**Process Status**

**To display the currently running process :**

```
hepzi@ubuntu:~$ ps
   PID   TTY       TIME     CMD
  2351  pts/0     00:00:00    bash
  4223  pts/0     00:00:00    ps
```

The output consists of four columns :
**`PID`** - Process ID
**`TTY`** - Terminal associated with process
**`TIME`** - Time in minutes and seconds (CPU utilization time for process)
**`CMD`** - Executable command

**To view all running process except session leaders(where the process id as same as the session id and skip any processes which do not have a controlling terminal) :**

```
hepzi@ubuntu:~$ ps -a
   PID TTY       TIME       CMD
  1630 tty2     00:00:00  gnome-session-b
  4224 pts/0    00:00:00  ps
```

**To view all process associated with the terminal :**

```
hepzi@ubuntu:~$ ps -T
   PID    SPID TTY       TIME   CMD
  2351    2351 pts/0    00:00:00    bash
  4230    4230 pts/0    00:00:00    ps
```

**To view process without TTY : (S means interruptible sleep waiting for some event)**

```
hepzi@ubuntu:~$ ps -x
   PID TTY      STAT   TIME COMMAND
  1585 ?        Ss     0:03 /lib/systemd/systemd --user
  1586 ?        S      0:00 (sd-pam)
  1592 ?        S<sl   0:00 /usr/bin/pipewire
  1593 ?        Ssl    0:00 /usr/bin/pipewire-media-session

  5134 pts/0    R+     0:00 ps -x
```

**To only list running process : (STAT  R actively running, + foreground process)**

```
hepzi@ubuntu:~$ ps -r
   PID   TTY      STAT   TIME  COMMAND
  3492   pts/0       R+     0:00        ps -r
```

**To display most amount of information a user to understand current  state of the system running process :**
**a-** running process from all users
**u-** shows user
**x-** process those have not been executed

```
hepzi@ubuntu:~$ ps aux
USER     PID  %CPU  %MEM   VSZ   RSS  TTY   STAT  START   TIME    COMMAND
root       1        0.2      0.3    166664  8008    ?       Ss      19:17      0:04        /sbin/init sp
root       2        0.0      0.0         0       0       ?       S       19:17      0:00        [kthreadd]
root       3  0.0  0.0      0     0 ?       I<   19:17    0:00          0.00      0.00           [rcu_gp]
root       4  0.0  0.0      0     0 ?       I<   19:17    0:00          0.00      0.00          [rcu_par_gp]
root       5  0.0  0.0      0     0 ?       I<   19:17   0:00          0.00      0.00        [slub_flushwq]

hepzi      3737  0.0  0.0  21324  1600 pts/0    R+   19:49       19:49         0:00             ps aux
```

- **USER**: Name of the user
- **%CPU**: CPU usage percentage
- **%MEM**: Memory usage percentage
- **VSZ**: Total virtual memory used by the process, in kb
- **RSS**: RAM occupied by the process
- **STAT**: Current process state
- **START**: Time the process was started.

**To view the process that use the most CPU power :**

```
hepzi@ubuntu:~$ ps aux --sort=-pcpu
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
hepzi       1753 20.4 11.3 4241904 230164 ?      Ssl  19:19  14:42 /usr/bin/gnom
hepzi       2340  7.1  9.5 1114028 192496 ?      Sl   19:20   5:02 /usr/lib/libr
hepzi       2493  4.5 14.8 3292212 301172 ?      Sl   19:24   3:05 /snap/firefox
hepzi       3394  2.7  7.2 2482792 146496 ?      Sl   19:25   1:50 /snap/firefox
mysql       1134  2.4  7.7 1790896 156408 ?      Ssl  19:18   1:45 /usr/sbin/mys
systemd+     621  0.8  0.1  14824  2624 ?        Ss   19:18   0:36 /lib/systemd/
hepzi       1967  0.7  0.3 249748  7476 ?        Sl   19:19   0:32 /usr/bin/ibus
hepzi       4031  0.0  0.1  21660  3804 pts/0    R+   20:31   0:00 ps aux –sort
```

**To view memory usage for each running process :**

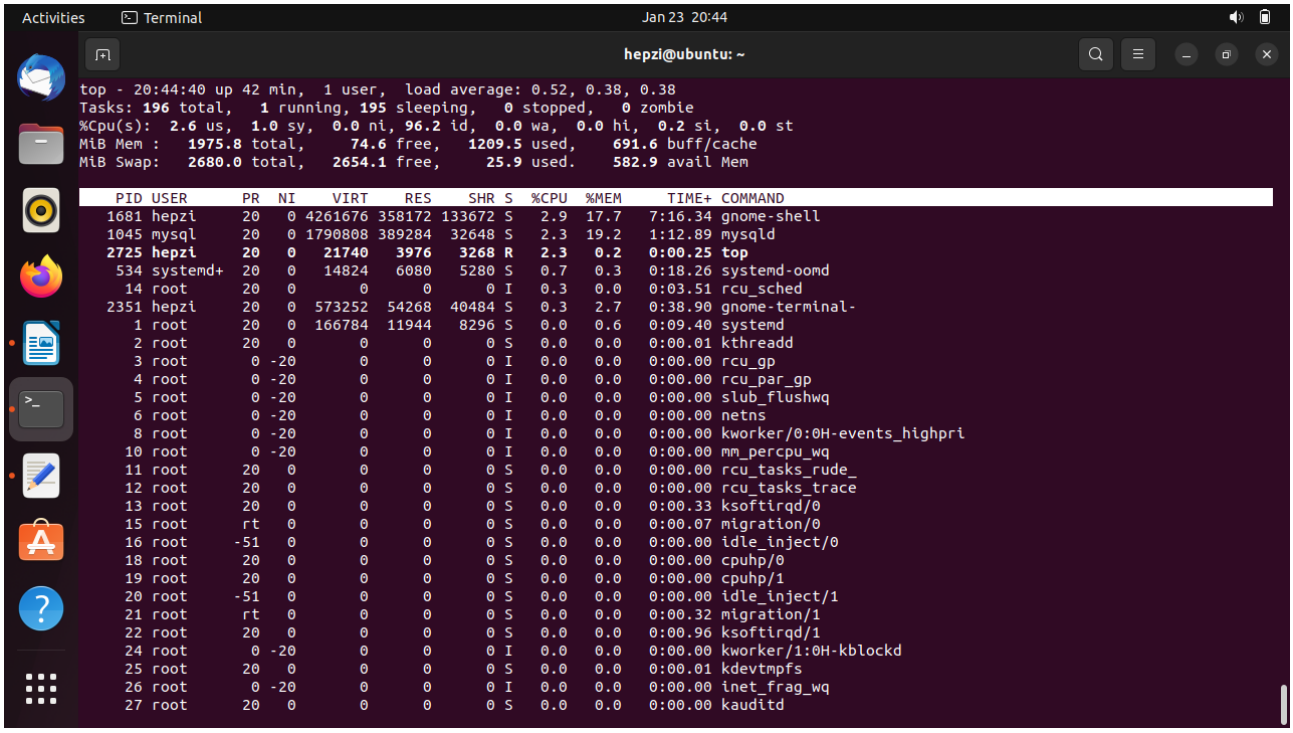
```
hepzi@ubuntu:~$ ps aux --sort=+pmem
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME  COMMAND
root         2  0.0  0.0      0     0 ?        S    19:17   0:00                [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   19:17   0:00                 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   19:17   0:00               [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   19:17   0:00               [slub_flushwq]
```

hepzi 2493 4.4 14.0 3290148 284616 ? Sl 19:24 3:09 /snap/firefox

## Checking system performance

**To display the realtime view of running process :**

hepzi@ubuntu:~$ top




hepzi@ubuntu:/$ sudo apt install htop

Reading package lists... Done

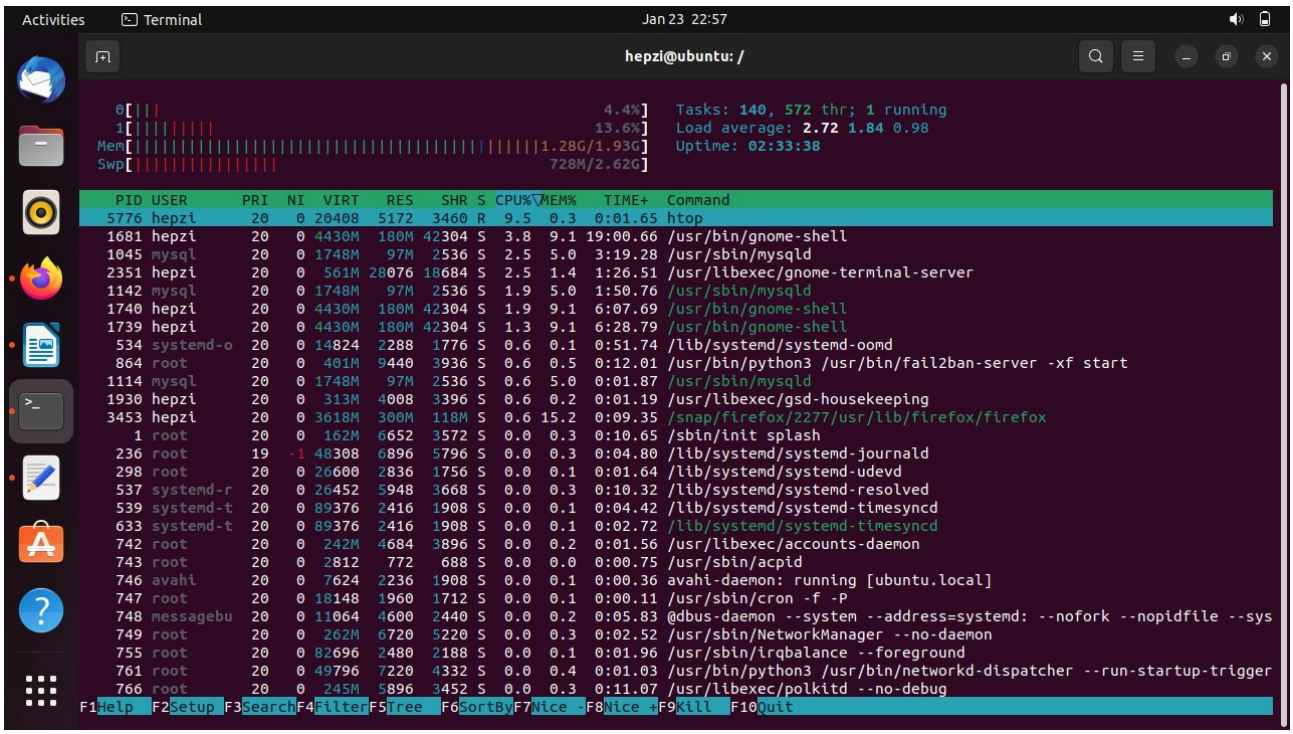
Building dependency tree... Done

Reading state information... Done

Processing triggers for man-db (2.10.2-1) ...

**Allows the user to interaction on process monitoring(similar to task manager) :**

hepzi@ubuntu:/$ htop




hepzi@ubuntu:/$ sudo apt install nmon

Reading package lists... Done

Building dependency tree... Done

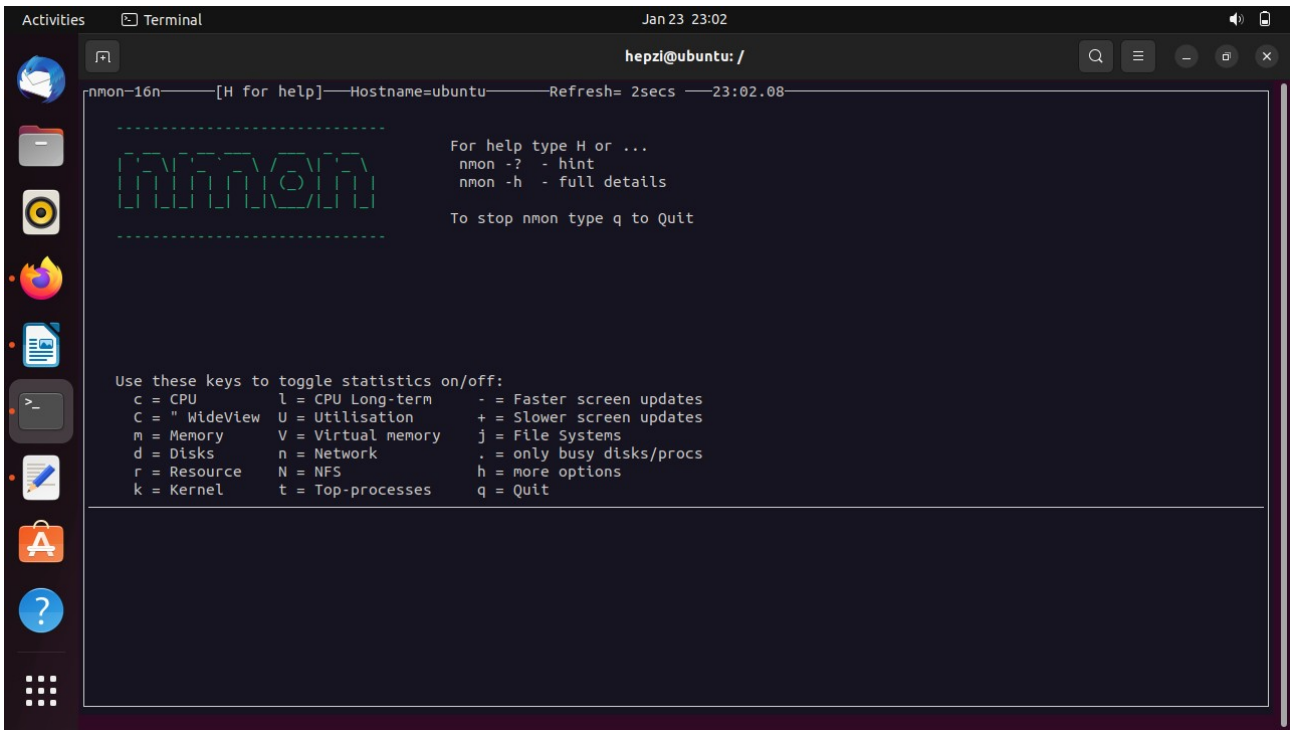
Reading state information... Done

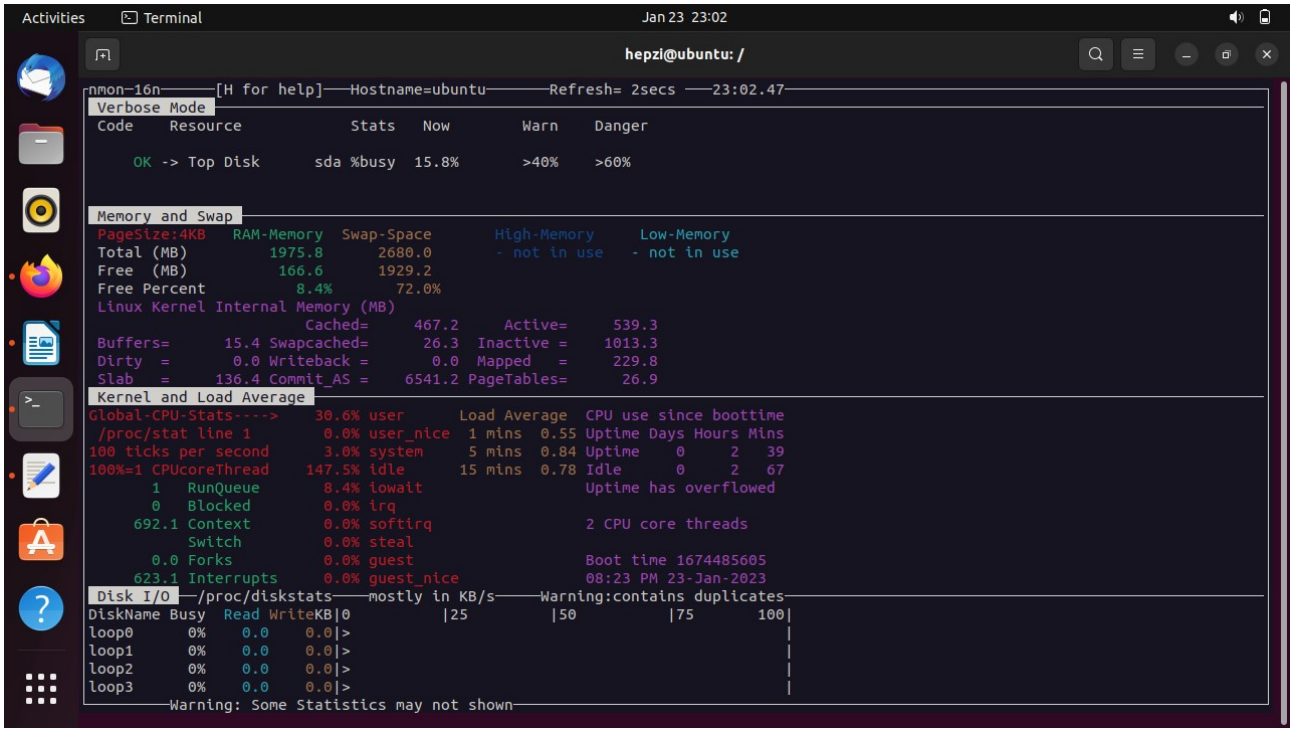
Processing triggers for man-db (2.10.2-1) ...

**System monitoring tool :**

hepzi@ubuntu:/$ nmon

**To monitor the performance of the system such as cpu, memory usage, disk space, network and so on.**




## Check the drive space usage

**To display usage in bytes :**

```
hepzi@ubuntu:~$ df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs             202324     1552    200772   1% /run
/dev/sda3       25106692 13233096  10572912  56% /
```

```
tmpfs          1011608       0   1011608   0% /dev/shm
tmpfs             5120       4      5116   1% /run/lock
/dev/sda2       524252    5364    518888   2% /boot/efi
tmpfs           202320    2416    199904   2% /run/user/1000
```

**To display usage in MB :**

```
hepzi@ubuntu:~$ df -m
Filesystem     1M-blocks  Used Available Use% Mounted on
tmpfs              198     2       197   1% /run
/dev/sda3        24519 12923     10326  56% /
tmpfs              988     0       988   0% /dev/shm
tmpfs                5     1         5   1% /run/lock
/dev/sda2          512     6       507   2% /boot/efi
tmpfs              198     3       196   2% /run/user/1000
```

**To display the file system type :**

```
hepzi@ubuntu:~$ df -T
Filesystem     Type  1K-blocks     Used Available Use% Mounted on
tmpfs          tmpfs    202324     1552    200772   1% /run
/dev/sda3      ext4   25106692 13233100  10572908  56% /
tmpfs          tmpfs   1011608        0   1011608   0% /dev/shm
tmpfs          tmpfs      5120        4      5116   1% /run/lock
/dev/sda2      vfat     524252     5364    518888   2% /boot/efi
tmpfs          tmpfs    202320     2416    199904   2% /run/user/1000
```

**To display the temporary file systems :**

```
hepzi@ubuntu:~$ df -t tmpfs
Filesystem     1K-blocks  Used Available Use% Mounted on
tmpfs             202324  1552    200772   1% /run
tmpfs            1011608     0   1011608   0% /dev/shm
tmpfs               5120     4      5116   1% /run/lock
tmpfs             202320  2416    199904   2% /run/user/1000
```

**To display filesystems but without the specified type :**

```
hepzi@ubuntu:~$ df -x tmpfs
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/sda3       25106692 13233104  10572904  56% /
/dev/sda2         524252     5364    518888   2% /boot/efi
```

**To displays information abot used and unused memory and swap usage :**

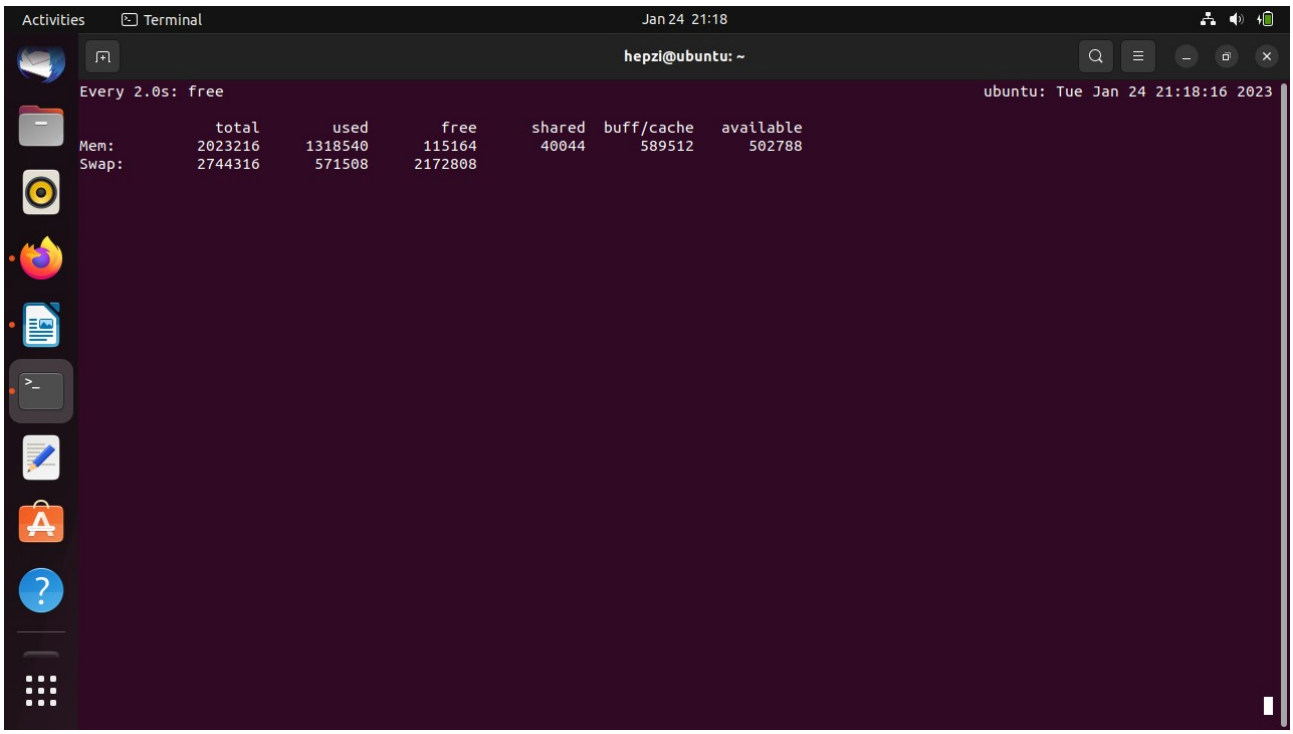
```
hepzi@ubuntu:~$ free
               total        used        free      shared  buff/cache   available
Mem:         2023216     1319944      115416       39288      587856      502204
Swap:        2744316      571508     2172808
```

**To view Human readable output :**

```
hepzi@ubuntu:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           1.9Gi       1.3Gi       112Mi        38Mi       574Mi       490Mi
Swap:          2.6Gi       558Mi       2.1Gi
```

**Watch will run until stopped by ctrl + c**

```
hepzi@ubuntu:~$ watch free
```




**To track the changes in output every 2secs :**

**To see what has changed (d- differences will highlight the changes) :**

hepzi@ubuntu:~$ watch -d free

**The output updated after every 5secs :**

hepzi@ubuntu:~$ watch -n 5 -d free

**Crontab(Scheduling tasks)**

**-is a command that we want to executed on a regular schedule by the OS at a specified time.**

**To view crontab status :**

hepzi@ubuntu:~$ sudo systemctl status cron

● cron.service - Regular background program processing daemon

Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: >

Active: active (running) since Tue 2023-01-24 19:18:19 IST; 2h 47min ago

Docs: man:cron(8)

Main PID: 746 (cron)

Tasks: 1 (limit: 2287)

Memory: 1.9M

```
      CPU: 755ms
   CGroup: /system.slice/cron.service
           └─746 /usr/sbin/cron -f -P
```

**To view the crontab file :**

```
hepzi@ubuntu:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin




# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root     test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root     test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root     test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

#

**To edit system crontabs:**

hepzi@ubuntu:~$ crontab -e
crontab: installing new crontab

**To view crontab file contents :**

hepzi@ubuntu:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
 * * * * * date >> date.txt
# m h  dom mon dow   command

```
hepzi@ubuntu:~$ cat date.txt
Tuesday 24 January 2023 11:04:01 PM IST
Tuesday 24 January 2023 11:05:01 PM IST

hepzi@ubuntu:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:

# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
 * * * 1 * echo 'Hello' >> /tmp/test.txt
# m h  dom mon dow   command
```

**To be executed at a specified time :**

```
hepzi@ubuntu:~$ cat /tmp/test.txt
Hello
```