<u>Question 4 (2024eb03003):</u>

You must perform all operations without making any configuration changes to the system.

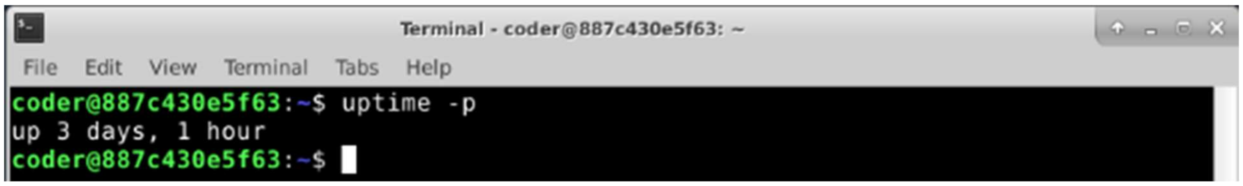
These tasks should be executed within your own user account. Since uptime, processes, memory usage,disk usage, and background jobs vary across systems and users, each student's output will be unique.

1. System Uptime Verification

Display the time elapsed since the system was last booted.

uptime -p

- uptime shows how long the system has been running since the last boot.
- The -p option prints this elapsed time in a human-readable "pretty" format, such as up 3 days, 1 hours.


Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

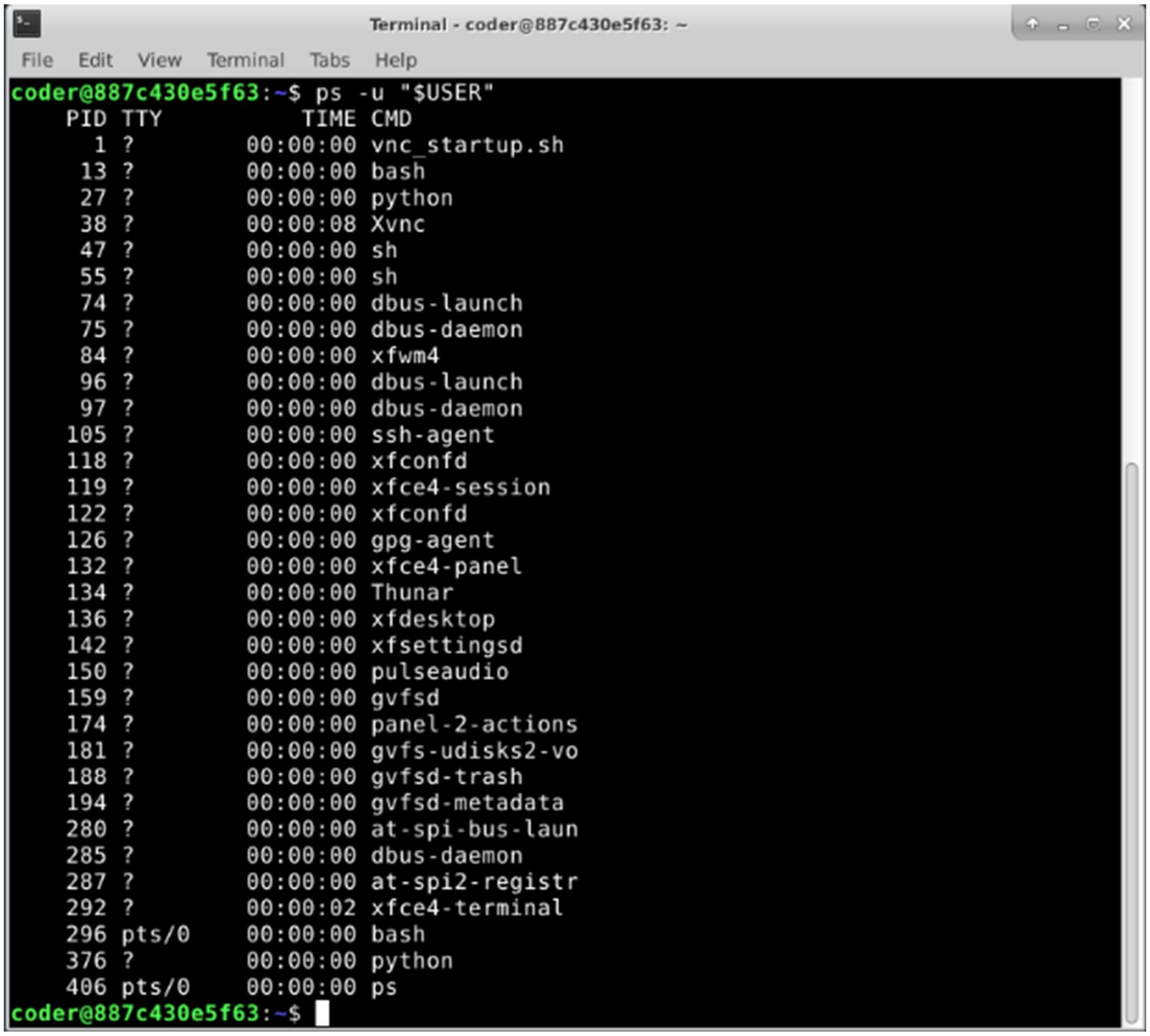
```
coder@887c430e5f63:~$ uptime -p
up 3 days, 1 hour
coder@887c430e5f63:~$ 
```

2. User Process Listing

List all processes currently running under your user account.

ps -u "$USER"

- ps shows process status.
- -u "$USER" restricts the list to processes owned by user account, displaying its PID, CPU time, and the command that started each process.


Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

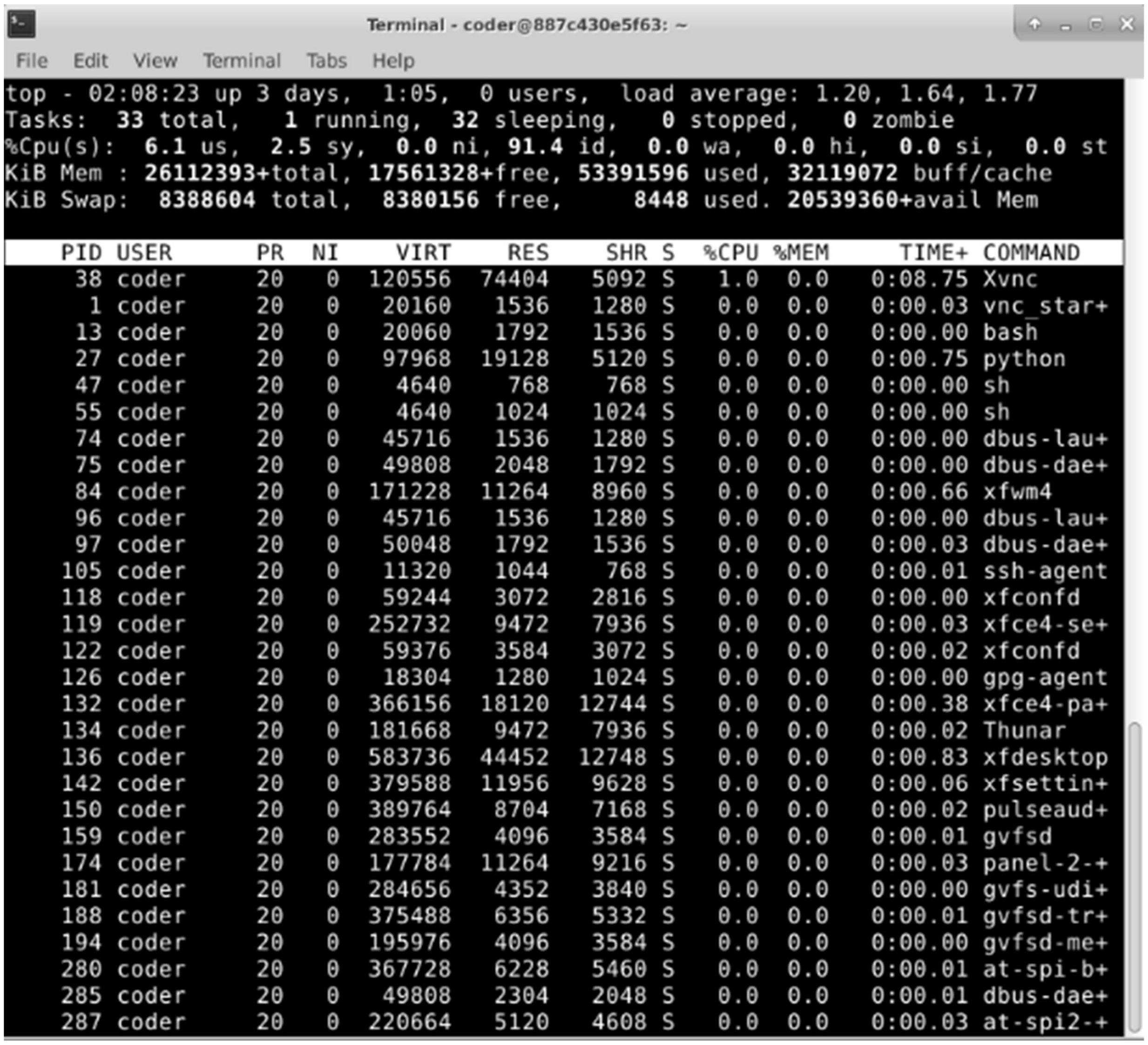
```
coder@887c430e5f63:~$ ps -u "$USER"
    PID TTY          TIME CMD
      1 ?        00:00:00 vnc_startup.sh
     13 ?        00:00:00 bash
     27 ?        00:00:00 python
     38 ?        00:00:08 Xvnc
     47 ?        00:00:00 sh
     55 ?        00:00:00 sh
     74 ?        00:00:00 dbus-launch
     75 ?        00:00:00 dbus-daemon
     84 ?        00:00:00 xfwm4
     96 ?        00:00:00 dbus-launch
     97 ?        00:00:00 dbus-daemon
    105 ?        00:00:00 ssh-agent
    118 ?        00:00:00 xfconfd
    119 ?        00:00:00 xfce4-session
    122 ?        00:00:00 xfconfd
    126 ?        00:00:00 gpg-agent
    132 ?        00:00:00 xfce4-panel
    134 ?        00:00:00 Thunar
    136 ?        00:00:00 xfdesktop
    142 ?        00:00:00 xfsettingsd
    150 ?        00:00:00 pulseaudio
    159 ?        00:00:00 gvfsd
    174 ?        00:00:00 panel-2-actions
    181 ?        00:00:00 gvfs-udisks2-vo
    188 ?        00:00:00 gvfsd-trash
    194 ?        00:00:00 gvfsd-metadata
    280 ?        00:00:00 at-spi-bus-laun
    285 ?        00:00:00 dbus-daemon
    287 ?        00:00:00 at-spi2-registr
    292 ?        00:00:02 xfce4-terminal
    296 pts/0    00:00:00 bash
    376 ?        00:00:00 python
    406 pts/0    00:00:00 ps
coder@887c430e5f63:~$
```

3. CPU Usage Analysis

Identify the process that is consuming the highest CPU usage among your running processes.

top -o %CPU

- top shows running processes in real time, with a %CPU column.
- -o %CPU sorts processes by CPU usage, so the first process in the list is the one consuming the most CPU among all processes, including mine.

- 

Terminal - coder@887c430e5f63: ~

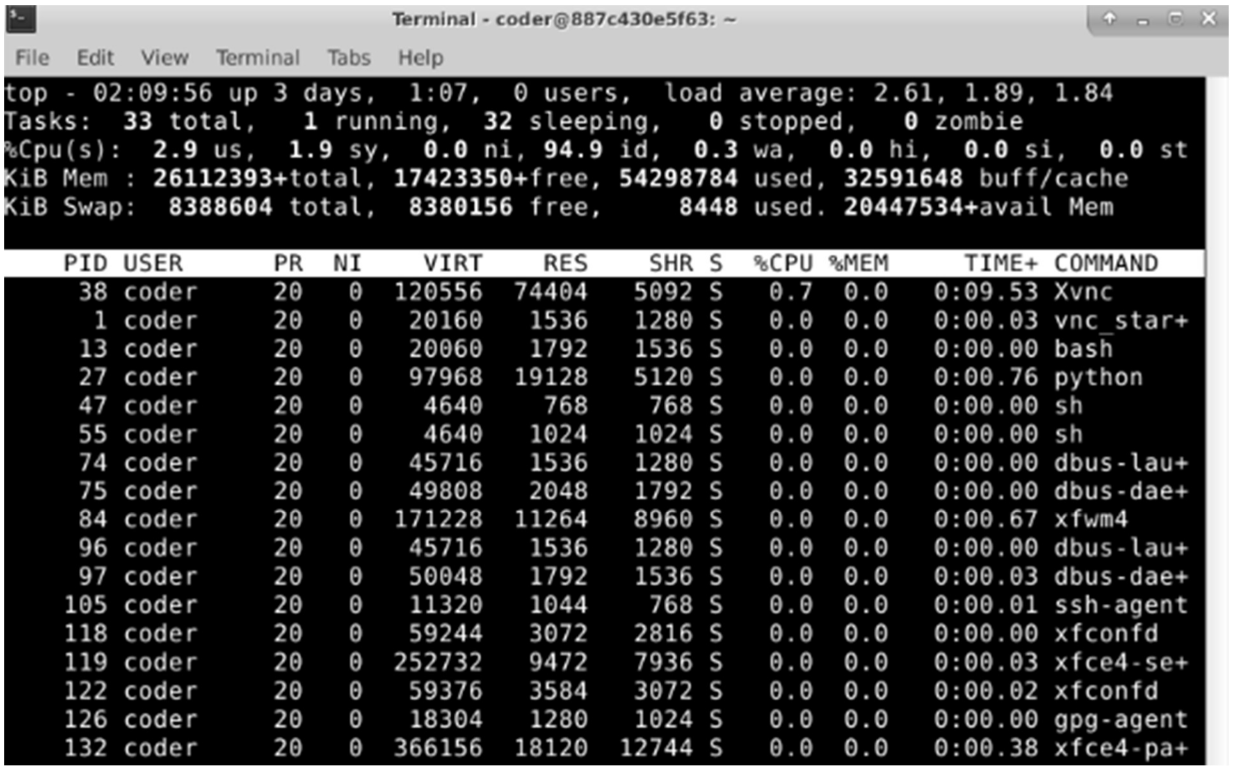
File Edit View Terminal Tabs Help

```
top - 02:08:23 up 3 days,  1:05,  0 users,  load average: 1.20, 1.64, 1.77
Tasks:  33 total,   1 running,  32 sleeping,   0 stopped,   0 zombie
%Cpu(s):  6.1 us,  2.5 sy,  0.0 ni, 91.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 26112393+total, 17561328+free, 53391596 used, 32119072 buff/cache
KiB Swap:  8388604 total,  8380156 free,     8448 used. 20539360+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   38 coder     20   0  120556  74404   5092 S   1.0  0.0   0:08.75 Xvnc
    1 coder     20   0   20160   1536   1280 S   0.0  0.0   0:00.03 vnc_star+
   13 coder     20   0   20060   1792   1536 S   0.0  0.0   0:00.00 bash
   27 coder     20   0   97968  19128   5120 S   0.0  0.0   0:00.75 python
   47 coder     20   0    4640    768    768 S   0.0  0.0   0:00.00 sh
   55 coder     20   0    4640   1024   1024 S   0.0  0.0   0:00.00 sh
   74 coder     20   0   45716   1536   1280 S   0.0  0.0   0:00.00 dbus-lau+
   75 coder     20   0   49808   2048   1792 S   0.0  0.0   0:00.00 dbus-dae+
   84 coder     20   0  171228  11264   8960 S   0.0  0.0   0:00.66 xfwm4
   96 coder     20   0   45716   1536   1280 S   0.0  0.0   0:00.00 dbus-lau+
   97 coder     20   0   50048   1792   1536 S   0.0  0.0   0:00.03 dbus-dae+
  105 coder     20   0   11320   1044    768 S   0.0  0.0   0:00.01 ssh-agent
  118 coder     20   0   59244   3072   2816 S   0.0  0.0   0:00.00 xfconfd
  119 coder     20   0  252732   9472   7936 S   0.0  0.0   0:00.03 xfce4-se+
  122 coder     20   0   59376   3584   3072 S   0.0  0.0   0:00.02 xfconfd
  126 coder     20   0   18304   1280   1024 S   0.0  0.0   0:00.00 gpg-agent
  132 coder     20   0  366156  18120  12744 S   0.0  0.0   0:00.38 xfce4-pa+
  134 coder     20   0  181668   9472   7936 S   0.0  0.0   0:00.02 Thunar
  136 coder     20   0  583736  44452  12748 S   0.0  0.0   0:00.83 xfdesktop
  142 coder     20   0  379588  11956   9628 S   0.0  0.0   0:00.06 xfsettin+
  150 coder     20   0  389764   8704   7168 S   0.0  0.0   0:00.02 pulseaud+
  159 coder     20   0  283552   4096   3584 S   0.0  0.0   0:00.01 gvfsd
  174 coder     20   0  177784  11264   9216 S   0.0  0.0   0:00.03 panel-2-+
  181 coder     20   0  284656   4352   3840 S   0.0  0.0   0:00.00 gvfs-udi+
  188 coder     20   0  375488   6356   5332 S   0.0  0.0   0:00.01 gvfsd-tr+
  194 coder     20   0  195976   4096   3584 S   0.0  0.0   0:00.00 gvfsd-me+
  280 coder     20   0  367728   6228   5460 S   0.0  0.0   0:00.01 at-spi-b+
  285 coder     20   0   49808   2304   2048 S   0.0  0.0   0:00.01 dbus-dae+
  287 coder     20   0  220664   5120   4608 S   0.0  0.0   0:00.03 at-spi2-+
```

To restrict to only my user's processes, note my username (say coder) and run:

top -u "$coder" -o %CPU

Now the top row in the display is the highest-CPU process owned by my account, showing its PID, command name, and live CPU usage percentage.


Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

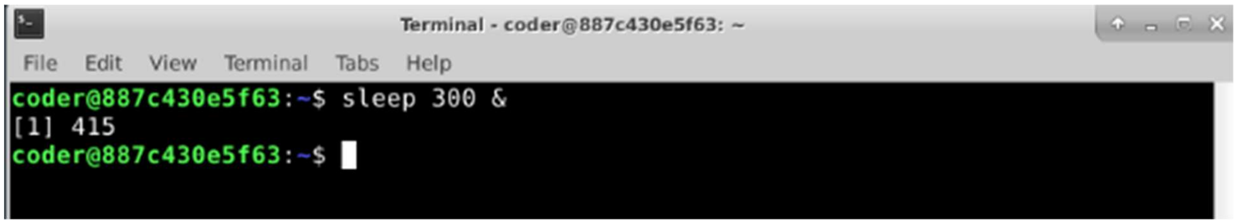
```
top - 02:09:56 up 3 days,  1:07,  0 users,  load average: 2.61, 1.89, 1.84
Tasks:  33 total,   1 running,  32 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.9 us,  1.9 sy,  0.0 ni, 94.9 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 26112393+total, 17423350+free, 54298784 used, 32591648 buff/cache
KiB Swap:  8388604 total,  8380156 free,     8448 used. 20447534+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   38 coder     20   0  120556  74404   5092 S   0.7  0.0   0:09.53 Xvnc
    1 coder     20   0   20160   1536   1280 S   0.0  0.0   0:00.03 vnc_star+
   13 coder     20   0   20060   1792   1536 S   0.0  0.0   0:00.00 bash
   27 coder     20   0   97968  19128   5120 S   0.0  0.0   0:00.76 python
   47 coder     20   0    4640    768    768 S   0.0  0.0   0:00.00 sh
   55 coder     20   0    4640   1024   1024 S   0.0  0.0   0:00.00 sh
   74 coder     20   0   45716   1536   1280 S   0.0  0.0   0:00.00 dbus-lau+
   75 coder     20   0   49808   2048   1792 S   0.0  0.0   0:00.00 dbus-dae+
   84 coder     20   0  171228  11264   8960 S   0.0  0.0   0:00.67 xfwm4
   96 coder     20   0   45716   1536   1280 S   0.0  0.0   0:00.00 dbus-lau+
   97 coder     20   0   50048   1792   1536 S   0.0  0.0   0:00.03 dbus-dae+
  105 coder     20   0   11320   1044    768 S   0.0  0.0   0:00.01 ssh-agent
  118 coder     20   0   59244   3072   2816 S   0.0  0.0   0:00.00 xfconfd
  119 coder     20   0  252732   9472   7936 S   0.0  0.0   0:00.03 xfce4-se+
  122 coder     20   0   59376   3584   3072 S   0.0  0.0   0:00.02 xfconfd
  126 coder     20   0   18304   1280   1024 S   0.0  0.0   0:00.00 gpg-agent
  132 coder     20   0  366156  18120  12744 S   0.0  0.0   0:00.38 xfce4-pa+
```

4. Background Process Execution

Start a command in the background and verify that it is running.

sleep 300 &

- sleep 300 & starts a 5-minute sleep process in the background and immediately returns a prompt.

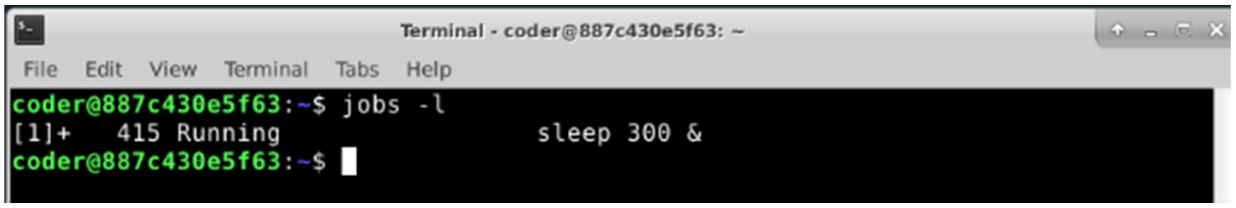

Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

```
coder@887c430e5f63:~$ sleep 300 &
[1] 415
coder@887c430e5f63:~$ 
```

Verify it is running:

jobs -l

- jobs -l shows background jobs for the current shell, including their job ID, PID, and status (e.g., Running), confirming background process is active.



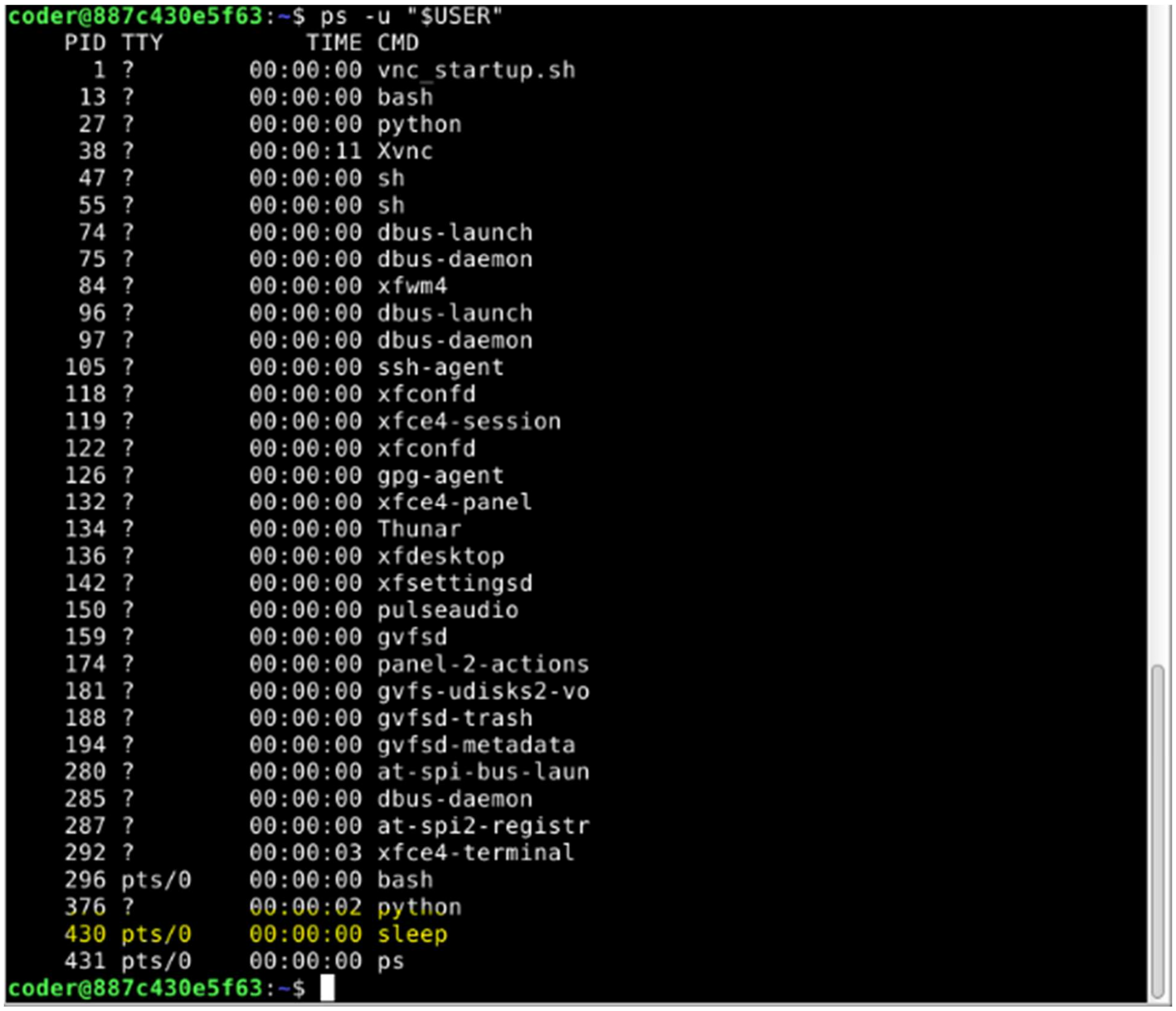
```
coder@887c430e5f63:~$ jobs -l
[1]+   415 Running                 sleep 300 &
coder@887c430e5f63:~$ 
```

5. Process Priority Management

Change the priority (niceness) of one of the running processes and display the updated priority.

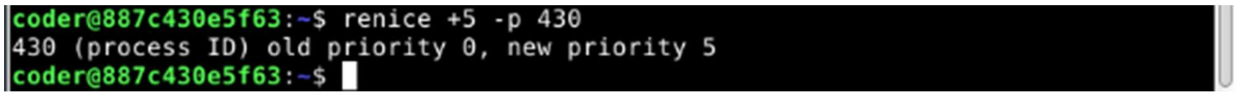
ps -u "$USER"

Note the PID we want to adjust (say it is 415).



```
coder@887c430e5f63:~$ ps -u "$USER"
    PID TTY          TIME CMD
      1 ?        00:00:00 vnc_startup.sh
     13 ?        00:00:00 bash
     27 ?        00:00:00 python
     38 ?        00:00:11 Xvnc
     47 ?        00:00:00 sh
     55 ?        00:00:00 sh
     74 ?        00:00:00 dbus-launch
     75 ?        00:00:00 dbus-daemon
     84 ?        00:00:00 xfwm4
     96 ?        00:00:00 dbus-launch
     97 ?        00:00:00 dbus-daemon
    105 ?        00:00:00 ssh-agent
    118 ?        00:00:00 xfconfd
    119 ?        00:00:00 xfce4-session
    122 ?        00:00:00 xfconfd
    126 ?        00:00:00 gpg-agent
    132 ?        00:00:00 xfce4-panel
    134 ?        00:00:00 Thunar
    136 ?        00:00:00 xfdesktop
    142 ?        00:00:00 xfsettingsd
    150 ?        00:00:00 pulseaudio
    159 ?        00:00:00 gvfsd
    174 ?        00:00:00 panel-2-actions
    181 ?        00:00:00 gvfs-udisks2-vo
    188 ?        00:00:00 gvfsd-trash
    194 ?        00:00:00 gvfsd-metadata
    280 ?        00:00:00 at-spi-bus-laun
    285 ?        00:00:00 dbus-daemon
    287 ?        00:00:00 at-spi2-registr
    292 ?        00:00:03 xfce4-terminal
    296 pts/0    00:00:00 bash
    376 ?        00:00:02 python
    430 pts/0    00:00:00 sleep
    431 pts/0    00:00:00 ps
coder@887c430e5f63:~$ 
```

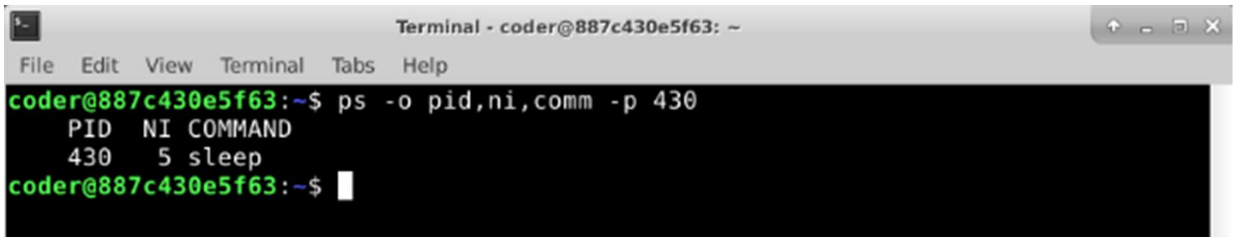
renice +5 -p 415

- This increases the nice value by 5, lowering the process priority (making it more “polite” with CPU time).
- 

```
coder@887c430e5f63:~$ renice +5 -p 430
430 (process ID) old priority 0, new priority 5
coder@887c430e5f63:~$
```

ps -o pid,ni,comm -p 415

- ni shows the nice value; we should now see the new niceness (e.g., 5 instead of 0) for that PID.



Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

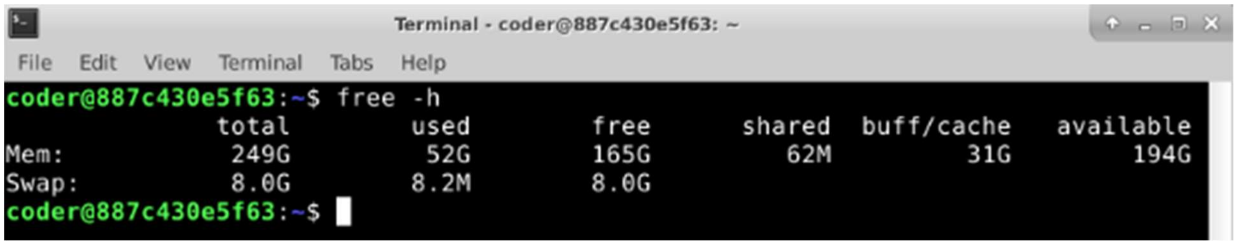
```
coder@887c430e5f63:~$ ps -o pid,ni,comm -p 430
    PID  NI COMMAND
    430   5 sleep
coder@887c430e5f63:~$
```

6. Memory Usage Monitoring

Display memory usage information in a human-readable format.

free -h

- free shows total, used, and free physical RAM and swap.
- -h prints sizes in a human-readable format (KB/MB/GB), making it easy to understand overall memory usage and available



Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

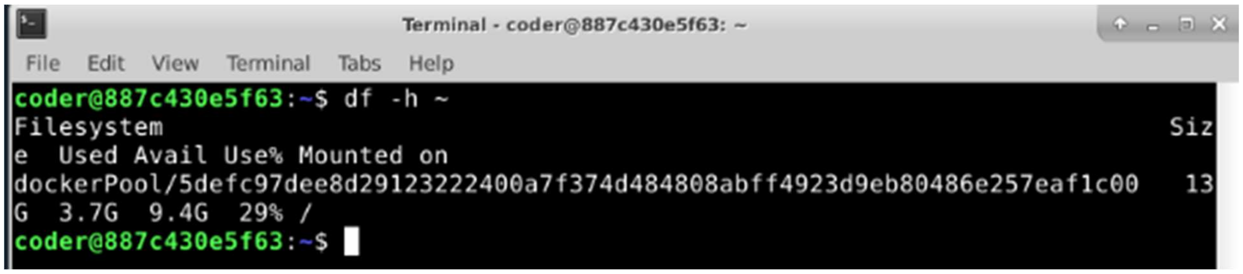
```
coder@887c430e5f63:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:            249G         52G        165G         62M         31G        194G
Swap:           8.0G        8.2M        8.0G
coder@887c430e5f63:~$
```

7. Disk Space Inspection

Display the disk space usage of the filesystem where your home directory resides.

df -h ~

- df reports disk usage for the filesystem that contains the given path.
- -h shows sizes in a human-readable format (KB/MB/GB), so this output tells us total, used, and available space, plus usage percentage, for the filesystem where home directory lives.



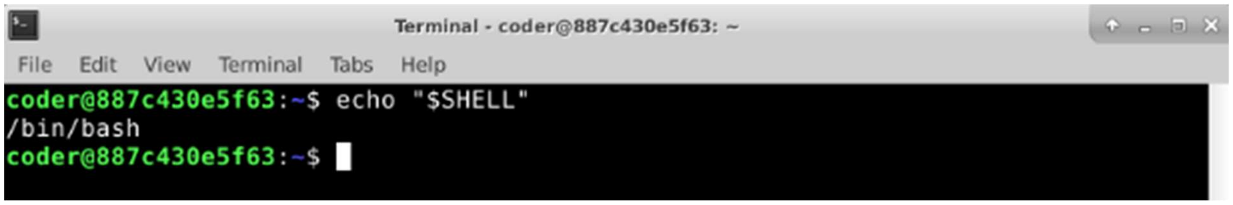
```
coder@887c430e5f63:~$ df -h ~
Filesystem                                                                        Siz
e  Used Avail Use% Mounted on
dockerPool/5defc97dee8d29123222400a7f374d484808abff4923d9eb80486e257eaf1c00   13
G  3.7G  9.4G  29% /
coder@887c430e5f63:~$ 
```

8. Shell Identification

Display the name of the shell currently in use.

echo "$SHELL"

- $SHELL is an environment variable that stores the path to default login shell, such as /bin/bash or /usr/bin/zsh.
- The last component of that path (e.g., bash, zsh) is the name of the shell we are currently using for that session.



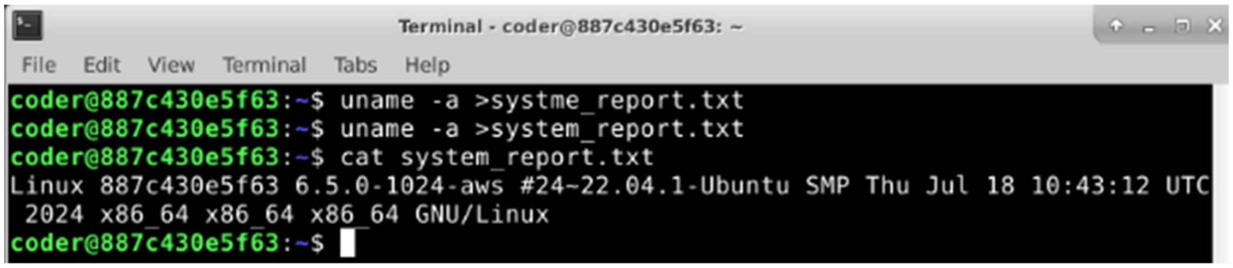
```
coder@887c430e5f63:~$ echo "$SHELL"
/bin/bash
coder@887c430e5f63:~$ 
```

9. Output Redirection

Redirect the output of a system information command of your choice into a file named system_report.txt.

uname -a > system_report.txt {This command displays key system details like kernel version, hostname, architecture, and uptime in a single line. The > operator redirects standard output (stdout) to system_report.txt, overwriting the file if it exists.}

cat system_report.txt (afterward to confirm the output was saved correctly)



```
coder@887c430e5f63:~$ uname -a >systme_report.txt
coder@887c430e5f63:~$ uname -a >system_report.txt
coder@887c430e5f63:~$ cat system_report.txt
Linux 887c430e5f63 6.5.0-1024-aws #24~22.04.1-Ubuntu SMP Thu Jul 18 10:43:12 UTC
 2024 x86_64 x86_64 x86_64 GNU/Linux
coder@887c430e5f63:~$ 
```

10. Disk Usage Visualization

Demonstrate the usage of the ncdu tool using appropriate options and briefly explain what it shows.

ncdu (NCurses Disk Usage) is a terminal-based disk usage analyzer that provides an interactive, visual interface for exploring directory sizes on Linux systems.

Basic Usage

We run **ncdu /path/to/directory** to scan a specific path, or just ncdu for the current directory; it quickly builds a navigable tree sorted by size with bar graphs showing proportions. Use arrow keys to navigate, Enter to drill into directories, and 'q' to quit.

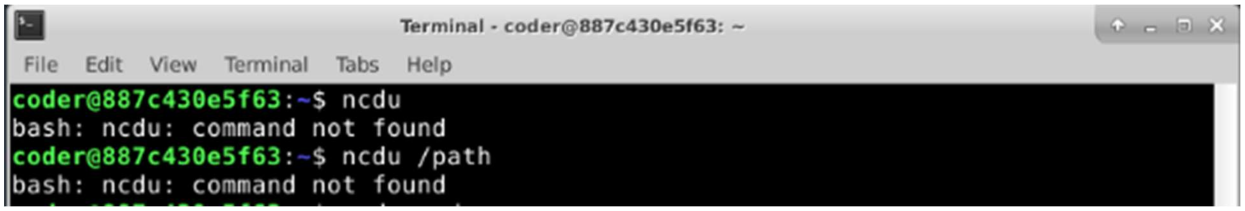
Key Options

- -x: Limits analysis to the same filesystem, excluding mounts.
- -o filename: Exports scan data for later reloading with -f filename.
- -e: Enables extended info like permissions and timestamps.

What It Shows

The interface displays directories and files by size (largest first), with human-readable units (e.g., MiB), percentages, and graphical bars for quick visualization of space hogs. Toggle views with 'g' for graphs/percentages, 's' to sort by size, or 'a' for apparent vs. disk usage. Press 'd' to delete items cautiously, and 'i' for details on selections.

Unable to share screenshot due Coursera lab system doesn't have ncdu package installed.


Terminal - coder@887c430e5f63: ~
File Edit View Terminal Tabs Help

```
coder@887c430e5f63:~$ ncdu
bash: ncdu: command not found
coder@887c430e5f63:~$ ncdu /path
bash: ncdu: command not found
```