CS 250 OPERATING SYSTEM

2021-22

Assignment: 01

Name: **Sudhir Sharma**

email: sudhirsharma@iitbhlai.ac.in

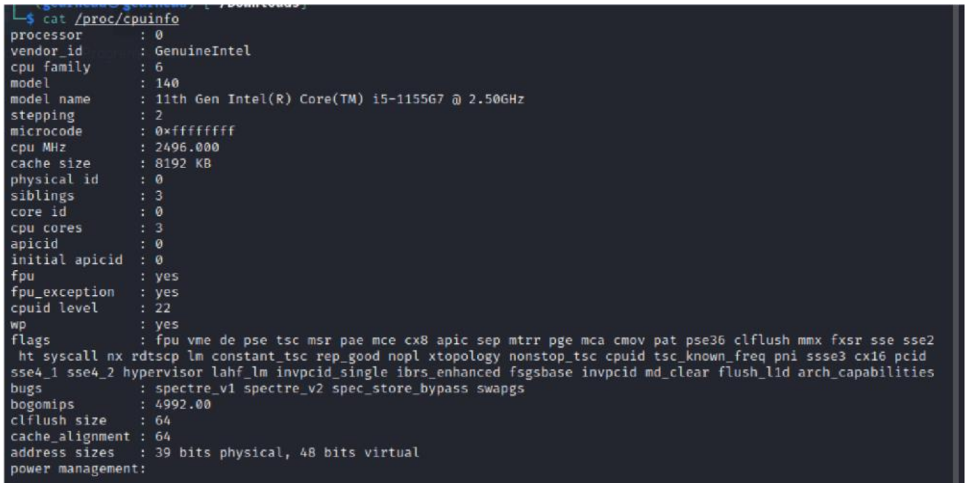
Roll No: 12041500

Report

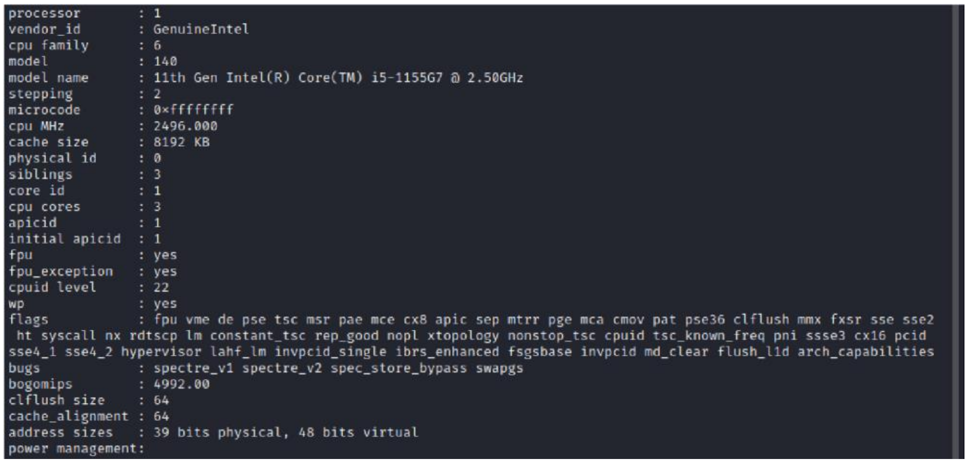
Part A: System Related Information [Total: 8 Points]

Proc file system (procfs) is virtual file system created on fly when system boots and is dissolved at time of system shut down.

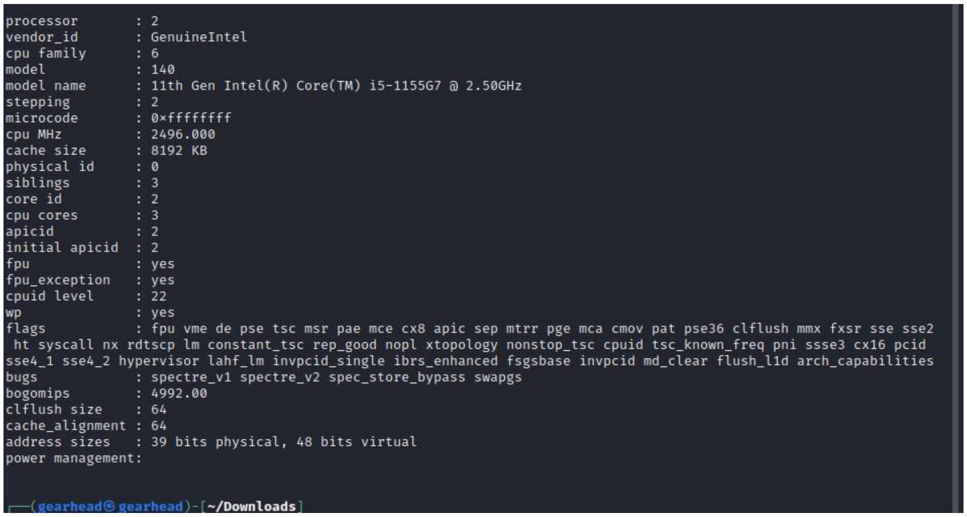
It contains useful information about the processes that are currently running, it is regarded as control and information center for kernel.



```
└─$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
model name      : 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz
stepping        : 2
microcode       : 0xffffffff
cpu MHz         : 2496.000
cache size      : 8192 KB
physical id     : 0
siblings        : 3
core id         : 0
cpu cores       : 3
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni ssse3 cx16 pcid
sse4_1 sse4_2 hypervisor lahf_lm invpcid_single ibrs_enhanced fsgsbase invpcid md_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs
bogomips        : 4992.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```



```
processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
model name      : 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz
stepping        : 2
microcode       : 0xffffffff
cpu MHz         : 2496.000
cache size      : 8192 KB
physical id     : 0
siblings        : 3
core id         : 1
cpu cores       : 3
apicid          : 1
initial apicid  : 1
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni ssse3 cx16 pcid
sse4_1 sse4_2 hypervisor lahf_lm invpcid_single ibrs_enhanced fsgsbase invpcid md_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs
bogomips        : 4992.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```



```
processor       : 2
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
model name      : 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz
stepping        : 2
microcode       : 0xffffffff
cpu MHz         : 2496.000
cache size      : 8192 KB
physical id     : 0
siblings        : 3
core id         : 2
cpu cores       : 3
apicid          : 2
initial apicid  : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni ssse3 cx16 pcid
sse4_1 sse4_2 hypervisor lahf_lm invpcid_single ibrs_enhanced fsgsbase invpcid md_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs
bogomips        : 4992.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:


┌──(gearhead㉿gearhead)-[~/Downloads]
```

1.

a) No of processors in my sys is 3 as it is VM

b) For each processor

physical address size:- 39 bits

Virtual address size:- 48 bits

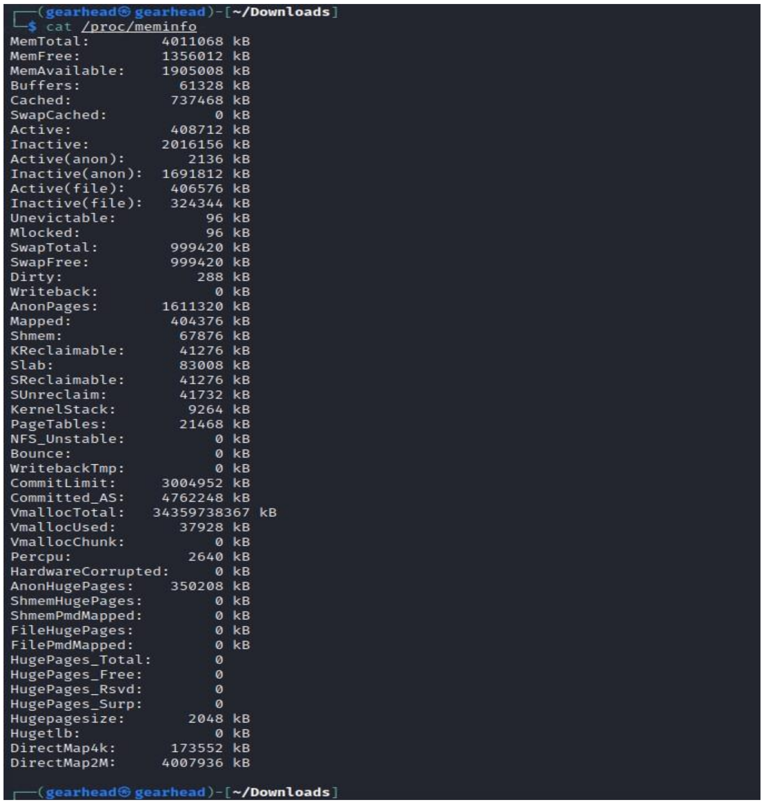
c) Frequency of each processor is :-2495.998 MHz

2.

a) physical memory 4011068 kB

b) RAM isn't being used 1356012 kB

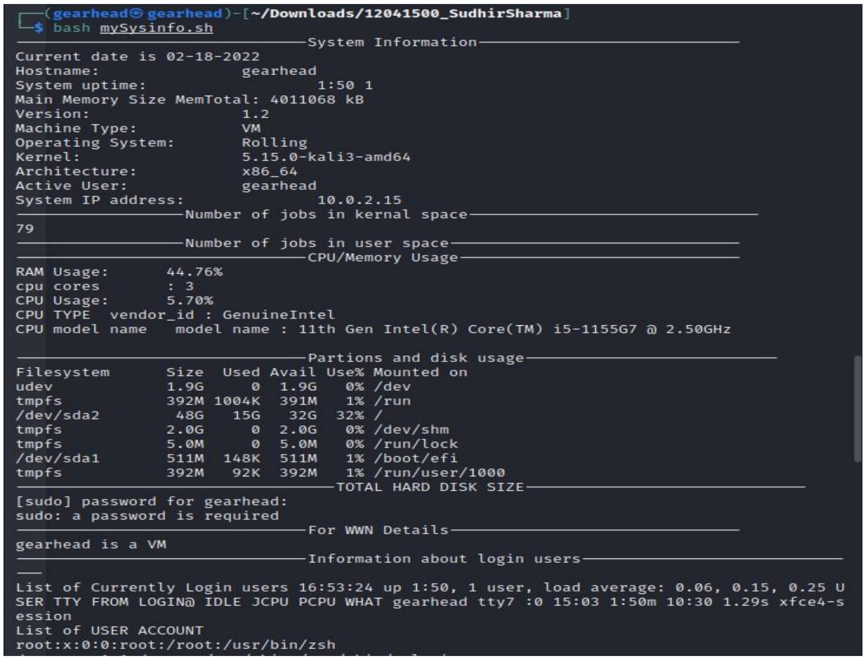
c) RAM is used by buffer 61328 kB



```
┌──(gearhead㉿gearhead)-[~/Downloads]
└─$ cat /proc/meminfo
MemTotal:        4011068 kB
MemFree:         1356012 kB
MemAvailable:    1905008 kB
Buffers:           61328 kB
Cached:           737468 kB
SwapCached:            0 kB
Active:           408712 kB
Inactive:        2016156 kB
Active(anon):       2136 kB
Inactive(anon):  1691812 kB
Active(file):     406576 kB
Inactive(file):   324344 kB
Unevictable:          96 kB
Mlocked:              96 kB
SwapTotal:        999420 kB
SwapFree:         999420 kB
Dirty:               288 kB
Writeback:             0 kB
AnonPages:       1611320 kB
Mapped:           404376 kB
Shmem:             67876 kB
KReclaimable:      41276 kB
Slab:              83008 kB
SReclaimable:      41276 kB
SUnreclaim:        41732 kB
KernelStack:        9264 kB
PageTables:        21468 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     3004952 kB
Committed_AS:    4762248 kB
VmallocTotal:   34359738367 kB
VmallocUsed:       37928 kB
VmallocChunk:          0 kB
Percpu:             2640 kB
HardwareCorrupted:     0 kB
AnonHugePages:    350208 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:      173552 kB
DirectMap2M:     4007936 kB

┌──(gearhead㉿gearhead)-[~/Downloads]
```

3

For running mySysinfo.sh file you have to enter your root password for showing the total hard Disk size.

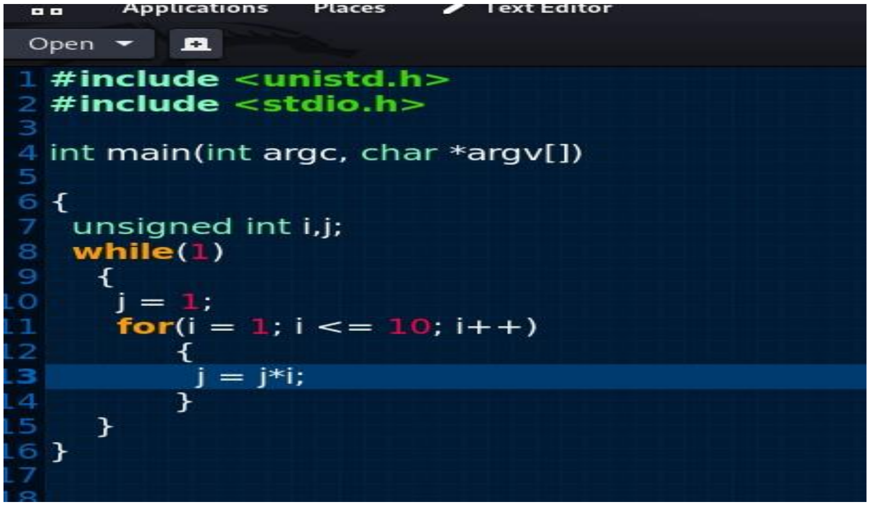


```
┌──(gearhead㉿gearhead)-[~/Downloads/12041500_SudhirSharma]
└─$ bash mySysinfo.sh
──────────────────────────────System Information───────────────────────────
Current date is 02-18-2022
Hostname:                gearhead
System uptime:                   1:50 1
Main Memory Size MemTotal: 4011068 kB
Version:                 1.2
Machine Type:            VM
Operating System:        Rolling
Kernel:                  5.15.0-kali3-amd64
Architecture:            x86_64
Active User:             gearhead
System IP address:               10.0.2.15
─────────────────Number of jobs in kernal space───────────────────────────────
79
─────────────────Number of jobs in user space─────────────────────────────
──────────────────────────────CPU/Memory Usage────────────────────────────
RAM Usage:      44.76%
cpu cores       : 3
CPU Usage:      5.70%
CPU TYPE  vendor_id : GenuineIntel
CPU model name   model name : 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz

──────────────────────────────Partions and disk usage──────────────────────────
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M 1004K  391M   1% /run
/dev/sda2        48G   15G   32G  32% /
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/sda1       511M  148K  511M   1% /boot/efi
tmpfs           392M   92K  392M   1% /run/user/1000
─────────────────────────────────TOTAL HARD DISK SIZE──────────────────────────────
[sudo] password for gearhead:
sudo: a password is required
──────────────────────────────For WWN Details──────────────────────────────
gearhead is a VM
──────────────────────────────Information about login users────────────────────────────
───
List of Currently Login users 16:53:24 up 1:50, 1 user, load average: 0.06, 0.15, 0.25 U
SER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT gearhead tty7 :0 15:03 1:50m 10:30 1.29s xfce4-s
ession
List of USER ACCOUNT
root:x:0:0:root:/root:/usr/bin/zsh
```

Part B Process Related Information

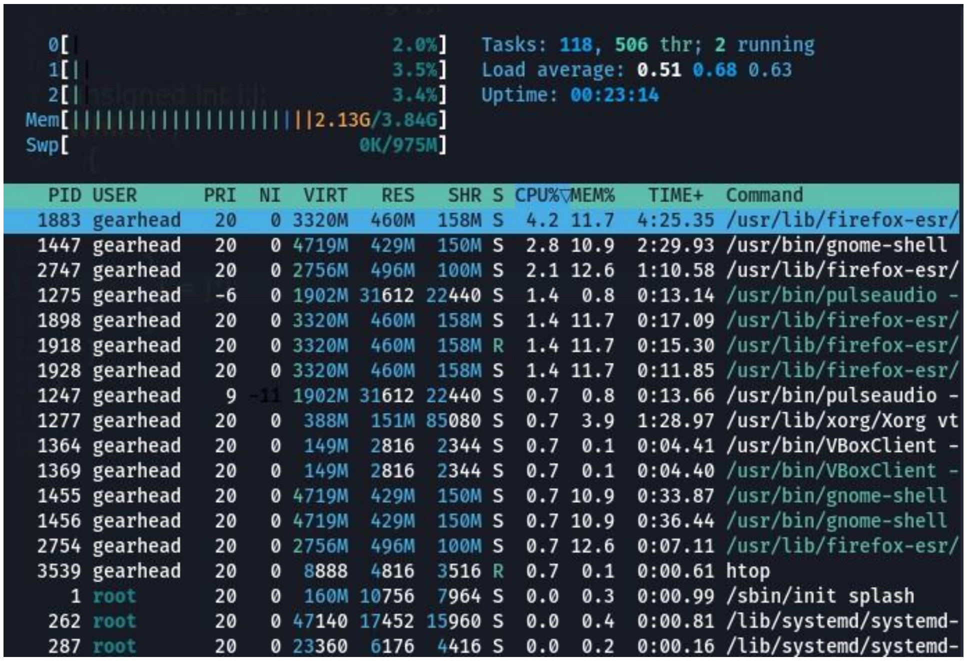
1.

Programname.c contain an infinite loop c program



```
#include <unistd.h>
#include <stdio.h>

int main(int argc, char *argv[])

{
 unsigned int i,j;
 while(1)
   {
    j = 1;
    for(i = 1; i <= 10; i++)
        {
          j = j*i;
        }
   }
}
```

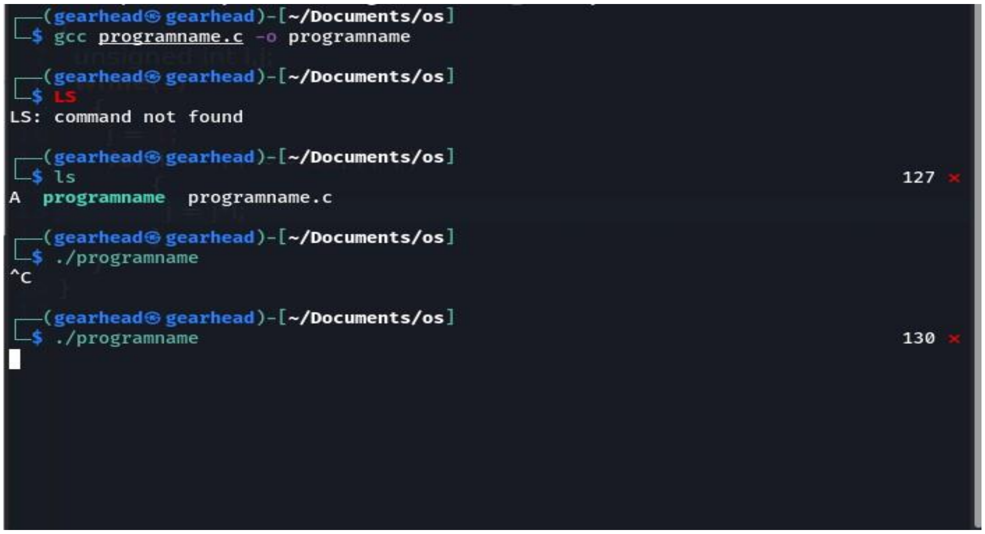
Before running the Programname.c file, the htop command shows the following output



```
    0[|                          2.0%]   Tasks: 118, 506 thr; 2 running
    1[||                         3.5%]   Load average: 0.51 0.68 0.63
    2[||                         3.4%]   Uptime: 00:23:14
  Mem[|||||||||||||||||||||2.13G/3.84G]
  Swp[                        0K/975M]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
 1883 gearhead   20   0 3320M  460M  158M S  4.2 11.7  4:25.35 /usr/lib/firefox-esr/
 1447 gearhead   20   0 4719M  429M  150M S  2.8 10.9  2:29.93 /usr/bin/gnome-shell
 2747 gearhead   20   0 2756M  496M  100M S  2.1 12.6  1:10.58 /usr/lib/firefox-esr/
 1275 gearhead   -6   0 1902M 31612 22440 S  1.4  0.8  0:13.14 /usr/bin/pulseaudio -
 1898 gearhead   20   0 3320M  460M  158M S  1.4 11.7  0:17.09 /usr/lib/firefox-esr/
 1918 gearhead   20   0 3320M  460M  158M R  1.4 11.7  0:15.30 /usr/lib/firefox-esr/
 1928 gearhead   20   0 3320M  460M  158M S  1.4 11.7  0:11.85 /usr/lib/firefox-esr/
 1247 gearhead    9 -11 1902M 31612 22440 S  0.7  0.8  0:13.66 /usr/bin/pulseaudio -
 1277 gearhead   20   0  388M  151M 85080 S  0.7  3.9  1:28.97 /usr/lib/xorg/Xorg vt
 1364 gearhead   20   0  149M  2816  2344 S  0.7  0.1  0:04.41 /usr/bin/VBoxClient -
 1369 gearhead   20   0  149M  2816  2344 S  0.7  0.1  0:04.40 /usr/bin/VBoxClient -
 1455 gearhead   20   0 4719M  429M  150M S  0.7 10.9  0:33.87 /usr/bin/gnome-shell
 1456 gearhead   20   0 4719M  429M  150M S  0.7 10.9  0:36.44 /usr/bin/gnome-shell
 2754 gearhead   20   0 2756M  496M  100M S  0.7 12.6  0:07.11 /usr/lib/firefox-esr/
 3539 gearhead   20   0  8888  4816  3516 R  0.7  0.1  0:00.61 htop
    1 root       20   0  160M 10756  7964 S  0.0  0.3  0:00.99 /sbin/init splash
  262 root       20   0 47140 17452 15960 S  0.0  0.4  0:00.81 /lib/systemd/systemd-
  287 root       20   0 23360  6176  4416 S  0.0  0.2  0:00.16 /lib/systemd/systemd-
```

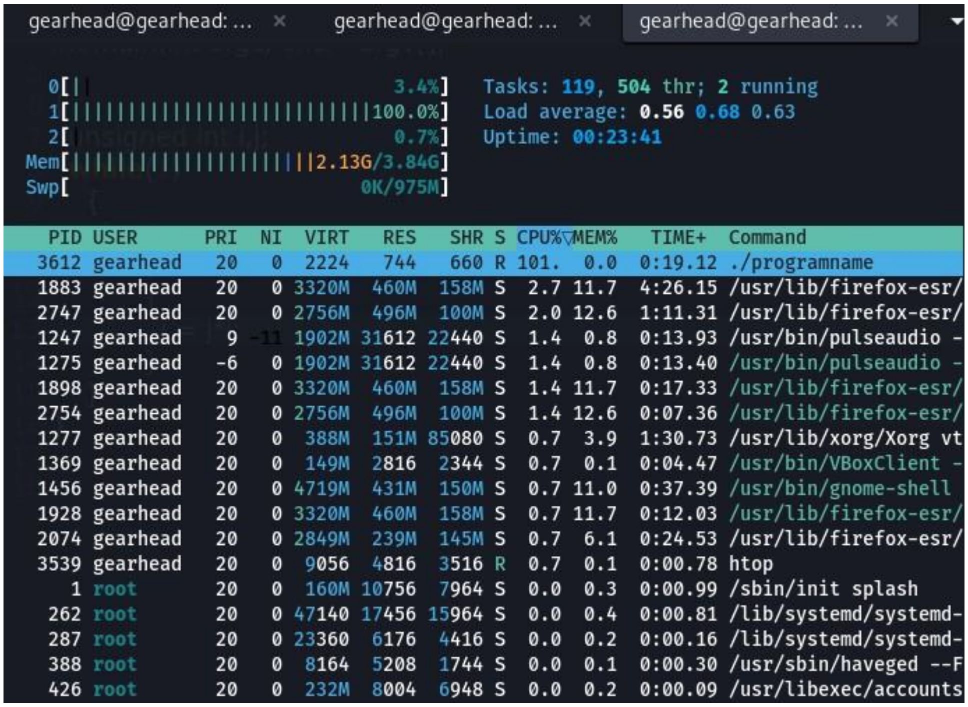
Here CPU usage :– 4.2%

After running the file



```
┌──(gearhead㉿gearhead)-[~/Documents/os]
└─$ gcc programname.c -o programname

┌──(gearhead㉿gearhead)-[~/Documents/os]
└─$ LS
LS: command not found

┌──(gearhead㉿gearhead)-[~/Documents/os]
└─$ ls                                                                  127 ×
A  programname  programname.c

┌──(gearhead㉿gearhead)-[~/Documents/os]
└─$ ./programname
^C

┌──(gearhead㉿gearhead)-[~/Documents/os]
└─$ ./programname                                                       130 ×
█
```

htop command shows the following output



```
    0[||                        3.4%]   Tasks: 119, 504 thr; 2 running
    1[||||||||||||||||||||||||||100.0%]   Load average: 0.56 0.68 0.63
    2[|                         0.7%]   Uptime: 00:23:41
  Mem[|||||||||||||||||||||2.13G/3.84G]
  Swp[                      0K/975M]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
   3612 gearhead   20   0  2224   744   660 R 101.  0.0  0:19.12 ./programname
   1883 gearhead   20   0 3320M  460M  158M S  2.7 11.7  4:26.15 /usr/lib/firefox-esr/
   2747 gearhead   20   0 2756M  496M  100M S  2.0 12.6  1:11.31 /usr/lib/firefox-esr/
   1247 gearhead    9 -11 1902M 31612 22440 S  1.4  0.8  0:13.93 /usr/bin/pulseaudio -
   1275 gearhead   -6   0 1902M 31612 22440 S  1.4  0.8  0:13.40 /usr/bin/pulseaudio -
   1898 gearhead   20   0 3320M  460M  158M S  1.4 11.7  0:17.33 /usr/lib/firefox-esr/
   2754 gearhead   20   0 2756M  496M  100M S  1.4 12.6  0:07.36 /usr/lib/firefox-esr/
   1277 gearhead   20   0  388M  151M 85080 S  0.7  3.9  1:30.73 /usr/lib/xorg/Xorg vt
   1369 gearhead   20   0  149M  2816  2344 S  0.7  0.1  0:04.47 /usr/bin/VBoxClient -
   1456 gearhead   20   0 4719M  431M  150M S  0.7 11.0  0:37.39 /usr/bin/gnome-shell
   1928 gearhead   20   0 3320M  460M  158M S  0.7 11.7  0:12.03 /usr/lib/firefox-esr/
   2074 gearhead   20   0 2849M  239M  145M S  0.7  6.1  0:24.53 /usr/lib/firefox-esr/
   3539 gearhead   20   0  9056  4816  3516 R  0.7  0.1  0:00.78 htop
      1 root       20   0  160M 10756  7964 S  0.0  0.3  0:00.99 /sbin/init splash
    262 root       20   0 47140 17456 15964 S  0.0  0.4  0:00.81 /lib/systemd/systemd-
    287 root       20   0 23360  6176  4416 S  0.0  0.2  0:00.16 /lib/systemd/systemd-
    388 root       20   0  8164  5208  1744 S  0.0  0.1  0:00.30 /usr/sbin/haveged --F
    426 root       20   0  232M  8004  6948 S  0.0  0.2  0:00.09 /usr/libexec/accounts
```

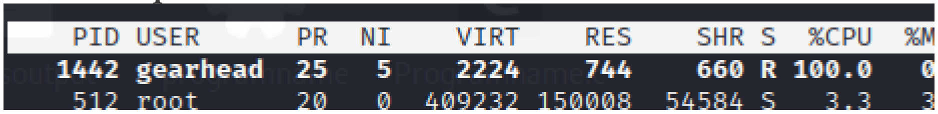
Here CPU usage :- 100%

REASON:

This type of endless loop is generally calculated at the user layer and is not called. **System Call**, **In addition to the expiration of time slice, the kernel (called by the System) will also cause process scheduling. This is missing from the endless loop program.**

a)

PID of cpu =1442



```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %M
 1442 gearhead  25   5    2224    744    660 R 100.0   0
  512 root      20   0  409232 150008  54584 S   3.3   3
```

b) Its consuming 100 % of CPU and 0.00% of Memory

c) Priority of that process is 25

d) The current state is in running state (R)

In my system Three different states are showing

1) Running (R) :: it is the process that is currently served by the CPU
   Its flag is R.
2) Sleeping (S) :: it is the process who waits for the resources to run which lead CPU to send signal and goes to sleep mode. Once the resources get stopped it get awaked and start running [Queue].
3) Interruptible (I) :: if the process is on sleep state waiting for some signal or command to arrive the interruptible condition occurs, our kernel sets the process's states to Running Task.

2.

a) The command to show the processes running in the current shell is `ps`.

b) The command to show all the processes associated with the current terminal is `ps T`.

c) The command to Search PID of a particular process. `ps -C [name of the process] –o pid=`

d) The command to Display child process of a parent process. `ps --ppid [perent id]`

3.

Glances is a cross-platform command-line curses-based system monitoring tool written in **Python** language which use the **psutil** library to grab information from the system. With Glance, we can monitor **CPU**, **Load Average**, **Memory**, **Network Interfaces**, **Disk I/O**, **Processes** and **File System** spaces utilization.

It can also work in client/server mode. Remote monitoring could be done via terminal, Web interface or API (XML-

RPC and RESTful).

Glances is written in Python and uses the psutil library to get information from your system.

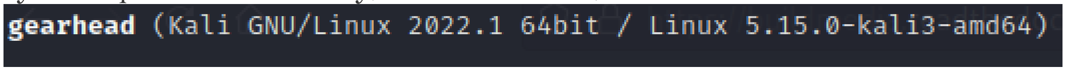
Stats can also be exported to external time/value databases.
Glances is IPv6 compatible.

A)

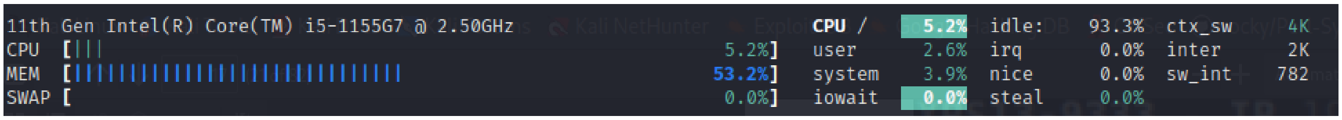
1. CPU Information (user related applications, system core programs and idle programs.
2. Total memory Information including RAM, Swap, Free memory etc.
3. The average CPU load for the past 1min, 5mins and 15 mins.
4. Network Download/Upload rates of network connections.
5. Total number of processes, active ones, sleeping processes etc.
6. Disk I/O related (read or write) speed details
7. Currently mounted devices disk usages.

8. Top processes with their CPU/Memory usages, Names and location of application.
9. Shows the current date and time at bottom.
10 Highlights processes in Red that consumes highest system resources.

The header shows the hostname, OS name, release version, platform architecture IP addresses (private and public) and
system uptime. Additionally, on GNU/Linux, it also shows the kernel version.



```
gearhead (Kali GNU/Linux 2022.1 64bit / Linux 5.15.0-kali3-amd64)
```



```
11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz              CPU /      5.2%  idle:    93.3%  ctx_sw    4K
CPU  [|||                                          5.2%]    user       2.6%  irq       0.0%  inter     2K
MEM  [||||||||||||||||||||||||||||||              53.2%]    system     3.9%  nice      0.0%  sw_int   782
SWAP [                                             0.0%]    iowait     0.0%  steal     0.0%
```

CPU stats description:

• user: percent time spent in user space.

• system: percent time spent in kernel space. System CPU time is the time spent running code in the Operating

System kernel.

• idle: percent of CPU used by any program. Every program or task that runs on a computer system occupies a certain amount of processing time on the CPU. If the CPU has completed all tasks it is idle.

• nice (*nix): percent time occupied by user level processes with a positive nice value. The time the CPU has

spent running users' processes that have been niced.

• irq (Linux, *BSD): percent time spent servicing/handling hardware/software interrupts.

• iowait (Linux): percent time spent by the CPU waiting for I/O operations to complete.

• steal (Linux): percentage of time a virtual CPU waits for a real CPU while the hypervisor is servicing another

virtual processor.

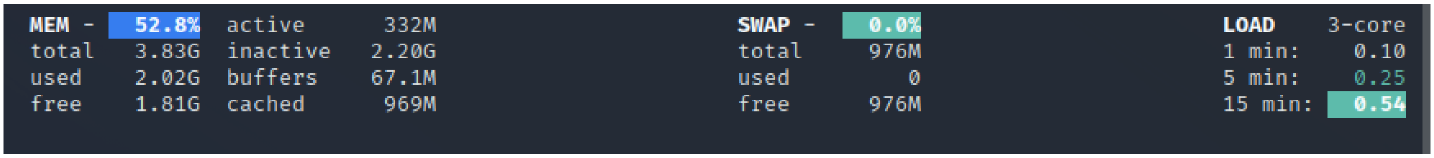
• ctx_sw: number of context switches (voluntary + involuntary) per second. A context switch is a procedure

that a computer's CPU (central processing unit) follows to change from one task (or process) to another while ensuring that the tasks do not conflict.

• inter: number of interrupts per second.

- sw_inter: number of software interrupts per second. Always set to 0 on Windows and SunOS.

- syscal: number of system calls per second. Do not displayed on Linux (always 0).



```
MEM -   52.8%  active      332M                         SWAP -    0.0%                    LOAD     3-core
total   3.83G  inactive   2.20G                         total     976M                    1 min:     0.10
used    2.02G  buffers    67.1M                         used         0                    5 min:     0.25
free    1.81G  cached      969M                         free      976M                    15 min:    0.54
```

Stats description:

- percent: the percentage usage calculated as (total-available)/total*100.

- total: total physical memory available.

- used: memory used, calculated differently depending on the platform and designed for informational purposes only.

- free: memory not being used at all (zeroed) that is readily available; note that this doesn't reflect the actual

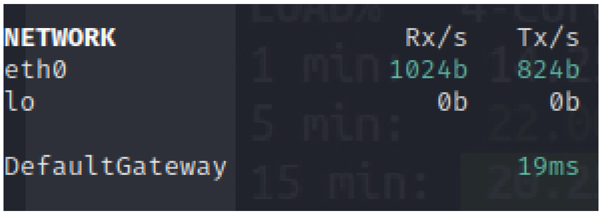
memory available (use 'available' instead).

- active: (UNIX): memory currently in use or very recently used, and so it is in RAM.

- inactive: (UNIX): memory that is marked as not used.

- buffers: (Linux, BSD): cache for things like file system metadata.

- cached: (Linux, BSD): cache for various things.

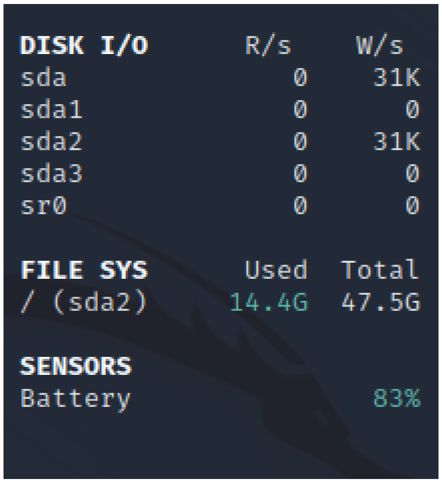


```
NETWORK             Rx/s    Tx/s
eth0               1024b    824b
lo                    0b      0b

DefaultGateway              19ms
```

Glances displays the network interface bit rate. The unit is adapted dynamically (bit/s, kbit/s, Mbit/s, etc).

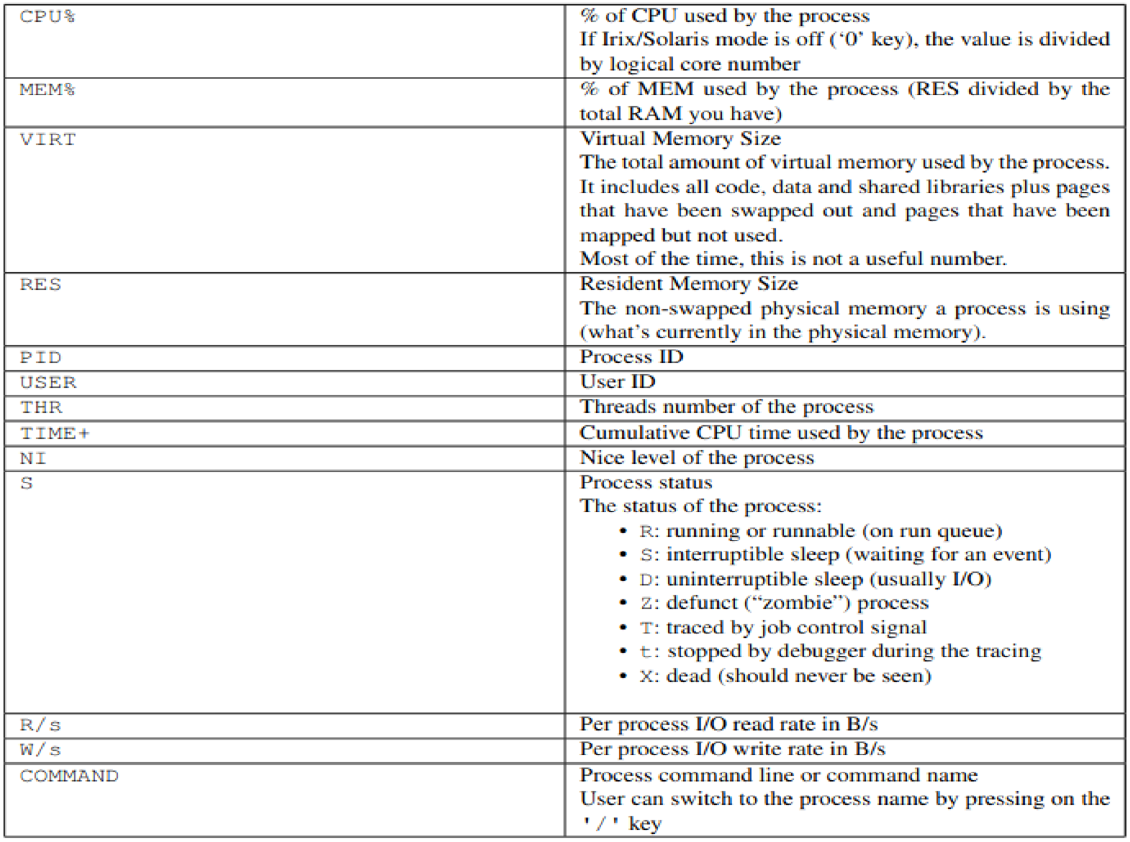
If the interface speed is detected (not on all systems), the defaults thresholds are applied (70% for careful, 80% warning and 90% critical). It is possible to define this percent thresholds from the configuration file. It is also possible to define

per interface bit rate thresholds. In this case thresholds values are defined in bps.



```
DISK I/O       R/s    W/s
sda              0    31K
sda1             0      0
sda2             0    31K
sda3             0      0
sr0              0      0

FILE SYS      Used  Total
/ (sda2)     14.4G  47.5G

SENSORS
Battery                83%
```

Glances displays the disk I/O throughput. The unit is adapted dynamically.Also displays the used and total file system disk space. The unit is adapted dynamically.

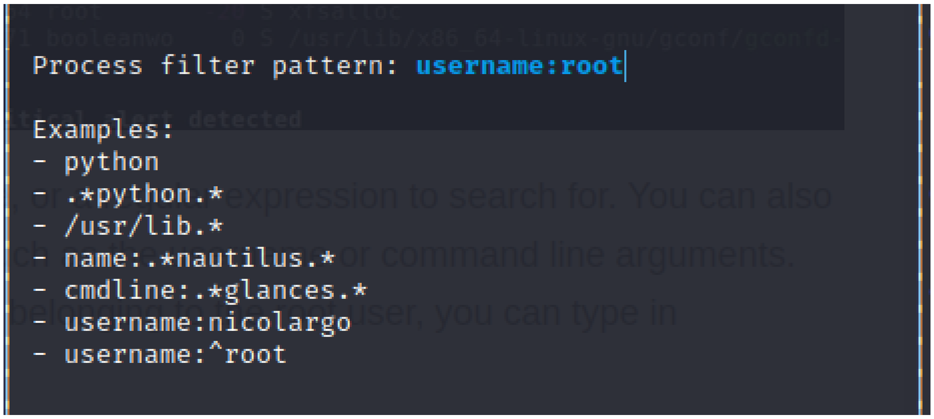


CPU%	% of CPU used by the process If Irix/Solaris mode is off ('0' key), the value is divided by logical core number
MEM%	% of MEM used by the process (RES divided by the total RAM you have)
VIRT	Virtual Memory Size The total amount of virtual memory used by the process. It includes all code, data and shared libraries plus pages that have been swapped out and pages that have been mapped but not used. Most of the time, this is not a useful number.
RES	Resident Memory Size The non-swapped physical memory a process is using (what's currently in the physical memory).
PID	Process ID
USER	User ID
THR	Threads number of the process
TIME+	Cumulative CPU time used by the process
NI	Nice level of the process
S	Process status The status of the process: • R: running or runnable (on run queue) • S: interruptible sleep (waiting for an event) • D: uninterruptible sleep (usually I/O) • Z: defunct ("zombie") process • T: traced by job control signal • t: stopped by debugger during the tracing • X: dead (should never be seen)
R/s	Per process I/O read rate in B/s
W/s	Per process I/O write rate in B/s
COMMAND	Process command line or command name User can switch to the process name by pressing on the '/' key

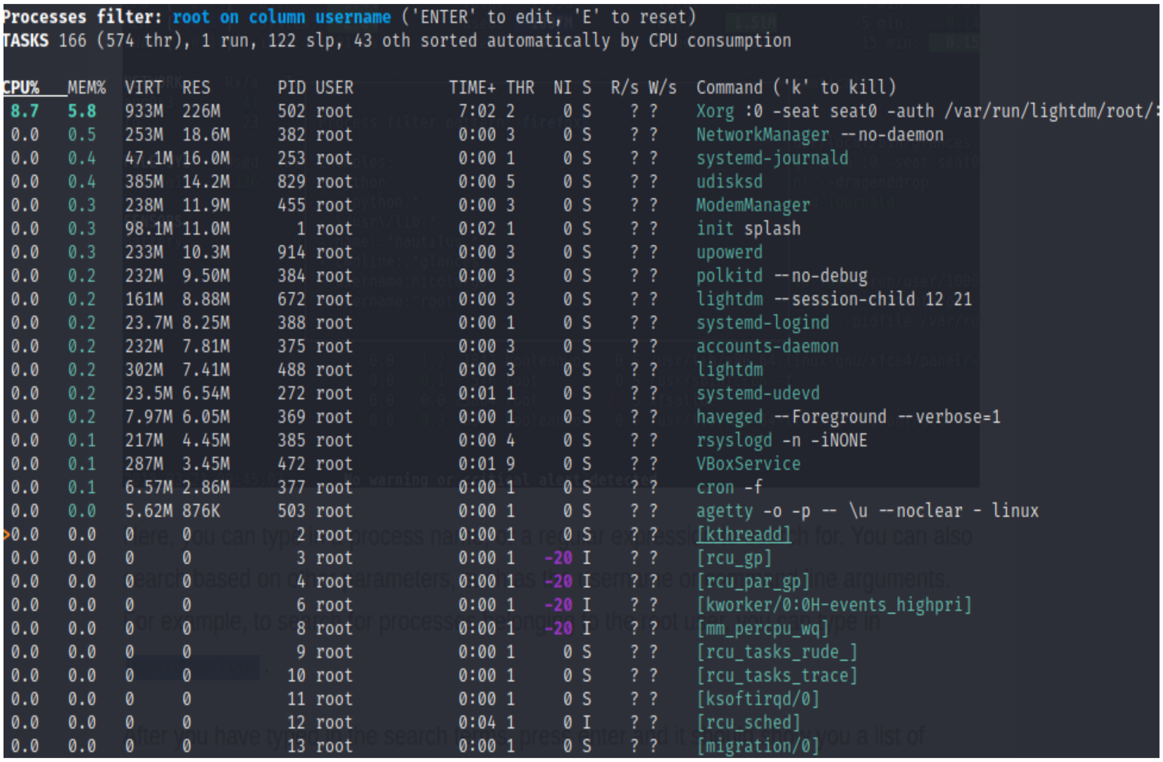
B) Meaning of Glances colour code:

GREEN: OK (everything is fine)
BLUE: CAREFUL (need attention)
VIOLET: WARNING (alert)
RED: CRITICAL (critical)

C)
Press `Enter ` during glances running



```
Process filter pattern: username:root

Examples:
- python
- .*python.*
- /usr/lib.*
- name:.*nautilus.*
- cmdline:.*glances.*
- username:nicolargo
- username:^root
```

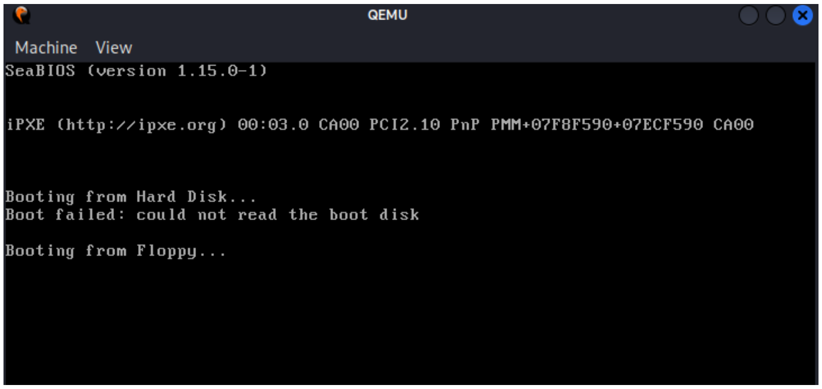


```
Processes filter: root on column username ('ENTER' to edit, 'E' to reset)
TASKS 166 (574 thr), 1 run, 122 slp, 43 oth sorted automatically by CPU consumption

CPU%  MEM%  VIRT  RES      PID USER        TIME+ THR  NI S  R/s W/s  Command ('k' to kill)
 8.7  5.8   933M  226M     502 root         7:02 2     0 S    ? ?   Xorg :0 -seat seat0 -auth /var/run/lightdm/root/:
 0.0  0.5   253M  18.6M    382 root         0:00 3     0 S    ? ?   NetworkManager --no-daemon
 0.0  0.4   47.1M 16.0M    253 root         0:00 1     0 S    ? ?   systemd-journald
 0.0  0.4   385M  14.2M    829 root         0:00 5     0 S    ? ?   udisksd
 0.0  0.3   238M  11.9M    455 root         0:00 3     0 S    ? ?   ModemManager
 0.0  0.3   98.1M 11.0M      1 root         0:02 1     0 S    ? ?   init splash
 0.0  0.3   233M  10.3M    914 root         0:00 3     0 S    ? ?   upowerd
 0.0  0.2   232M  9.50M    384 root         0:00 3     0 S    ? ?   polkitd --no-debug
 0.0  0.2   161M  8.88M    672 root         0:00 3     0 S    ? ?   lightdm --session-child 12 21
 0.0  0.2   23.7M 8.25M    388 root         0:00 1     0 S    ? ?   systemd-logind
 0.0  0.2   232M  7.81M    375 root         0:00 3     0 S    ? ?   accounts-daemon
 0.0  0.2   302M  7.41M    488 root         0:00 3     0 S    ? ?   lightdm
 0.0  0.2   23.5M 6.54M    272 root         0:01 1     0 S    ? ?   systemd-udevd
 0.0  0.2   7.97M 6.05M    369 root         0:00 1     0 S    ? ?   haveged --Foreground --verbose=1
 0.0  0.1   217M  4.45M    385 root         0:00 4     0 S    ? ?   rsyslogd -n -iNONE
 0.0  0.1   287M  3.45M    472 root         0:01 9     0 S    ? ?   VBoxService
 0.0  0.1   6.57M 2.86M    377 root         0:00 1     0 S    ? ?   cron -f
 0.0  0.0   5.62M 876K     503 root         0:00 1     0 S    ? ?   agetty -o -p -- \u --noclear - linux
>0.0  0.0   0     0          2 root         0:00 1     0 S    ? ?   [kthreadd]
 0.0  0.0   0     0          3 root         0:00 1   -20 I    ? ?   [rcu_gp]
 0.0  0.0   0     0          4 root         0:00 1   -20 I    ? ?   [rcu_par_gp]
 0.0  0.0   0     0          6 root         0:00 1   -20 I    ? ?   [kworker/0:0H-events_highpri]
 0.0  0.0   0     0          8 root         0:00 1   -20 I    ? ?   [mm_percpu_wq]
 0.0  0.0   0     0          9 root         0:00 1     0 S    ? ?   [rcu_tasks_rude_]
 0.0  0.0   0     0         10 root         0:00 1     0 S    ? ?   [rcu_tasks_trace]
 0.0  0.0   0     0         11 root         0:00 1     0 S    ? ?   [ksoftirqd/0]
 0.0  0.0   0     0         12 root         0:04 1     0 I    ? ?   [rcu_sched]
 0.0  0.0   0     0         13 root         0:00 1     0 S    ? ?   [migration/0]
```

Part C Understand the Bootloader [6 Points]

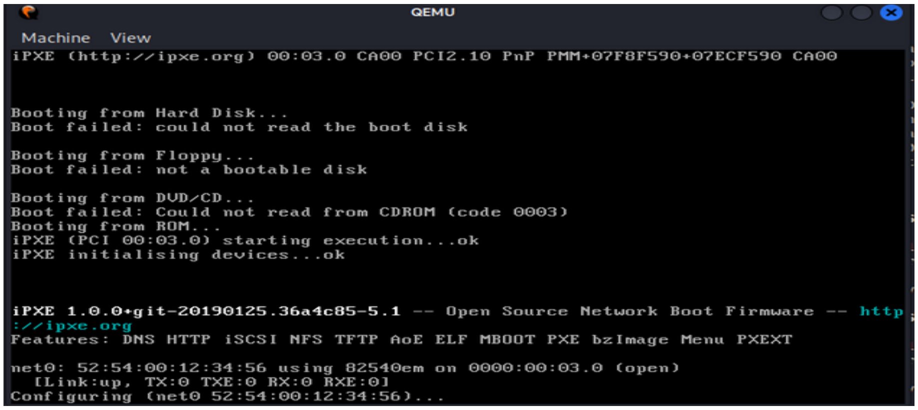
A boot sector or a boot block is a region on a bootable device that contains machine code to be loaded into RAM by a computer system's built-in firmware during its initialization. It is of 512 bytes on a floppy disk.

While Running boot_up1.bin it shows Booting from Floppy disk as my system is running on VM.



```
SeaBIOS (version 1.15.0-1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F590+07ECF590 CA00


Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from Floppy...
```

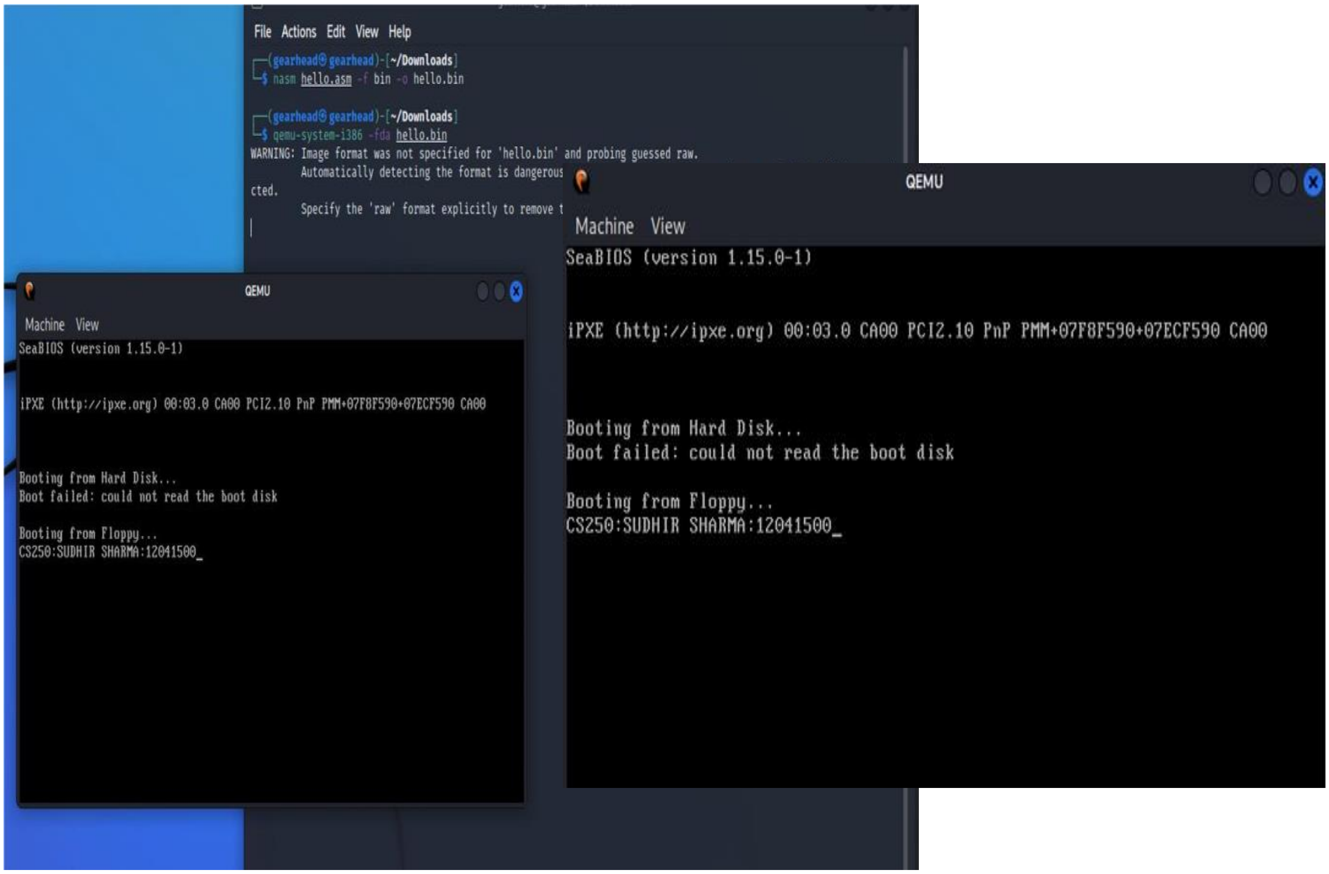
While running boot_up2.bin it shows booting from ROM



```
iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F590+07ECF590 CA00


Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from Floppy...
Boot failed: not a bootable disk

Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok


iPXE 1.0.0+git-20190125.36a4c85-5.1 -- Open Source Network Boot Firmware -- http
://ipxe.org
Features: DNS HTTP iSCSI NFS TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:12:34:56 using 82540em on 0000:00:03.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:12:34:56)...
```

The Difference I saw while running both the file is that boot_up1.bin file boots using Floppy disk and while running boot_up2.bin file its shows booting using Rom.

The changes we made in hello.asm file is that we added a line



```
┌──(gearhead㉿gearhead)-[~/Downloads]
└─$ nasm hello.asm -f bin -o hello.bin

┌──(gearhead㉿gearhead)-[~/Downloads]
└─$ qemu-system-i386 -fda hello.bin
WARNING: Image format was not specified for 'hello.bin' and probing guessed raw.
         Automatically detecting the format is dangerous
cted.
         Specify the 'raw' format explicitly to remove t
```

```
SeaBIOS (version 1.15.0-1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F590+07ECF590 CA00


Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from Floppy...
CS250:SUDHIR SHARMA:12041500_
```

```
SeaBIOS (version 1.15.0-1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8F590+07ECF590 CA00


Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from Floppy...
CS250:SUDHIR SHARMA:12041500_
```

`msg:   db "CS250:SUDHIR SHARMA:12041500", 0   ; Our actual message to print` in line no-15.

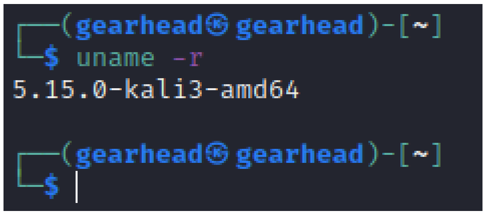
Lets see what are those t

```
db      0x55                ; just the byte 0x55
db      0x55,0x56,0x57      ; three bytes in succession
db      'a',0x55            ; character constants are OK
db      'hello',13,10,'$'   ; so are string constants
dw      0x1234              ; 0x34 0x12
dw      'A'                 ; 0x41 0x00 (it's just a number)
dw      'AB'                ; 0x41 0x42 (character constant)
dw      'ABC'               ; 0x41 0x42 0x43 0x00 (string)
```

DB = **define byte size variables**. DW = define word size (16 bits) variables. DD = define double word size (32 bits) variables.

Part D: Compilation and Installation of latest Linux kernel

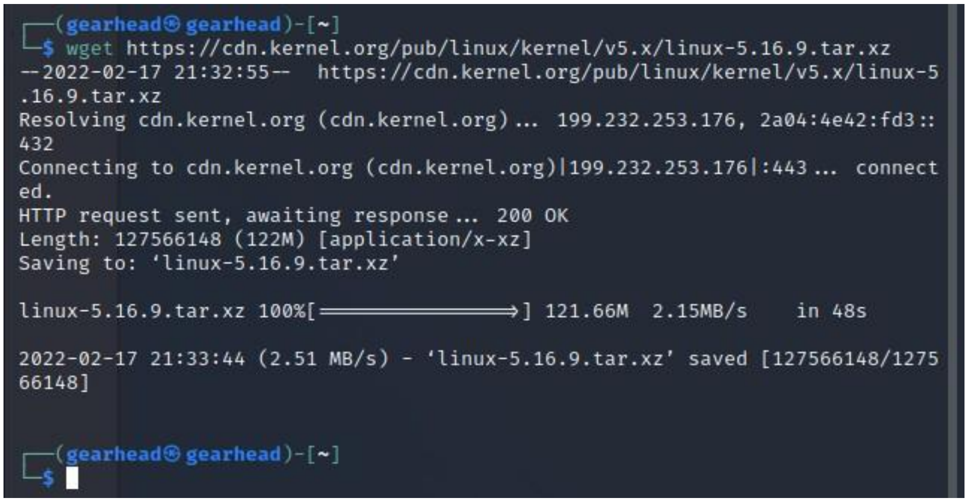
At present my sys shows



```
┌──(gearhead㉿gearhead)-[~]
└─$ uname -r
5.15.0-kali3-amd64

┌──(gearhead㉿gearhead)-[~]
└─$ |
```

Step 1. Get the latest Linux kernel source code

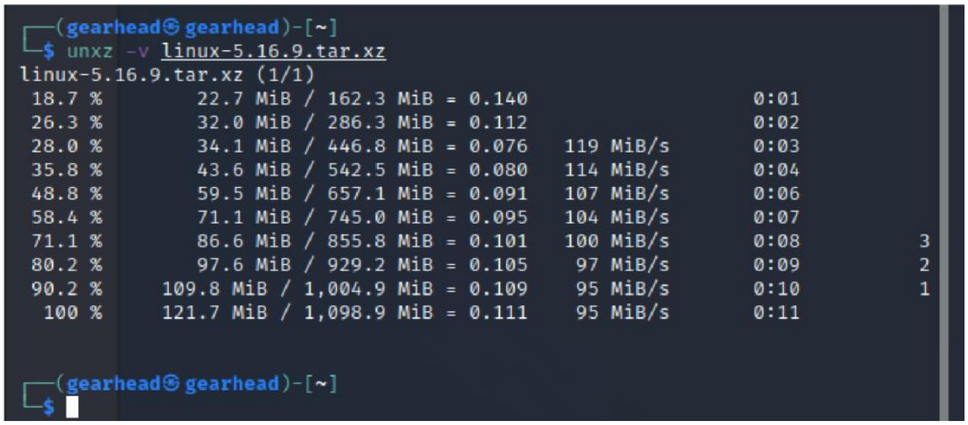
wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.16.9.tar.xz



```
┌──(gearhead㉿gearhead)-[~]
└─$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.16.9.tar.xz
--2022-02-17 21:32:55--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5
.16.9.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 199.232.253.176, 2a04:4e42:fd3::
432
Connecting to cdn.kernel.org (cdn.kernel.org)|199.232.253.176|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 127566148 (122M) [application/x-xz]
Saving to: ‘linux-5.16.9.tar.xz’

linux-5.16.9.tar.xz 100%[=================>] 121.66M  2.15MB/s    in 48s

2022-02-17 21:33:44 (2.51 MB/s) - ‘linux-5.16.9.tar.xz’ saved [127566148/1275
66148]


┌──(gearhead㉿gearhead)-[~]
└─$ █
```

Step 2. Extract tar.xz file

unxz -v linux-5.16.9.tar.xz



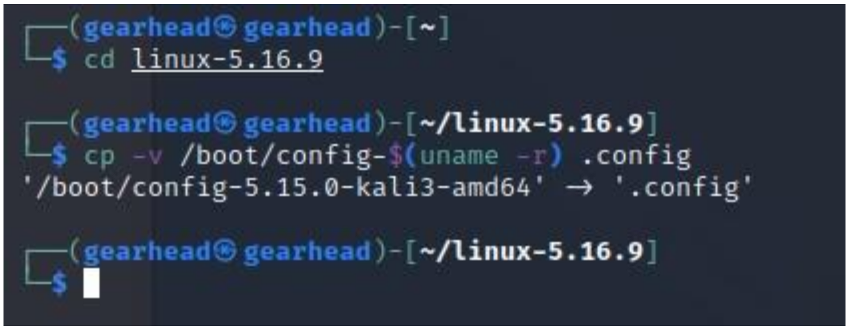
```
┌──(gearhead㉿gearhead)-[~]
└─$ unxz -v linux-5.16.9.tar.xz
linux-5.16.9.tar.xz (1/1)
 18.7 %       22.7 MiB / 162.3 MiB = 0.140                   0:01
 26.3 %       32.0 MiB / 286.3 MiB = 0.112                   0:02
 28.0 %       34.1 MiB / 446.8 MiB = 0.076   119 MiB/s       0:03
 35.8 %       43.6 MiB / 542.5 MiB = 0.080   114 MiB/s       0:04
 48.8 %       59.5 MiB / 657.1 MiB = 0.091   107 MiB/s       0:06
 58.4 %       71.1 MiB / 745.0 MiB = 0.095   104 MiB/s       0:07
 71.1 %       86.6 MiB / 855.8 MiB = 0.101   100 MiB/s       0:08          3
 80.2 %       97.6 MiB / 929.2 MiB = 0.105    97 MiB/s       0:09          2
 90.2 %     109.8 MiB / 1,004.9 MiB = 0.109    95 MiB/s       0:10          1
  100 %     121.7 MiB / 1,098.9 MiB = 0.111    95 MiB/s       0:11


┌──(gearhead㉿gearhead)-[~]
└─$ █
```

tar -xvf linux-5.16.9.tar

Step 3. Configure the Linux kernel features and modules

$ cd linux-5.16.9

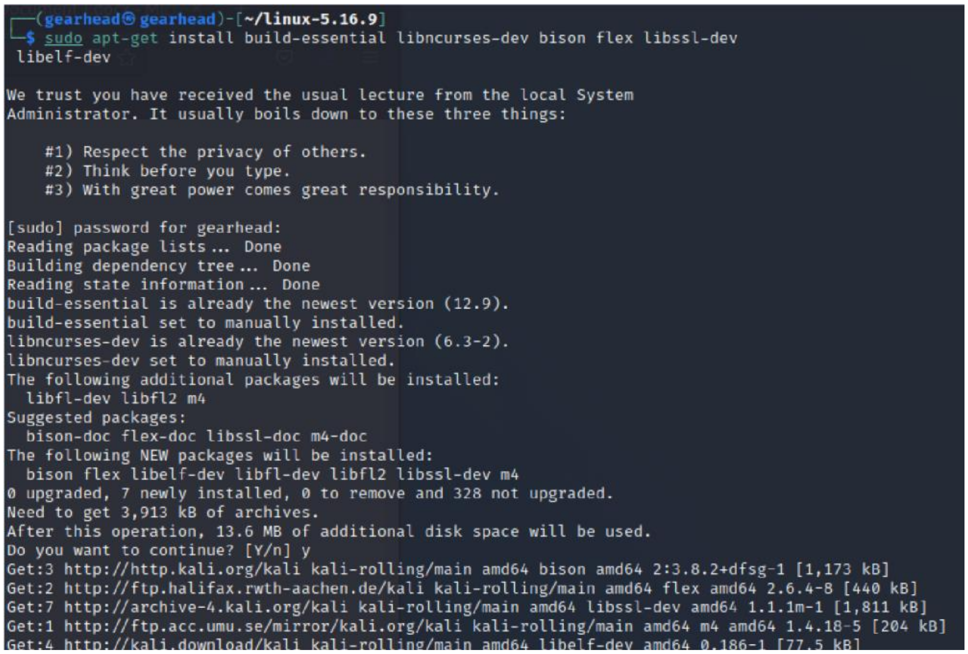
$ cp -v /boot/config-$(uname -r) .config



```
┌──(gearhead㉿gearhead)-[~]
└─$ cd linux-5.16.9

┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ cp -v /boot/config-$(uname -r) .config
'/boot/config-5.15.0-kali3-amd64' → '.config'

┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ █
```

Step 4. Install the required compilers and other tools

sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev

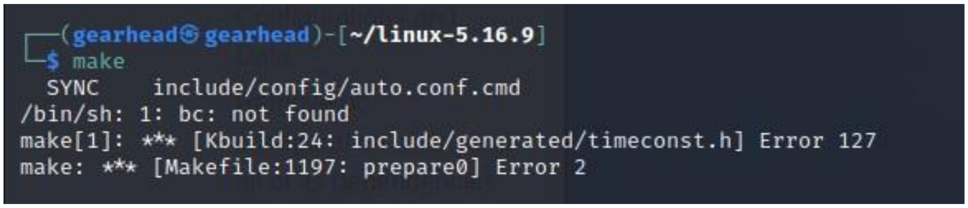


```
┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ sudo apt-get install build-essential libncurses-dev bison flex libssl-dev
 libelf-dev

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for gearhead:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
build-essential is already the newest version (12.9).
build-essential set to manually installed.
libncurses-dev is already the newest version (6.3-2).
libncurses-dev set to manually installed.
The following additional packages will be installed:
  libfl-dev libfl2 m4
Suggested packages:
  bison-doc flex-doc libssl-doc m4-doc
The following NEW packages will be installed:
  bison flex libelf-dev libfl-dev libfl2 libssl-dev m4
0 upgraded, 7 newly installed, 0 to remove and 328 not upgraded.
Need to get 3,913 kB of archives.
After this operation, 13.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:3 http://http.kali.org/kali kali-rolling/main amd64 bison amd64 2:3.8.2+dfsg-1 [1,173 kB]
Get:2 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 flex amd64 2.6.4-8 [440 kB]
Get:7 http://archive-4.kali.org/kali kali-rolling/main amd64 libssl-dev amd64 1.1.1m-1 [1,811 kB]
Get:1 http://ftp.acc.umu.se/mirror/kali.org/kali kali-rolling/main amd64 m4 amd64 1.4.18-5 [204 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 libelf-dev amd64 0.186-1 [77.5 kB]
```

Step 5. Configuring the kernel

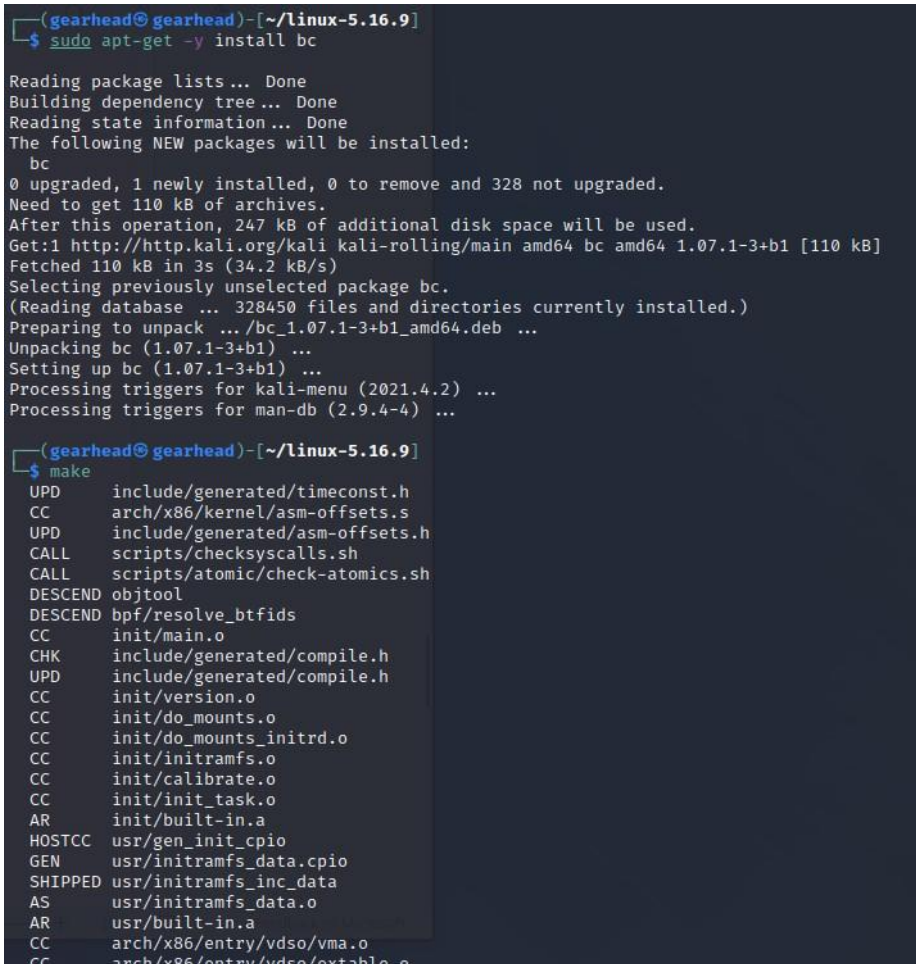
$ make

If you get these error

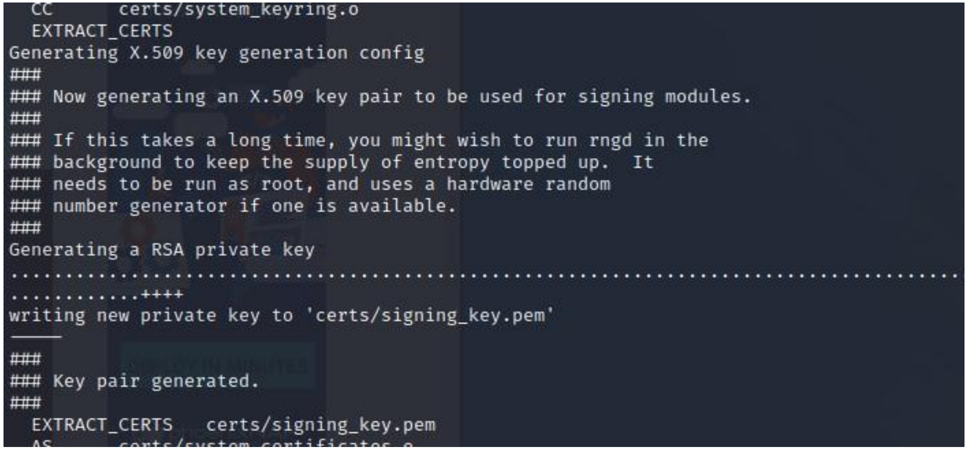


```
┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ make
  SYNC    include/config/auto.conf.cmd
/bin/sh: 1: bc: not found
make[1]: *** [Kbuild:24: include/generated/timeconst.h] Error 127
make: *** [Makefile:1197: prepare0] Error 2
```

Then use these



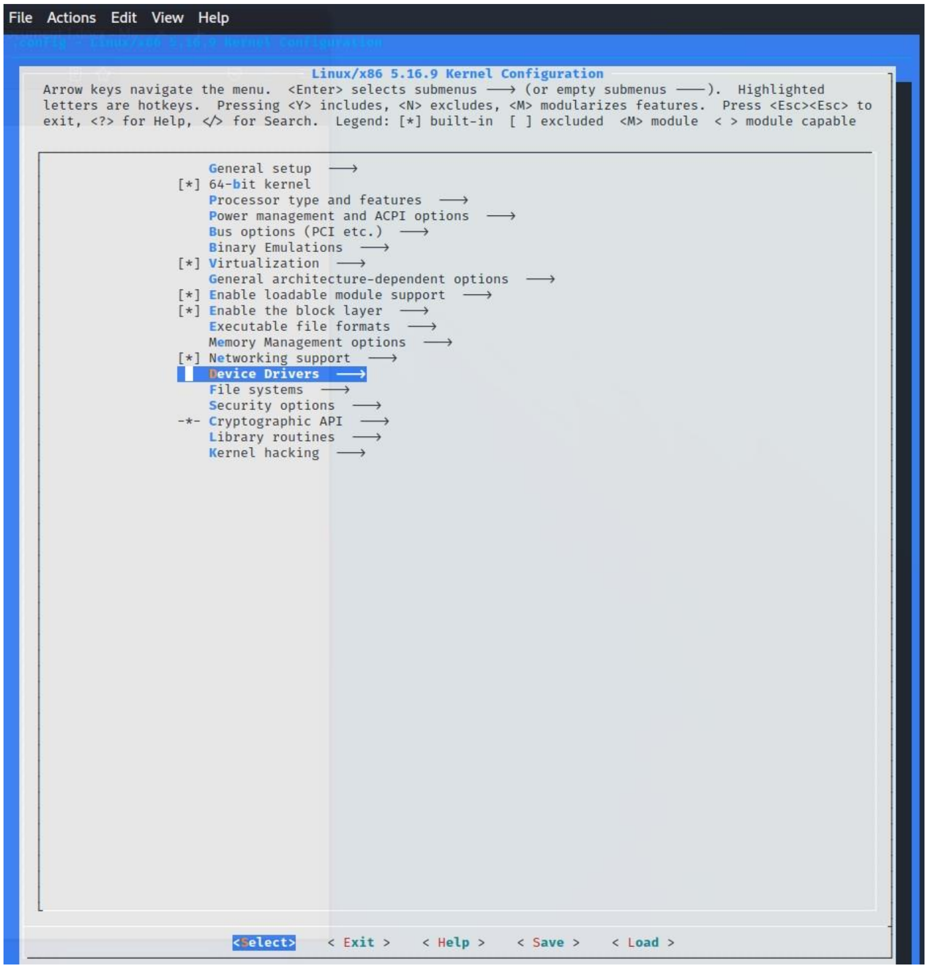
```
┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ sudo apt-get -y install bc

Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
  bc
0 upgraded, 1 newly installed, 0 to remove and 328 not upgraded.
Need to get 110 kB of archives.
After this operation, 247 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 bc amd64 1.07.1-3+b1 [110 kB]
Fetched 110 kB in 3s (34.2 kB/s)
Selecting previously unselected package bc.
(Reading database ... 328450 files and directories currently installed.)
Preparing to unpack .../bc_1.07.1-3+b1_amd64.deb ...
Unpacking bc (1.07.1-3+b1) ...
Setting up bc (1.07.1-3+b1) ...
Processing triggers for kali-menu (2021.4.2) ...
Processing triggers for man-db (2.9.4-4) ...

┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ make
  UPD     include/generated/timeconst.h
  CC      arch/x86/kernel/asm-offsets.s
  UPD     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  CALL    scripts/atomic/check-atomics.sh
  DESCEND objtool
  DESCEND bpf/resolve_btfids
  CC      init/main.o
  CHK     include/generated/compile.h
  UPD     include/generated/compile.h
  CC      init/version.o
  CC      init/do_mounts.o
  CC      init/do_mounts_initrd.o
  CC      init/initramfs.o
  CC      init/calibrate.o
  CC      init/init_task.o
  AR      init/built-in.a
  HOSTCC  usr/gen_init_cpio
  GEN     usr/initramfs_data.cpio
  SHIPPED usr/initramfs_inc_data
  AS      usr/initramfs_data.o
  AR      usr/built-in.a
  CC      arch/x86/entry/vdso/vma.o
  CC      arch/x86/entry/vdso/extable.o
```



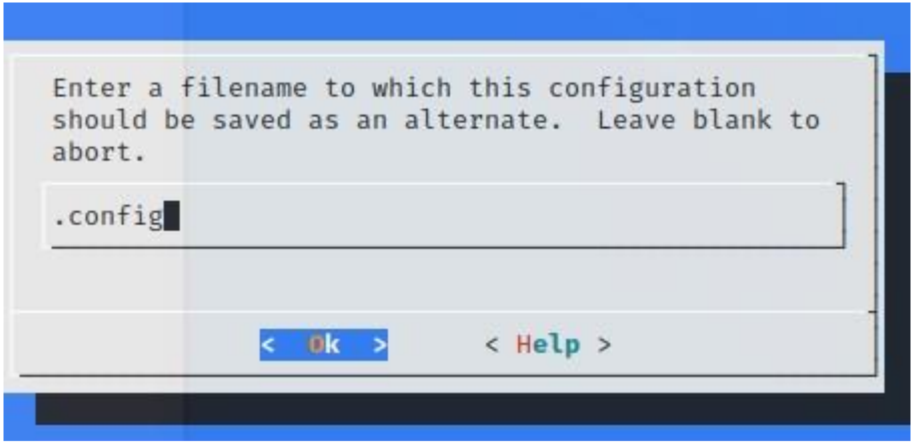
```
  CC      certs/system_keyring.o
  EXTRACT_CERTS
Generating X.509 key generation config
###
### Now generating an X.509 key pair to be used for signing modules.
###
### If this takes a long time, you might wish to run rngd in the
### background to keep the supply of entropy topped up.  It
### needs to be run as root, and uses a hardware random
### number generator if one is available.
###
Generating a RSA private key
..............................................................................................
...........++++
writing new private key to 'certs/signing_key.pem'
-----
###
### Key pair generated.
###
  EXTRACT_CERTS   certs/signing_key.pem
  AS      certs/system_certificates.o
```

```
$ make menuconfig
```

It takes some times to install



```
.config - Linux/x86 5.16.9 Kernel Configuration

                    Linux/x86 5.16.9 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                     General setup  --->
                 [*] 64-bit kernel
                     Processor type and features  --->
                     Power management and ACPI options  --->
                     Bus options (PCI etc.)  --->
                     Binary Emulations  --->
                 [*] Virtualization  --->
                     General architecture-dependent options  --->
                 [*] Enable loadable module support  --->
                 [*] Enable the block layer  --->
                     Executable file formats  --->
                     Memory Management options  --->
                 [*] Networking support  --->
                     Device Drivers  --->
                     File systems  --->
                     Security options  --->
                 -*- Cryptographic API  --->
                     Library routines  --->
                     Kernel hacking  --->

              <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
Enter a filename to which this configuration
should be saved as an alternate.  Leave blank to
abort.

.config

        <  Ok  >        < Help >
```

Finally

$ make modules_install

Step6: Install Kernel

After installing the modules we need to install Kernel by executing the below command:

$ sudo make install

Step7: Enable Kernel for boot

Once you are done with installing Kernel, then we have to enable Kernel for a boot, for which execute the below command:

$ sudo update-initramfs -c -k 5.14.13

Remember to replace the version in the above command with your version of the kernel you just compiled.

The next step is to update-grub for which type or copy the following command in your Ubuntu terminal and then press enter:

$ sudo update-grub

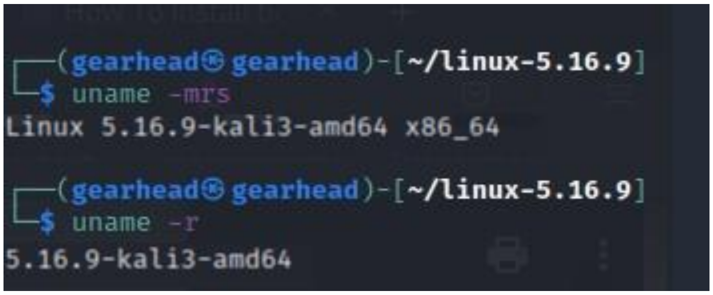
Step8: Reboot System

This step involves rebooting your system for which execute the reboot command in your terminal:

$ reboot

Step9: Verification of Linux Kernel

This last step involves verifying the new Linux Kernel version which can be achieved with the following command:



```
┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ uname -mrs
Linux 5.16.9-kali3-amd64 x86_64

┌──(gearhead㉿gearhead)-[~/linux-5.16.9]
└─$ uname -r
5.16.9-kali3-amd64
```