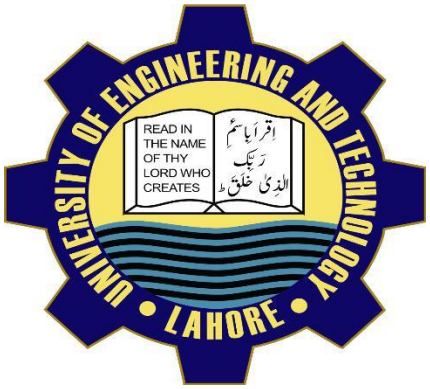
# Lab Report





| *Name* | *Registration No* | *Section* |
|---|---|---|
| *M. Bilal Khan* | 2017-EE-115 | D |
| *M. Yousaf Khan* | 2017-EE-89 | D |
| *Muhammad Saad* | 2017-EE-96 | D |


## EE431L Operating Systems
## Spring 2021
## Date Submitted: March 15, 2021


## Department of Electrical Engineering

## University of Engineering and Technology, Lahore

# Q No. 1

## Hardware Configuration using /proc filesystem

### Part (a)

```
xyz@ubuntu:~$ more /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 58
model name      : Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz
stepping        : 9
microcode       : 0x21
cpu MHz         : 2594.118
cache size      : 3072 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
```

*Figure 1*

**Processor:**

Provides each processor with an identifying number. If we have one processor it will display a 0. If we have more than one processor it will display number of processors counting the processors using zero notation. Since my computer has only 1 processor so it is showing 0.

**Cores:**

Provides the number of cores a processor has. Counting should include zero notation means counting will be started from 0 not from 1. Since my computer has 2 cores so it is showing 1.

## Verification from LSCPU:

```
xyz@ubuntu:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   45 bits physical, 48 bits virtual
CPU(s):                          2
On-line CPU(s) list:             0,1
Thread(s) per core:              1
Core(s) per socket:              1
Socket(s):                       2
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           58
Model name:                      Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz
Stepping:                        9
CPU MHz:                         2594.118
```

*Figure 2*

### Part (b) No. of Cores

As from above figure we can see that there are 2 sockets and each socket has 1 core. So total cores are 2.

No. of Sockets = 2

Cores per Socket = 1

Total No. of Cores = 2*1 = 2

### Part (c) No. of Processors

My machine has only 1 processor as we can see in Figure 1 it shows processor = 0 including zero notation.

### Part (d) Processor Frequency

As we can see in Figure 1 or Figure 2 CPU MHz is **2549.118 MHz** so this is my machines processor frequency.

### Part (e) Physical Memory

```
xyz@ubuntu:~$ more /proc/meminfo
MemTotal:        2005432 kB
MemFree:          135720 kB
MemAvailable:     783940 kB
Buffers:          102000 kB
Cached:           659588 kB
```

*Figure 3*

From Figure 3 Total available memory is **2005432 KB.**

### Part (f) Free Memory

From Figure 3 Total available memory is **135720 KB.**

### Part (g) Forks since reboot

```
xyz@ubuntu:~$ vmstat -f
         3970 forks
```

### Part (h) Context switches since reboot

## Q No. 2

### Monitoring status of Running Process

```
top - 09:05:46 up  2:49,  1 user,  load average: 0.97, 0.41, 0.15
Tasks: 281 total,   2 running, 279 sleeping,   0 stopped,   0 zombie
%Cpu(s): 50.5 us,  6.8 sy, 21.8 ni, 20.7 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   1958.4 total,    162.2 free,   1011.3 used,    784.9 buff/cache
MiB Swap:    923.3 total,    837.3 free,     85.9 used.    767.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   8314 xyz       20   0    2364    580    512 R  99.0   0.0   1:14.89 cpu
```

### Part (a) PID of CPU command

As we can see from above figure Process ID (PID) for CPU command is **8314.**

### Part (b) CPU and Memory consumption

CPU usage = **100%**

Memory Usage = **0%**

### Part (c) State of Process

Process is in **Running** state.

# Q No. 3

## How Linux Shell runs user commands

### Part (a) Finding PID using PS command

```
xyz@ubuntu:~$ ps -C "cpu-print" -o pid=
   9752
```

### Part (b) Finding ancestor PID of process with 9752 as PID.

```
xyz@ubuntu:~$ ps -f 9752
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
xyz         9752    3907 51 09:43 pts/0    R+     1:49 ./cpu-print
```

### Finding ancestor PID and then ancestor's ancestor PID until it reaches INIT process.

```
xyz@ubuntu:~$ ps -efj
UID          PID    PPID    PGID     SID  C STIME TTY          TIME CMD
root           1       0       1       1  0 06:16 ?        00:00:04 /sbin/init a
root           2       0       0       0  0 06:16 ?        00:00:00 [kthreadd]
```

PID = 9752 PPID = 3907

PID = 3907 PPID = 3899

PID = 3899 PPID = 1333

PID = 1333 PPID = 1

PID = 1 PPID = 0

### Part (c) Output Redirection

```
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ ./cpu-print > /tmp/tmp.txt &
[1] 9840
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ ls -l /proc/9840/fd
total 0
lrwx------ 1 xyz xyz 64 Mar 15 09:53 0 -> /dev/pts/2
l-wx------ 1 xyz xyz 64 Mar 15 09:53 1 -> /tmp/tmp.txt
lrwx------ 1 xyz xyz 64 Mar 15 09:53 2 -> /dev/pts/2
```

File descriptor 0 it holds the input address, file descriptor 1 it points towards the changed output destination which means output will be redirected and error file descriptor holds the input address.

### Part (d) Pipelining

```
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ ./cpu-print | grep hello &
[2] 9850
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ ls -l /proc/9850/fd
total 0
lr-x------ 1 xyz xyz 64 Mar 15 09:56 0 -> 'pipe:[112664]'
lrwx------ 1 xyz xyz 64 Mar 15 09:56 1 -> /dev/pts/2
lrwx------ 1 xyz xyz 64 Mar 15 09:56 2 -> /dev/pts/2
```

### Part (e) Types of commands

```
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ type cd
cd is a shell builtin
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ type ls
ls is aliased to `ls --color=auto'
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ type history
history is a shell builtin
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ type ps
ps is /usr/bin/ps
```

## Q No. 4

## Memory Allocation

```
Program : 'memory_1'
____________________


PID : 1986
Size of int : 4

Press Enter Key to exit.
```

```
xyz         1986  0.0  0.2   6284  4856 pts/0    S+   10:08   0:00 ./memory1
```

VSZ = 6284

RSS = 4856

RSS is the Resident Set Size and is used to show how much memory is allocated to that process and is in RAM. It does not include memory that is swapped out. It does include memory from shared libraries as long as the pages from those libraries are actually in memory. It does include all stack and heap memory.

VSZ is the Virtual Memory Size. It includes all memory that the process can access, including memory that is swapped out, memory that is allocated, but not used, and memory that is from shared libraries.

```
Program : 'memory_2'
____________________


PID : 2009
Size of int : 4

Press Enter Key to exit.
```

```
xyz         2009  0.0  0.2   6280  4868 pts/0    S+   10:12   0:00 ./memory2
```

VSZ = 6280

RSS = 4868

By looking into code, we can see that in memory2.c there is an extra computation in which array elements are being changed so memory2.c is using more RSS which means it is consuming more RAM than memory1.c because of extra computations.

## Q No. 5

### Disk Utilization

#### Disk.c

```
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ iostat
Linux 5.8.0-44-generic (ubuntu)         03/15/2021      _x86_64_        (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           4.36    0.06    8.66    2.29    0.00   84.62

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
loop0             0.05         0.37         0.00         0.00        359          0          0
loop1             0.06         1.12         0.00         0.00       1095          0          0
loop2             0.06         1.10         0.00         0.00       1073          0          0
loop3             0.05         0.37         0.00         0.00        362          0          0
loop4             0.48        14.12         0.00         0.00      13771          0          0
loop5             0.04         0.36         0.00         0.00        347          0          0
loop6             0.01         0.01         0.00         0.00         14          0          0
sda             143.45     10623.81      5601.13         0.00   10361822    5463009          0
scd0              0.02         0.00         0.00         0.00          3          0          0
```

- %user, is CPU utilization for the user,
- %nice, is the CPU utilization for apps with nice priority,
- %system, is the CPU being utilized by the system,
- %iowait, is the time percentage during which CPU was idle but there was an outstanding I/O request,

- %steal, percentage of time CPU was waiting as the hypervisor was working on another CPU,
- %idle, is the percentage of time system was idle with no outstanding request.

## Disk1.c

```
xyz@ubuntu:~/Downloads/OSLab/Lab1/intro-code$ iostat
Linux 5.8.0-44-generic (ubuntu)         03/15/2021      _x86_64_        (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           7.38    0.04   19.78    7.01    0.00   65.78

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
loop0             0.03         0.24         0.00         0.00        359          0          0
loop1             0.04         0.72         0.00         0.00       1095          0          0
loop2             0.04         0.71         0.00         0.00       1073          0          0
loop3             0.03         0.24         0.00         0.00        362          0          0
loop4             0.41        13.24         0.00         0.00      20029          0          0
loop5             0.03         0.23         0.00         0.00        347          0          0
loop6             0.01         0.01         0.00         0.00         14          0          0
sda             774.65     66860.12      3967.81         0.00  101164046    6003581          0
scd0              0.01         0.00         0.00         0.00          3          0          0
```

By comparison of above results we can clearly see that Disk.c file is utilizing less disk then Disk1.c and disk was idle for more time while executing Disk.c this time is more for Disk.c because iowait time is less which means that CPU was idle for less time when there was a pending I/O request.