

DTL – Assignment 6: Linux Commands

Amaan Jamadar

1] Sudo and apt command

```
(base) amaan@ubunx ~ ➤ sudo apt update
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:2 https://mirror.kku.ac.th/CTAN/systems/win32/miktex/setup/deb bionic InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: http://miktex.org/download/ubuntu/dists/bionic/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
N: Skipping acquire of configured file 'universe/binary-i386/Packages' as repository 'http://miktex.org/download/ubuntu bionic InRelease' doesn't support architecture 'i386'
```

2] PWD

```
(base) amaan@ubunx ~ ➤ pwd
/home/amaan
```

3] cd command and ls

```
(base) amaan@ubunx ~ ➤ cd Documents
(base) amaan@ubunx ~/Documents ➤ ls
112103008_FCS_TUT3 a-hello.o Cloud_AWS DSGT
FCS_sse.txt Impression ODMC SEM III
test.cpp
112103008_FCS_TUT3.zip a-hello.s docker.txt DTL
hello.c luffy.jpg point_to_look.txt 'SEM III.zip'
wallpaper a.out DSA_Assign 'FCS_112103008_Amaan Jamadar D
iv- I_S-1' IMP_PD.docX 'Machine Learning' PPL test.c
Zest.zip
```

4] Cat and tac

```
(base) amaan@ubunx ~/Documents ➤ cat hello.c
#include<stdio.h>
Examples:
int main(){
    int a[10];
    int b;
    return 0;
}
```

5] cp command

```
(base) amaan@ubunx ~/Documents/Cloud_AWS ls
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cloud_AWS
docker.txt
DSA_Assign
DSGT
DTL
'FCS_112103008_Amaan Jamadar Div- I_S-1'
FCS_sse.txt
hello.c
Use the cp command to copy files or directories and their contents to another location. The basic syntax is: cp [OPTION]... [SOURCE]... [DEST]
Examples:
• To copy one file from the current directory to another, enter the source name and the destination directory. For example: cp
(base) amaan@ubunx ~/Documents cp docker.txt ~/Documents/Cloud_AWS
(base) amaan@ubunx ~/Documents cd Cloud_AWS
(base) amaan@ubunx ~/Documents/Cloud_AWS ls
docker.txt Notes.txt
(base) amaan@ubunx ~/Documents/Cloud_AWS cp [OPTION]... [SOURCE]... [DEST]
```

```
(base) amaan@ubunx ~/Documents mv point_to_look.txt ~/Documents/Cloud_AWS/main_points.txt
(base) amaan@ubunx ~/Documents ls name files and directories. Note, it doesn't produce an execution, hence renaming happens.
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cloud_AWS
docker.txt
DSA_Assign
DSGT
DTL
'FCS_112103008_Amaan Jamadar Div- I_S-1'
FCS_sse.txt
hello.c
SYNTAX: mv [OPTION]... [SOURCE]... [DEST]
Examples
• To move filename.txt to /home/username/Documents directory:
mv filename.txt /home/username/Documents.
• To rename a file: mv old_name.txt new_filename.txt
(base) amaan@ubunx ~/Documents cd Cloud_AWS
(base) amaan@ubunx ~/Documents/Cloud_AWS ls
docker.txt main_points.txt Notes.txt
```

6] mkdir command

```
(base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cloud_AWS
(base) amaan@ubunx ~/Documents mkdir Sample
(base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cloud_AWS
(base) amaan@ubunx ~/Documents ... contd
To create one or more directories at once, use the -p option. This option creates the directory and its parent directories if they don't exist. If a directory already exists, it will not be created again.
Permissions for each of them. The user executing the command must have the privilege to make a new folder in the parent directory, or they may receive a denied error.
9. rm command
To remove a file or directory, use the rm command. It can be used to delete files, directories, and symbolic links. The basic syntax is: rm [OPTION]... [FILE]...
Parameters:
-f, --force: Force removal without prompting for confirmation.
-i, --interactive: Prompt for confirmation before removing each file.
-R, --recursive: Remove entire directory trees.
-v, --verbose: Show the names of the files being removed.
-w, --writable: Write to the standard output.
--help: Display help information.
--version: Display version information.
```

7] rmdir

```
(base) amaan@ubunx ~/Documents/Sample MODE (as in chmod), not
(base) amaan@ubunx ~ cd Documents
(base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3 docker.txt IMP_PD.docx
112103008_FCS_TUT3.zip DSA_Assign Impression
a-hello.i luffy.jpg
a-hello.o Machine Learning
a-hello.s ODMC
a.out PPL
Cloud_AWS SEM III
(base) amaan@ubunx ~/Documents
```

Example:

- To remove an empty directory, use `rmdir`.
For example, `rmdir -m777 directory_name`
- With `-p`, i.e. `rmdir -p` will be equivalent to `rmdir a/b/c`.

8] touch

```
base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3 docker.txt IMP_PD.docx
112103008_FCS_TUT3.zip DSA_Assign Impression
a-hello.i luffy.jpg
a-hello.o Machine Learning
a-hello.s ODMC
a.out PPL
Cloud_AWS SEM III
(base) amaan@ubunx ~/Documents touch hello_new.txt
(base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3 docker.txt IMP_PD.docx
112103008_FCS_TUT3.zip DSA_Assign Impression
a-hello.i luffy.jpg
a-hello.o Machine Learning
a-hello.s ODMC
a.out PPL
Cloud_AWS SEM III
```

9] plocate

```
amaan@ubunx:~/Documents$ plocate -i hello
/boot/grub/x86_64-efi/hello.mod
/home/amaan/Documents/.hello.c.swp
/home/amaan/Documents/a-hello.i
/home/amaan/Documents/a-hello.o
/home/amaan/Documents/a-hello.s
/home/amaan/Documents/hello.c
/home/amaan/Documents/hello_new.txt
/home/amaan/anaconda3/lib/python3.9/_phello_.foo.py
/home/amaan/anaconda3/lib/python3.9/_pycache_/_phello_.foo.cpython-39.pyc
/home/amaan/anaconda3/lib/python3.9/site-packages/boto/pyami/helloworld.py
/home/amaan/anaconda3/lib/python3.9/site-packages/boto/pyami/_pycache_/_helloworld.cpython-39.pyc
/home/amaan/anaconda3/lib/python3.9/site-packages/nbclient/tests/files/Helloworld.ipynb
/home/amaan/anaconda3/lib/python3.9/site-packages/nbclient/tests/files/.ipynb_checkpoints/Helloworld-chekpoint.ipynb
/home/amaan/anaconda3/lib/python3.9/site-packages/nbconvert/preprocessors/tests/files/Helloworld.ipynb
/home/amaan/anaconda3/lib/python3.9/site-packages/nbconvert/tests/files/hello.py
/home/amaan/anaconda3/lib/python3.9/site-packages/nbconvert/tests/files/_pycache_/_hello.cpython-39.py
/home/amaan/anaconda3/lib/python3.9/site-packages/spyder/plugins/help/utils/static/images/spyder-hello-locstring.png
/home/amaan/anaconda3/lib/tk8.6/demos/hello
/home/amaan/anaconda3/pkg/_openmp_mutex-4.5-1_gnu/info/recipe/parent/hello-world.cpp
/home/amaan/anaconda3/pkg/binaryornot-0.4.4-pyhd3eb1b0_1/info/test/tests/files/hello_world.py
/home/amaan/anaconda3/pkg/boto-2.49.0-py39h06a4308_0/lib/python3.9/site-packages/boto/pyami/helloworld.py
/home/amaan/anaconda3/pkg/boto-2.49.0-py39h06a4308_0/lib/python3.9/site-packages/boto/pyami/_pycache_/_helloworld.cpython-39.pyc
/home/amaan/anaconda3/pkg/cookiecutter-1.7.2-pyhd3eb1b0_0/info/test/tests/test-extensions/hello_extensions
```

10] tac

```
amaan@ubunx:~/Documents$ tac hello.c
SYNTAX : find [option] [path] [expression]
return F0;RMAL SYNTAX : find [dir_to_start][type of obj] [
int b;
int a[10];
EXAMPLES:
int main(){
#include<stdio.h>
amaan@ubunx:~/Documents$ -name "*.txt" => start with DTL dir, find file
c) find DTL -name file2 -exec rm -i {} \;
=> find files which match 'file2' in their name, then a
```

11] find

```
(base) amaan@ubunx ~ find Documents -name hello.ccc' other UNIX commands can be executed on t  
Documents/hello.c  
(base) amaan@ubunx ~ find Documents -name "*.txt"  
Documents/hello_new.txt  
Documents/Cloud_AWS/Notes.txt option] [path] [expression]  
Documents/Cloud_AWS/docker.txtX : find [dir_to_start][type of obj] [options] [name/pattern to fir  
Documents/Cloud_AWS/main_points.txt  
Documents/docker.txt  
Documents/FCS_sse.txtES:  
(base) amaan@ubunx ~ find Documents -name hello_new.txt -exec rm -i {} \;  
rm: remove regular empty file 'Documents/hello_new.txt'? y  
(base) amaan@ubunx ~ cd Documents -exec rm -i {} \  
(base) amaan@ubunx ~/Documents ls  
112103008_FCS_TUT3 docker.txt IMP_PD.docx 'SEM III.zip'  
112103008_FCS_TUT3.zip DSA_Assign Impression test.c  
a-hello.i DSGT luffy.jpg test.cpp  
a-hello.o it finds all .txt files in the system Machine Learning wallpaper  
a-hello.s e) find . -ty FCS_112103008_AmaanJamadarDiv- I_S-1! ODMCD dirs (-type d) Zest.zip  
a.out FCS_sse.txt PPL  
Cloud_AWS hello.c 'SEM III'  
(base) amaan@ubunx ~/Documents
```

```

(base) amaan@ubunx ~ Documents ➤ cd Documents
(base) amaan@ubunx ~/Documents ➤ ls
112103008_FCS_TUT3 docker.txt IMP_PD.docx 'SEM III.zip'
112103008_FCS_TUT3.zip DSA_Assign IMPRESSION test.c
a-hello.i It supports searching by file (-f), folder (-d), name (-name), creation date, modification date, owner and permissions. By using the '-exec' other UNIX commands can be executed or
a-hello.o found. DSGT
a-hello.s FCS_112103008_Amaan Jamadar Div- I_S-1' Machine Learning' be
a.out FCS_sse.txt ODMC
Cloud_AWS hello.c PPL
Cloud_AWS 'SEM III' wallpaper
Zest.zip
(base) amaan@ubunx ~/Documents ➤ find . -type d -name "*.txt"
INFORMAL SYNTAX : find [dir_to_start][type of obj] [options] [name/pattern to f
.
./test.c
./wallpaper EXAMPLES:
./a.out a) find DTL -name file1 => start with DTL dir, find file with name file1
./Cloud_AWS ./Cloud_AWS/Notes.txt b) find DTL -name "*.txt" => start with DTL dir, find file with name that matches *.
./Cloud_AWS/docker.txt 'L -name file2 -exec rm -i {} \;
./Cloud_AWS/main_points.txt which match 'file2' in their name, then execute the rm command if sea
./a-hello.o after asking(-i). {} represents the current value returned by find.
./hello.c
./ODMC d) find / -name "*.txt" => start from the root, find file with name that matches "*.t
./ODMC/Mass Spring- free Oscillations without damping.pdf
./ODMC/New Document(49).pdf ./ODMC/d -name "*.txt" => start from pwd, find dirs (-type d) with name th
./ODMC/ODEMC seating arrangement_T1.pdf
./ODMC/linearly Dep. & Indep functions.pdf
./ODMC/Homo. LDE.pdf
./ODMC/Method of reduction of order.pdf
./ODMC/Proofs to linearly indep solutions of Home.LDE.pdf
./ODMC/Transforms & inverse transforms of elementary functions.pdf
./ODMC/New Document(50).pdf
./ODMC/Tutorial

```

```

(base) amaan@ubunx ~/Documents ➤ find /var/log -mtime -7
/var/log
/var/log/apport.log.1
/var/log/dpkg.log
/var/log/apport.log search for files within a specific directory and perform subsequent operations.
/var/log/kern.log.1
/var/log/kern.log ports searching by file (-f), folder (-d), name (-name), creation date, modification
/var/log/ubuntu-advantage-timer.log and permissions. By using the '-exec' other UNIX commands can be executed or
/var/log/auth.log
find: '/var/log/private': Permission denied
/var/log/syslog.1 INFORMAL SYNTAX : find [dir to start][type of obj] [options] [name/pattern to f
/var/log/journal/197fe80350fa44ffa4275de8750c12d2
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@3368b42aa12a49b5a0438b7662f638d5-000000000000000005d-0005f13062334b35.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@5323aa30d3974b7c9f4b36caa02fe927-00000000000000000fc-0005f1924cb9c0cc.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@5323aa30d3974b7c9f4b36caa02fe927-00000000000000000ac-0005f1883d0eb955.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/user-1000@cf51cbd0d54e4564978a835f923a5ecd-0000000000000000071146-0005f130af35dcf8.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@3368b42aa12a49b5a0438b7662f638d5-000000000000000007ab-0005f11b846ae6b9.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/user-1000@cf51cbd0d54e4564978a835f923a5ecd-00000000000000000773e0-0005f1838cffb331.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@3368b42aa12a49b5a0438b7662f638d5-00000000000000000001-0005f11b72e81e0f.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/user-1000@cf51cbd0d54e4564978a835f923a5ecd-000000000000000006fe2c-0005f11cf427a8cc.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/user-1000@cf51cbd0d54e4564978a835f923a5ecd-0000000000000000071066-0005f13063a7a049.journal
/var/log/journal/197fe80350fa44ffa4275de8750c12d2/system@3368b42aa12a49b5a0438b7662f638d5-0000000000000000071066-0005f13063a7a049.journal

```

12] grep

```
base) amaan@ubunx ~/Documents ls amaan@ubunx:~/Documents
112103008_FCS_TUT3 docker.txt IMP_PD.docx 'SEM III.zip'
112103008_FCS_TUT3.zip DSA_Assign Impression test.c
a-hello.i DSGT luffy.jpg ... contd test.cpp
a-hello.o DTL 'Machine Learning' wallpaper
a-hello.s FCS_sse.txt ODMC Zest.zip
a.out hello.c
Cloud_AWS
(base) amaan@ubunx ~/Documents grep use docker.txt
least privileged user
use tags for vernerabilitiy
use copy instead of add
use a static code analysis
(base) amaan@ubunx ~/Documents
```

EXAMPLES:

```
(base) amaan@ubunx ~/Documents grep -i."use".*.txt and's output will display lines that contain blue
docker.txt:- Use of labels
docker.txt:- least privileged user
docker.txt:- use tags for vernerabilitiy
docker.txt:- use copy instead of add
docker.txt:- use a static code analysis
FCS_sse.txt:Overall, SSEs are an important measure of the performance of a feedback control system, and
they can be used to determine how well the system is tracking the desired input.
FCS_sse.txt:In a feedback control system, the static error constants (SECs) are used to describe the sys
tem's steady state error (SSE) in response to different types of inputs. There are four types of SECs:
Kp, Ki, Kd, and Kf, and they are defined as follows:
FCS_sse.txt:SECs can be used to analyze the performance of a feedback control system and to design controllers that can reduce or eliminate steady state errors.
```

```
base) amaan@ubunx ~/Documents grep "the" * grep: 112103008_FCS_TUT3: Is a directory
grep: Cloud_AWS: Is a directory
grep: docker.txt: - find the vulnerability : Trivy tool
grep: DSA_Assign: Is a directory | regular expression print - find strings that match a regex, by
grep: DSGT: Is a directory searching through all the text in a specific file. Once the grep command
grep: DTL: Is a directory
grep: FCS_112103008_Amaan Jamadar Div- I_S-1: Is a directory
grep: FCS_sse.txt:In a feedback control system, the steady state error (SSE) is the difference between the de
sired input and the steady state output of the system. The steady state output is the output of the sys
tem when it has reached a steady state, or when it is no longer changing in response to a change in the
input.
FCS_sse.txt: Zero steady state error: This occurs when the steady state output of the system exactly
matches the desired input. In this case, the error is zero, and the system is said to be tracking the
input perfectly. | grep blue notepad.txt => To search for the word blue in
FCS_sse.txt: Finite steady state error: This occurs when the steady state output of the system is no
t exactly equal to the desired input, but the error is constant and finite. In this case, the system is
not tracking the input perfectly, but the error is small and the system is still performing well.
FCS_sse.txt: Infinite steady state error: This occurs when the steady state output of the system is
not equal to the desired input, and the error is infinite. In this case, the system is not performing
well and is not tracking the input at all. | grep -i "the" * is for all files in pwd.
FCS_sse.txt:Overall, SSEs are an important measure of the performance of a feedback control system, and
they can be used to determine how well the system is tracking the desired input.
FCS_sse.txt:In a feedback control system, the static error constants (SECs) are used to describe the sys
tem's steady state error (SSE) in response to different types of inputs. There are four types of SECs:
Kp, Ki, Kd, and Kf, and they are defined as follows:
FCS_sse.txt: Kp: The proportional static error constant is the steady state error when the input is
a step function. It is equal to the steady state error divided by the step size.
FCS_sse.txt: Ki: The integral static error constant is the steady state error when the input is a r
```

13]df command

```
(base) amaan@ubunx ~/Documents > df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1.6G   2.5M  1.6G  1% /run
/dev/nvme0n1p5  47G   13G   32G  29% /
tmpfs           7.8G   0    7.8G  0% /dev/shm
tmpfs           5.0M   4.0K  5.0M  1% /run/lock
/dev/nvme0n1p7  944M  242M  637M  28% /boot
/dev/nvme0n1p8  296G  7.7G  273G  3% /home
/dev/nvme0n1p1  256M  55M   202M  22% /boot/efi
tmpfs           1.6G   2.4M  1.6G  1% /run/user/1000
(base) amaan@ubunx ~/Documents > df -m
Filesystem      1M-blocks  Used Available Use% Mounted on
tmpfs           1579       3       1577  1% /run
/dev/nvme0n1p5  47753     13008  32288  29% /
tmpfs           7893       0       7893  0% /dev/shm
tmpfs           5         1       5   1% /run/lock
/dev/nvme0n1p7  944       242     637  28% /boot
/dev/nvme0n1p8  302138    7871   278849  3% /home
/dev/nvme0n1p1  256       55     202  22% /boot/efi
tmpfs           1579       3       1577  1% /run/user/1000
(base) amaan@ubunx ~/Documents >
```

```
(base) amaan@ubunx ~/Documents ➤ df -k
Filesystem      1K-blocks   Used Available Use% Mounted on
tmpfs          1616336    2476  1613860  1% /run EXAMPLES:
/dev/nvme0n1p5 48898724 13320048 33062300 29% /
tmpfs          8081660     0  8081660  0% /dev/shm
tmpfs           5120      4    5116  1% /run/lock
/dev/nvme0n1p7  965872   247784  651684 28% /boot
/dev/nvme0n1p8 309388908 8059520 285540076 3% /home
/dev/nvme0n1p1  262144   55388  206756 22% /boot/efi as:
tmpfs          1616332    2456  1613876  1% /run/user/1000
(base) amaan@ubunx ~/Documents ➤ df -T
Filesystem      Type  1K-blocks   Used Available Use% Mounted on
tmpfs          tmpfs  1616336    2472  1613864  1% /run
/dev/nvme0n1p5 ext4  48898724 13320048 33062300 29% /
tmpfs          tmpfs  8081660     0  8081660  0% /dev/shm
tmpfs          tmpfs   5120      4    5116  1% /run/lock
/dev/nvme0n1p7 ext4   965872   247784  651684 28% /boot
/dev/nvme0n1p8 ext4 309388908 8059520 285540076 3% /home
/dev/nvme0n1p1 vfat  262144   55388  206756 22% /boot/efi
tmpfs          tmpfs  1616332    2456  1613876  1% /run/user/1000
(base) amaan@ubunx ~/Documents ➤
```

14] du

```
base) amaan@ubunx ~Documents du -m ~/Documents/Cloud_AWS  
/home/amaan/Documents/Cloud_AWS  
base) amaan@ubunx ~Documents du -s ~/Documents/Cloud_AWS  
/home/amaan/Documents/Cloud_AWS  
base) amaan@ubunx ~Documents du -k ~/Documents/Cloud_AWS  
/home/amaan/Documents/Cloud_AWS  
base) amaan@ubunx ~Documents du -h ~/Documents/Cloud_AWS  
/home/amaan/Documents/Cloud_AWS  
base) amaan@ubunx ~Documents [options] [file]
```

15] head

```
(base) amaan@ubunx ~/Documents ➤ head docker.txt
Best Practices :
  Use of labels
  least privileged user head
  use tags for vernerabilitiy command allows you to view the first
  use copy instead of add
  don't leak important info(credentials , config file ,git repo) to docker images
docker ignore file - for ignore some files from big repo
  Docker ignore file - for ignoring some files from big repo output piped
  data to the CLI.

  Multi-stage build
(base) amaan@ubunx ~/Documents ➤ head docker.txt -n 5
Best Practices :
  Use of labels
  least privileged user
  use tags for vernerabilitiy note.txt => To view the first ten lines
  use copy instead of add note.txt in pwd .
(base) amaan@ubunx ~/Documents ➤ head docker.txt -c 24
Best Practices :
  Use of labels
  Below are some options you can add:
  - Use o%
(base) amaan@ubunx ~/Documents ➤ head docker.txt -q
Best Practices :
  -c or -bytes prints the first customized number
  Use of labels
  of bytes.
  least privileged user
  use tags for vernerabilitiy
  use copy instead of add
  don't leak important info(credentials , config file ,git repo) to docker images
docker ignore file - for ignore some files from big repo
```

16] tail

```
(base) ✘ amaan@ubunx ~/Documents ➤ tail docker.txt
17. head
  The head command allows you to view the f
  Practice: playwithdocker.com
  + code to check:
    -copy
    -from
    -add
    -entrypoint
  The head command is also used to output pi
  data to the CLI.

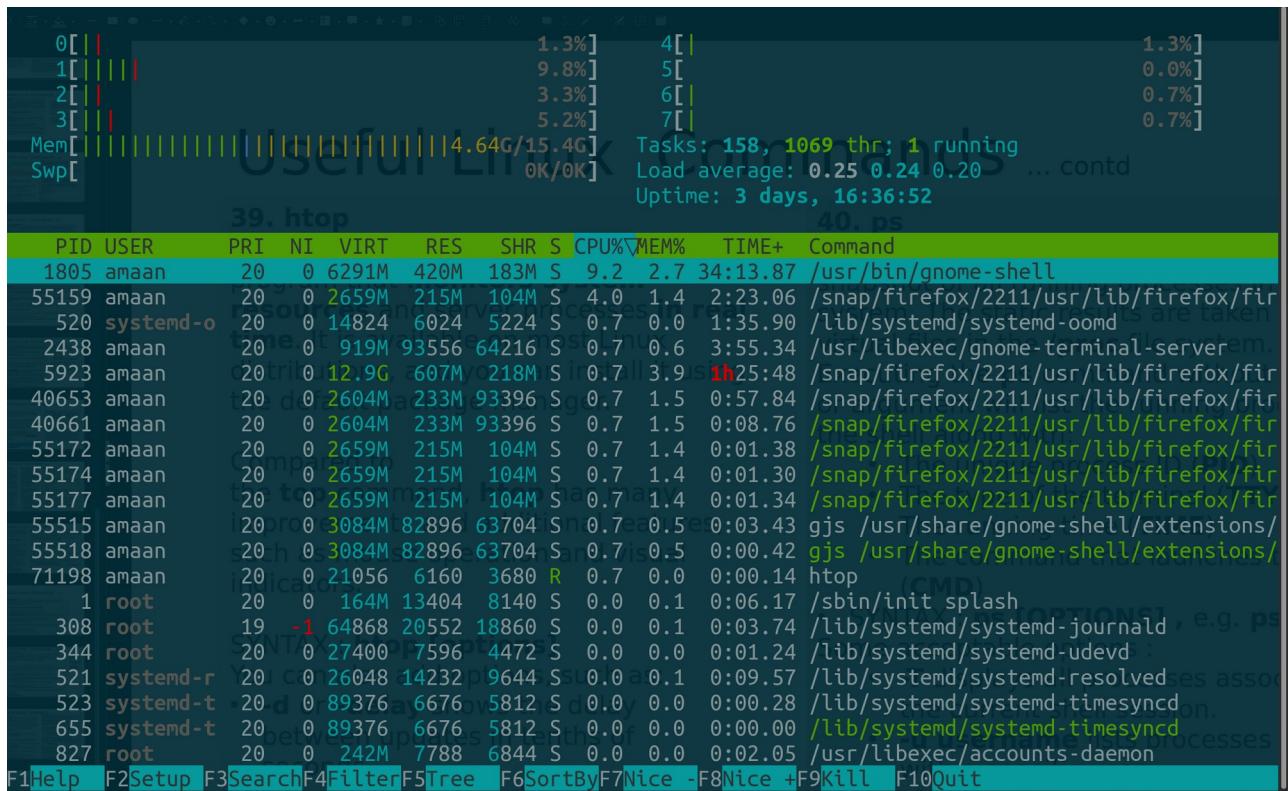
  SYNTAX : head [option] [file]
```

17] top

```
top - 16:34:56 up 3 days, 16:36, 1 user, load average: 0.22, 0.22, 0.20
Tasks: 317 total, 1 running, 315 sleeping, 0 stopped, 1 zombie
%Cpu(s): 2.8 us, 0.7 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 15784.5 total, 3235.9 free, 3662.1 used, 8886.5 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 10693.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2438	amaan	20	0	941276	93556	64216	S	14.3	0.6	3:54.61	gnome-terminal-
1805	amaan	20	0	6433584	429832	187884	S	7.3	2.7	34:11.73	gnome-shell
245	root	-51	0	0	0	0	S	2.3	0.0	1:37.81	irq/155-ELAN066
55159	amaan	20	0	2723128	221464	107400	S	2.3	1.4	2:21.72	Isolated Web Co
5923	amaan	20	0	12.6g	621728	224164	S	0.7	3.8	85:48.07	firefox
520	systemd+	20	0	14824	6024	5224	S	0.3	0.0	1:35.86	systemd-oomd
69603	root	0	-20	0	0	0	I	0.3	0.0	0:00.79	kworker/u17:0-i915_flip
70791	root	20	-20	0	0	0	I	0.3	0.0	0:00.45	kworker/1:0-events
71027	amaan	20	0	21868	4284	3428	R	0.3	0.0	0:00.02	top
1	root	20	0	168160	13404	8140	S	0.0	0.1	0:06.17	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.09	kthread
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
12	root	20	0	0	0	0	S	0.0	0.0	0:00.50	ksoftirqd/0
13	root	20	0	0	0	0	I	0.0	0.0	0:34.12	rcu_sched
14	root	rt	-20	0	0	0	I	0.0	0.0	0:00.48	migration/0
15	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1

18] htop



19] diff

```
(base) amaan@ubunx ~/Documents diff cybersecurity.txt FCS_sse.txt  
1,4c1,3  
< First Course on - Basics of IT  
< second course on - basics of cybersecurity  
< choose a path in cybersecurity  
< 'comptia security +' -- certification  
--  
> Steady state errors (SSE) for feedback systems:  
>  
> In a feedback control system, the steady state error (SSE) is the difference between the desired input and the steady state output of the system. The steady state output is the output of the system when it has reached a steady state, or when it is no longer changing in response to a change in the input.  
5a5,32  
> There are three types of SSEs that can occur in a feedback control system:  
>  
> Zero steady state error: This occurs when the steady state output of the system exactly matches the desired input. In this case, the error is zero, and the system is said to be tracking the input perfectly.  
>  
> Finite steady state error: This occurs when the steady state output of the system is not exactly equal to the desired input, but the error is constant and finite. In this case, the system is not tracking the input perfectly, but the error is small and the system is still performing well.
```

20] tar

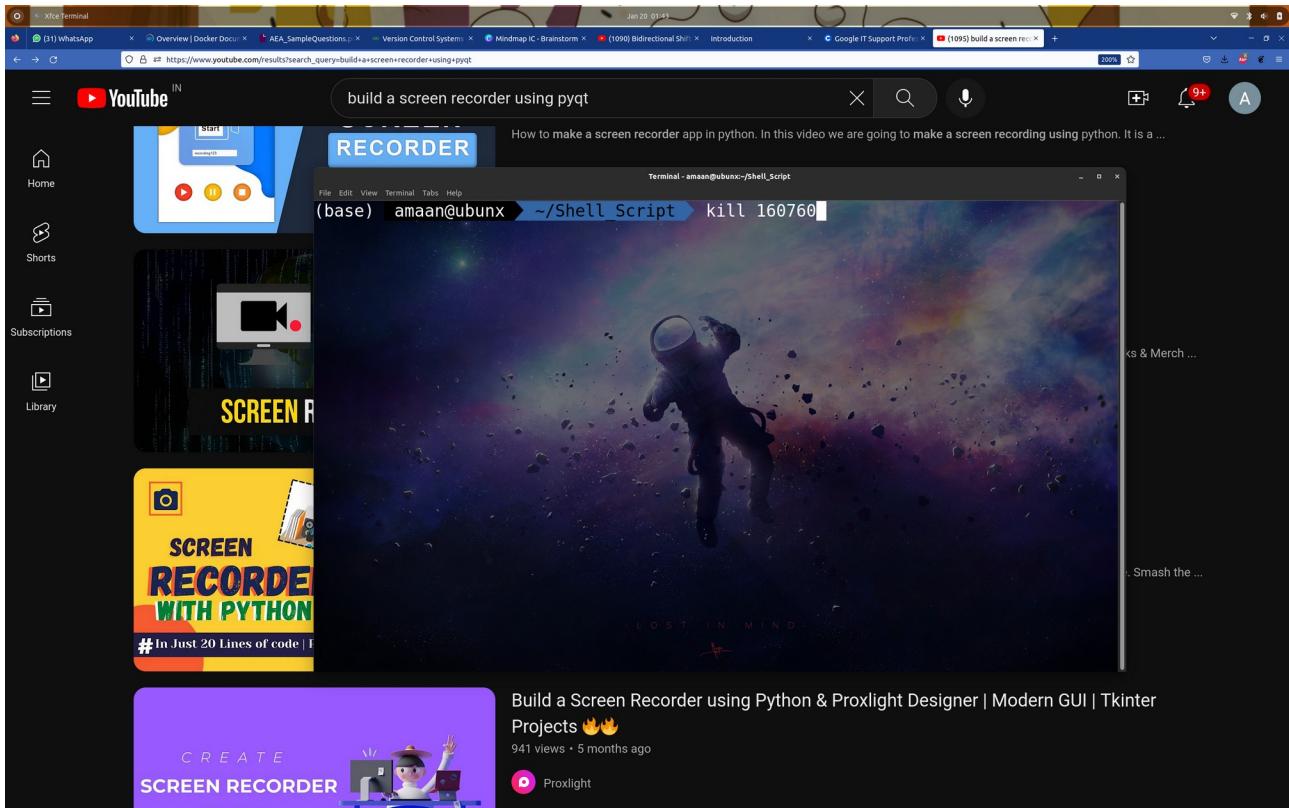
```
(base) amaan@ubunx ~/Documents ls  
112103008_FCS_TUT3  
112103008_FCS_TUT3.zip  
a-hello.i  
a-hello.o  
a-hello.s  
a.out  
Cofsug  
cybersecurity.txt  
DevOps  
DSA_Assign  
DSA_Course  
DSGT  
DTL  
'FCS_112103008_Amaan Jamadar Div- I_S-1'  
(base) amaan@ubunx ~/Documents tar -cvf DTL_arc.tar ~/Documents/DTL  
tar: Removing leading '/' from member names  
/home/amaan/Documents/DTL/  
/home/amaan/Documents/DTL/Research_paper_Assignment_6.zip  
/home/amaan/Documents/DTL/Assignment4DTL/  
/home/amaan/Documents/DTL/Assignment4DTL/assignment4.synctex.gz  
/home/amaan/Documents/DTL/Assignment4DTL/assignment4.log  
/home/amaan/Documents/DTL/Assignment4DTL/assignment4.aux  
/home/amaan/Documents/DTL/Assignment4DTL/texput.log
```

```
/home/amaan/Documents/DTL/Assignment3DTL/DTL_Assignment3.pdf
/home/amaan/Documents/DTL/Assignment3DTL/DTL_Assignment3.tex
/home/amaan/Documents/DTL/Assignment3DTL/Bugati.jpg
/home/amaan/Documents/DTL/Assignment3DTL/DTL_Assignment3.synctex.gz
/home/amaan/Documents/DTL/Assignment3DTL/watch.jpeg
/home/amaan/Documents/DTL/Assignment3DTL/DTL_Assignment3.toc
/home/amaan/Documents/DTL/Assignment3DTL/DTL_Assignment3.aux
/home/amaan/Documents/DTL/sample report.png
(base) amaan@ubunx ~/Documents ls
112103008_FCS_TUT3 FCS_112103008_Amaan Jamadar Div- I_S-I
112103008_FCS_TUT3.zip FCS_sse.txt
a-hello.i hello.c
a-hello.o IMP_PD.docx
a-hello.s Impression
a.out luffy.jpg
Cofsug ODMC
cybersecurity.txt PPL
DevOps SEM III
DSA_Assign SEM III.zip
DSA_Course test.c
DSGT test.cpp
DTL wallpaper
DTL_arc.tar Zest.zip
(base) amaan@ubunx ~/Documents
```

21] chmod

```
(base) amaan@ubunx ~/Documents cd ..
(base) amaan@ubunx ~ cd Shell_Script
(base) amaan@ubunx ~/Shell_Script ls
bbc_scraper.sh
(base) amaan@ubunx ~/Shell_Script chmod u+x bbc_scraper.sh
(base) amaan@ubunx ~/Shell_Script
```

22] kill



A screenshot of a presentation slide titled "Useful Linux Commands". The slide is numbered 24. The content discusses the "kill" command, explaining its purpose and how to use it. It includes a section on "SYNAX" (after knowing the program's PID) and a list of signal types, including SIGTERM and SIGKILL. A terminal window in the background shows the command "kill 160760" being run. The presentation slide has a sidebar with multiple slides visible, and the status bar at the bottom indicates "Slide 21 of 29".

24. kill
To terminate an application
signal misbehaving applications
close their processes
To kill a program
identification number

If you don't know the process ID
ps ux

SYNAX (after knowing the program's PID) :

There are **64 signals** available which are among the most common
• **SIGTERM** request to stop a process gives it some time to finish what it is doing. The system will use the signal when the process does not respond to the signal when it receives it.
• **SIGKILL** forces programs to stop, and you will lose unsaved progress.
• **kill SIGKILL 63773 =>** program's PID is **63773**, and you want to force it to stop:

23] ping

```
(base) amaan@ubunx ~/Shell Script ping google.com
PING google.com (142.250.192.142) 56(84) bytes of data.
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=1 ttl=114 time=18.9 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=2 ttl=114 time=14.4 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=3 ttl=114 time=41.4 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=4 ttl=114 time=58.7 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=5 ttl=114 time=69.4 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=6 ttl=114 time=9.53 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=7 ttl=114 time=12.3 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=8 ttl=114 time=11.9 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=9 ttl=114 time=14.0 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=10 ttl=114 time=89.3 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=11 ttl=114 time=11.5 ms
```

```
time=15.0 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=28 ttl=114 time=13.2 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=29 ttl=114 time=37.5 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=30 ttl=114 time=11.8 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=31 ttl=114 time=58.3 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=32 ttl=114 time=10.9 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=33 ttl=114 time=9.36 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=34 ttl=114 time=16.9 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=35 ttl=114 time=10.6 ms
64 bytes from bom12s18-in-f14.1e100.net (142.250.192.142): icmp_seq=36 ttl=114 time=11.5 ms
^C
--- google.com ping statistics ---
36 packets transmitted, 36 received, 0% packet loss, time 35057ms
rtt min/avg/max/mdev = 7.123/21.648/89.252/19.396 ms
(base) amaan@ubunx ~/Shell Script
```

24] wget

```
(base) amaan@ubunx ➤ ~/Shell Script ➤ wget https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf
--2023-01-20 01:45:42-- https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf
Resolving www.loggly.com (www.loggly.com)... 23.212.5.48, 23.212.5.40
Connecting to www.loggly.com (www.loggly.com)|23.212.5.48|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 987562 (964K) [application/pdf]
Saving to: 'Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf'

Linux-Cheat-Sheet-S 100%[=====] 964.42K 668KB/s in 1.4s

2023-01-20 01:45:45 (668 KB/s) - 'Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf' saved [987562/987562]

(base) amaan@ubunx ➤ ~/Shell Script ➤
```

```
(base) amaan@ubunx ➤ ~/Shell Script ➤ wget https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf
--2023-01-20 01:45:42-- https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf
Resolving www.loggly.com (www.loggly.com)... 23.212.5.48, 23.212.5.40
Connecting to www.loggly.com (www.loggly.com)|23.212.5.48|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 987562 (964K) [application/pdf]
Saving to: 'Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf'

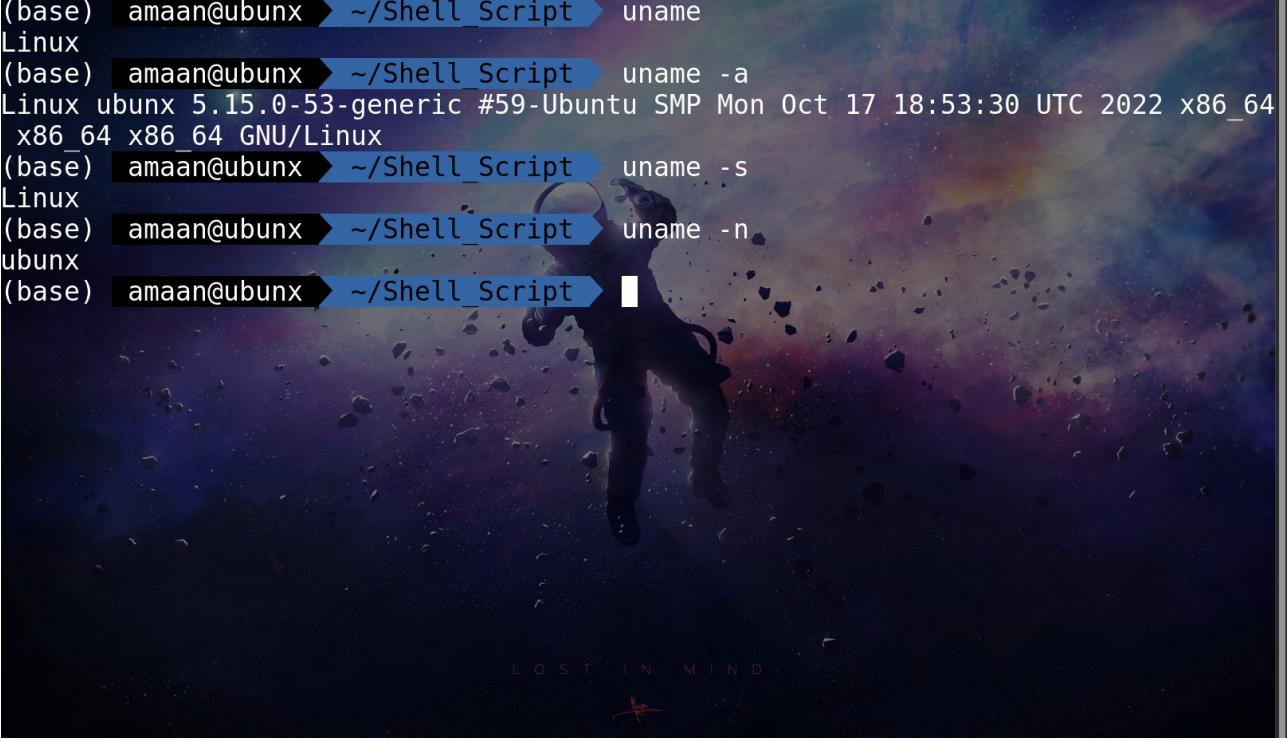
Linux-Cheat-Sheet-S 100%[=====] 964.42K 668KB/s in 1.4s

2023-01-20 01:45:45 (668 KB/s) - 'Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf' saved [987562/987562]

(base) amaan@ubunx ➤ ~/Shell Script ➤ ls
bbc_scraper.sh Linux-Cheat-Sheet-Sponsored-By-Loggly.pdf
(base) amaan@ubunx ➤ ~/Shell Script ➤
```

25] uname

```
(base) amaan@ubunx ~/Shell Script uname
Linux
(base) amaan@ubunx ~/Shell Script uname -a
Linux ubunx 5.15.0-53-generic #59-Ubuntu SMP Mon Oct 17 18:53:30 UTC 2022 x86_64
x86_64 x86_64 GNU/Linux
(base) amaan@ubunx ~/Shell Script uname -s
Linux
(base) amaan@ubunx ~/Shell Script uname -n
ubunx
(base) amaan@ubunx ~/Shell Script █
```



26] history

```
(base) amaan@ubunx ~/Shell Script history █
```

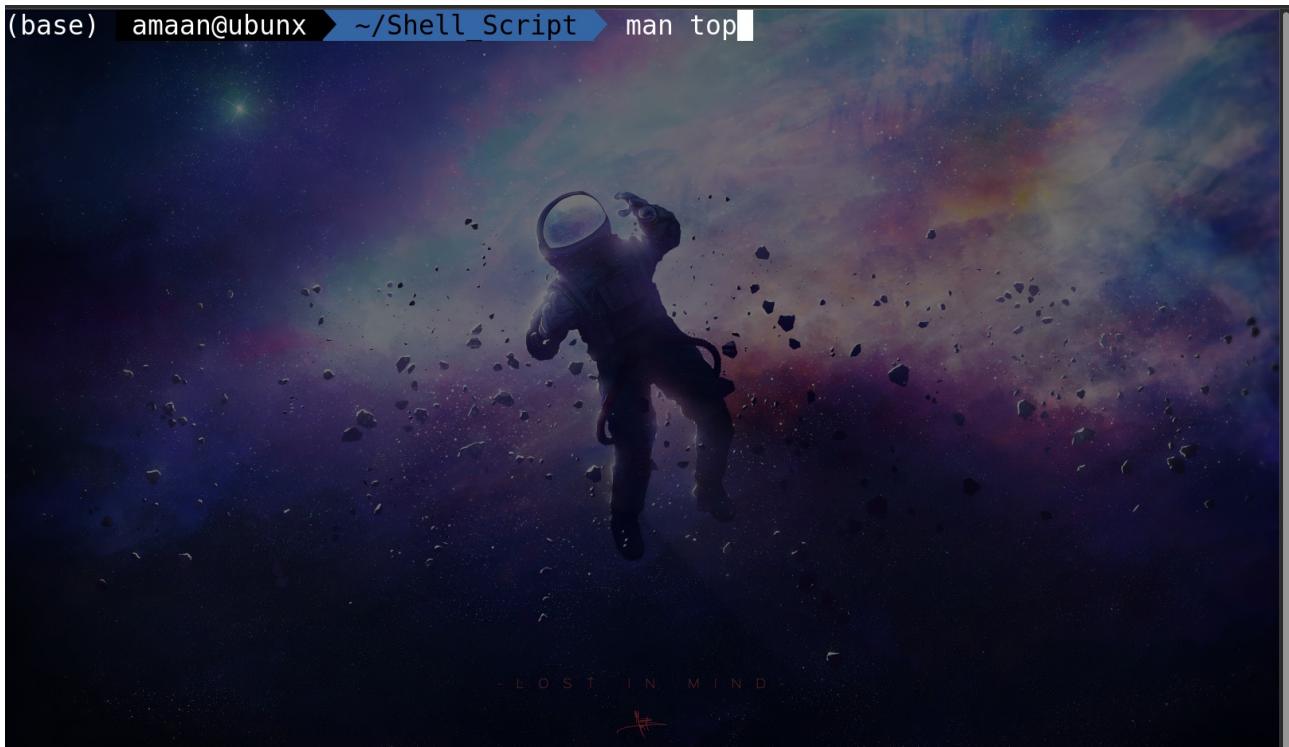


```
1700 ls
1701 chmod u+x bbc_scraper.sh
1702 clear
1703 jobs -l
1704 clear
1705 jobs -l
1706 clear
1707 top
1708 clear
1709 kill 160760
1710 clear
1711 ping google
1712 clear
1713 ping google.com
1714 clear
1715 wget https://www.loggly.com/wp-content/uploads/2015/05/Linux-Cheat-Sheet-
Sponsored-By-Loggly.pdf
1716 ls
1717 clear
1718 uname
1719 uname -a
1720 uname -s
1721 uname -n
1722 clear
```

- LOST IN MIND -

27] man

```
(base) amaan@ubunx ~/Shell_Script man top
```



TOP(1)

User Commands

TOP(1)

NAME

`top` - display Linux processes

SYNOPSIS

`top [-hv|-bcEeHiOSs1 -d secs -n max -u|U user -p pids -o field -w [cols]]`

The traditional switches `-' and whitespace are optional.

DESCRIPTION

The `top` program provides a dynamic real-time view of a running system. It can display `system` summary information as well as a list of `processes` or `threads` currently being managed by the Linux kernel. The types of system summary information shown and the types, order and size of information displayed for processes are all user configurable and that configuration can be made persistent across restarts.

The program provides a limited interactive interface for process manipulation as well as a much more extensive interface for personal configuration -- encompassing every aspect of its operation. And while `top` is referred to throughout this document, you are free to name the program anything you wish. That new name, possibly an alias, will

Manual page `top(1)` line 1 (press h for help or q to quit)

28] echo

```
(base) amaan@ubunx ~/Shell_Script> echo " DTL ASSIGNMENT Amaan "
DTL ASSIGNMENT Amaan
(base) amaan@ubunx ~/Shell_Script> echo -e " DTL ASSIGNMENT \bAmaan "
DTL ASSIGNMENTAmaan
(base) amaan@ubunx ~/Shell_Script> echo -e " DTL ASSIGNMENT \cAmaan "
DTL ASSIGNMENT %
(base) amaan@ubunx ~/Shell_Script> echo -e " DTL \nASSIGNMENT \nAmaan "
DTL
ASSIGNMENT
Amaan
(base) amaan@ubunx ~/Shell_Script>
```

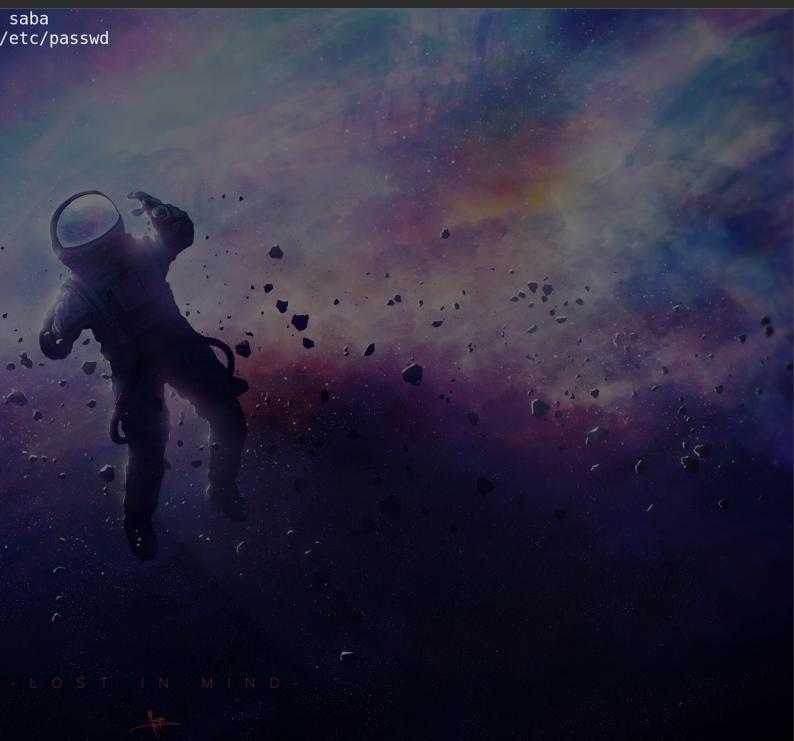
29] hostname

```
(base) amaan@ubunx ~/Documents> hostname
ubunx
(base) amaan@ubunx ~/Documents> hostname -A
ubunx ubunx
(base) amaan@ubunx ~/Documents> hostname -i
127.0.1.1
(base) amaan@ubunx ~/Documents> hostname -I
10.100.106.248 172.17.0.1
(base) amaan@ubunx ~/Documents>
```

30] zip

```
base) amaan@ubunx ➤ ~/Documents ➤ ls
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cofsug
cybersecurity.txt
DevOps
DSA_Assign
DSA_Course
DSGT
DTL
FCS_112103008_Amaan Jamadar Div- I_S-1'
base) amaan@ubunx ➤ ~/Documents ➤ zip new_zif.zip test.c test.cpp
adding: test.c (deflated 30%)
adding: test.cpp (deflated 30%)
base) amaan@ubunx ➤ ~/Documents ➤ ls
112103008_FCS_TUT3
112103008_FCS_TUT3.zip
a-hello.i
a-hello.o
a-hello.s
a.out
Cofsug
cybersecurity.txt
DevOps
DSA_Assign
DSA_Course
DSGT
DTL
FCS_112103008_Amaan Jamadar Div- I_S-1'
base) amaan@ubunx ➤ ~/Documents ➤ ls
FCS_sse.txt
hello.c
IMP_PD.docx
Impression
luffy.jpg
ODMC
PPL
'SEM III'
'SEM III.zip'
test.c
test.cpp
wallpaper
Zest.zip
```

31] useradd and userdel



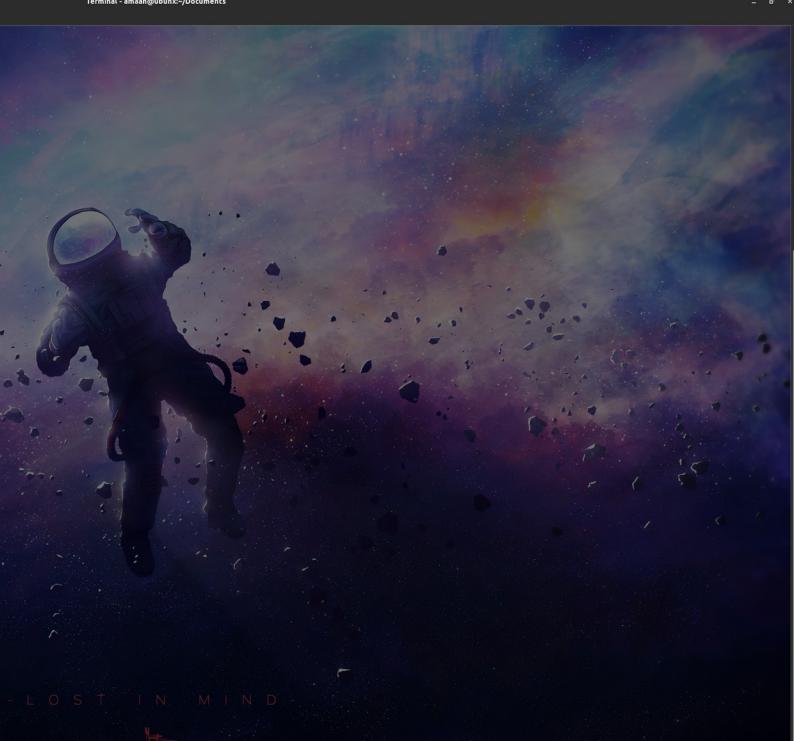
Xfce Terminal

File Edit View Terminal Tabs Help

Terminal - amaan@ubunx:~/Documents

```
(base) amaan@ubunx ~/Documents sudo userdel saba
(base) amaan@ubunx ~/Documents cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
tss
uuid
systemd-oom
tcpdump
avahi-autoipd
usbmux
dnsmasq
kernoops
avahi
cups-pk-helper
rtkit
whoopsie
```

- LOST IN MIND -



Xfce Terminal

File Edit View Terminal Tabs Help

Terminal - amaan@ubunx:~/Documents

```
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
tss
uuid
systemd-oom
tcpdump
avahi-autoipd
usbmux
dnsmasq
kernoops
avahi
cups-pk-helper
rtkit
whoopsie
sssd
speech-dispatcher
nm-openvpn
saned
colord
geoclue
pulse
gnome-initial-setup
hplip
gdm
amaan
fwupd-refresh
_flatpak
mysql
saba
(base) amaan@ubunx ~/Documents
```

- LOST IN MIND -



A screenshot of an Xfce terminal window titled "Xfce Terminal". The terminal is displaying a list of system services or processes. A tooltip "Screenshot captured" is visible at the top right. The background of the terminal window features a dark, space-themed image of an astronaut floating in a nebula.

```
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
tss
uidd
systemd-oom
tcpdump
avahi-autoipd
usbmux
dnsmasq
kernoops
avahi
cups-pk-helper
rtkit
whoopsie
sssd
speech-dispatcher
nm-openvpn
saned
colord
geoclue
pulse
gnome-initial-setup
hplip
gdm
amaan
fwupd-refresh
_flatpak
mysql
(base) amaan@ubunx ~/Documents
(base) amaan@ubunx ~/Documents
```

32] vim , nano editors



A screenshot of an Xfce terminal window titled "Terminal - amaan@ubunx ~/Documents". The terminal is showing the output of the "ls" command, listing various files and folders in the current directory. The background of the terminal window features a dark, space-themed image of an astronaut floating in a nebula.

```
(base) amaan@ubunx ~/Documents > ls
112103008_FCS_TUT3      a.out          DSA_Course
112103008_FCS_TUT3.zip   Cofsgug       DSGT
a-hello.i                cybersecurity.txt  DTL
a-hello.o                DevOps        'FCS_112103008_Amaan Jamadar Div- I_S-1'
a-hello.s                DSA_Assign    FCS_sse.txt
(base) amaan@ubunx ~/Documents > vim hello.c
```

Xfce Terminal

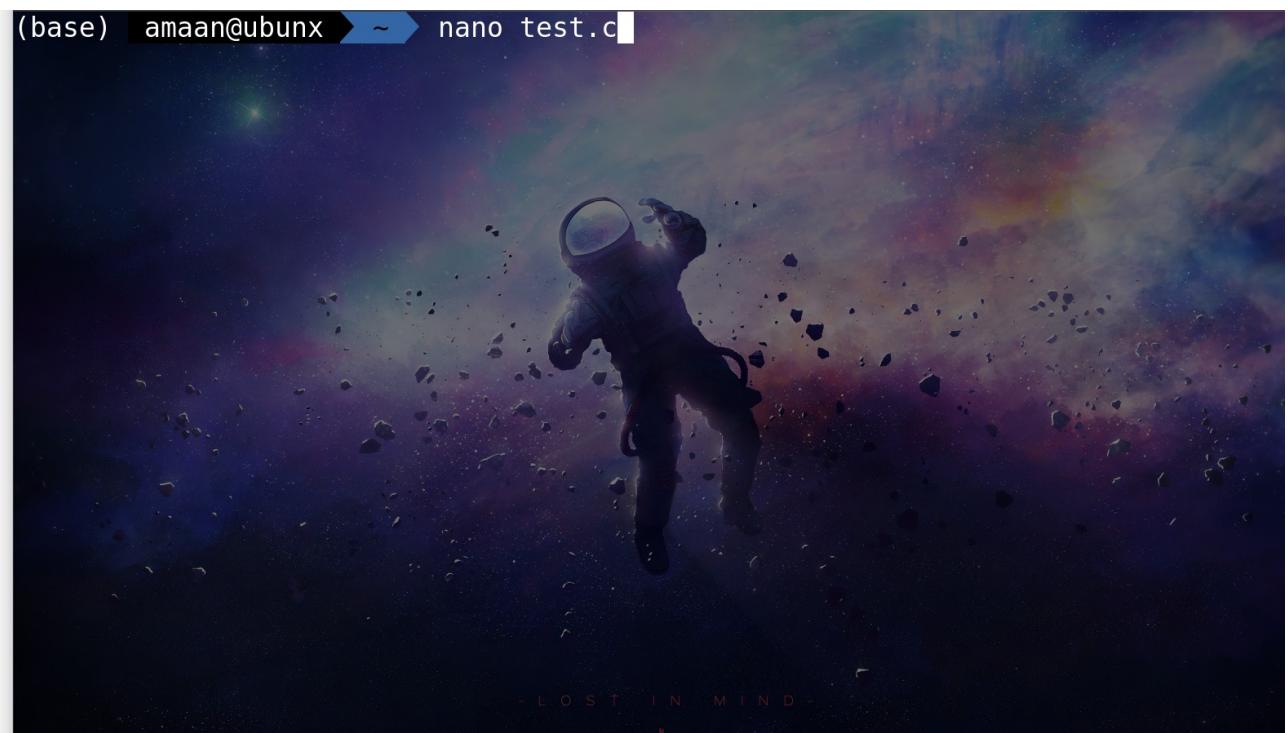
File Edit View Terminal Tabs Help

1
2 |
3 int main(){
4 int a[10];
5 int b;
6 return 0;
7 }

-- INSERT --

2,1 All

This screenshot shows a terminal window titled 'Xfce Terminal' running the 'vim' editor. The buffer contains a simple C program with line numbers 1 through 7. The cursor is at line 2. The status bar at the bottom indicates '-- INSERT --'. The window title bar also shows 'Terminal - vim helloc'. The terminal interface includes a menu bar with File, Edit, View, Terminal, Tabs, and Help.



GNU nano 6.2 test.c

```
#include<stdio.h>

int f(int *x, int c){
    c= c-1;
    if(c == 0) return 1;
    *x = *x + 1 ;
    return (*x) * f(x, c);
}

int main(){
//    int array[5] = {1,2,3,4,5};
//    int p = 5;
//    p = (array + 1);
//    printf("%d",*p );
//    printf("%d", f(&p,p));
    return 0;
}
```

File to insert [from ./]: M-F New Buffer ^X Execute Command

33] alias

```
(base)  amaan@ubunx ~/Documents alias fi=figlet
(base)  amaan@ubunx ~/Documents fi HI
```



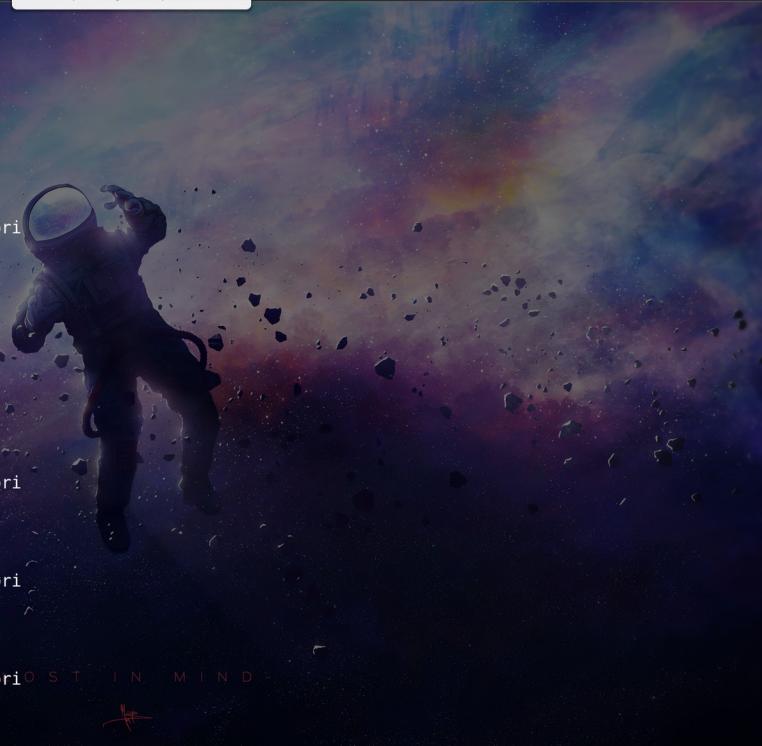
```
(base)  amaan@ubunx ~/Documents
```

- LOST IN MIND -

34]ps command



```
(base) amaan@ubunx ~$ ps -A
  PID TTY      TIME CMD
 199185 pts/1    00:00:00 Socket Process
 199204 pts/1    00:00:06 WebExtensions
 199224 pts/1    00:00:03 Privileged Cont
 199506 pts/1    00:00:00 RDD Process
 199510 pts/1    00:00:00 Utility Process
 199642 pts/1    00:00:17 Isolated Web Co
 199648 pts/1    00:00:46 Isolated Web Co
 199723 ?        00:00:00 kworker/0:0-events
 200167 ?        00:00:01 kworker/3:2-cgroup_destroy
 200300 ?        00:00:00 kworker/1:0-events
 200632 ?        00:00:00 kworker/7:2-events
 200639 ?        00:00:00 kworker/5:2-events
 200714 ?        00:00:00 kworker/4:2-events
 200752 ?        00:00:00 kworker/u16:2-events_unbound
 201072 ?        00:00:00 kworker/u16:3-flush-259:0
 201087 ?        00:00:00 kworker/u17:2-i915_flip
 201179 ?        00:00:00 kworker/u16:4-events_unbound
 201206 ?        00:00:00 kworker/7:0-events
 201213 ?        00:00:00 kworker/5:0-kec_query
 201234 ?        00:00:01 kworker/4:0-cgroup_destroy
 201272 ?        00:00:00 kworker/6:0-events
 201338 ?        00:00:00 kworker/2:0-mm_percpu_wq
 201395 pts/1    00:00:00 Web Content
 201431 ?        00:00:00 kworker/0:1-events
 201453 pts/1    00:00:00 Web Content
 201565 ?        00:00:00 kworker/3:0-cgroup_destroy
 201566 ?        00:00:00 kworker/1:1-events
 201684 pts/0    00:00:00 zsh
 201795 ?        00:00:00 kworker/2:2-events
 201874 ?        00:00:00 kworker/1:2-events
 201924 ?        00:00:00 kworker/3:1-events
 201925 ?        00:00:00 kworker/u17:1
 201952 pts/1    00:00:00 Web Content
 201966 ?        00:00:00 kworker/5:1
 202048 ?        00:00:00 kworker/4:1-events
 202056 ?        00:00:00 kworker/4:3
 202080 pts/0    00:00:00 ps
(base) amaan@ubunx ~$
```



```
(base) amaan@ubunx ~$ ps
  PID TTY      TIME CMD
 201684 pts/0    00:00:00 zsh
 202073 pts/0    00:00:00 ps
(base) amaan@ubunx ~$ ps -A
  PID TTY      TIME CMD
  1 ?        00:00:16 systemd
  2 ?        00:00:00 kthreadd
  3 ?        00:00:00 rcu_gp
  4 ?        00:00:00 rcu_par_gp
  5 ?        00:00:00 netns
  7 ?        00:00:00 kworker/0:0H-events_highpri
  9 ?        00:00:00 mm_percpu_wq
 10 ?       00:00:00 rcu_tasks_rude_
 11 ?       00:00:00 rcu_tasks_trace
 12 ?       00:00:03 ksoftirqd/0
 13 ?       00:03:00 rcu_sched
 14 ?       00:00:02 migration/0
 15 ?       00:00:00 idle_inject/0
 17 ?       00:00:00 cpuhp/0
 18 ?       00:00:00 cpuhp/1
 19 ?       00:00:00 idle_inject/1
 20 ?       00:00:02 migration/1
 21 ?       00:00:02 ksoftirqd/1
 23 ?       00:00:00 kworker/1:0H-events_highpri
 24 ?       00:00:00 cpuhp/2
 25 ?       00:00:00 idle_inject/2
 26 ?       00:00:02 migration/2
 27 ?       00:00:02 ksoftirqd/2
 29 ?       00:00:00 kworker/2:0H-events_highpri
 30 ?       00:00:00 cpuhp/3
 31 ?       00:00:00 idle_inject/3
 32 ?       00:00:02 migration/3
 33 ?       00:00:01 ksoftirqd/3
 35 ?       00:00:00 kworker/3:0H-events_highpri
 36 ?       00:00:00 cpuhp/4
 37 ?       00:00:00 idle_inject/4
 38 ?       00:00:02 migration/4
(base) amaan@ubunx ~$
```