Mar. 17, 2024].

[3] "mv(1) - Linux man page," man7.org. [Online]. Available: https://man7.org/linux/man-pages/man1/mv.1.html . [Accessed: Mar. 17, 2024].

[4] "Which Command in Linux With Examples," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/which-command-in-linux-with-examples/ . [Accessed: Mar. 17, 2024].

[5] "Linux cat," javatpoint. [Online]. Available: https://www.javatpoint.com/linux-cat . [Accessed: Mar. 17, 2024].

[6] "What Is chmod?," Nexcess. [Online]. Available: https://www.nexcess.net/help/what-is-chmod/ . [Accessed: Mar. 17, 2024].

[7] "Piping in Unix or Linux," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/piping-in-unix-or-linux/ . [Accessed: Mar. 17, 2024].

[8] "Linux Redirection," guru99. [Online]. Available: https://www.guru99.com/linux-redirection.html . [Accessed: Mar. 17, 2024].

[9] "uname," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Uname . [Accessed: Mar. 17, 2024].

[10] "How to Use the Linux free Command," Turing. [Online]. Available: https://www.turing.com/kb/how-to-use-the-linux-free-command . [Accessed: Mar. 17, 2024].

[11] "How to use the df command," Red Hat. [Online]. Available: https://www.redhat.com/sysadmin/linux-df-command . [Accessed: Mar. 17, 2024].

[12] "How to Use the vmstat Command," phoenixNAP. [Online]. Available: https://phoenixnap.com/kb/vmstat-command . [Accessed: Mar. 17, 2024].

[13] "The Linux Filesystem Hierarchy," tldp.org. [Online]. Available: https://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html . [Accessed: Mar. 17, 2024].

[14] "Understanding Linux Process States," Baeldung. [Online]. Available: https://www.baeldung.com/linux/process-states . [Accessed: Mar. 17, 2024].