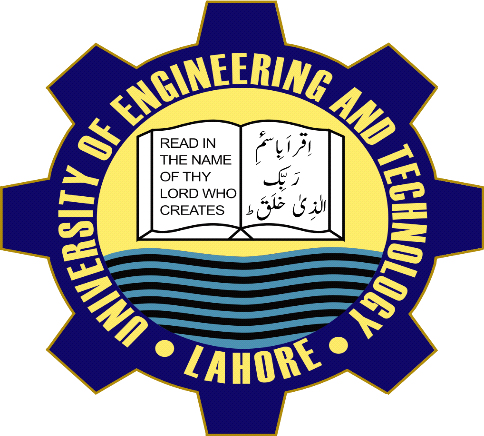
**Lab 1: Introduction to Operating System**



**Semester: Spring 2021**

**Submitted by:**

**2017-EE-161 Syeda Zainab Ali**

**2017-EE-165 Muhammad Ismail Khan**

**2017-EE-193 Anas Ali**

**Department of Electrical Engineering**

**University of Engineering and Technology, Lahore**

**Lab 1: Introduction to Operating System**

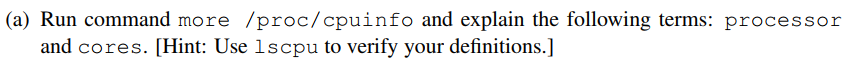
**Leaning Objectives:**

* To understand the basics of Linux terminal and how to use the terminal
* To understand and practice the basic Liunx terminal commands

**Exercise1:**

**Tasks:**

**Task A:**



**Solution:**

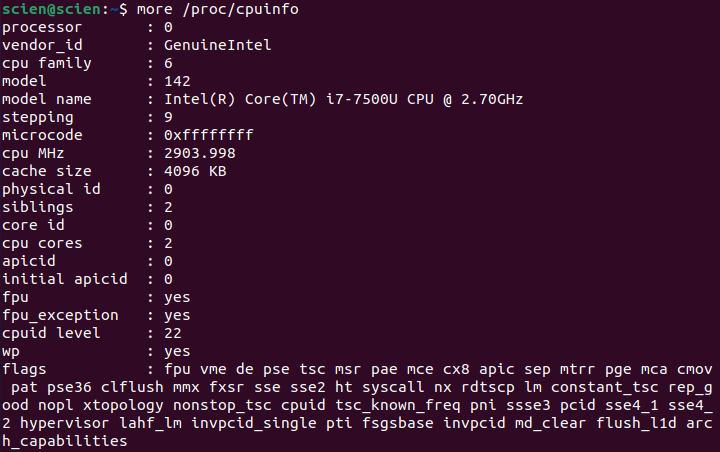
**command:** more /proc/cpuinfo 

Figure No. 1

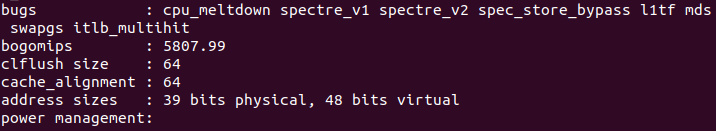


Figure No. 2

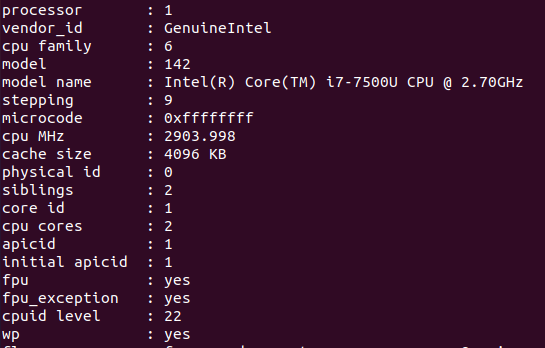


Figure No. 3

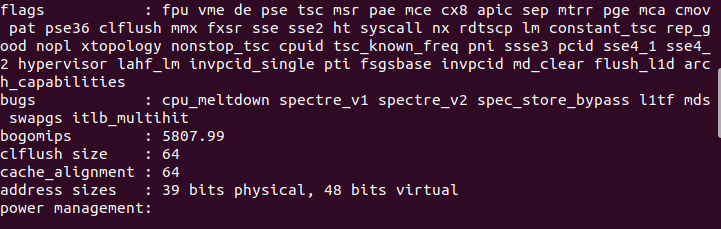


Figure No. 4

Command: lscpu

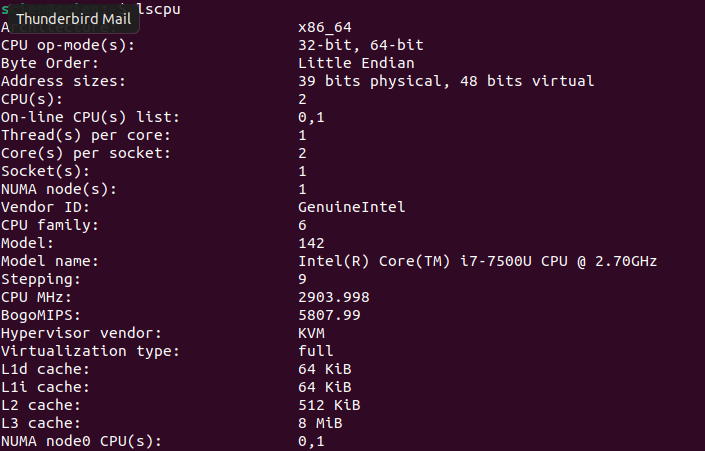


Figure No. 5

**Task B:**



**Solution:**

From Figure No. 1 and Figure No. 3, we can see that there are 2 cores. Also, Figure No. 5 verifies that there are 2 cores per socket, and there is one socket. So, there are two cores in total.

**Task C:**



**Solution:**

There are 2 processors.

**Task D:**



**Solution:**

The frequency is 2.70 GHz.

**Task E:**



**Solution:**

Command: more /prco/meminfo command

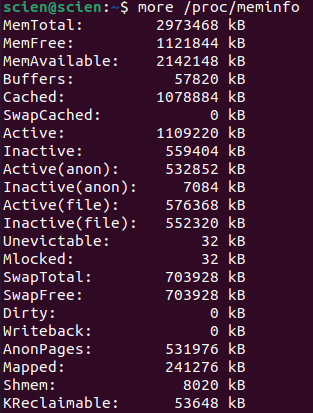


Figure No. 6

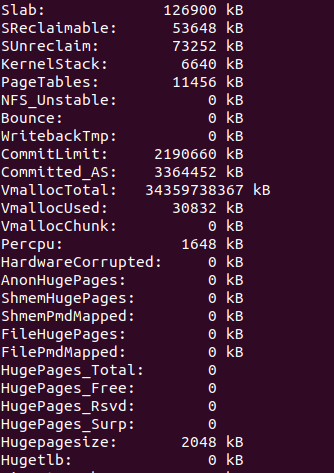


Figure No. 7



Figure No. 8

**Solution:**

According to Figure No. 6, 2973468 KBs of physical memory is there.

**Task F:**



**Solution:**

According to Figure No. 6, 1121844 KBs of memory is free.

**Task G:**



**Solution:**

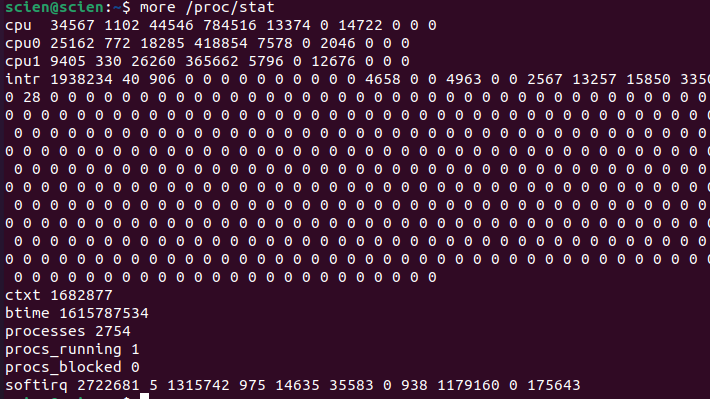


Figure No. 9

According to Figure No. 9, There have been 2754 forks(processes) since boot.

**Task H:**



**Solution:**

According to Figure No. 9, there have been 1682877 context switches performed by system since bootup.

**Exercise 2:**

**Tasks:**

**File Compilation Command:**



Figure No. 10

**Command: top**

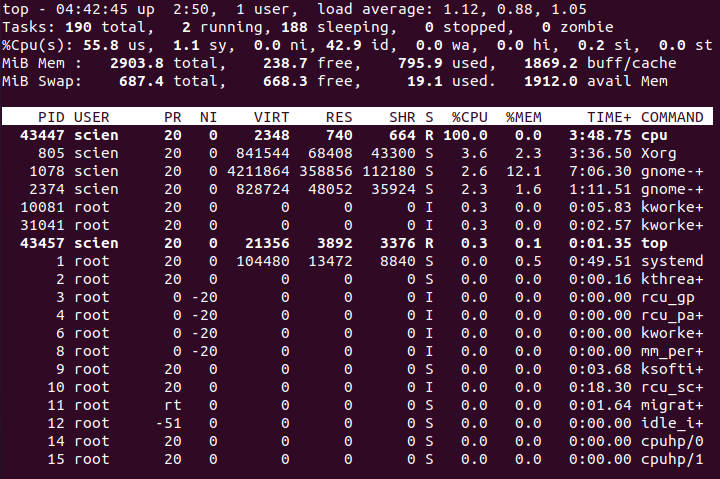


Figure No. 11

**Task A:**



**Solution:**

According to Figure No. 11, The corresponding PID in command column by name ‘cpu’ is 43447.

**Task B:**



**Solution:**

This process with PID=43447 is consuming almost 100% of CPU, and 0 % of memory.

**Task C:**

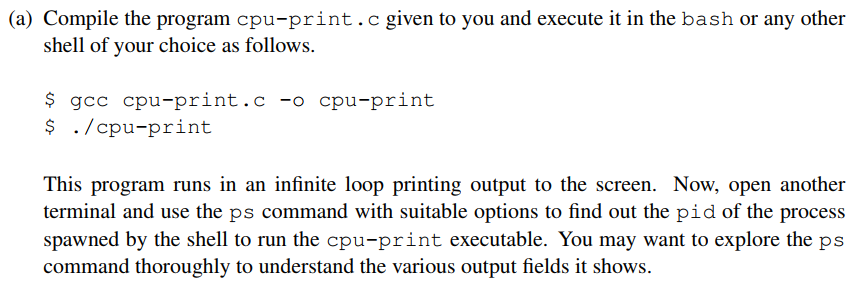


**Solution:**

The ‘S’ column gives state of the process. According to value at PID=43447, the process is in running state.

**Exercise 3:**

**Task A:**



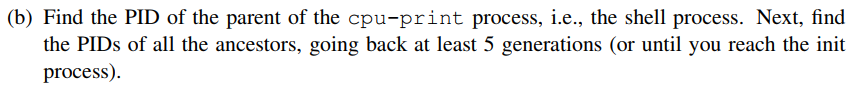
**Solution:**

To find PID, we run the following command: ps -e.



Figure No. 12

**Task B:**



**Solution:**

From the above command in Figure No. 12, we get PID:

cpu-print PID: 1586



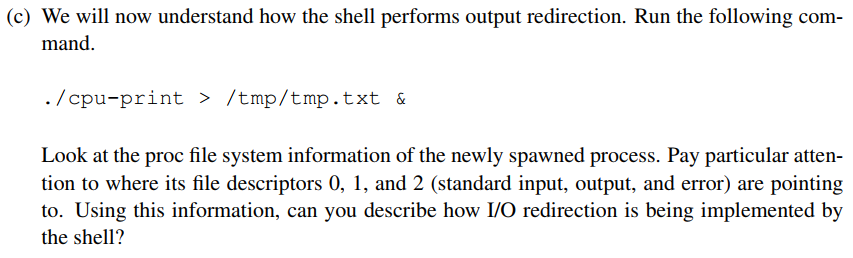
Figure No. 13

The command to find Ancestor tree for process with PID=1586:



Figure No. 14

**Task C:**



**Solution:**

****

Figure No. 15

The command in Figure No. 16 provides file descriptors.

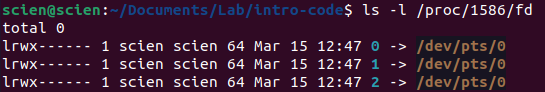
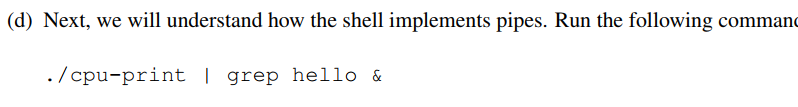
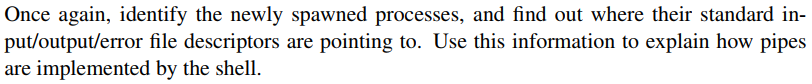


Figure No. 16

**Task D:**





**Solution:**

Here, the shell was started again, so the PID was changed to 4520.



Figure No. 17



Figure No. 18

Figure No. 19 shows the pipe file descriptors.

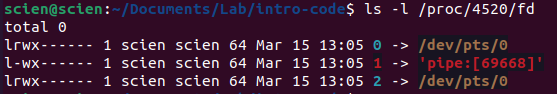
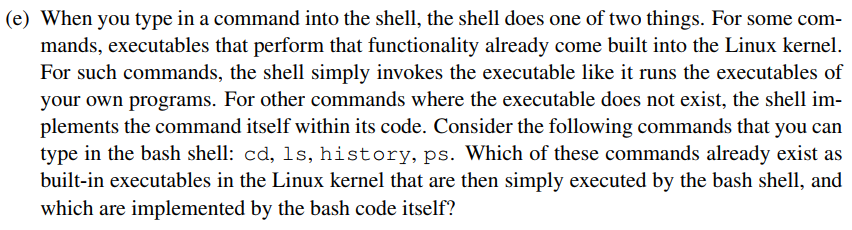


Figure No. 19

**Task E:**



**Solution:**

**Command: cd**



Figure No. 20

**Command: ls**

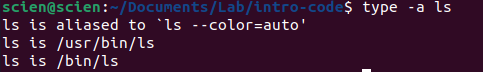


Figure No. 21

So, ls is executed by bash code itself

**Command: history**



Figure No. 22

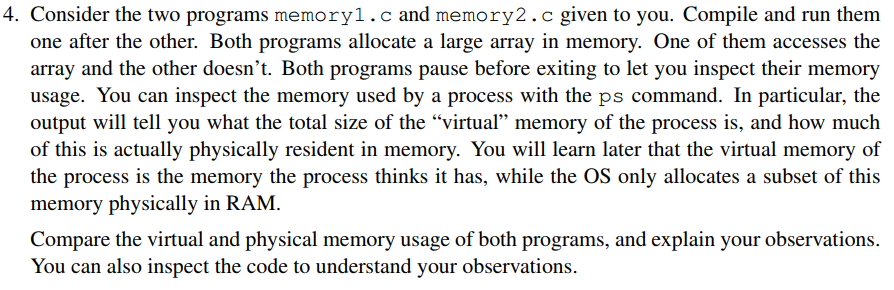
**Command: ps**



Figure No. 23

So, ps command is executed by bash code itself.

**Exercise 4:**



**Solution:**

Command:



Figure No. 24

This command provides the physical memory as well as virtual memory used by the processes.

**For Memory 1:**

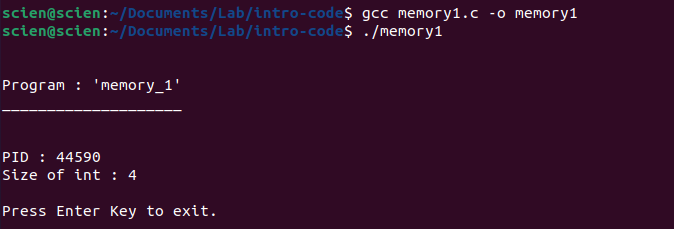


Figure No. 25



Figure No. 26

|  |  |  |
| --- | --- | --- |
| PID | Virtual Memory | Physical Memory |
| 44590 | 6286 bytes | 5156 bytes |

Table: For memory1.c

**For Memory 2:**

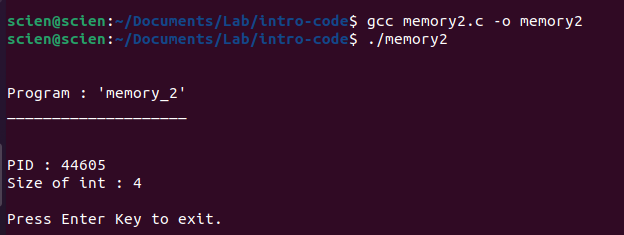


Figure No. 27



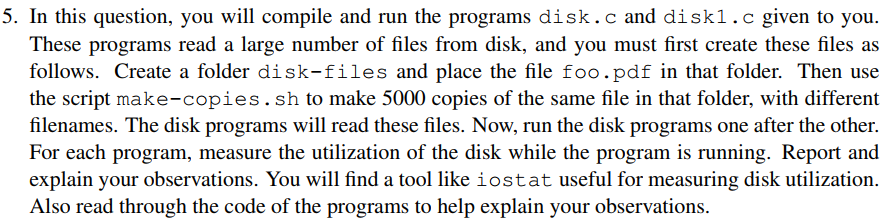
Figure No. 28

|  |  |  |
| --- | --- | --- |
| PID | Virtual Memory | Physical Memory |
| 44605 | 6264 bytes | 5060 bytes |

Table: For memory2.c

By observing the memory allocation, it is clear that the virtual program needed more memory than actual physical memory allocated. So, virtual memory is larger than physical memory in both cases.

**Exercise 5:**



**Solution:**

**To make new files:**



Figure No. 29

**For disk.c**

File Compilation Command:



Figure No. 30

Command: iostat

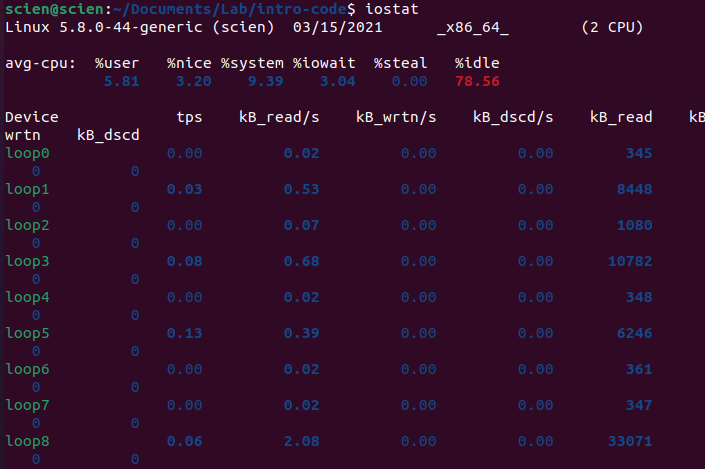


Figure No. 31

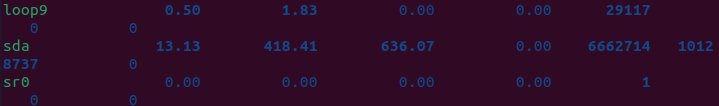


Figure No. 32

**For disk1.c**

File Compilation command:



Figure No. 33

iostat command:

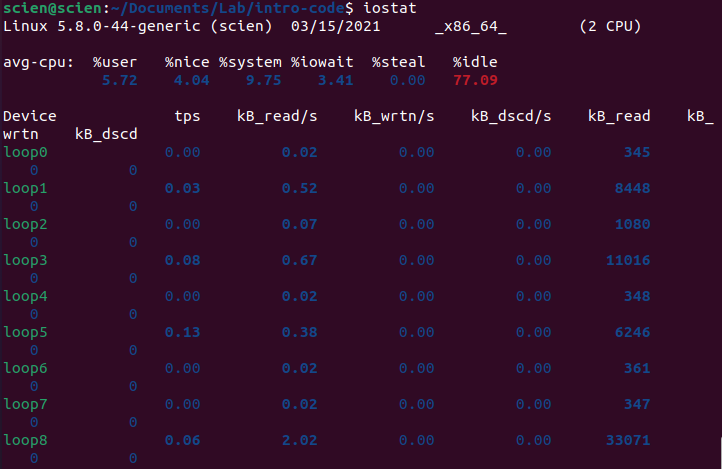


Figure No. 34

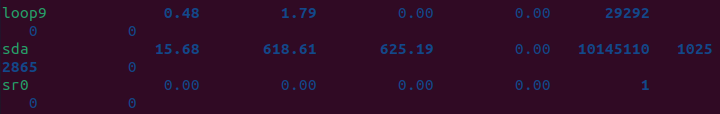


Figure No. 35

Now the comparison of transfer per second, and reading speed of disk.c and disk1.c is made:

|  |  |  |
| --- | --- | --- |
| Parameter | Disk.c | Disk1.c |
| tps | 13.13 | 15.68 |
| Data read | 418.41kb/s | 618.61kb/s |

Table. Comparison between different parameters

From tps, we deduce that cpu is more busier executing dik1.c than it was while executing disk.c

Also, data read speed for disk.c is less than data read speed of disk1.c. This can be the consequence of cpu allocating more resources to disk1.c than disk.c.