OPS361

Pieter Johannes Swart

(600640)

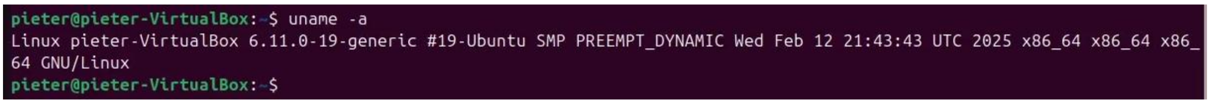
2025/03/19

Table of Contents

Creating and Navigating Directories

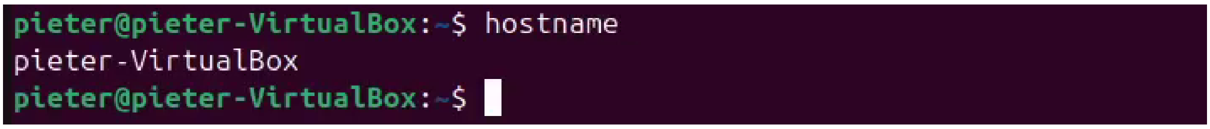
1. System Information and Navigation

- Displaying System Information => the next command to gather basic system information
  “ **uname -a** ”
  - “uname” = stands for Unix Name and is used to display system information.
  - The “-a” flag shows all available system details, including kernel name, version, machine hardware.



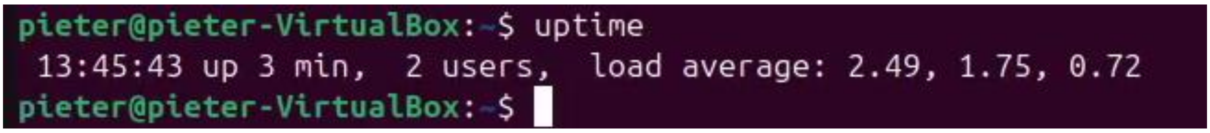
```
pieter@pieter-VirtualBox:~$ uname -a
Linux pieter-VirtualBox 6.11.0-19-generic #19-Ubuntu SMP PREEMPT_DYNAMIC Wed Feb 12 21:43:43 UTC 2025 x86_64 x86_64 x86_
64 GNU/Linux
pieter@pieter-VirtualBox:~$
```

- Display system’s hostname
  “ **hostname** ”
  - This command displays the name of the system “host” on the network.
  - Useful for identifying the machine, especially in networked environments.



```
pieter@pieter-VirtualBox:~$ hostname
pieter-VirtualBox
pieter@pieter-VirtualBox:~$ 
```

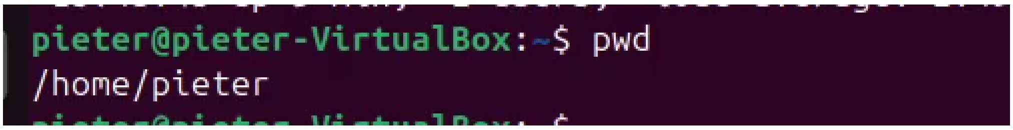
- Display system uptime
  “ **uptime** ”
  - Shows how long the system has been running since the last reboot.
  - Also displays the number of logged-in users and the system’s load average.



```
pieter@pieter-VirtualBox:~$ uptime
 13:45:43 up 3 min,  2 users,  load average: 2.49, 1.75, 0.72
pieter@pieter-VirtualBox:~$ 
```

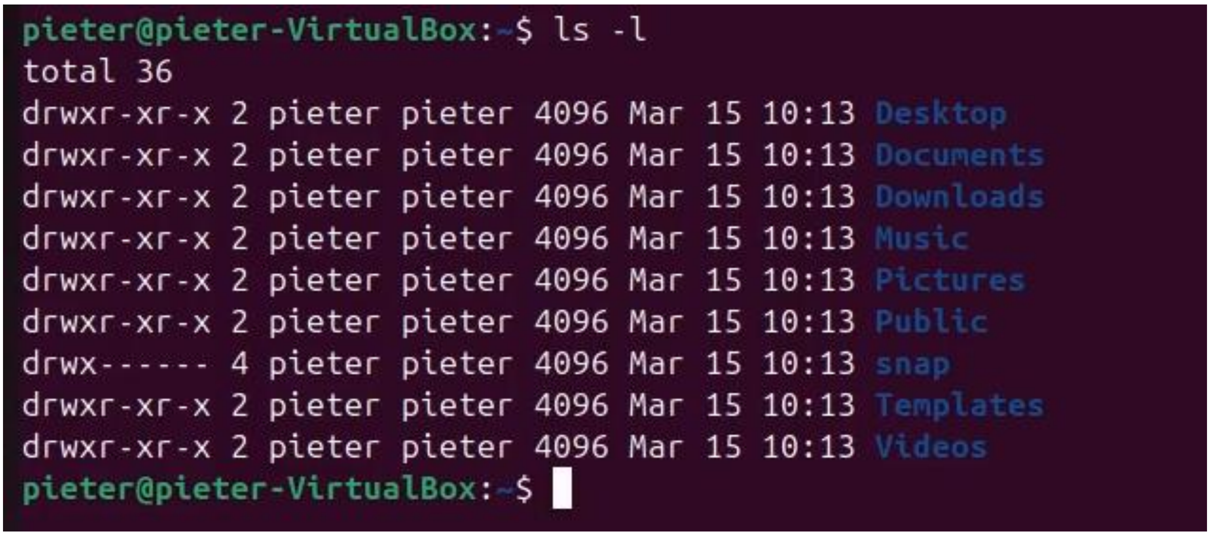
2. Navigating the File System

- Print Working Directory
  “ **pwd** ”
  - Displays the current directory you are in.
  - Helps confirm your location in the file system.



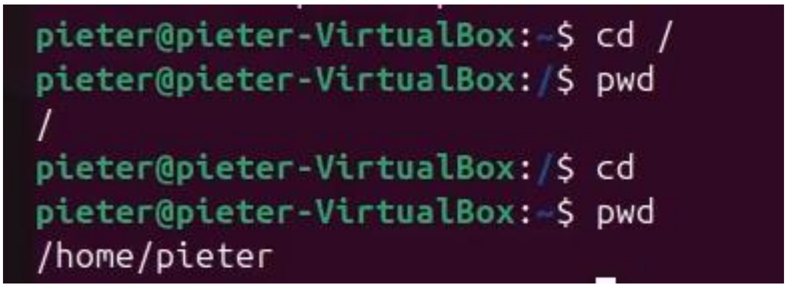
```
pieter@pieter-VirtualBox:~$ pwd
/home/pieter
```

- List directory contents
  - It Shows the contents of the directory in long format, displaying details like file permissions, ownership, size, and modification date.



```
pieter@pieter-VirtualBox:~$ ls -l
total 36
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Desktop
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Documents
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Downloads
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Music
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Pictures
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Public
drwx------ 4 pieter pieter 4096 Mar 15 10:13 snap
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Templates
drwxr-xr-x 2 pieter pieter 4096 Mar 15 10:13 Videos
pieter@pieter-VirtualBox:~$
```

- Change Directory
  “ **cd** ”
  - “cd” is used to navigate between directories.
  - In this case, it moves to the “/home” directory.
  - Running “pwd” after confirms the new location.

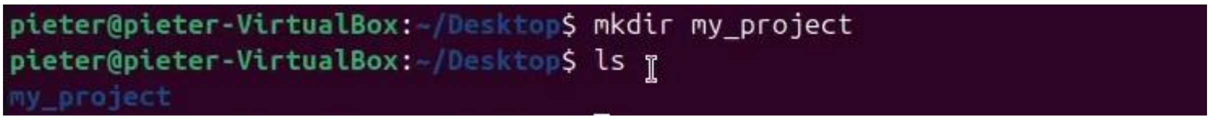


```
pieter@pieter-VirtualBox:~$ cd /
pieter@pieter-VirtualBox:/$ pwd
/
pieter@pieter-VirtualBox:/$ cd
pieter@pieter-VirtualBox:~$ pwd
/home/pieter
```

File and Directory Management

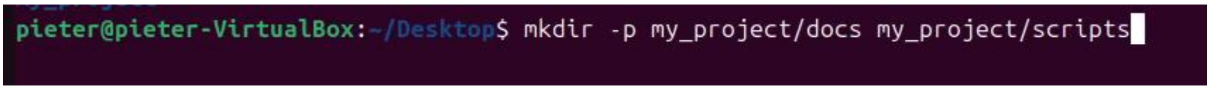
1. Creating and Navigating Directories

- Create a directory using mkdir
  “ **mkdir my_project** ”
  - The “mkdir” command creates a new directory named “my_project”.

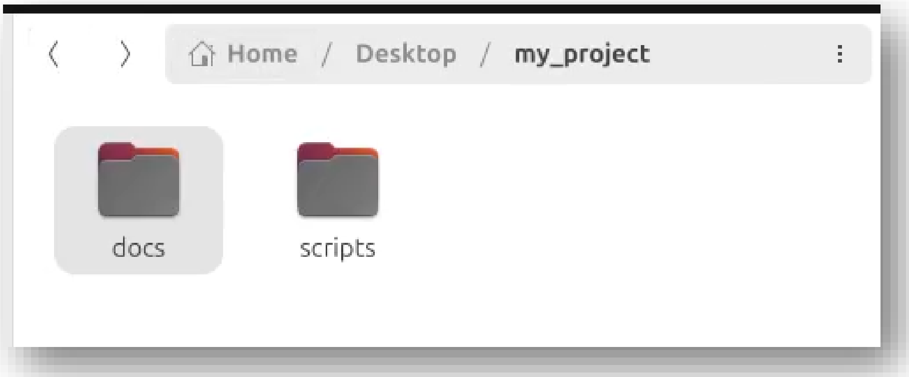


```
pieter@pieter-VirtualBox:~/Desktop$ mkdir my_project
pieter@pieter-VirtualBox:~/Desktop$ ls
my_project
```

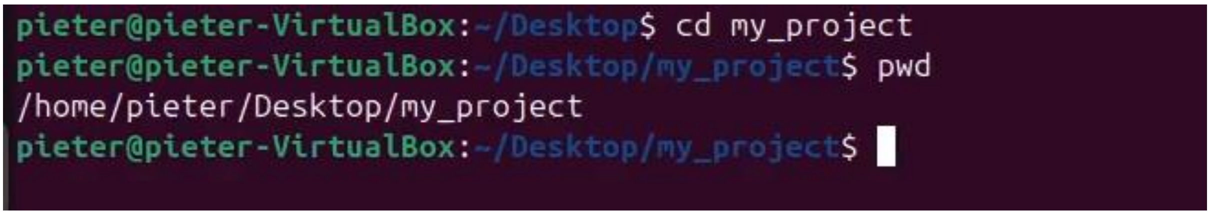
- Create subdirectories using
  =>command: “ **mkdir -p** ”
  “ **mkdir -p my_project/docs my_project/scripts** ”
  - The -p option creates parent directories if they don’t exist.
  - This command creates two subdirectories: docs and scripts inside my_project.



```
pieter@pieter-VirtualBox:~/Desktop$ mkdir -p my_project/docs my_project/scripts
```



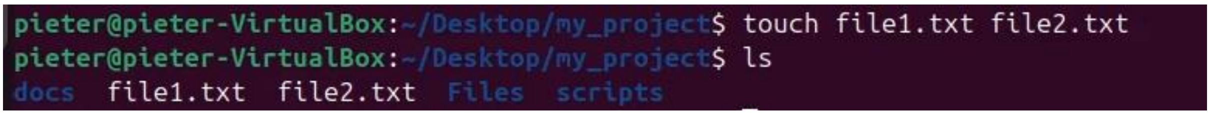
- Navigate between directories using “cd”
  cd my_project
  pwd
  - cd my_project moves into the my_project directory.
  - pwd confirms the current location.



```
pieter@pieter-VirtualBox:~/Desktop$ cd my_project
pieter@pieter-VirtualBox:~/Desktop/my_project$ pwd
/home/pieter/Desktop/my_project
pieter@pieter-VirtualBox:~/Desktop/my_project$ 
```

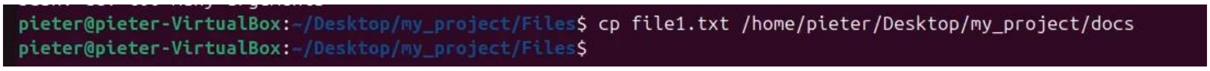
2. Creating and Managing Files

- Create files using:
  “ **touch file1.txt file2.txt** ”
  - “touch” creates new empty files named file1.txt and file2.txt.



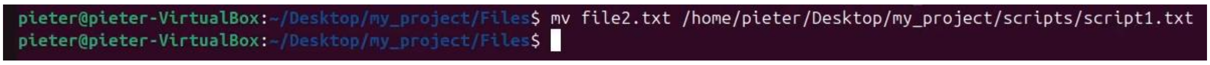
```
pieter@pieter-VirtualBox:~/Desktop/my_project$ touch file1.txt file2.txt
pieter@pieter-VirtualBox:~/Desktop/my_project$ ls
docs  file1.txt  file2.txt  Files  scripts
```

- Copy files using cp
  “ cp file1.txt docs/ ”
  - “cp” copies “file1.txt” into the “docs/” directory.



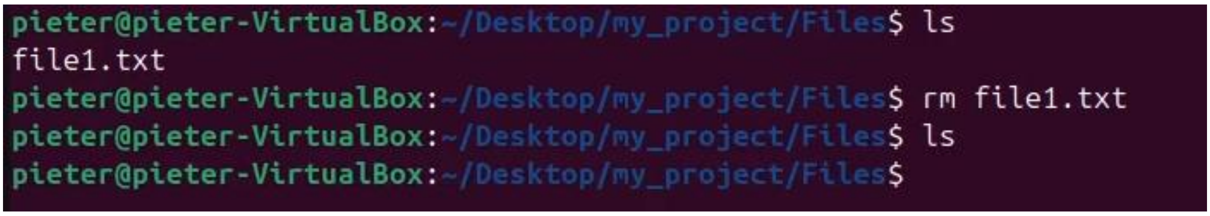
```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ cp file1.txt /home/pieter/Desktop/my_project/docs
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$
```

- Move/Rename files using mv
  “ **mv file2.txt scripts/script1.txt** ”
  - “mv” moves file2.txt to the “scripts/” directory and renames it to script1.txt.



```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ mv file2.txt /home/pieter/Desktop/my_project/scripts/script1.txt
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ 
```

- Remove a file using rm
  “ **rm file1.txt** ”
  - “rm” deletes “file1.txt”



```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ ls
file1.txt
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ rm file1.txt
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ ls
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$
```

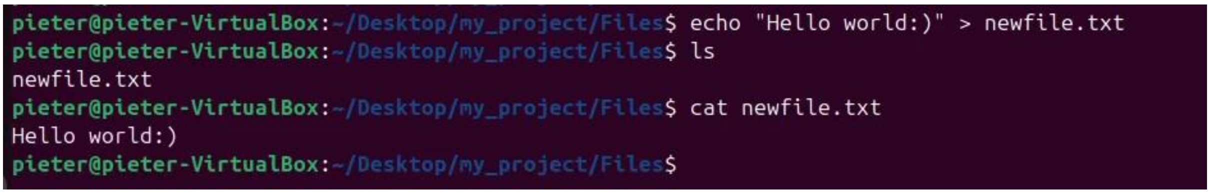
3. Viewing File Contents

- Use “cat” to view file content

  echo "Hello world:)" > testfile.txt

  cat testfile.txt

  - “echo” adds text to “testfile.txt”
  - “cat” displays the content of “testfile.txt”

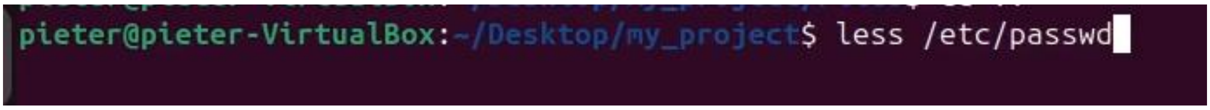


```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ echo "Hello world:)" > newfile.txt
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ ls
newfile.txt
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ cat newfile.txt
Hello world:)
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$
```

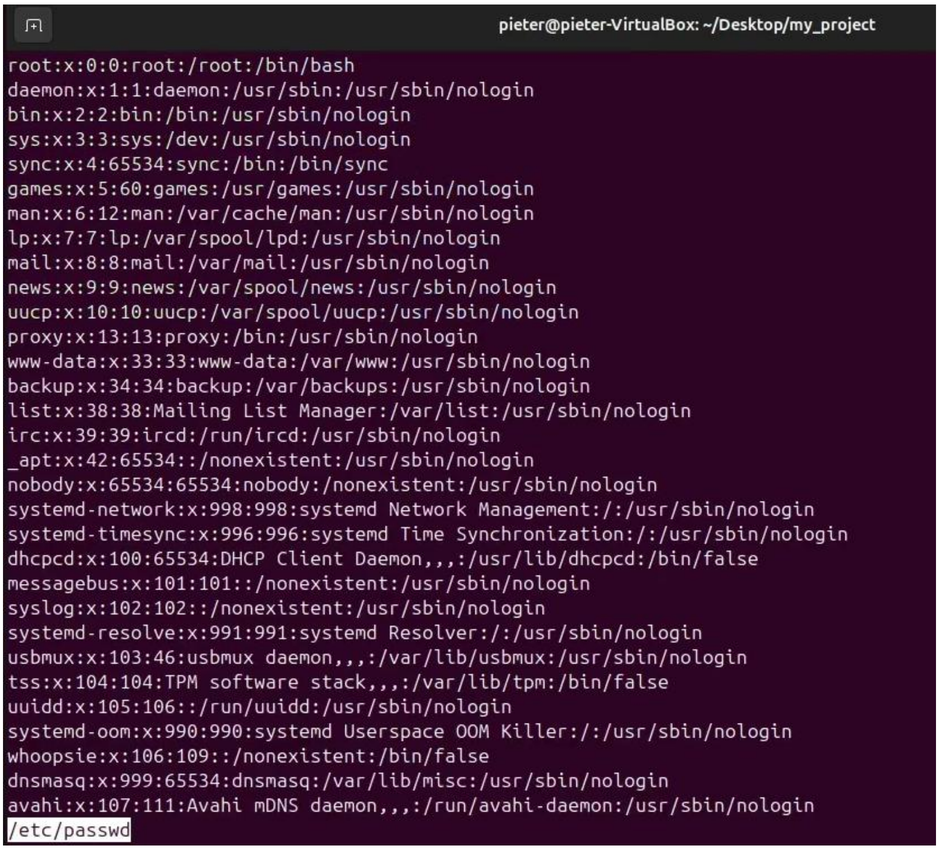
- Use “less” for large files

  “ **less /etc/passwd** ”

  - less allows scrolling through large files.



```
pieter@pieter-VirtualBox:~/Desktop/my_project$ less /etc/passwd
```

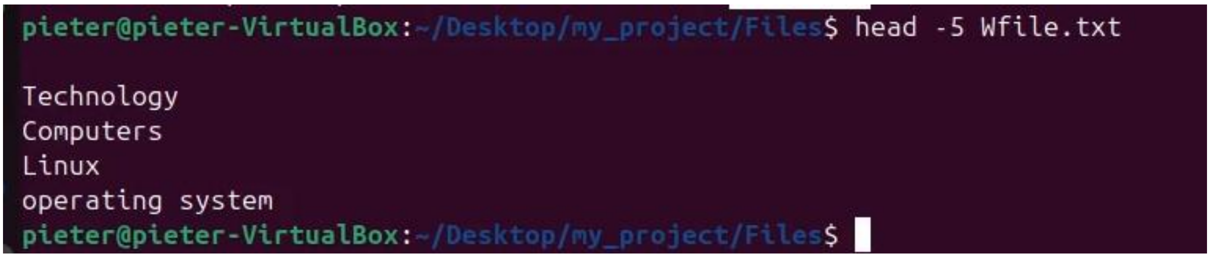


```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin
usbmux:x:103:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tss:x:104:104:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:105:106::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:990:990:systemd Userspace OOM Killer:/:/usr/sbin/nologin
whoopsie:x:106:109::/nonexistent:/bin/false
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:107:111:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
/etc/passwd
```

- Use head to show first 5 lines

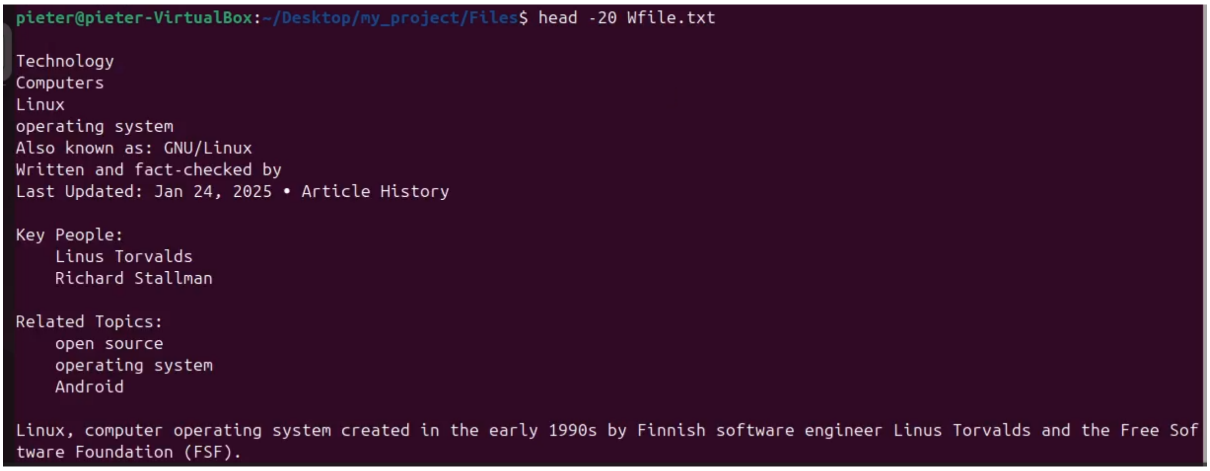
  “ **head -5 Wfile.txt** ”

  - “head -5” shows the *first 5 lines* of the “Wfile.txt” file.



```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ head -5 Wfile.txt

Technology
Computers
Linux
operating system
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ 
```

“ **head -20 Wfile.txt** ”



```
pieter@pieter-VirtualBox:~/Desktop/my_project/Files$ head -20 Wfile.txt

Technology
Computers
Linux
operating system
Also known as: GNU/Linux
Written and fact-checked by
Last Updated: Jan 24, 2025 • Article History

Key People:
    Linus Torvalds
    Richard Stallman

Related Topics:
    open source
    operating system
    Android

Linux, computer operating system created in the early 1990s by Finnish software engineer Linus Torvalds and the Free Sof
tware Foundation (FSF).
```

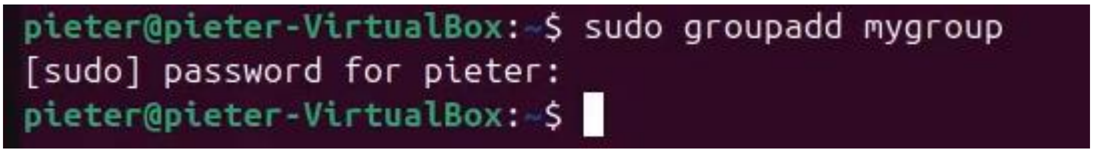
User and Group Management

1. Creating a New User and Group

- Create a new group

  “ **sudo groupadd mygroup** ”

  - “groupadd” creates a new group named *mygroup*.
  - “sudo” is required to run administrative commands.

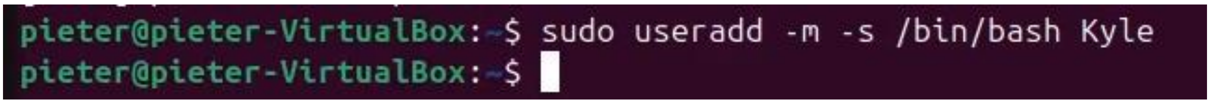


```
pieter@pieter-VirtualBox:~$ sudo groupadd mygroup
[sudo] password for pieter: 
pieter@pieter-VirtualBox:~$ 
```

- Create a new user

  “ **sudo useradd -m -s /bin/bash newuser** ”

  - “useradd” creates a new user named **newuser**.
  - “-m” creates a **home directory** (/home/newuser).
  - “-s /bin/bash” sets **Bash** as the default shell.



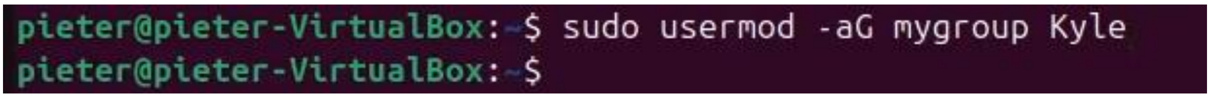
```
pieter@pieter-VirtualBox:~$ sudo useradd -m -s /bin/bash Kyle
pieter@pieter-VirtualBox:~$ 
```

2. Adding the User to the Group and Verifying Membership

- Add user to the group

  " **sudo usermod -aG mygroup newuser** "

  - "usermod -aG" adds newuser to mygroup without removing existing groups.

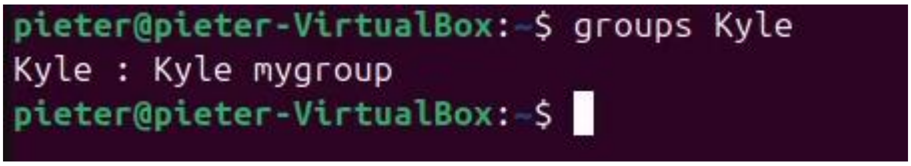


```
pieter@pieter-VirtualBox:~$ sudo usermod -aG mygroup Kyle
pieter@pieter-VirtualBox:~$
```

- Verify user's group membership

  " **groups newuser** "

  - Displays all groups associated with newuser.



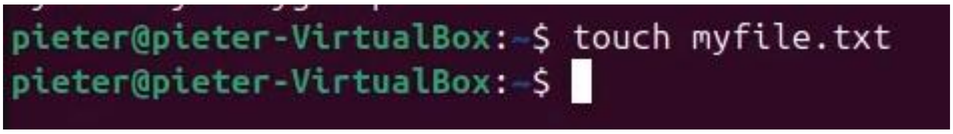
```
pieter@pieter-VirtualBox:~$ groups Kyle
Kyle : Kyle mygroup
pieter@pieter-VirtualBox:~$
```

3. Changing File Ownership and Permissions

- Create a test file

  " touch myfile.txt "

  - Creates an empty file named "myfile.txt"

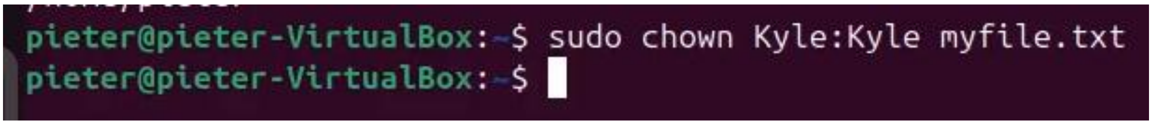


```
pieter@pieter-VirtualBox:~$ touch myfile.txt
pieter@pieter-VirtualBox:~$
```

- Change file ownership using chown

  " **sudo chown newuser:newuser myfile.txt** "

  - chown changes ownership to newuser.
  - newuser:newuser means both user and group ownership are set to newuser.



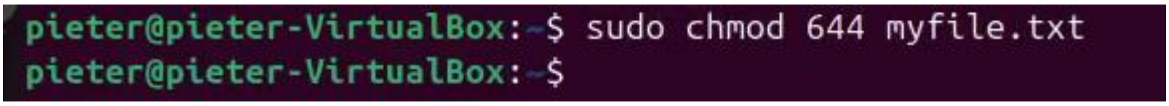
```
pieter@pieter-VirtualBox:~$ sudo chown Kyle:Kyle myfile.txt
pieter@pieter-VirtualBox:~$
```

- Modify file permissions using chmod

  " **chmod 644 myfile.txt** "

  644 sets read/write permissions for the owner, and read-only for others:

  - **Owner =** Read & Write (rw-)
  - **Group =** Read-only (r--)
  - **Others =** Read-only (r--)

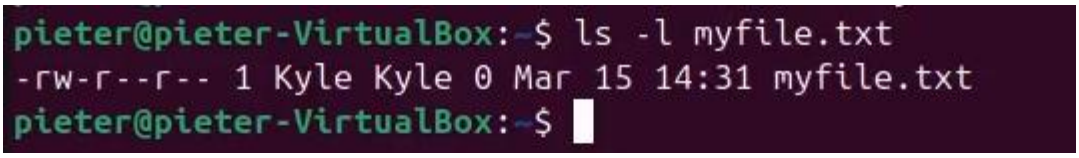


```
pieter@pieter-VirtualBox:~$ sudo chmod 644 myfile.txt
pieter@pieter-VirtualBox:~$
```

- Verify file permissions

  " **ls -l myfile.txt** "

  - Displays the file permissions, owner, and group.



```
pieter@pieter-VirtualBox:~$ ls -l myfile.txt
-rw-r--r-- 1 Kyle Kyle 0 Mar 15 14:31 myfile.txt
pieter@pieter-VirtualBox:~$ 
```

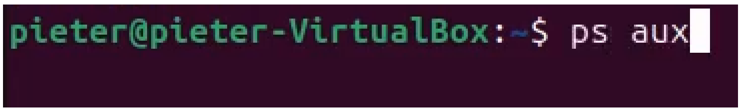
Process and Task Management

1. Listing Running Processes

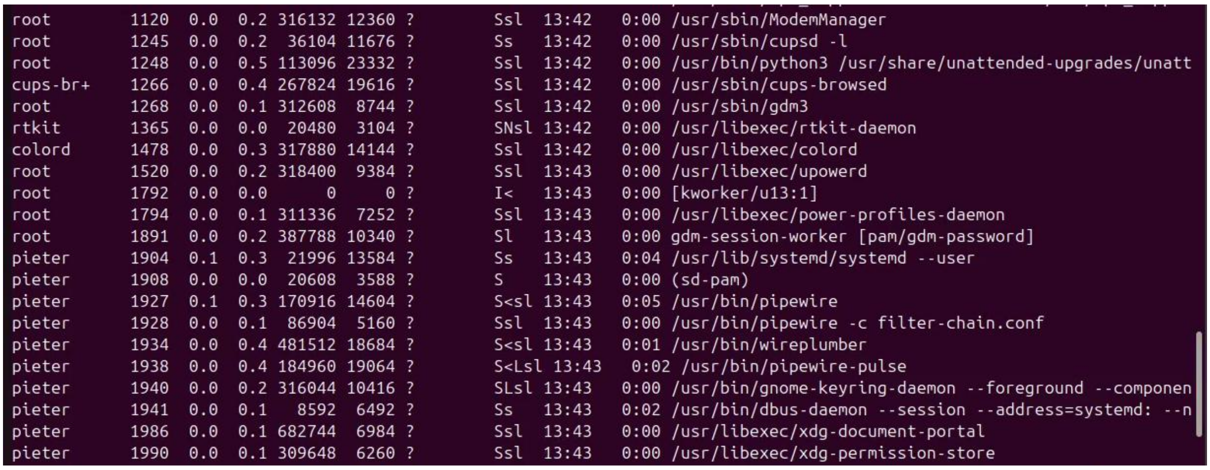
- Using “ps” to list processes

  “ **ps aux** ”

  - “ps” displays active processes.
  - “a” shows processes from all users.
  - “u” displays user-oriented output (with owner, CPU usage, memory usage).
  - “x” includes processes not attached to a terminal.



```
pieter@pieter-VirtualBox:~$ ps aux
```

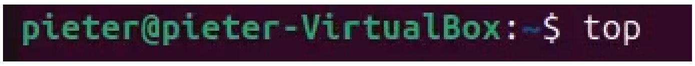


```
root         1120  0.0  0.2 316132 12360 ?        Ssl  13:42   0:00 /usr/sbin/ModemManager
root         1245  0.0  0.2  36104 11676 ?        Ss   13:42   0:00 /usr/sbin/cupsd -l
root         1248  0.0  0.5 113096 23332 ?        Ssl  13:42   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
cups-br+     1266  0.0  0.4 267824 19616 ?        Ssl  13:42   0:00 /usr/sbin/cups-browsed
root         1268  0.0  0.1 312608  8744 ?        Ssl  13:42   0:00 /usr/sbin/gdm3
rtkit        1365  0.0  0.0  20480  3104 ?        SNsl 13:42   0:00 /usr/libexec/rtkit-daemon
colord       1478  0.0  0.3 317880 14144 ?        Ssl  13:42   0:00 /usr/libexec/colord
root         1520  0.0  0.2 318400  9384 ?        Ssl  13:43   0:00 /usr/libexec/upowerd
root         1792  0.0  0.0      0     0 ?        I<   13:43   0:00 [kworker/u13:1]
root         1794  0.0  0.1 311336  7252 ?        Ssl  13:43   0:00 /usr/libexec/power-profiles-daemon
root         1891  0.0  0.2 387788 10340 ?        Sl   13:43   0:00 gdm-session-worker [pam/gdm-password]
pieter       1904  0.1  0.3  21996 13584 ?        Ss   13:43   0:04 /usr/lib/systemd/systemd --user
pieter       1908  0.0  0.0  20608  3588 ?        S    13:43   0:00 (sd-pam)
pieter       1927  0.1  0.3 170916 14604 ?        S<sl 13:43   0:05 /usr/bin/pipewire
pieter       1928  0.0  0.1  86904  5160 ?        Ssl  13:43   0:00 /usr/bin/pipewire -c filter-chain.conf
pieter       1934  0.0  0.4 481512 18684 ?        S<sl 13:43   0:01 /usr/bin/wireplumber
pieter       1938  0.0  0.4 184960 19064 ?        S<Lsl 13:43   0:02 /usr/bin/pipewire-pulse
pieter       1940  0.0  0.2 316044 10416 ?        SLsl 13:43   0:00 /usr/bin/gnome-keyring-daemon --foreground --componen
pieter       1941  0.0  0.1   8592  6492 ?        Ss   13:43   0:02 /usr/bin/dbus-daemon --session --address=systemd: --n
pieter       1986  0.0  0.1 682744  6984 ?        Ssl  13:43   0:00 /usr/libexec/xdg-document-portal
pieter       1990  0.0  0.1 309648  6260 ?        Ssl  13:43   0:00 /usr/libexec/xdg-permission-store
```

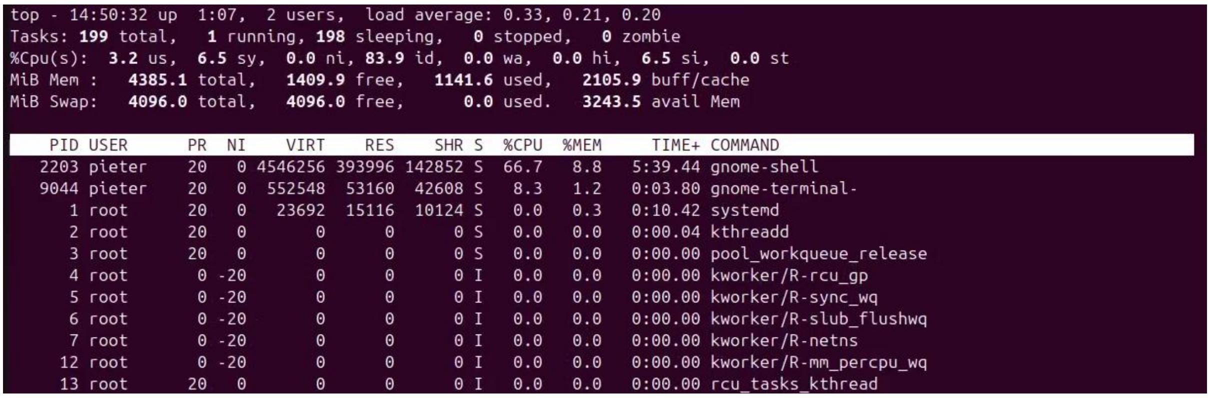
- Using top to monitor system processes

  “ **top** ”

  - Displays real-time information on CPU, memory usage, and running processes.
  - Press “q” to exit.



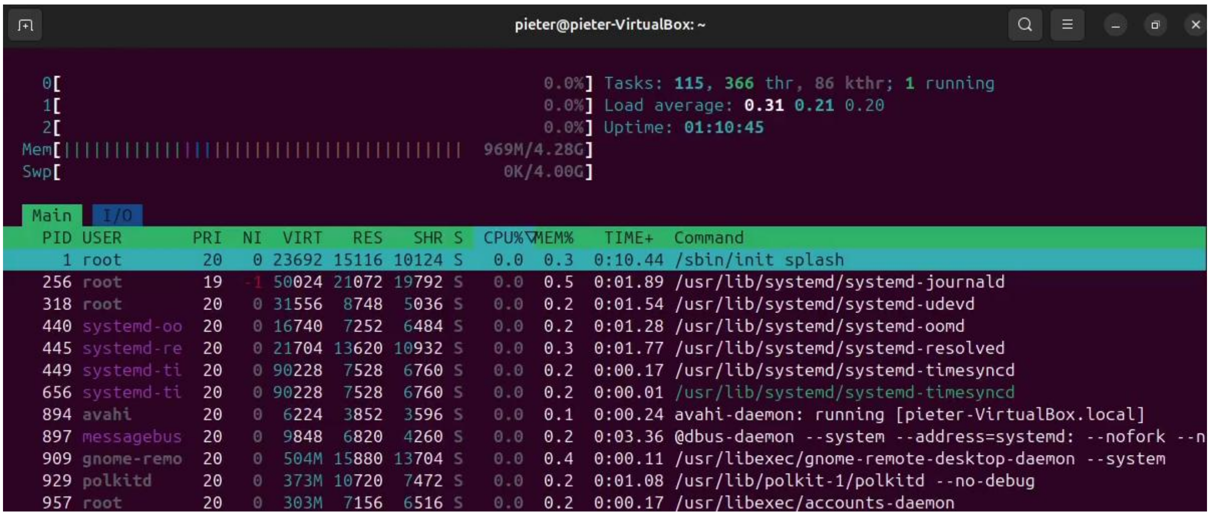
```
pieter@pieter-VirtualBox:~$ top
```



```
top - 14:50:32 up  1:07,  2 users,  load average: 0.33, 0.21, 0.20
Tasks: 199 total,   1 running, 198 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.2 us,  6.5 sy,  0.0 ni, 83.9 id,  0.0 wa,  0.0 hi,  6.5 si,  0.0 st
MiB Mem :   4385.1 total,   1409.9 free,   1141.6 used,   2105.9 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.   3243.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2203 pieter    20   0 4546256 393996 142852 S  66.7   8.8   5:39.44 gnome-shell
   9044 pieter    20   0  552548  53160  42608 S   8.3   1.2   0:03.80 gnome-terminal-
      1 root      20   0   23692  15116  10124 S   0.0   0.3   0:10.42 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_flushwq
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
     12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_percpu_wq
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
```

- Using htop
  =>Install htop: **sudo apt install htop** => type this command:
  " **htop** "
  - A more user-friendly version of top with color coding and interactive features.
  - Navigate using arrow keys; press F9 to kill a process, and q to quit.

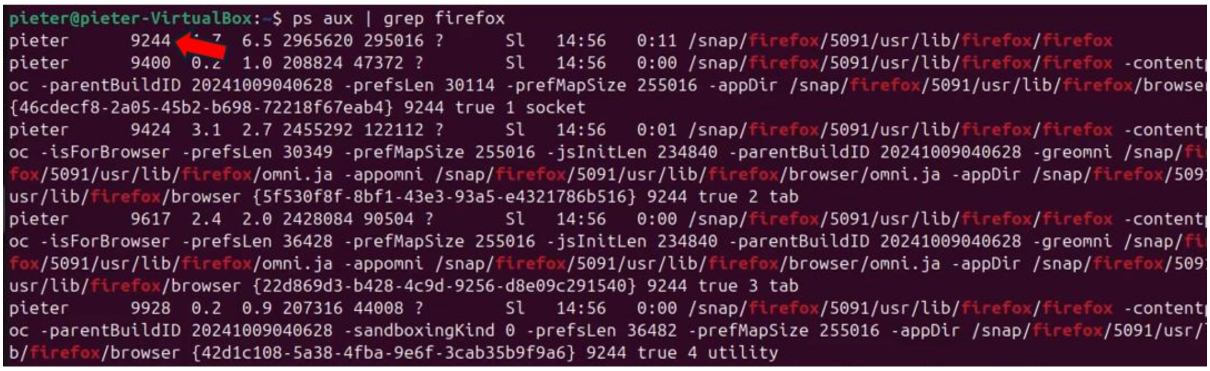
```
pieter@pieter-VirtualBox:~$ htop
```



```
    0[                                              0.0%] Tasks: 115, 366 thr, 86 kthr; 1 running
    1[                                              0.0%] Load average: 0.31 0.21 0.20
    2[                                              0.0%] Uptime: 01:10:45
  Mem[||||||||||||||||||||||||||||||||||||||||  969M/4.28G]
  Swp[                                            0K/4.00G]

 Main  I/O
    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
      1 root        20   0 23692 15116 10124 S   0.0  0.3  0:10.44 /sbin/init splash
    256 root        19  -1 50024 21072 19792 S   0.0  0.5  0:01.89 /usr/lib/systemd/systemd-journald
    318 root        20   0 31556  8748  5036 S   0.0  0.2  0:01.54 /usr/lib/systemd/systemd-udevd
    440 systemd-oo  20   0 16740  7252  6484 S   0.0  0.2  0:01.28 /usr/lib/systemd/systemd-oomd
    445 systemd-re  20   0 21704 13620 10932 S   0.0  0.3  0:01.77 /usr/lib/systemd/systemd-resolved
    449 systemd-ti  20   0 90228  7528  6760 S   0.0  0.2  0:00.17 /usr/lib/systemd/systemd-timesyncd
    656 systemd-ti  20   0 90228  7528  6760 S   0.0  0.2  0:00.01 /usr/lib/systemd/systemd-timesyncd
    894 avahi       20   0  6224  3852  3596 S   0.0  0.1  0:00.24 avahi-daemon: running [pieter-VirtualBox.local]
    897 messagebus  20   0  9848  6820  4260 S   0.0  0.2  0:03.36 @dbus-daemon --system --address=systemd: --nofork --n
    909 gnome-remo  20   0  504M 15880 13704 S   0.0  0.4  0:00.11 /usr/libexec/gnome-remote-desktop-daemon --system
    929 polkitd     20   0  373M 10720  7472 S   0.0  0.2  0:01.08 /usr/lib/polkit-1/polkitd --no-debug
    957 root        20   0  303M  7156  6516 S   0.0  0.2  0:00.17 /usr/libexec/accounts-daemon
```

2. Terminating a Process

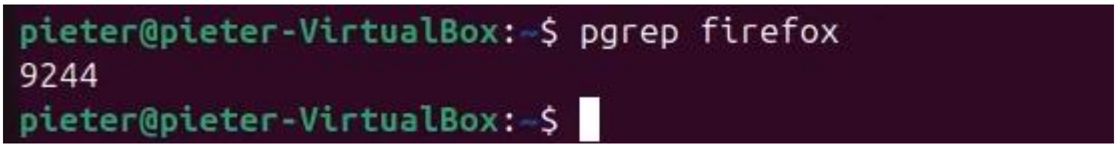
- Find process ID
  " **ps aux | grep firefox** "
  - Searches for all running firefox processes.
  - The **PID** (Process ID) is in the second column.



```
pieter@pieter-VirtualBox:~$ ps aux | grep firefox
pieter      9244  1.7  6.5 2965620 295016 ?      Sl   14:56   0:11 /snap/firefox/5091/usr/lib/firefox/firefox
pieter      9400  0.2  1.0 208824 47372 ?        Sl   14:56   0:00 /snap/firefox/5091/usr/lib/firefox/firefox -contentpr
oc -parentBuildID 20241009040628 -prefsLen 30114 -prefMapSize 255016 -appDir /snap/firefox/5091/usr/lib/firefox/browser
{46cdecf8-2a05-45b2-b698-72218f67eab4} 9244 true 1 socket
pieter      9424  3.1  2.7 2455292 122112 ?      Sl   14:56   0:01 /snap/firefox/5091/usr/lib/firefox/firefox -contentpr
oc -isForBrowser -prefsLen 30349 -prefMapSize 255016 -jsInitLen 234840 -parentBuildID 20241009040628 -greomni /snap/fi
fox/5091/usr/lib/firefox/omni.ja -appomni /snap/firefox/5091/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/509
usr/lib/firefox/browser {5f530f8f-8bf1-43e3-93a5-e4321786b516} 9244 true 2 tab
pieter      9617  2.4  2.0 2428084 90504 ?       Sl   14:56   0:00 /snap/firefox/5091/usr/lib/firefox/firefox -contentpr
oc -isForBrowser -prefsLen 36428 -prefMapSize 255016 -jsInitLen 234840 -parentBuildID 20241009040628 -greomni /snap/fi
fox/5091/usr/lib/firefox/omni.ja -appomni /snap/firefox/5091/usr/lib/firefox/browser/omni.ja -appDir /snap/firefox/509
usr/lib/firefox/browser {22d869d3-b428-4c9d-9256-d8e09c291540} 9244 true 3 tab
pieter      9928  0.2  0.9 207316 44008 ?        Sl   14:56   0:00 /snap/firefox/5091/usr/lib/firefox/firefox -contentpr
oc -parentBuildID 20241009040628 -sandboxingKind 0 -prefsLen 36482 -prefMapSize 255016 -appDir /snap/firefox/5091/usr/l
b/firefox/browser {42d1c108-5a38-4fba-9e6f-3cab35b9f9a6} 9244 true 4 utility
```

Using pgrep
“ **pgrep firefox** ”

- This command directly print the PID of Firefox

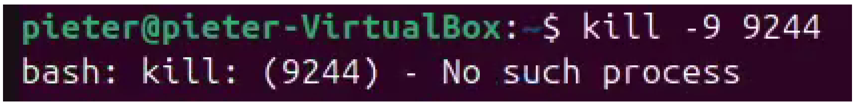


```
pieter@pieter-VirtualBox:~$ pgrep firefox
9244
pieter@pieter-VirtualBox:~$ 
```

- Terminate a specific process using “kill”
  “ **kill 9244** ”
  - 9244 is the PID.
  - Sends a termination signal SIGTERM to the process.

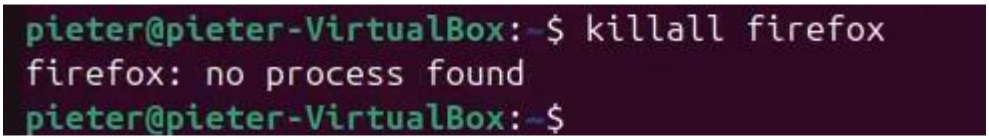
```
pieter@pieter-VirtualBox:~$ kill 9244
```

- Forcefully terminate a process using “kill -9”
  “ kill -9 9244 ”
  - “-9” sends the “SIGKILL” signal, which immediately stops the process.



```
pieter@pieter-VirtualBox:~$ kill -9 9244
bash: kill: (9244) - No such process
```

- Kill all processes with a specific name using killall
  “ **killall firefox** ”
  - Kills all instances of firefox.

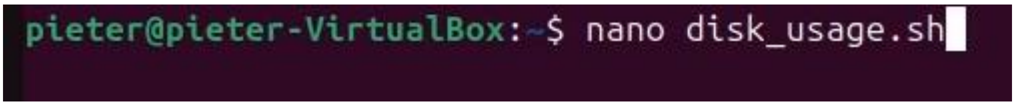


```
pieter@pieter-VirtualBox:~$ killall firefox
firefox: no process found
pieter@pieter-VirtualBox:~$
```

Basic Shell Scripting

1. Create the script file

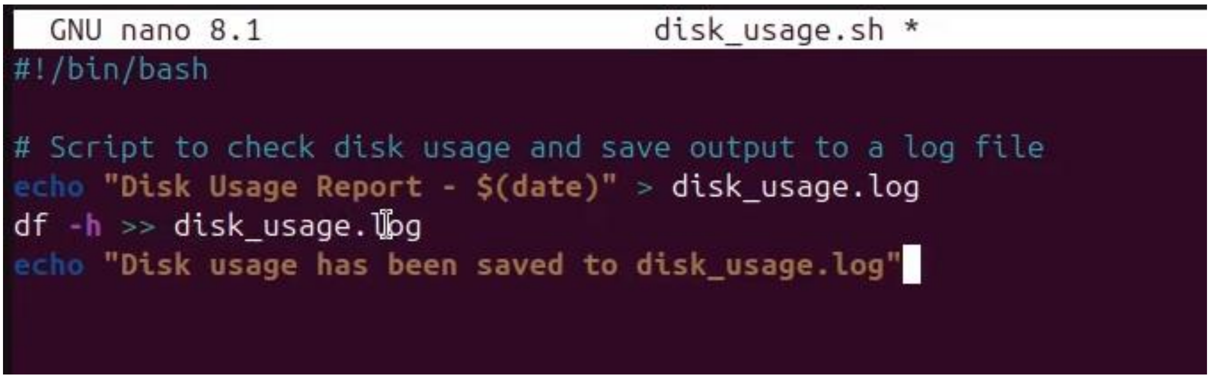
- Opens the Nano text editor to create the script.
  “ **nano disk_usage.sh** ”



```
pieter@pieter-VirtualBox:~$ nano disk_usage.sh
```

2. Write the script

```
#!/bin/bash

# Script to check disk usage and save output to a log file

echo "Disk Usage Report - $(date)" > disk_usage.log
df -h >> disk_usage.log
echo "Disk usage has been saved to disk_usage.log"
```

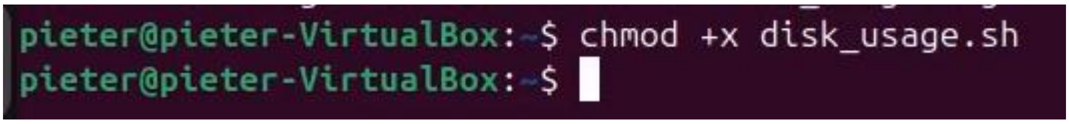


```
GNU nano 8.1                          disk_usage.sh *
#!/bin/bash

# Script to check disk usage and save output to a log file
echo "Disk Usage Report - $(date)" > disk_usage.log
df -h >> disk_usage.log
echo "Disk usage has been saved to disk_usage.log"
```

Ctrl + X to Exit => type "Y" to save and press "Enter"

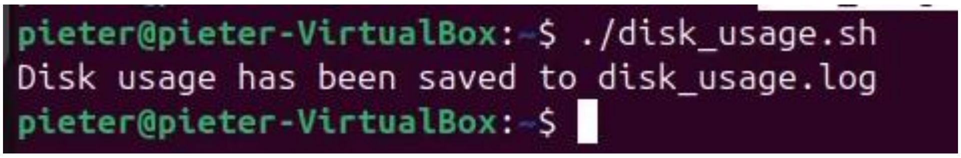
- **#!/bin/bash** => Shebang line indicating the script uses Bash.
- **echo "Disk Usage Report - $(date)" > disk_usage.log** => Adds a timestamp to the log file.
- **df -h >> disk_usage.log** => Appends the disk usage report to the log file.
- **echo "Disk usage has been saved to disk_usage.log**" => Displays a confirmation message.

3. Make the script executable
   " chmod +x disk_usage.sh "
   - Grants **execute permissions** to the script.



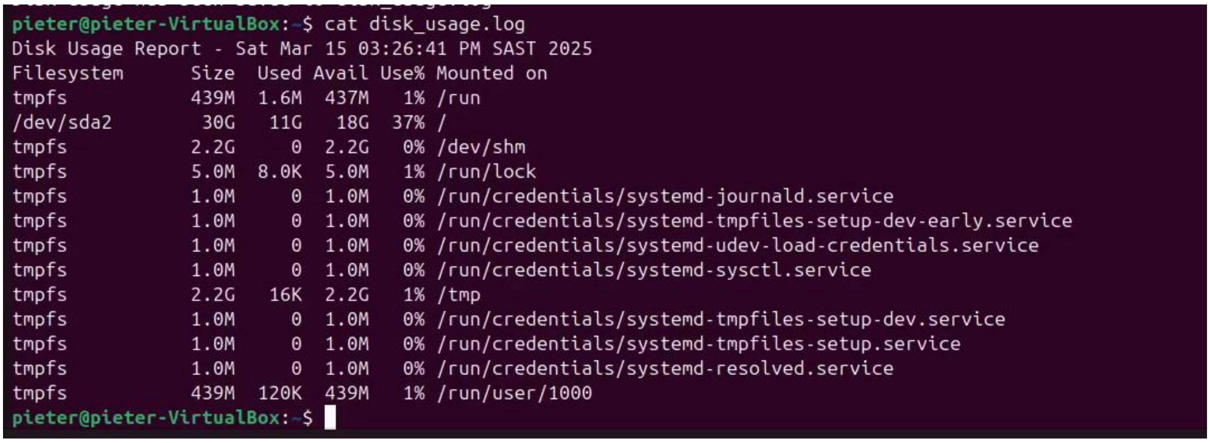
```
pieter@pieter-VirtualBox:~$ chmod +x disk_usage.sh
pieter@pieter-VirtualBox:~$
```

4. Run the script
   " **./disk_usage.sh** "
   - Executes the script.



```
pieter@pieter-VirtualBox:~$ ./disk_usage.sh
Disk usage has been saved to disk_usage.log
pieter@pieter-VirtualBox:~$
```

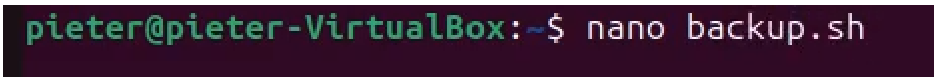
5. Verify the output
   " **cat disk_usage.log** "
   - Displays the saved disk usage report.



```
pieter@pieter-VirtualBox:~$ cat disk_usage.log
Disk Usage Report - Sat Mar 15 03:26:41 PM SAST 2025
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           439M  1.6M  437M   1% /run
/dev/sda2        30G   11G   18G  37% /
tmpfs           2.2G     0  2.2G   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-udev-load-credentials.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-sysctl.service
tmpfs           2.2G   16K  2.2G   1% /tmp
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-resolved.service
tmpfs           439M  120K  439M   1% /run/user/1000
pieter@pieter-VirtualBox:~$
```

Simple File Backup Script

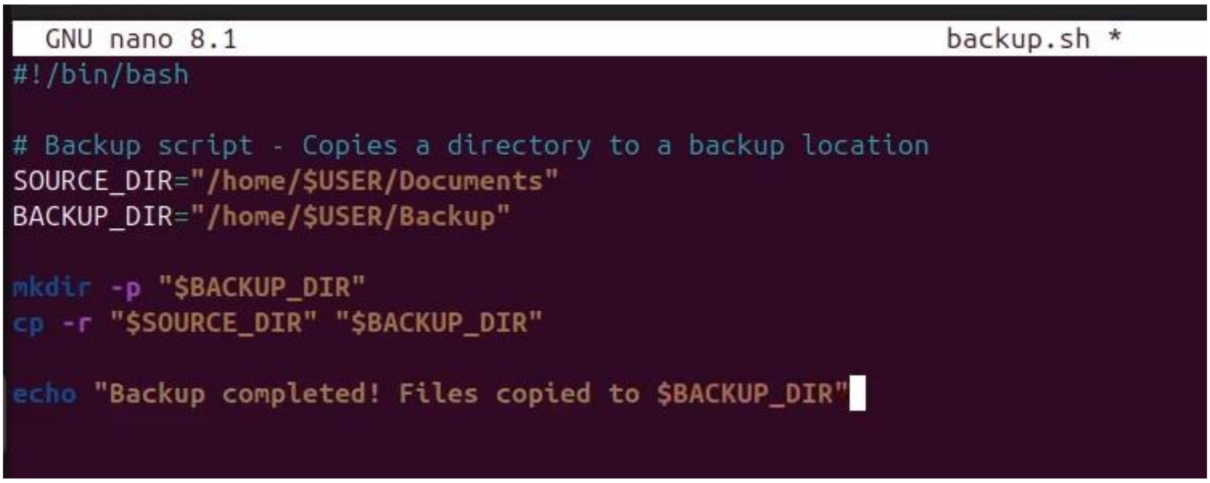
1. Create the script
   “ **nano backup.sh** ”



```
pieter@pieter-VirtualBox:~$ nano backup.sh
```

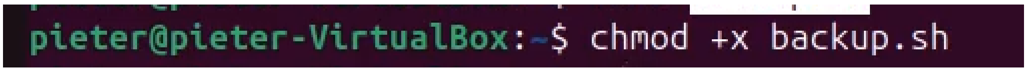
2. Write the script

```
#!/bin/bash

# Backup script - Copies a directory to a backup location

SOURCE_DIR="/home/$USER/Documents"
BACKUP_DIR="/home/$USER/Backup"

mkdir -p "$BACKUP_DIR"
cp -r "$SOURCE_DIR" "$BACKUP_DIR"

echo "Backup completed! Files copied to $BACKUP_DIR"
```



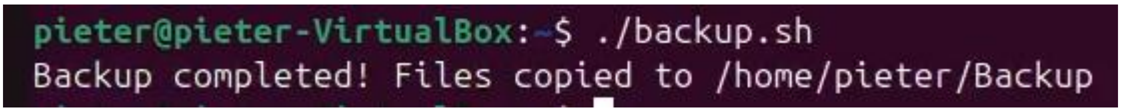
```
  GNU nano 8.1                                              backup.sh *
#!/bin/bash

# Backup script - Copies a directory to a backup location
SOURCE_DIR="/home/$USER/Documents"
BACKUP_DIR="/home/$USER/Backup"

mkdir -p "$BACKUP_DIR"
cp -r "$SOURCE_DIR" "$BACKUP_DIR"

echo "Backup completed! Files copied to $BACKUP_DIR"
```

3. Make the script executable
   “ **chmod +x backup.sh** ”



```
pieter@pieter-VirtualBox:~$ chmod +x backup.sh
```

4. Run the script
   “ **./backup.sh** ”



```
pieter@pieter-VirtualBox:~$ ./backup.sh
Backup completed! Files copied to /home/pieter/Backup
```

References

Garn, D., 2021. How to manage users and groups in Linux. 14 Oct, pp. https://www.redhat.com/en/blog/linux-user-group-management.

Hira, Z., 2022. Shell Scripting for Beginners – How to Write Bash Scripts in Linux. 31 March, pp. https://www.freecodecamp.org/news/shell-scripting-crash-course-how-to-write-bash-scripts-in-linux/.

Kenlon, S., 2019. Navigating your filesystem in the Linux terminal. 11 June, pp. https://www.redhat.com/en/blog/navigating-filesystem-linux-terminal#:~:text=The%20cd%20(change%20directory)%20command%20moves%20you%20into%20a%20different,is%20like%20navigating%20the%20internet..

Unknown, 2022. Commands for Process Management in Linux. 4 Aug, pp. https://www.digitalocean.com/community/tutorials/process-management-in-linux.

Unknown, 2022. Linux - File Management. 11 Jan, pp. https://www.tutorialspoint.com/unix/unix-file-management.htm.

Unknown, 2024. Linux Shell Script to Backup Files and Directory. 15 May, pp. https://www.geeksforgeeks.org/linux-shell-script-to-backup-files-and-directory/.