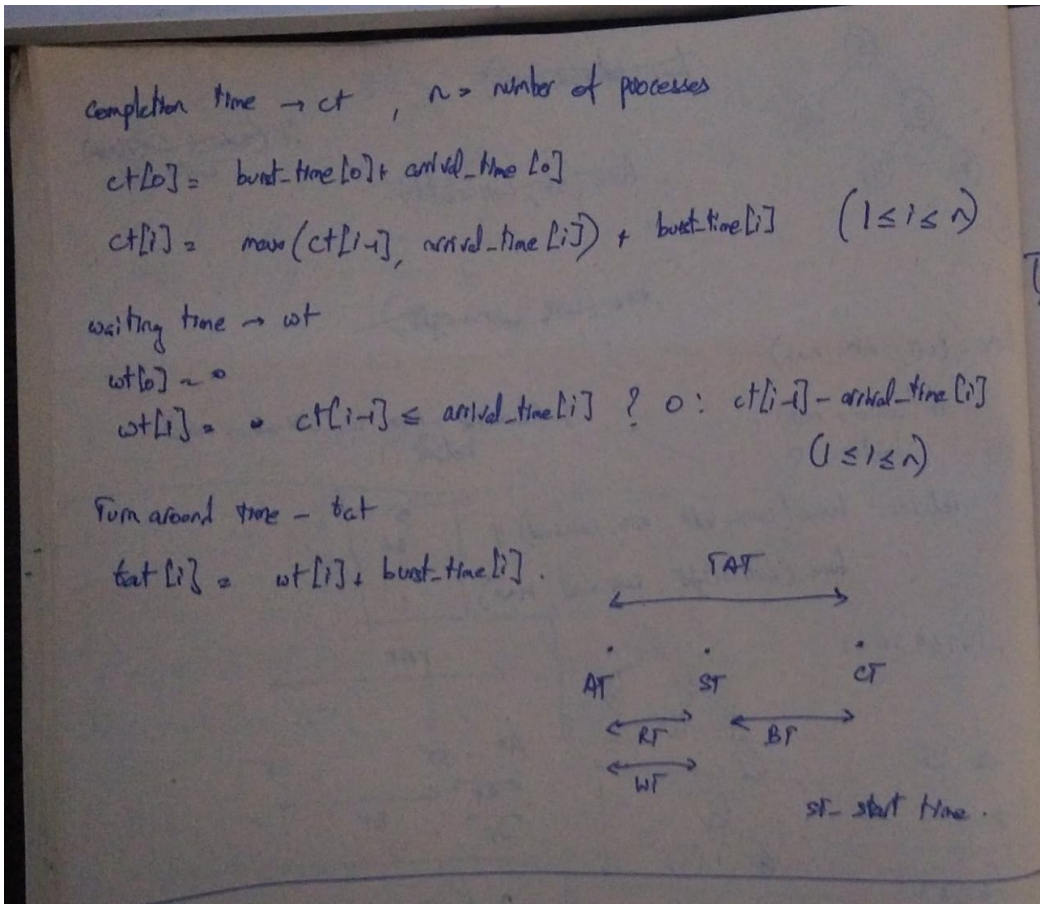
Lab Assignment - 2
(Operating Systems Practice)

Name: - M. Binesh
Roll : - CED19I036

1) FCFS: -

Formulae used in fcfs approach



I) All arrival times are 0:

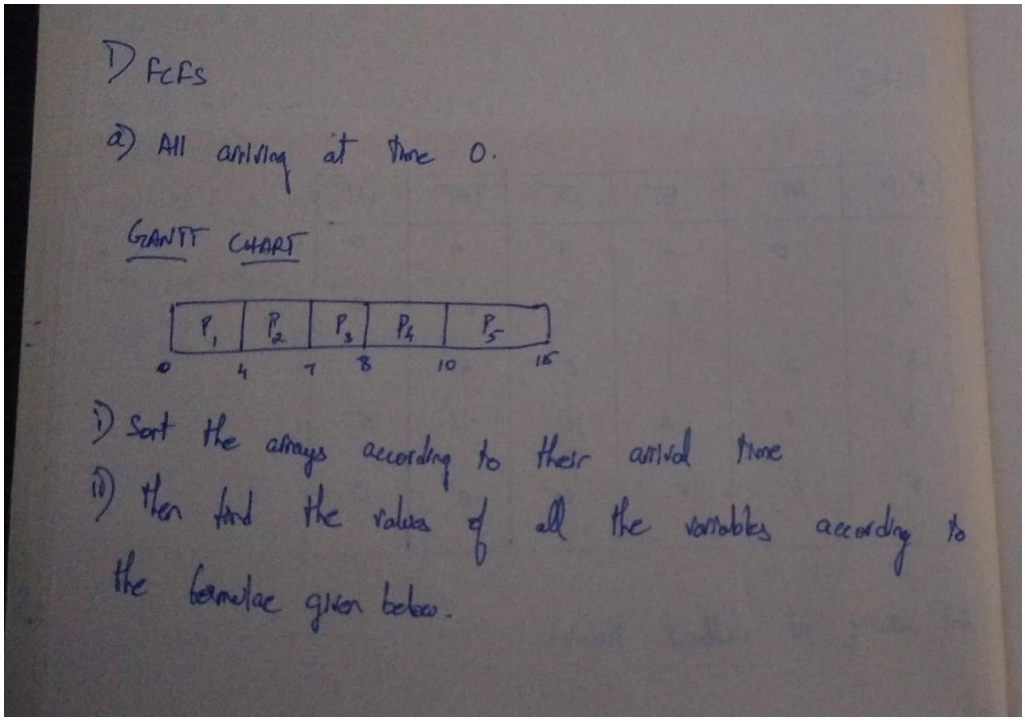
osp > lab2 > ≡ inp.txt

```
5
0 4
0 3
0 1
0 2
0 5


arrival_time  burst_time
--first example
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
kuries@Beast:~/reside/sem/osp/lab2$ g++ CED19I036_lab2_Q1.cpp && ./a.out < inp.txt
Enter the number of processes
Processes       Arrival Time    Burst time      Completion Time  Waiting time    Turn Around Time
1                   0               4               4               0               4
2                   0               3               7               4               7
3                   0               1               8               7               8
4                   0               2               10              8               10
5                   0               5               15              10              15
Average Turn Around Time: 8.8
Average Waiting Time: 5.8
```

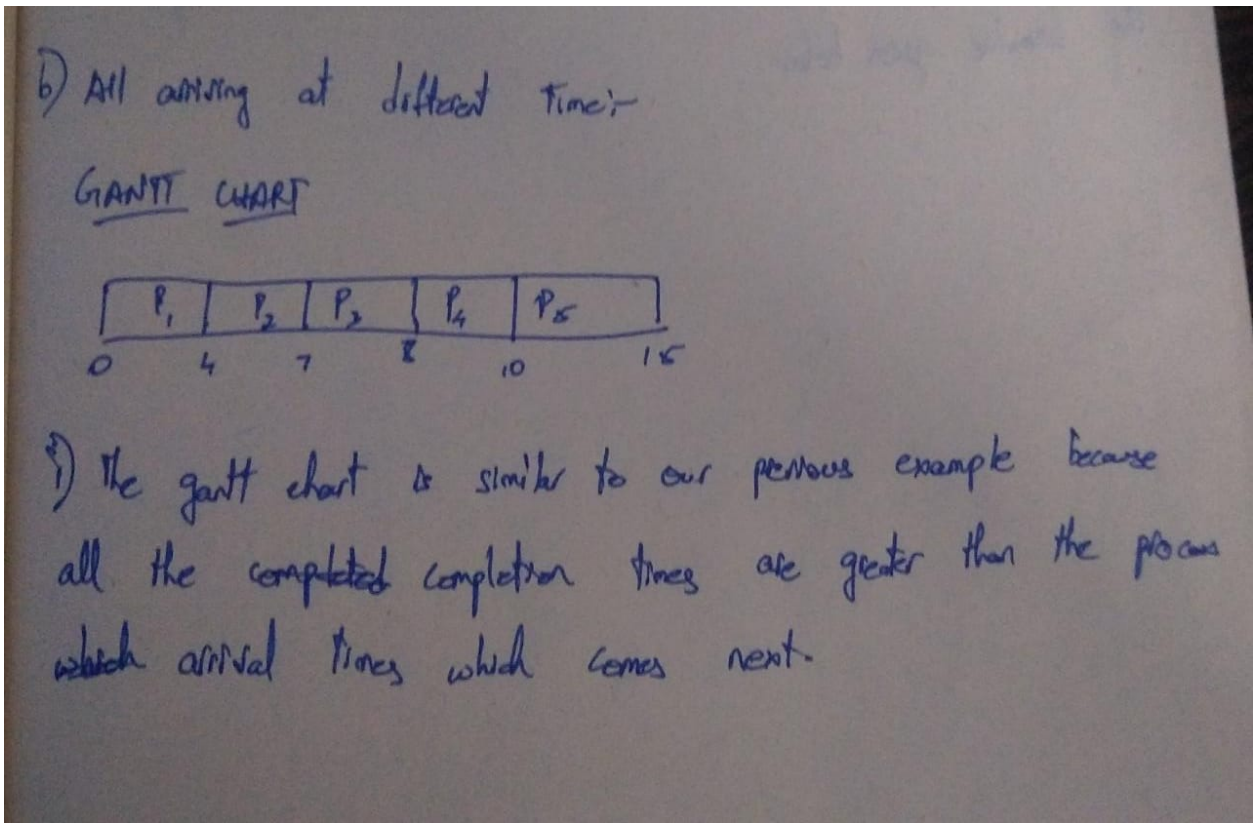


II) All arrival times are different:

```
osp > lab2 > inp.txt
1   5
2   0 4
3   1 3
4   2 1
5   3 2
6   4 5
7
8
9   arrival_time  burst_time
10  --second example
```

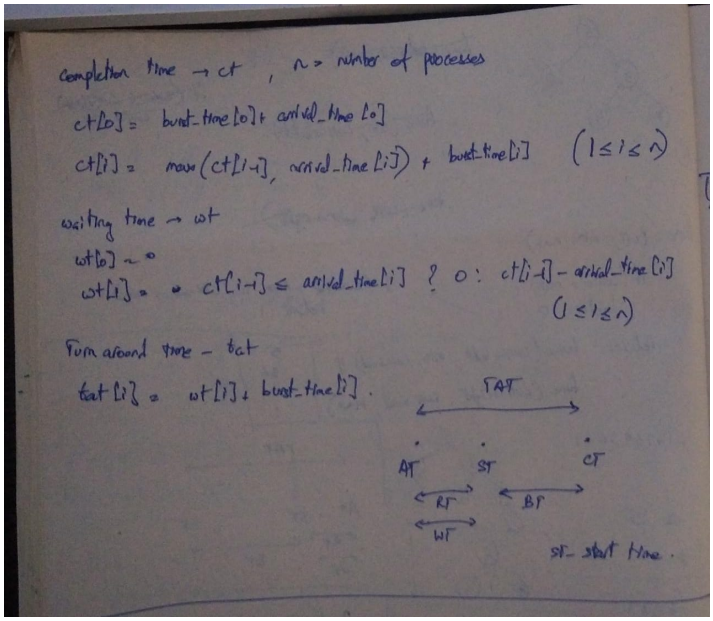
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
kuries@Beast:~/reside/sem/osp/lab2$ g++ CED19I036_lab2_Q1.cpp && ./a.out < inp.txt
Enter the number of processes
Processes       Arrival Time    Burst time      Completion Time  Waiting time   Turn Around Time
1                   0               4               4               0               4
2                   1               3               7               3               6
3                   2               1               8               5               6
4                   3               2               10              5               7
5                   4               5               15              6               11
Average Turn Around Time: 6.8
Average Waiting Time: 3.8
```



2) Shortest Job First: -

Some of the formulae are borrowed from the fcfs approach



I) All processes arrive at time 0

```
6
0 1
0 5
0 1
0 3
0 6
0 10


arrival_time  burst_time
--third example
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
kuries@Beast:~/reside/sem/osp/lab2$ g++ CED19I036_lab2_Q2.cpp && ./a.out < inp.txt
Enter the number of processes
Processes       Arrival Time    Burst time      Completion Time  Waiting time    Turn Around Time
1                   0               1               1               0               1
3                   0               1               2               1               2
4                   0               3               5               2               5
2                   0               5               10              5               10
5                   0               6               16              10              16
6                   0               10              26              16              26
Average Turn Around Time: 10
Average Waiting Time: 5.66667
```

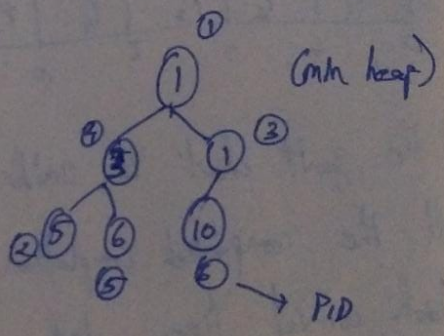
the formulae given below.

---

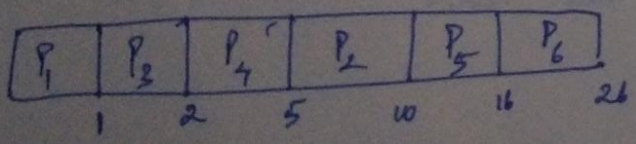
2) SJF

a) All arriving at time 0.

Priority queue created from input :-



GANTT CHART :-



II) All processes arrive at different times

osp > lab2 > inp.txt

```
8
2 1
1 5
4 1
2 3
0 6
120 1
105 10
100 1

arrival_time  burst_time
--fourth example
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

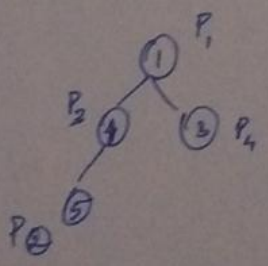
```
kuries@Beast:~/reside/sem/osp/lab2$ g++ CED19I036_lab2_Q2.cpp && ./a.out < inp.txt
Enter the number of processes
Processes       Arrival Time    Burst time      Completion Time  Waiting time   Turn Around Time
5                   0               6               6               0               6
1                   2               1               7               4               5
3                   4               1               8               3               4
4                   2               3               11              6               9
2                   1               5               16              10              15
8                   100             1               101             0               1
7                   105             10              115             0               10
6                   120             1               121             0               1
Average Turn Around Time: 6.375
Average Waiting Time: 2.875
```

2) All arrive at different times:-

1) After sorting the processes w.r.t their arrival time we get

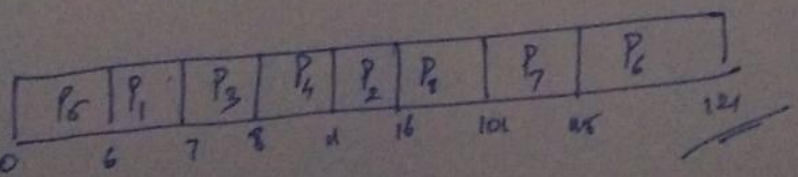
arrival time	burst time	Process Id
0	6	5
1	5	2
2	1	1
2	3	4
4	1	3
100	1	8
105	10	7
120	1	6

- First P_5 will be chosen and after that process is done CT would be 6.
- At time = 6, (P_2, P_1, P_4, P_3) will be made available.
- Priority queue for the above process is



- After all of them are processed, we will be left with P_8, P_7, P_6 at time = 16. Here CPU will be idle until the next process comes into ready state.

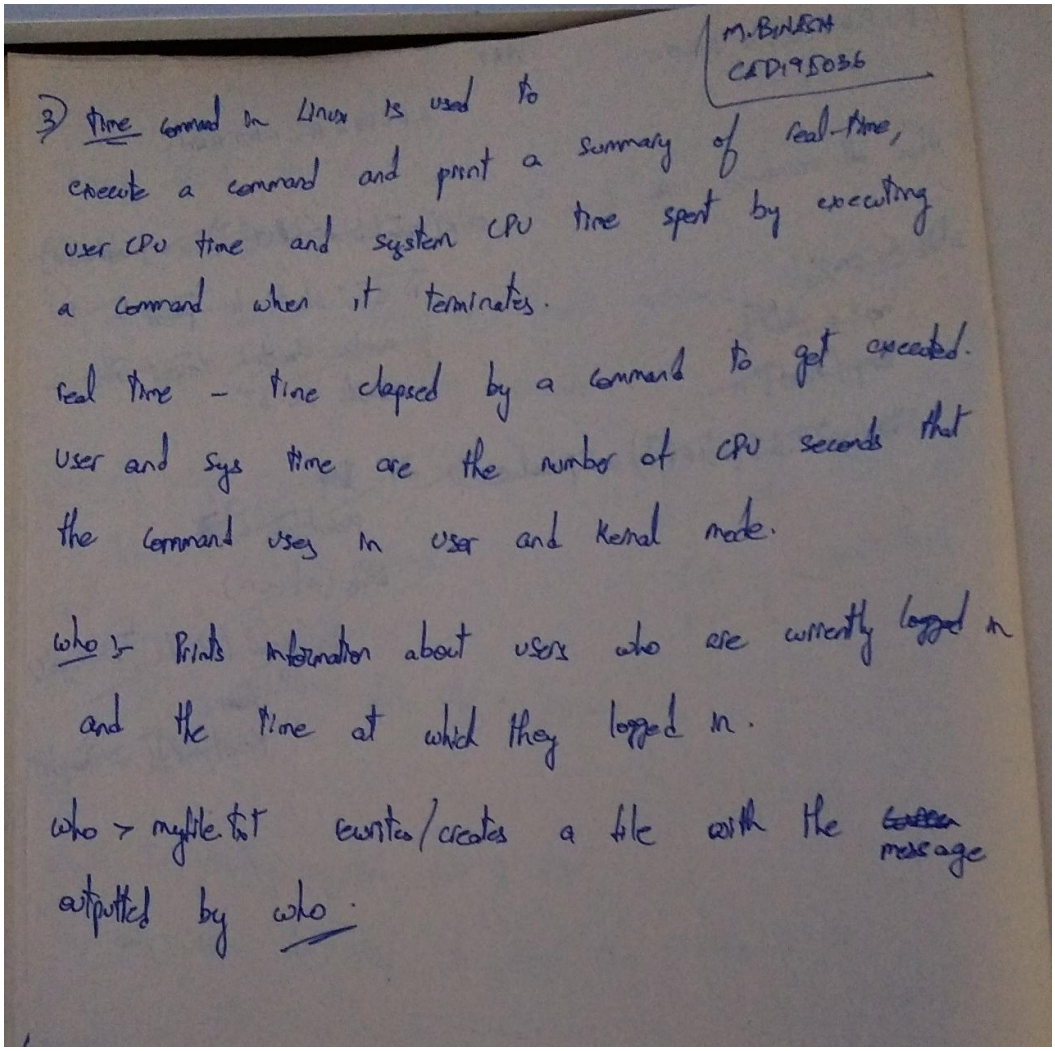
GANTT CHART :-



3) Display time in terminal and who in text file

```
kuries@Beast:~/reside/discard$ ls
kuries@Beast:~/reside/discard$ time who > myfile.txt

real    0m0.002s
user    0m0.002s
sys     0m0.000s
kuries@Beast:~/reside/discard$ more myfile.txt
kuries   :0           2021-08-15 18:31 (:0)
```



M. Binesh
CED19I5036

3) time command in Linux is used to execute a command and print a summary of real-time, user CPU time and system CPU time spent by executing a command when it terminates.

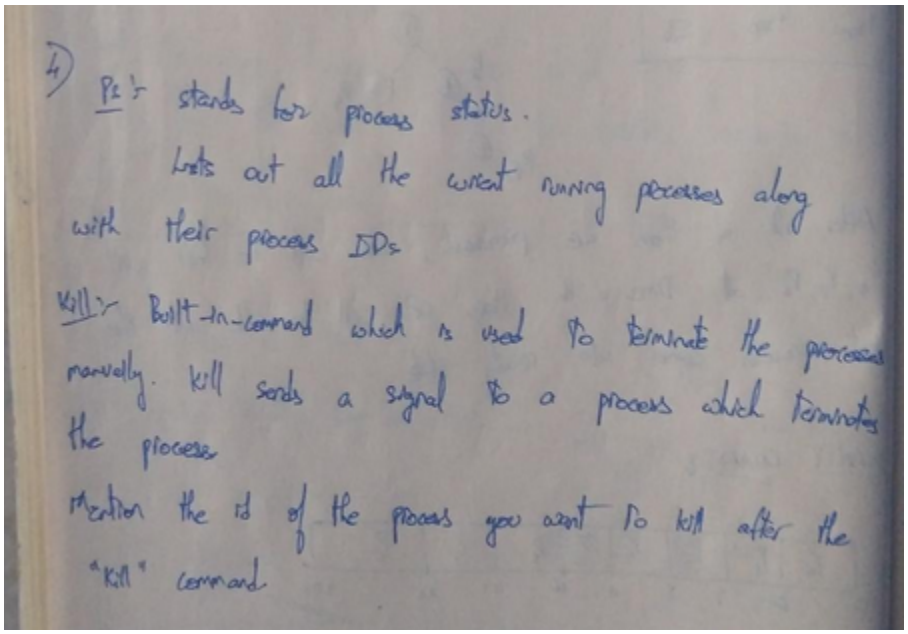
real time - time elapsed by a command to get executed.

user and sys time are the number of CPU seconds that the command uses in user and kernal mode.

who :- Prints information about users who are currently logged in and the time at which they logged in.

who > myfile.txt writes/creates a file with the ~~output~~ message outputted by who.

4) ps, kill, pkill, top, glances, pgrep, date, whoami



ps, kill

```
kuries@Beast:~/reside/program_files/c++$ ps
    PID TTY          TIME CMD
   6690 pts/0    00:00:00 bash
  60092 pts/0    00:00:00 gedit
  60126 pts/0    00:00:00 ps
kuries@Beast:~/reside/program_files/c++$ kill 60092
kuries@Beast:~/reside/program_files/c++$ ps
    PID TTY          TIME CMD
   6690 pts/0    00:00:00 bash
  60152 pts/0    00:00:00 ps
[1]+  Terminated              gedit B.cpp
```

K. Bindu
CSD19D036

Top:-

Top is used to show the Linux processes.
It provides dynamic real-time view of the running system.
Usually this shows the summary information of the system and the list of processes or threads which are currently managed by the Linux kernel.

Glances:-

System monitoring tool for the Linux machines, it is used to monitor system resources in web server mode or through web browser. It is an alternative to top & htop.

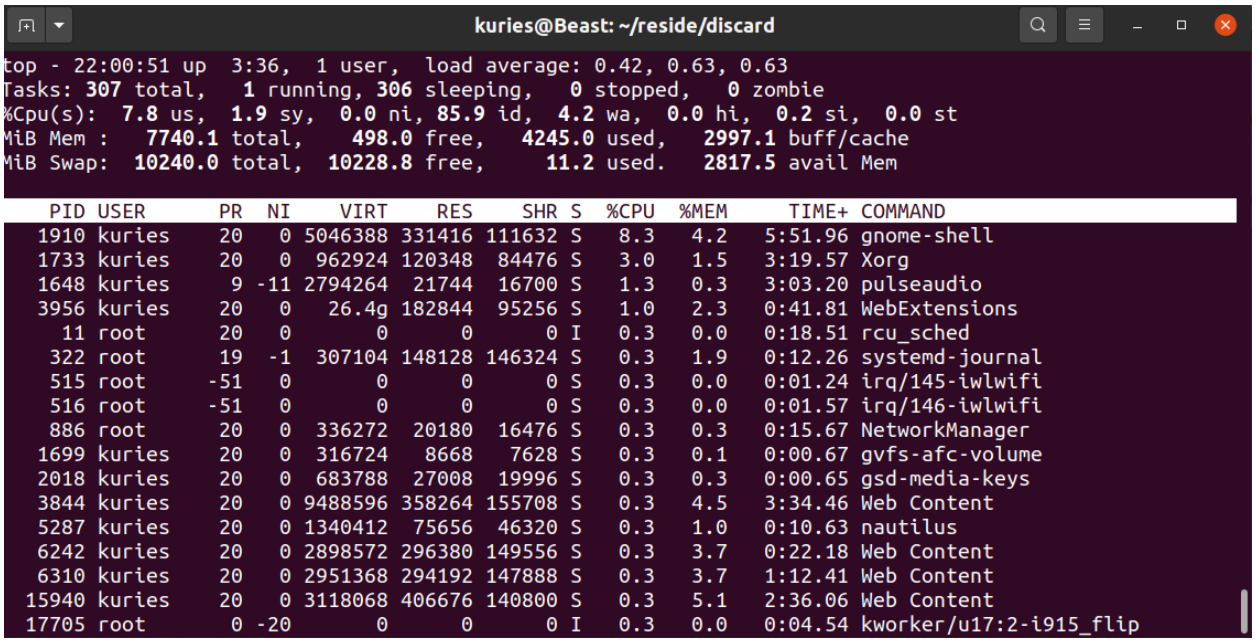
Features:-

1) Average CPU Load 2) Sleeping processes 3) Monitors 16+ metrics on the system.
4) Highly configurable 5) System info & Uptime
7) Supports exporting data to different services & databases.

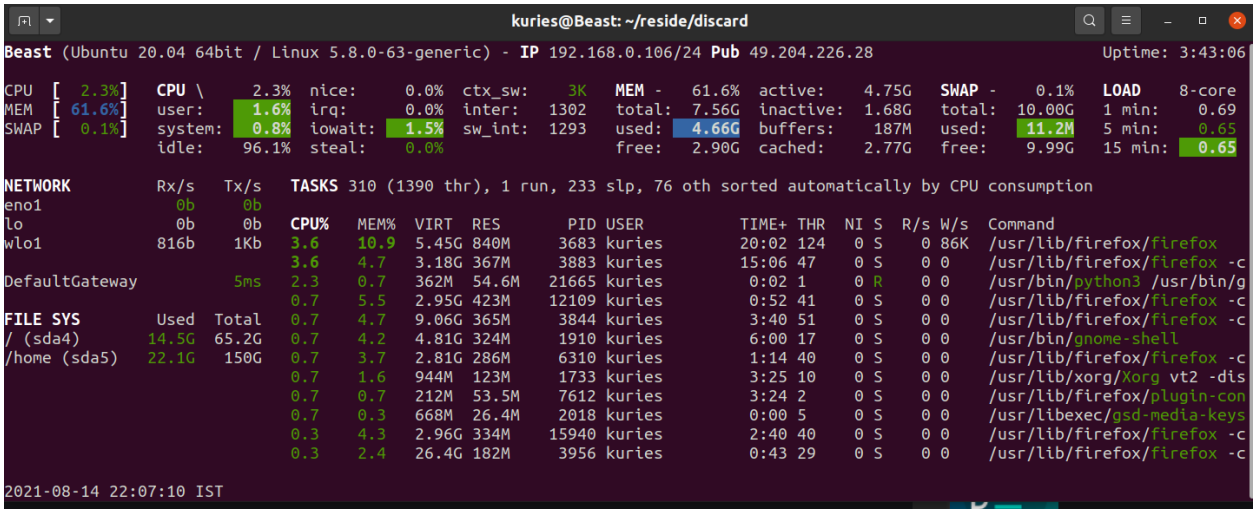
Pgrep:-

It finds the process IDs of a running program based on given criteria such as name, user running process, other attr.

top


kuries@Beast: ~/reside/discard
top - 22:00:51 up 3:36, 1 user, load average: 0.42, 0.63, 0.63
Tasks: 307 total, 1 running, 306 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.8 us, 1.9 sy, 0.0 ni, 85.9 id, 4.2 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 7740.1 total, 498.0 free, 4245.0 used, 2997.1 buff/cache
MiB Swap: 10240.0 total, 10228.8 free, 11.2 used. 2817.5 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1910 kuries 20 0 5046388 331416 111632 S 8.3 4.2 5:51.96 gnome-shell
1733 kuries 20 0 962924 120348 84476 S 3.0 1.5 3:19.57 Xorg
1648 kuries 9 -11 2794264 21744 16700 S 1.3 0.3 3:03.20 pulseaudio
3956 kuries 20 0 26.4g 182844 95256 S 1.0 2.3 0:41.81 WebExtensions
11 root 20 0 0 0 0 I 0.3 0.0 0:18.51 rcu_sched
322 root 19 -1 307104 148128 146324 S 0.3 1.9 0:12.26 systemd-journal
515 root -51 0 0 0 0 S 0.3 0.0 0:01.24 irq/145-iwlwifi
516 root -51 0 0 0 0 S 0.3 0.0 0:01.57 irq/146-iwlwifi
886 root 20 0 336272 20180 16476 S 0.3 0.3 0:15.67 NetworkManager
1699 kuries 20 0 316724 8668 7628 S 0.3 0.1 0:00.67 gvfs-afc-volume
2018 kuries 20 0 683788 27008 19996 S 0.3 0.3 0:00.65 gsd-media-keys
3844 kuries 20 0 9488596 358264 155708 S 0.3 4.5 3:34.46 Web Content
5287 kuries 20 0 1340412 75656 46320 S 0.3 1.0 0:10.63 nautilus
6242 kuries 20 0 2898572 296380 149556 S 0.3 3.7 0:22.18 Web Content
6310 kuries 20 0 2951368 294192 147888 S 0.3 3.7 1:12.41 Web Content
15940 kuries 20 0 3118068 406676 140800 S 0.3 5.1 2:36.06 Web Content
17705 root 0 -20 0 0 0 I 0.3 0.0 0:04.54 kworker/u17:2-i915_flip

glances


kuries@Beast: ~/reside/discard
Beast (Ubuntu 20.04 64bit / Linux 5.8.0-63-generic) - IP 192.168.0.106/24 Pub 49.204.226.28
Uptime: 3:43:06
CPU [2.3%] CPU \ 2.3% nice: 0.0% ctx_sw: 3K MEM - 61.6% active: 4.75G SWAP - 0.1% LOAD 8-core
MEM [61.6%] user: 1.6% irq: 0.0% inter: 1302 total: 7.56G inactive: 1.68G total: 10.00G 1 min: 0.69
SWAP [0.1%] system: 0.8% iowait: 1.5% sw_int: 1293 used: 4.66G buffers: 187M used: 11.2M 5 min: 0.65
idle: 96.1% steal: 0.0% free: 2.90G cached: 2.77G free: 9.99G 15 min: 0.65
NETWORK Rx/s Tx/s
eno1 0b 0b
lo 0b 0b
wlo1 816b 1Kb
DefaultGateway 5ms
FILE SYS Used Total
/ (sda4) 14.5G 65.2G
/home (sda5) 22.1G 150G
TASKS 310 (1390 thr), 1 run, 233 slp, 76 oth sorted automatically by CPU consumption
CPU% MEM% VIRT RES PID USER TIME+ THR NI S R/s W/s Command
3.6 10.9 5.45G 840M 3683 kuries 20:02 124 0 S 0 86K /usr/lib/firefox/firefox
3.6 4.7 3.18G 367M 3883 kuries 15:06 47 0 S 0 0 /usr/lib/firefox/firefox -c
2.3 0.7 362M 54.6M 21665 kuries 0:02 1 0 R 0 0 /usr/bin/python3 /usr/bin/g
0.7 5.5 2.95G 423M 12109 kuries 0:52 41 0 S 0 0 /usr/lib/firefox/firefox -c
0.7 4.7 9.06G 365M 3844 kuries 3:40 51 0 S 0 0 /usr/lib/firefox/firefox -c
0.7 4.2 4.81G 324M 1910 kuries 6:00 17 0 S 0 0 /usr/bin/gnome-shell
0.7 3.7 2.81G 286M 6310 kuries 1:14 40 0 S 0 0 /usr/lib/firefox/firefox -c
0.7 1.6 944M 123M 1733 kuries 3:25 10 0 S 0 0 /usr/lib/xorg/Xorg vt2 -dis
0.7 0.7 212M 53.5M 7612 kuries 3:24 2 0 S 0 0 /usr/lib/firefox/plugin-con
0.7 0.3 668M 26.4M 2018 kuries 0:00 5 0 S 0 0 /usr/libexec/gsd-media-keys
0.3 4.3 2.96G 334M 15940 kuries 2:40 40 0 S 0 0 /usr/lib/firefox/firefox -c
0.3 2.4 26.4G 182M 3956 kuries 0:43 29 0 S 0 0 /usr/lib/firefox/firefox -c
2021-08-14 22:07:10 IST

pgrep


kuries@Beast: ~/reside/discard
kuries@Beast:~/reside/discard$ pgrep -u kuries
1642
1643
1648
1650
1653
1669
1674
1681
1687
1691
1695
1699
1704
1708
1716
1729
1733
1761
1829
1850
1855
1870
1890

M. BINESH
CBDA5036

PKILL :-

It sends signals to the processes of a running program based on given criteria.

It is a wrapper around the pgrep command that only prints a list of matching processes.

Commonly used signals :-

1 (HUP) : reload process
9 (KILL) : kill process
15 (TERM) : To gracefully stop a process

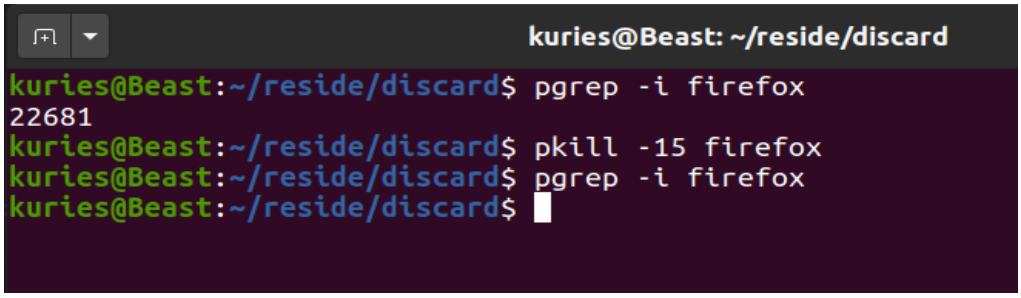
Date :-

Used to display system date & time.
It can also be used to change the date & time of System

whoami :-

It displays the username of the current user when this command is invoked.

pkill



date, whoami