

# Operating Systems Practical File

B.Tech. CSE

Daksh Verma

Instructor:

Mr. Amit Chauhan

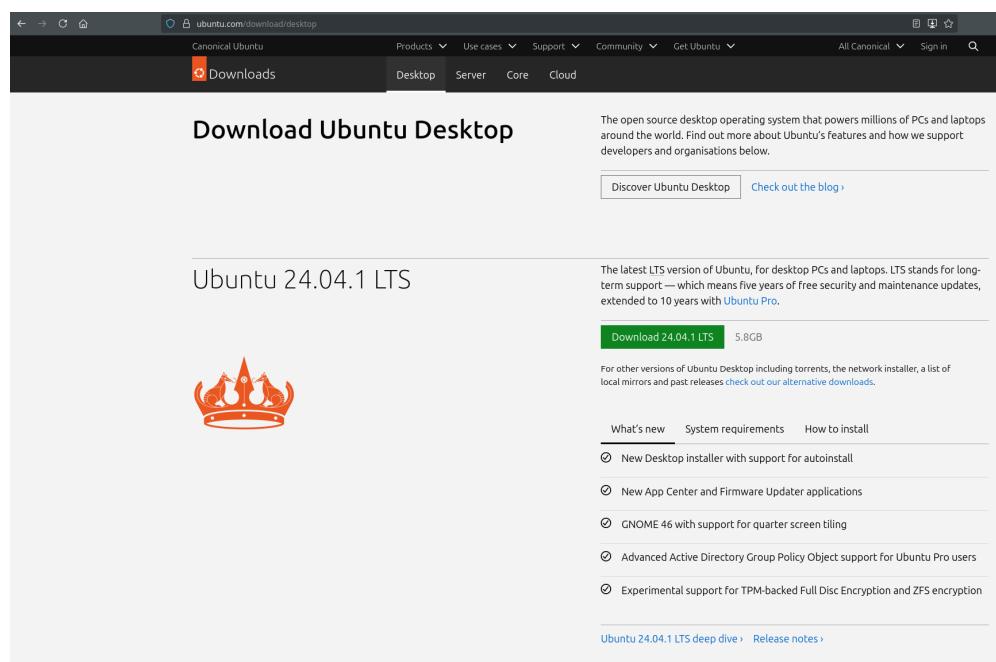
## Experiment 1

Aim: Installing and exploring various operating systems on a physical or virtual machine ( linux/windows )

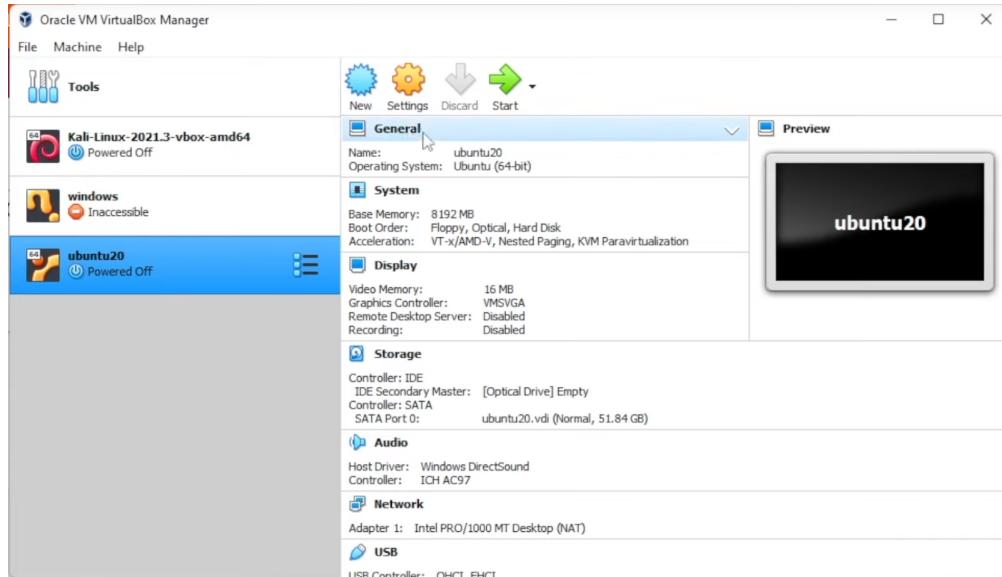
Procedure:

Virtual machine:

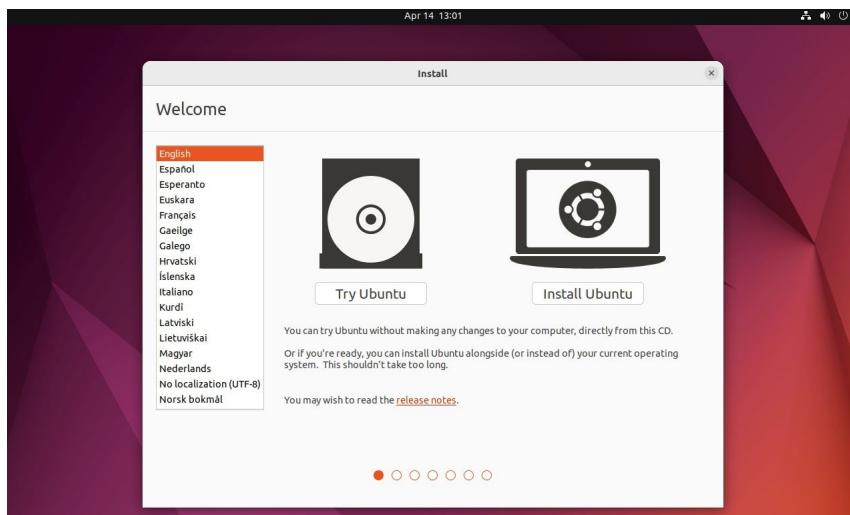
1. If you want to install the operating system on a virtual machine get a VM like VirtualBox, VMware, Hyper-V, Gnome Boxes or similar.
2. Download the ISO of the desired Operating System. In this case we are gonna get the Ubuntu ISO from the Official website.



3. Once we get the ISO file we will launch the Virtual machine, allocate the necessary resources and load the installer ISO into it.



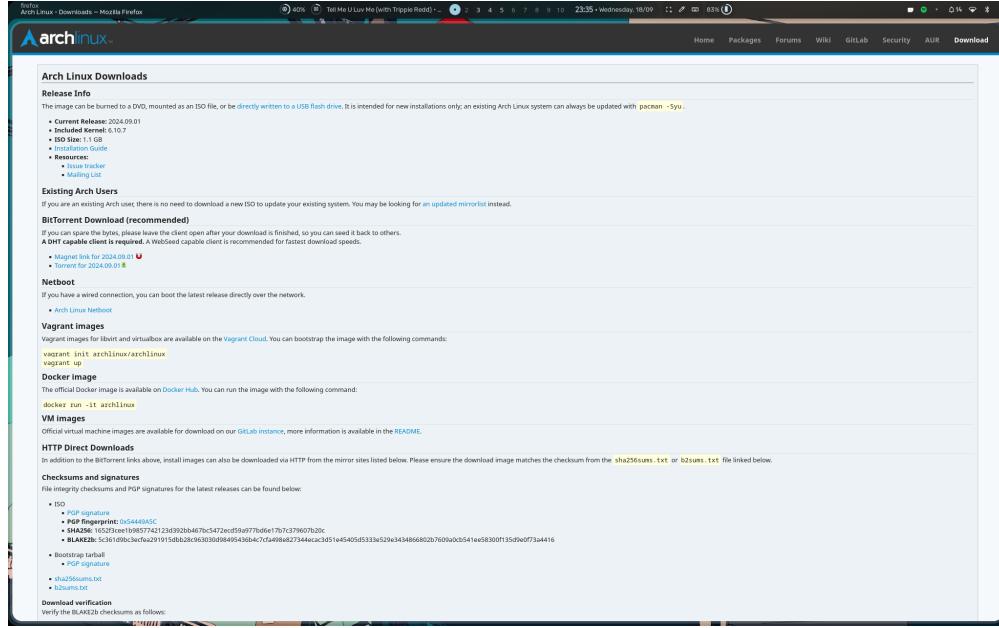
- After running the installation and rebooting the machine we will find ourself working on a live installation of our chosen operating system. (Note that the installation will differ from OS to OS and version to version, refer to official docs in case of any changes)



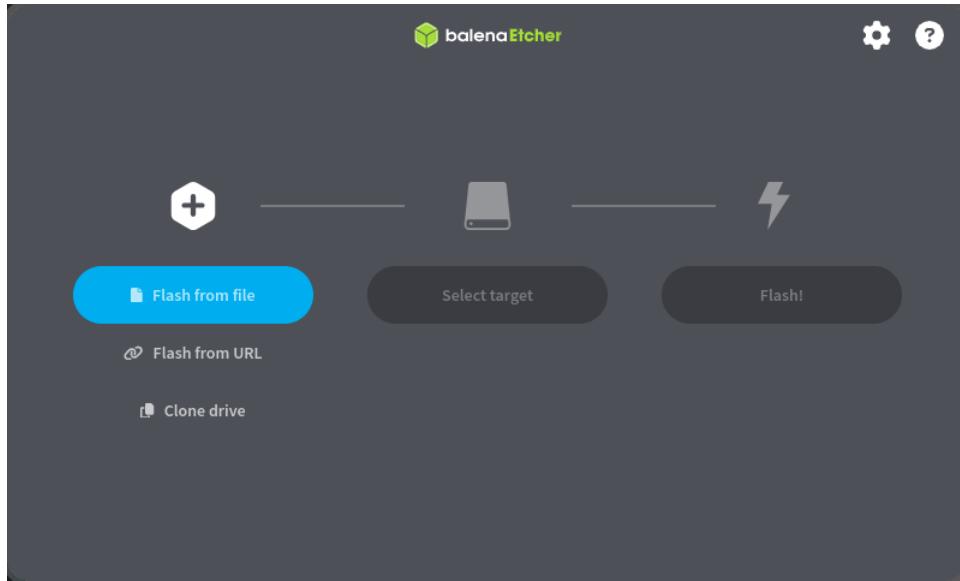
- Congratulations! you have sucessfully installed an OS on a virtual machine.

## Bare Metal Installation:

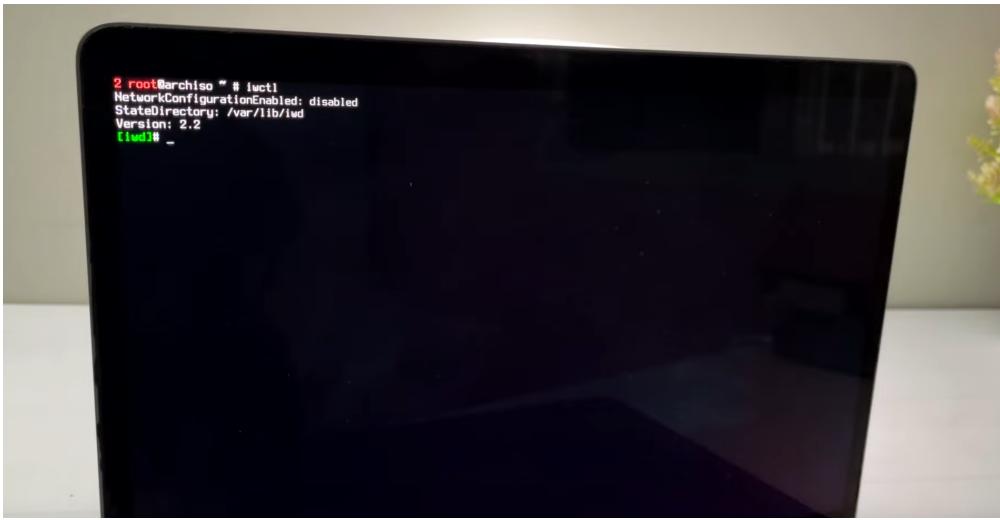
1. If you want to install the operating system on bare metal get a sufficiently large flash drive and a tool like rufus or balena etcher to burn the ISO file to the flash drive.
2. After collecting the necessary paraphanelia, download the ISO of the chosen OS.



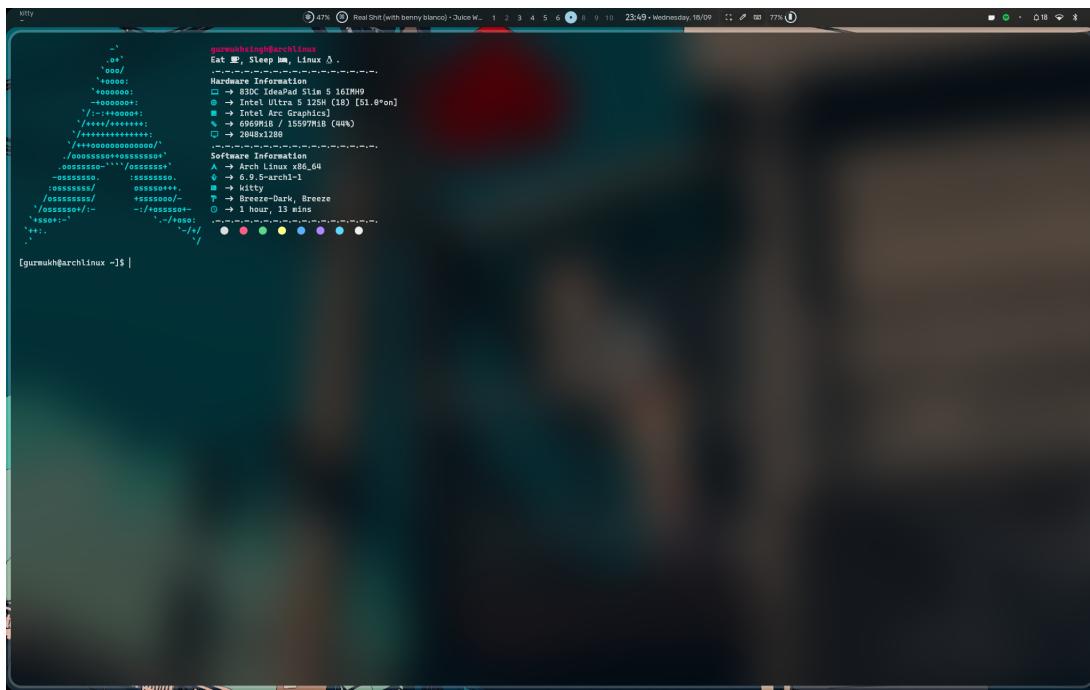
3. Etch the ISO onto the collected flash drive and make it bootable. This can be done with either rufus or balena etcher.



4. Boot the USB into the chosen computer and boot into the USB. The process would depend on your Computer. Refer to the official docs for the procedure on your system
5. Follow the instructions to install the OS on your system on bare metal.



6. After following all the installation procedure correctly, reboot your machine.
7. Congratulations! you have sucessfully installed the OS on bare metal.



## Experiment 2

Aim: To study and implement about the various basic linux commands. Procedure:

1. Linux based distributions come with many in built commands for communication directly with your computer. Bash was one of the first things Linus Torvalds, the creator of linux ported to linux when creating it. The purpose of this experiment is to get to know how to communicate with your computer
2. To Start, Open the terminal. (The terminal emulator depends on your choice of Desktop Environment)
3. Type out the following commands:

- (a) ls : list subdirectories

The screenshot shows a terminal window with a dark background. On the left, there is a large, stylized ASCII art representation of a tree or plant. On the right, the terminal output is displayed. It starts with the command 'ls' which lists several subdirectories: 'AirpodsBluetoothAddress.txt', 'Desktop', 'Developer', 'Documents', 'Games', 'help.txt', 'Downloads', 'go', 'Music', 'Pictures', 'Study', 'dvorak.sh', 'HCL', 'octave-workspace', 'Public', and 'Templates'. Below the list, there is a small decorative footer consisting of colored dots (red, green, yellow, blue, purple, white).

```
[gurmukh@archlinux ~]$ ls
AirpodsBluetoothAddress.txt  Documents  Games  help.txt      packettracer  spirograph  Videos
Desktop                      Downloads  go     Music       Pictures    Study
Developer                    dvorak.sh HCL   octave-workspace Public    Templates
[gurmukh@archlinux ~]$ 
```

- (b) mkdir : make directory jname;

```
[gurmukh@archlinux Desktop]$ ls
[gurmukh@archlinux Desktop]$ mkdir new_dir
[gurmukh@archlinux Desktop]$ ls
new_dir
[gurmukh@archlinux Desktop]$ ]
```

(c) rmdir : remove directory [name]

```
[gurmukh@archlinux Desktop]$ ls
new_dir
[gurmukh@archlinux Desktop]$ rmdir new_dir/
[gurmukh@archlinux Desktop]$ ls
[gurmukh@archlinux Desktop]$ ]
```

(d) cal : show the calender of the current month or the argument provided

```

Usage:
  cal [options] [[day] month] year
  cal [options] <timestamp|monthname>

Display a calendar, or some part of it.
Without any arguments, display the current month.

Options:
  -1, --one           show only a single month (default)
  -3, --three         show three months spanning the date
  -n, --months <num> show num months starting with date's month
  -S, --span          span the date when displaying multiple months
  -s, --sunday        Sunday as first day of week
  -m, --monday        Monday as first day of week
  -j, --julian        use day-of-year for all calendars
  --reform <val>     Gregorian reform date (1752|gregorian|iso|julian)
  --iso              alias for --reform=iso
  -y, --year          show the whole year
  -Y, --twelve         show the next twelve months
  -w, --week[=<num>]  show US or ISO-8601 week numbers
  -v, --vertical      show day vertically instead of line
  -c, --columns <width> amount of columns to use
  --color[=<when>]   colorize messages (auto, always or never)
                     colors are enabled by default

  -h, --help          display this help
  -V, --version       display version

For more details see cal(1).
[lgurmukh@archlinux Desktop]$ cal
September 2024
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

```

(e) date : give the date and time snapshot of a particular moment or the current moment

```

Usage: date [OPTION]... [-FORMAT]
  or: date [-u|-utc|-universal] [MMDDhhmm[[CC]YY][.ss]]
Display date and time in the given FORMAT.
With -, or with MMDDhhmm[[CC]YY][.ss], set the date and time.

Mandatory arguments to long options are mandatory for short options too.
-d, --date=STRING      display time described by STRING, not 'now'
--date=              assume now
--date=+N             warn about questionable usage to stderr
-f, --file=DATEFILE   like -date, once for each line of DATEFILE
-i[PMT], --iso-8601[=PMT] output date/time in ISO 8601 format
  PMT: date for date only (the default),
        'hours', 'minutes', 'seconds' or 'ns'
        for date and time, microseconds precision.
        Example: 2006-08-18T02:34:56.000000
--resolution          output the available resolution of timestamps
                      Example: 0.000000000
-R, --rfc=mail          output date/time in RFC 5322 format
  Example: Mon, 14 Aug 2006 02:34:56 -0600
--rfc=3339[PMT]         output date/time in RFC 3339 format
  PMT: date for date only (the default),
        for date and time to the indicated precision.
  Example: 2006-08-18 02:34:56-06:00
--reference=FILE       display current time or time of FILE
--set=STRING            set time described by STRING
-u, --utc, --universal  print or set Coordinated Universal Time (UTC)
--help                 display this help and exit
--version              output version information and exit

All options that specify the date to display are mutually exclusive.
I.e.: --date, --file, --reference, --resolution.

FORMAT controls the output. Interpreted sequences are:

%% a literal %
%a locale's abbreviated weekday name (e.g., Sun)
%b locale's full month name (e.g., January)
%B locale's abbreviated month name (e.g., Jan)
%c locale's full date and time (e.g., Thu Mar 3 23:00:28 2005)
%e day of month (0..31)
%F date; same as %Y-%m-%d
%g year, space padded (0..99); same as %_G
%G full date; like %Y-%m-%d
%h last two digits of year of ISO week number (see %W)
%j year, space padded (0..99); normally useful only with %W
%k same as %h
%l hour (00..23)
%L hour (00..59)
%V week of year (001..365)
%u hour, space padded (0 ..23); same as %_H
%U hour, space padded (1 ..32); same as %_I
%w week of year (0..52)
%W minute (00..59)
%a a newline
%n a new line
%p locale's equivalent of either AM or PM; blank if not known
%P like %p, but lower case

```

(f) ps : give a snapshot of some or all the active processes on the system

```
[guru@archlinux ~]$ ps -e
 PID TTY          TIME CMD
 1 ?    00:00:09 systemd
 2 ?    00:00:00 systemd-journal
 3 ?    00:00:00 pool_workqueue_release
 4 ?    00:00:00 kworker/0:rcu_g
 5 ?    00:00:00 kworker/0:rcu_g
 6 ?    00:00:00 kworker/0:rcu_g
 7 ?    00:00:00 kworker/0:rcu_g
 8 ?    00:00:00 kworker/0:rcu_events_highpri
 9 ?    00:00:00 kworker/0:rcu_mu
10 ?   00:00:00 rcu_tasks_kt_rude_lthread
11 ?   00:00:00 rcu_tasks_kt_trace_lthread
12 ?   00:00:00 rcu_exp_pax_sp_lthread
13 ?   00:00:00 rcu_exp_pax_sp_lthread
14 ?   00:00:00 rCU_taskx_trace_lthread
15 ?   00:00:00 rcu_taskx_trace_lthread
16 ?   00:00:00 ksoftirqd/0
17 ?   00:00:00 ksoftirqd/1
18 ?   00:00:00 rcu_sched
19 ?   00:00:00 rcub/1
20 ?   00:00:00 rcu_exp_pax_sp_lthread_wether/1
21 ?   00:00:00 rCU_taskx_pax_sp_lthread_wether/1
22 ?   00:00:00 migration/0
23 ?   00:00:00 idlet_inject/0
24 ?   00:00:00 cpuhp/1
25 ?   00:00:00 idlet_inject/1
26 ?   00:00:00 cpuhp/2
27 ?   00:00:00 ksoftirqd/2
28 ?   00:00:00 ksoftirqd/3
29 ?   00:00:00 ksoftirqd/4
30 ?   00:00:00 ksoftirqd/5
31 ?   00:00:00 idlet_inject/3
32 ?   00:00:00 migration/2
33 ?   00:00:00 ksoftirqd/6
34 ?   00:00:00 ksoftirqd/7
35 ?   00:00:00 ksoftirqd/8
36 ?   00:00:00 cpuhp/6
37 ?   00:00:00 idlet_inject/6
38 ?   00:00:00 cpuhp/7
39 ?   00:00:00 ksoftirqd/6
40 ?   00:00:00 ksoftirqd/7
41 ?   00:00:00 ksoftirqd/8
42 ?   00:00:00 ksoftirqd/9
43 ?   00:00:00 idlet_inject/8
44 ?   00:00:00 migration/7
45 ?   00:00:00 ksoftirqd/10
46 ?   00:00:00 ksoftirqd/11
47 ?   00:00:00 ksoftirqd/12
48 ?   00:00:00 cpuhp/9
49 ?   00:00:00 idlet_inject/9
50 ?   00:00:00 ksoftirqd/10
51 ?   00:00:00 ksoftirqd/11
52 ?   00:00:00 ksoftirqd/12
53 ?   00:00:00 ksoftirqd/13
54 ?   00:00:00 ksoftirqd/14
55 ?   00:00:00 rCU_taskx_pax_sp_lthread_worker/2
56 ?   00:00:00 cpuhp/10
57 ?   00:00:00 idlet_inject/10
58 ?   00:00:00 ksoftirqd/10
59 ?   00:00:00 migration/10
60 ?   00:00:00 ksoftirqd/10
61 ?   00:00:00 ksoftirqd/11
62 ?   00:00:00 cpuhp/11
63 ?   00:00:00 idlet_inject/11
64 ?   00:00:00 migration/11
65 ?   00:00:00 ksoftirqd/11
66 ?   00:00:00 ksoftirqd/11
67 ?   00:00:00 ksoftirqd/11
68 ?   00:00:00 cpuhp/12

[guru@archlinux ~]$ ps -e -1
 PID TTY          TIME CMD
 98281 pts/2    00:00:00 bash
 99584 pts/2    00:00:00 ps
[guru@guru@archlinux ~]$
```

(g) top : give a real time intel of all the processes running on the system.

top - 08:51:39 up 10:14, 2 users, load average: 1.20, 1.10, 1.10													
Tasks: 350 total, 1 running, 349 sleeping, 0 stopped, 0 zombie													
%Cpu(s): 2.2 us, 0.7 sy, 0.0 ni, 96.9 id, 0.1 wa, 0.1 hi, 0.1 si, 0.0 st													
MiB Mem : 15597.0 total, 5283.4 free, 7107.8 used, 4998.5 buff/cache													
MiB Swap: 12288.0 total, 12288.0 free, 0.0 used. 8489.2 avail Mem													
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND		
1452	gurmukh	20	0	4210216	350312	66688	S	18.0	2.2	24:39.22	ags		
96705	gurmukh	20	0	51.5g	487292	128836	S	7.7	3.1	1:24.05	spotify		
1310	gurmukh	20	0	1171672	101152	66664	S	6.0	0.6	23:56.48	Hyprland		
96558	gurmukh	20	0	4619198	285828	192356	S	6.0	1.8	0:31.27	spotify		
12938	gurmukh	20	0	277948	109904	11588	S	5.3	0.7	4:37.61	nvim		
1274	gurmukh	20	0	678612	520568	7348	S	1.7	3.3	1:53.44	pipewire-pulse		
96601	gurmukh	20	0	34.7g	175136	101816	S	1.7	1.1	0:18.90	spotify		
12910	gurmukh	20	0	2765932	769728	52760	S	1.0	4.8	6:36.23	neovide		
87566	root	0	-20	0	0	0	D	0.7	0.0	0:03.23	kworker/u76:1+i915_flip		
17	root	-2	0	0	0	0	I	0.3	0.0	0:09.12	rcu_preempt		
665	root	-51	0	0	0	0	S	0.3	0.0	0:11.95	irq/162-FTCS0038:00		
825	root	20	0	2444368	69448	22752	S	0.3	0.4	1:31.19	warp-svc		
1273	gurmukh	20	0	575692	30688	18452	S	0.3	0.2	0:28.48	wireplumber		
13536	gurmukh	20	0	21500	17612	6812	S	0.3	0.1	0:29.41	perl		
26089	root	20	0	0	0	0	I	0.3	0.0	0:06.40	kworker/u72:2-i915		
71654	root	0	-20	0	0	0	I	0.3	0.0	0:00.98	kworker/u75:2-hci0		
98192	gurmukh	20	0	605044	109208	72932	S	0.3	0.7	0:00.38	kitty		
100309	gurmukh	20	0	9684	7200	5040	R	0.3	0.0	0:00.02	top		
1	root	20	0	27932	18292	8868	S	0.0	0.1	0:09.51	systemd		
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	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_g		
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slub_		
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netsns		
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri		
11	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_pe		
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread		
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread		
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread		
16	root	20	0	0	0	0	S	0.0	0.0	0:00.23	ksoftirqd/0		
18	root	-2	0	0	0	0	S	0.0	0.0	0:00.00	rcub/1		
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_exp_par_gp_kthread_worker/1		
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_exp_gp_kthread_worker		
21	root	rt	0	0	0	0	S	0.0	0.0	0:00.91	migration/0		
22	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0		
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0		
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1		
25	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1		
26	root	rt	0	0	0	0	S	0.0	0.0	0:00.47	migration/1		
27	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/1		
29	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri		
30	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3		
31	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/3		
32	root	rt	0	0	0	0	S	0.0	0.0	0:00.45	migration/3		
33	root	20	0	0	0	0	S	0.0	0.0	0:00.05	ksoftirqd/3		
35	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/3:0H-events_highpri		
36	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/6		
37	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/6		
38	root	rt	0	0	0	0	S	0.0	0.0	0:00.34	migration/6		
39	root	20	0	0	0	0	S	0.0	0.0	0:00.04	ksoftirqd/6		
41	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/6:0H-events_highpri		
42	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/8		