Operating Systems Lab

Lab 1 Report

Mohammad Sameer

190010024

CSE, 2019

Question Number 1:

a. Output of **`more /proc/cpuinfo`** command

```
sameer@sam-ubuntu:~$ more /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 23
model           : 24
model name      : AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx
stepping        : 1
microcode       : 0x8108102
cpu MHz         : 1273.088
cache size      : 512 KB
physical id     : 0
siblings        : 8
core id         : 0
cpu cores       : 4
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
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 mmxext fxsr_opt pdpe1gb rdtscp
lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand
lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb hw
_pstate sme ssbd sev ibpb vmmcall sev_es fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap clflushopt sha_ni xsaveopt xsavec xgetbv1 xsaves clzero irperf xsaveerptr a
rat npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif overflow_recov succor smca
bugs            : sysret_ss_attrs null_seg spectre_v1 spectre_v2 spec_store_bypass
bogomips        : 4192.08
TLB size        : 2560 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 43 bits physical, 48 bits virtual
power management: ts ttp tm hwpstate eff_freq_ro [13] [14]

processor       : 1
vendor_id       : AuthenticAMD
cpu family      : 23
model           : 24
model name      : AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx
stepping        : 1
microcode       : 0x8108102
cpu MHz         : 1400.000
cache size      : 512 KB
physical id     : 0
--More--(0%)
```

Processor: A processor is the part of a computer that performs logical and arithmetical operations on the data as specified in the instructions. Modern computers may have multiple cores and perform numerous operations in parallel.

Core: A core is the CPU's fundamental compute unit; it may run a single program (or numerous if it supports hardware threads such as hyperthreading), keeping track of program state, registers, and execution order, and executing operations via Arithmetic Logic Units.

In the days when computers were first built, they had processors with a single compute unit, so the core and processor had the same meaning back then. These days when we say processor, we usually mean some number of cores.

Output of **`lscpu`** command.

```
sameer@sam-ubuntu:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   43 bits physical, 48 bits virtual
CPU(s):                          8
On-line CPU(s) list:             0-7
Thread(s) per core:              2
Core(s) per socket:              4
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       AuthenticAMD
CPU family:                      23
Model:                           24
Model name:                      AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx
Stepping:                        1
Frequency boost:                 enabled
CPU MHz:                         1400.000
CPU max MHz:                     2100.0000
CPU min MHz:                     1400.0000
BogoMIPS:                        4192.08
Virtualization:                  AMD-V
L1d cache:                       128 KiB
L1i cache:                       256 KiB
L2 cache:                        2 MiB
L3 cache:                        4 MiB
NUMA node0 CPU(s):               0-7
Vulnerability Itlb multihit:     Not affected
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Full AMD retpoline, IBPB conditional, STIBP disabled, RSB filling
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Not affected
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 mmxext fxsr_opt
                                 pdpe1gb rdtscp lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 sse4_
                                 1 sse4_2 movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch os
                                 vw skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb hw_pstate sme ssbd sev ibpb vmmcall sev_es fsgsba
                                 se bmi1 avx2 smep bmi2 rdseed adx smap clflushopt sha_ni xsaveopt xsavec xgetbv1 xsaves clzero irperf xsaveerptr arat npt lbrv s
                                 vm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif overflow_reco
                                 v succor smca
sameer@sam-ubuntu:~$
```

---

b. On running **`lscpu`** the following fields in the output:

```
Core(s) per socket:      4
Socket(s):               1
```

This effectively gives us Socket(s) * Core (s) per socket = 1 * 4 = 4 cores, therefore my machine has 4 cores.

---

c. The command used is **`cat /proc/cpuinfo | grep processor`**. Here from the **`/proc/cpuinfo`** we are listing all of the processors of the machine, the output of this command is:

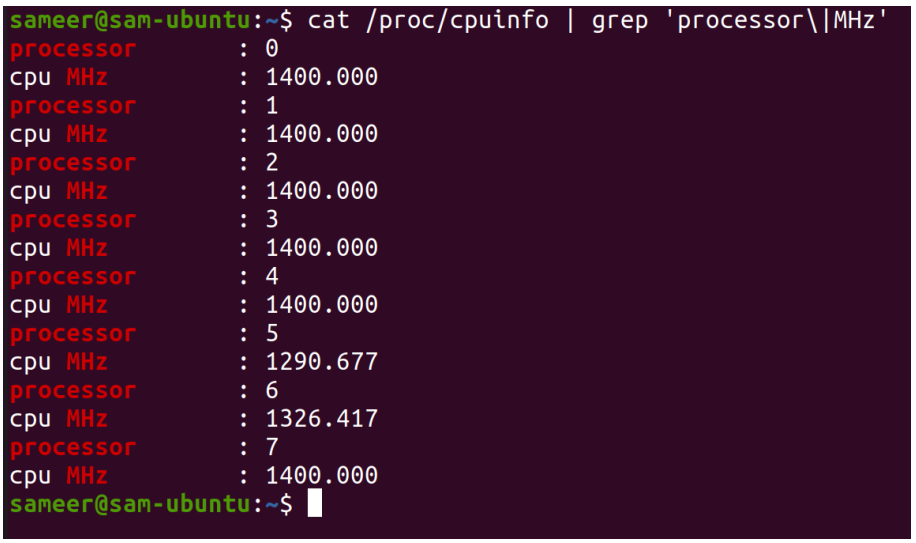
```
sameer@sam-ubuntu:~$ cat /proc/cpuinfo | grep processor
processor       : 0
processor       : 1
processor       : 2
processor       : 3
processor       : 4
processor       : 5
processor       : 6
processor       : 7
sameer@sam-ubuntu:~$
```

As we can see the processors are numbered 0 to 7, making a total of 8 processors.

---

d. The command used is **`cat /proc/cpuinfo | grep 'processor\|MHz'`**. Here from **`/proc/cpuinfo`** we are listing all occurrences of the processor and its speed in MHz. The output

of this gives us the speed of each processor as follows:




```
processor       : 0
cpu MHz         : 1400.000
processor       : 1
cpu MHz         : 1400.000
processor       : 2
cpu MHz         : 1400.000
processor       : 3
cpu MHz         : 1400.000
processor       : 4
cpu MHz         : 1400.000
processor       : 5
cpu MHz         : 1290.677
processor       : 6
cpu MHz         : 1326.417
processor       : 7
cpu MHz         : 1400.000
```

Even though we have slightly different in the processor speeds, there standard speed is 1400 Mhz

---

e. The command used is **`more /proc/meminfo`**. This command gives us memory related information. Output of **`more /proc/meminfo`** command

```
sameer@sam-ubuntu:~$ more /proc/meminfo
MemTotal:       14260024 kB
MemFree:         4005780 kB
MemAvailable:    8425844 kB
Buffers:          223728 kB
Cached:          4874560 kB
SwapCached:            0 kB
Active:          1909872 kB
Inactive:        7516732 kB
Active(anon):       4856 kB
Inactive(anon):  4985672 kB
Active(file):    1905016 kB
Inactive(file):  2531060 kB
Unevictable:          48 kB
Mlocked:              48 kB
SwapTotal:       6779900 kB
SwapFree:        6779900 kB
Dirty:              5820 kB
Writeback:             0 kB
AnonPages:       4328432 kB
Mapped:          1192100 kB
Shmem:            671464 kB
KReclaimable:     316972 kB
Slab:             472156 kB
SReclaimable:     316972 kB
SUnreclaim:       155184 kB
KernelStack:       25904 kB
PageTables:        54604 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    13909912 kB
Committed_AS:   13626068 kB
VmallocTotal:   34359738367 kB
VmallocUsed:       57068 kB
VmallocChunk:          0 kB
Percpu:            18048 kB
HardwareCorrupted:     0 kB
AnonHugePages:      2048 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
```

My machine has 14260024 kB (14.26 GB) of physical memory as seen from the above figure. Technically it has to be around 16 GB, but some of it reserved by AMD VEGA Graphics.

---

f. Out of the above memory, 4005780 kB (4.00 GB) of memory is free as seen from the above figure (Used **`more /proc/meminfo`** command to get it).

---

g. **`vmstat -f`** command is used to get the total number of forks since the boot in the system. Output of this command is:

```
sameer@sam-ubuntu:~$ vmstat -f
        4408 forks
sameer@sam-ubuntu:~$
```

As seen from the above figure, the total number of forks since the boot in the system is 4408.

---

h. **`more /proc/stat`** command gives the information about total context switches that the system has performed since bootup. The output of this command is:

As seen from the above figure, the total number of context switches is **`ctxt 12434946`**.

Question Number 2:

Output of **`top`** command:

```
top - 18:39:23 up 4 min,  1 user,  load average: 2.44, 1.64, 0.69
Tasks: 348 total,   3 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s): 13.3 us,  0.2 sy,  0.0 ni, 85.2 id,  0.1 wa,  0.0 hi,  1.2 si,  0.0 st
MiB Mem :  13925.8 total,   8024.3 free,   3136.7 used,   2764.8 buff/cache
MiB Swap:   6621.0 total,   6621.0 free,      0.0 used.   9922.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11889 sameer    20   0    2364    580    512 R  99.7   0.0   0:47.68 cpu
   2132 sameer    20   0 5993760 316880 124492 S   4.6   2.2   0:19.09 gnome-shell
   1944 sameer    20   0 2793904 167708 112284 S   2.6   1.2   0:18.91 Xorg
   2574 sameer    20   0 4504224 510640 241124 S   1.3   3.6   0:54.38 GeckoMain
   2850 sameer    20   0 3189332 558252 200928 S   1.0   3.9   0:29.63 Isolated Web Co
    125 root      20   0       0      0      0 I   0.3   0.0   0:00.23 kworker/u32:1-phy0
    594 root      -2   0       0      0      0 S   0.3   0.0   0:01.61 gfx
    899 root      20   0  336456  20068  16624 S   0.3   0.1   0:00.99 NetworkManager
   2796 sameer    20   0   32.4g 157744  99852 S   0.3   1.1   0:10.48 WebExtensions
   2856 sameer    20   0 3106468 362032 138564 S   0.3   2.5   0:26.08 Isolated Web Co
   3140 sameer    20   0 3074040 338884 146276 S   0.3   2.4   0:24.69 Isolated Web Co
   4298 systemd+  20   0   23928  11896   7968 S   0.3   0.1   0:00.08 systemd-resolve
  11902 sameer    20   0   12132   4244   3296 R   0.3   0.0   0:00.52 top
      1 root      20   0  169456  13624   8384 S   0.0   0.1   0:03.11 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      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      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0-cgroup_destroy
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.01 kworker/0:1-events
      8 root      20   0       0      0      0 I   0.0   0.0   0:00.46 kworker/u32:0-events_unbound
      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   0:00.05 ksoftirqd/0
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.25 rcu_sched
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     15 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     18 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     19 root      rt   0       0      0      0 S   0.0   0.0   0:00.23 migration/1
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.04 ksoftirqd/1
     21 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0-kec_query
     22 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
```

a. From the above figure, the PID of the process running cpu is **`11889`**.
b. From the above figure, it uses **`99.7 %`** of CPU and **`0.0 %`** of Memory.
c. From the above figure, the current state of this process is **`R`**, meaning **`RUNNING`**

Question Number 3:

a. To view the PID of process running **cpu-print**, we can either use **ps -fC cpu-print** or

ps aux | grep cpu-print

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ps -fC cpu-print
UID          PID    PPID  C STIME TTY          TIME CMD
sameer     12676   12595 52 18:50 pts/0    00:02:41 ./cpu-print
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ps aux | grep cpu-print
sameer     12676 52.5  0.0   2496   720 pts/0    S+   18:50   3:19 ./cpu-print
sameer     12884  0.0  0.0   9040   736 pts/3    S+   18:56   0:00 grep --color=auto cpu-print
```

The PID of the **cpu-print** process is **12676**.

---

b. To view the PIDs of the parent of the **cpu-print** process and its ancessestors we use pstree command, along with the **-p** and **-s** options. The **-s** option shows parent processes of the specified process, so we also include the PID of the **cpu-print** process and **-p** option enables showing of the PIDS. The output of **pstree -ps 12676** command is:

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ pstree -ps 12676
systemd(1)---systemd(1842)---gnome-terminal-(4935)---bash(12595)---cpu-print(12676)
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$
```

From the above figure, we can see that PID of parent of **cpu-print** (bash terminal) is **12595**, PID of its parent (gnome-terminal) is **4935**, PID of its parent (systemd) is **1842** and PID of its parent (systemd) is **1**.

---

c. After running the **./cpu-print > /tmp/tmp.txt &** command in the terminal window, it gave the PID of the **cpu-print** process as **14040**. Using this and **/proc/[PID]/fd** command I was able to access the directory of all the file descriptors of the cpu-print command. Since it is a directory, to print its contents I have used the **ls -la** command. Overall the command was **ls -la /proc/14040/fd**. Its output is:

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ls -la /proc/14040/fd
total 0
dr-x------ 2 sameer sameer  0 Jan 12 19:30 .
dr-xr-xr-x 9 sameer sameer  0 Jan 12 19:30 ..
lrwx------ 1 sameer sameer 64 Jan 12 19:30 0 -> /dev/pts/0
l-wx------ 1 sameer sameer 64 Jan 12 19:30 1 -> /tmp/tmp.txt
lrwx------ 1 sameer sameer 64 Jan 12 19:30 2 -> /dev/pts/0
```

From the above figure,

- the 0 file descriptor (standard input) was pointing to **/dev/pts/0**
- the 2 file descriptor (standard error) was pointing to **/dev/pts/0**

- but the 1 file descriptor (standard output) was pointing to **`/tmp/tmp.txt`**

  This is because we have specified a redirection of output to /tmp/tmp.txt in the command line, instead of the STDOUT.

  So the shell usually has

- standard input file descriptor (0) pointing to **`STDIN`**,
- standard output file descriptor (1) pointing to **`STDOUT`** and
- standard error file descriptor (2) pointing to **`STDERR`**.

  If we specify any sort of redirection (eg., > to indicate output redirection), it sets these file descriptors to the specified descriptor and performs the redirection.

---

d. After running the **`./cpu-print | grep hello &`** command in the terminal window, it gave the PID of the **`grep`** process as **`14373`**. Using this PID and the command mentioned in the above (c.) point, I was able to access the file descriptor of the **`grep`** process. The output of **`ls -la /proc/14373/fd`** command is:

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ls -la /proc/14373/fd
total 0
dr-x------ 2 sameer sameer  0 Jan 12 19:44 .
dr-xr-xr-x 9 sameer sameer  0 Jan 12 19:44 ..
lr-x------ 1 sameer sameer 64 Jan 12 19:44 0 -> 'pipe:[148826]'
lrwx------ 1 sameer sameer 64 Jan 12 19:44 1 -> /dev/pts/0
lrwx------ 1 sameer sameer 64 Jan 12 19:44 2 -> /dev/pts/0
```

From the above figure,

- the 1 file descriptor (standard output) was pointing to **`/dev/pts/0`**
- the 2 file descriptor (standard error) was pointing to **`/dev/pts/0`**
- but the 0 file descriptor (standard input) was pointing to **`pipe:[148826]`**

Note that here file descriptor 0 pointing to pipe indicates **`grep`** process is taking input from pipe via the **`cpu-print`** process.

So the shell usually has

- standard input file descriptor (0) pointing to **`STDIN`**,
- standard output file descriptor (1) pointing to **`STDOUT`** and
- standard error file descriptor (2) pointing to **`STDERR`**.

If we specify any sort of redirection (eg., | to indicate input redirection), it sets these file descriptors to the specified descriptor and performs the redirection.

---

e. To know if a command is a shell inbuilt or executables executed by the bash shell, we can use type

command. If a command is inbuilt it prints a message saying the command is inbuilt, else it prints another message indicating that the command is not shell inbuilt.

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ type cd
cd is a shell builtin
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ type ls
ls is aliased to `ls --color=auto'
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ type history
history is a shell builtin
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ type ps
ps is /usr/bin/ps
```

From the above figure we can see that **`cd`** and **`history`** are shell inbuilt commands, while **`ls`** and **`ps`** are executables executed by the bash shell.

Question Number 4:

After compiling and running the **`memory1.c`** file, I have inspected its memory usage using the **`ps`** command to know its PID and **`more /proc/PID/status`** to know about its memory usage.

Output of the commands are as follows:

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ps aux | grep memory1
sameer      6014  0.0  0.0   6284  4856 pts/1    S+   13:24   0:00 ./memory1
sameer      6018  0.0  0.0   8908   724 pts/0    S+   13:24   0:00 grep --color=auto memory1
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ more /proc/6014/status
Name:   memory1
Umask:  0002
State:  S (sleeping)
Tgid:   6014
Ngid:   0
Pid:    6014
PPid:   4788
TracerPid:      0
Uid:    1000    1000    1000    1000
Gid:    1000    1000    1000    1000
FDSize: 256
Groups: 4 24 27 30 46 120 131 132 1000
NStgid: 6014
NSpid:  6014
NSpgid: 6014
NSsid:  4788
VmPeak:     6284 kB
VmSize:     6284 kB
VmLck:         0 kB
VmPin:         0 kB
VmHWM:      4856 kB
VmRSS:      4856 kB
RssAnon:             3984 kB
RssFile:              872 kB
RssShmem:               0 kB
VmData:      180 kB
VmStk:      3920 kB
VmExe:         4 kB
VmLib:      1652 kB
VmPTE:        48 kB
VmSwap:        0 kB
HugetlbPages:          0 kB
CoreDumping:    0
THP_enabled:    1
Threads:        1
SigQ:   0/55447
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
```

After compiling and running the **`memory2.c`** file, I have inspected its memory usage using the **`ps`** command to know its PID and **`more /proc/PID/status`** to know about its memory usage.

Output of the commands are as follows:

```
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ ps aux | grep memory2
sameer      6025  0.0  0.0   6288  4944 pts/1    S+   13:24   0:00 ./memory2
sameer      6029  0.0  0.0   8908   728 pts/0    S+   13:24   0:00 grep --color=auto memory2
sameer@sam-ubuntu:~/Documents/Semester 6/OS Lab/Assignment 1/intro-code$ more /proc/6025/status
Name:   memory2
Umask:  0002
State:  S (sleeping)
Tgid:   6025
Ngid:   0
Pid:    6025
PPid:   4788
TracerPid:      0
Uid:    1000    1000    1000    1000
Gid:    1000    1000    1000    1000
FDSize: 256
Groups: 4 24 27 30 46 120 131 132 1000
NStgid: 6025
NSpid:  6025
NSpgid: 6025
NSsid:  4788
VmPeak:     6288 kB
VmSize:     6288 kB
VmLck:         0 kB
VmPin:         0 kB
VmHWM:      4944 kB
VmRSS:      4944 kB
RssAnon:             3980 kB
RssFile:              964 kB
RssShmem:               0 kB
VmData:      180 kB
VmStk:      3924 kB
VmExe:         4 kB
VmLib:      1652 kB
VmPTE:        52 kB
VmSwap:        0 kB
HugetlbPages:          0 kB
CoreDumping:    0
THP_enabled:    1
Threads:        1
SigQ:   0/55447
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
```

From the above figures, we can see

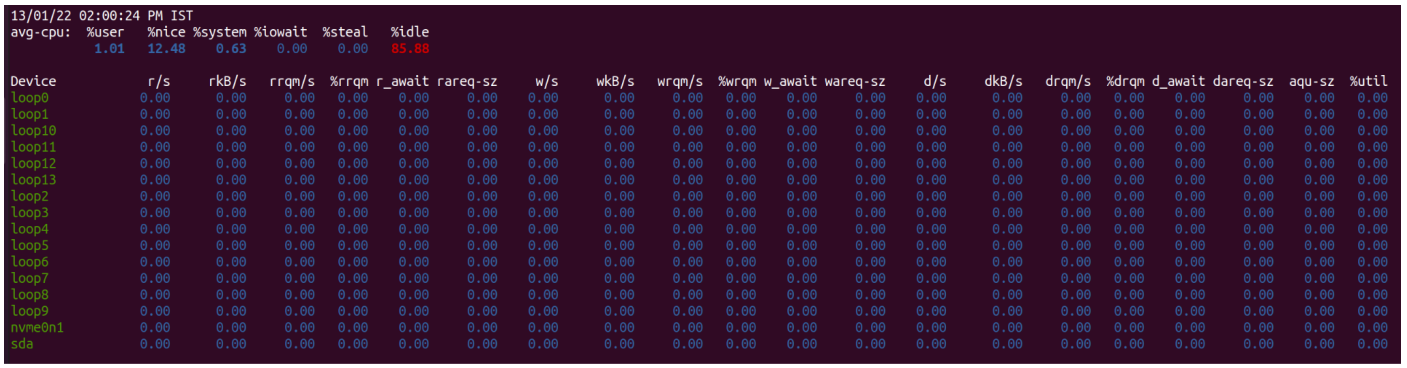
- Memory usage of **`memory1.c`** is **`VmSize: 6284 kB`** and **`VmRSS: 4856 kB`**
- Memory usage of **`memory2.c`** is **`VmSize: 6288 kB`** and **`VmRSS: 4944 kB`**

Here VmSize indicates the Virtual Memory usage and VmRSS indicates how much RAM the process is actually using. Both the processes have almost the same virtual memory size but memory2 process uses more RAM since it accesses the array elements whereas memory1 process doesn't access the array elements.

Question Number 5:

As indicated the files were made in the disk-files folder and then the **`disk.c`** and **`disk1.c`** files were compiled and run. To view the memory statistics **`iostat -xtc 1`** command was used. The below figure shows the memory statistics before running the **`disk.c`** file. As we can see from the below figure,

- the %system is 0.63, meaning only 0.63 % of the system was under utilization at this moment
- the rkB/s and %util columns of the nvme0n1 (disk) are both 0.00, indicating disk is not under utilization with read from disk at 0.00 kB per second.



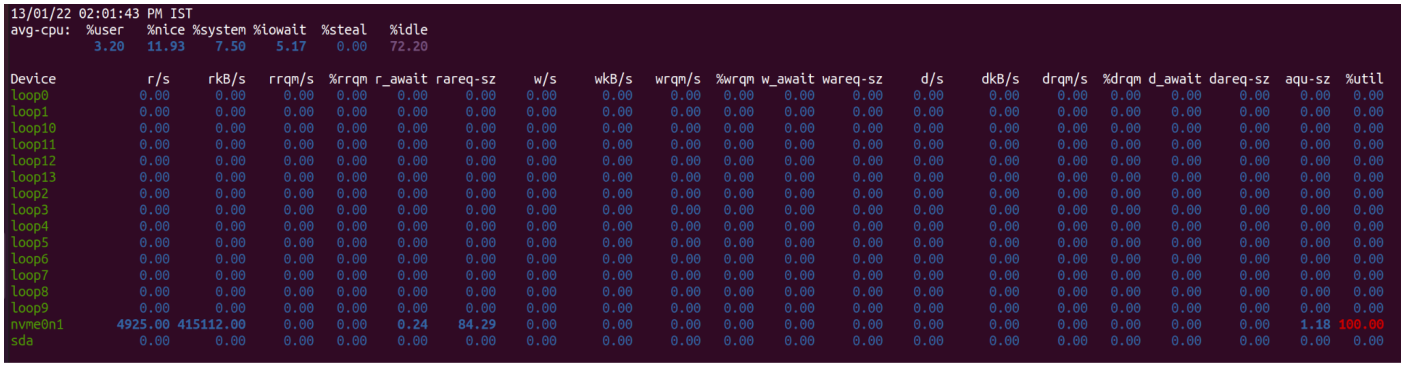
```
13/01/22 02:00:24 PM IST
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.01   12.48    0.63    0.00    0.00   85.88

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
loop0            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop1            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop10           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop11           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop12           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop13           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop2            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop3            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop4            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop5            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop6            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop7            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop8            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop9            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
nvme0n1          0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
sda              0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
```

The below figure shows the memory statistics while the **`disk.c`** file was running. As seen from the figure below,

- the system is at 7.50 % utilization, indicating that the disk.c file is running
- the %util of nvme0n1 is 100 % indicating that the disk is at 100 % utilization
- the rkB/s of the same show 415112.00 kB per second, meaning a lot of reading operations are going on at this moment.

This is true because the disk.c file is constantly reading various files from the disk.



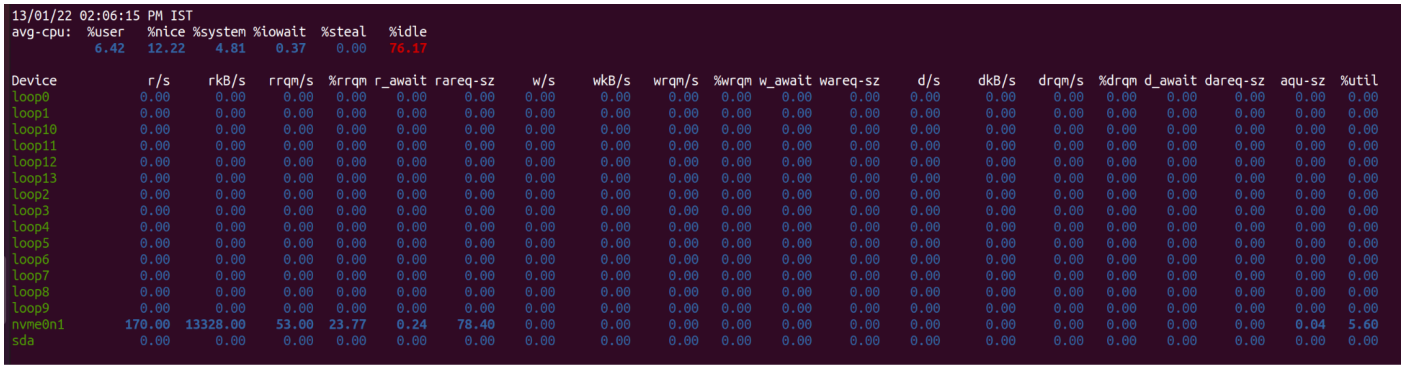
```
13/01/22 02:01:43 PM IST
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.20   11.93    7.50    5.17    0.00   72.20

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
loop0            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop1            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop10           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop11           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop12           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop13           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop2            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop3            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop4            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop5            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop6            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop7            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop8            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop9            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
nvme0n1       4925.00 415112.00     0.00   0.00    0.24    84.29    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    1.18 100.00
sda              0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
```

The below figure shows the memory statistics while the **`disk1.c`** file has just begun to run. As seen from the figure below,

- the system is at 4.81 % utilization, indicating that the disk1.c file is running
- the %util of nvme0n1 is 5.60 % indicating that the disk is under some utilization
- the rkB/s of the same show 13328.00 kB per second, meaning some reading operation is going on at this moment.

This is true because the **`disk1.c`** file, while started for the first time, reads one particular file from disk.



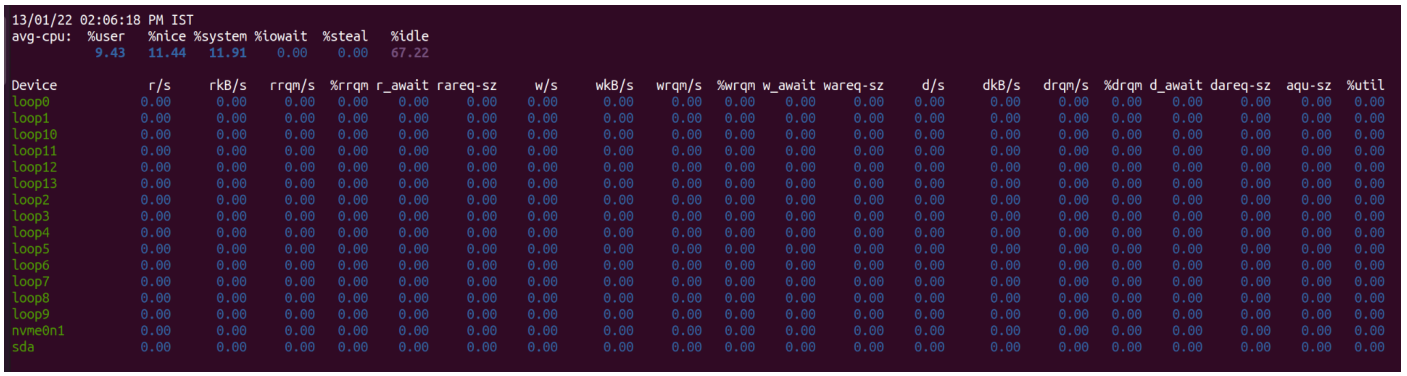
```
13/01/22 02:06:15 PM IST
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           6.42   12.22    4.81    0.37    0.00   76.17

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
loop0            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop1            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop10           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop11           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop12           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop13           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop2            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop3            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop4            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop5            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop6            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop7            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop8            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop9            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
nvme0n1        170.00  13328.00    53.00  23.77    0.24    78.40    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.04   5.60
sda              0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
```

The below figure shows the memory statistics while the **`disk1.c`** file has run for a few seconds. As seen from the figure below,

- the rkB/s and %util columns of the nvme0n1 are both 0.00, indicating disk is not under utilization with read from disk at 0.00 kB per second.

This is happening because the **`disk1.c`** file keeps reading the same file again and again, so once it reads the file from disk, it no longer requires to read the file from the disk, since it will be cached and will be read from cache.



```
13/01/22 02:06:18 PM IST
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           9.43   11.44   11.91    0.00    0.00   67.22

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   wrqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz  aqu-sz  %util
loop0            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop1            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop10           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop11           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop12           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop13           0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop2            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop3            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop4            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop5            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop6            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop7            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop8            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
loop9            0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
nvme0n1          0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
sda              0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00     0.00   0.00    0.00     0.00    0.00   0.00
```
