|  |
| --- |
|  |
| **Operating Systems Lab** |
| Lab 1 |

Sheharyar Chishti 2017-EE-121

Muhammad Irfan 2017-EE-124

Hamza Naseer 2017-EE-149



**Submitted by**

**Submitted to:** Nauman Ahmad

Department of Electrical Engineering

**University of Engineering and Technology Lahore**

Contents

[Lab 1: Introduction 3](#_Toc66742324)

[Question 1: /proc filesystem. 3](#_Toc66742325)

[a) Processor and Core: 3](#_Toc66742326)

[b) How many cores? 3](#_Toc66742327)

[c) How many processors? 3](#_Toc66742328)

[d) Frequency of each Processor. 3](#_Toc66742329)

[e) Total Physical Memory. 3](#_Toc66742330)

[f) Free Memory. 4](#_Toc66742331)

[g) Number of Forks since boot. 4](#_Toc66742332)

[h) Context Switches since bootup. 4](#_Toc66742333)

[Question 2: 4](#_Toc66742334)

[i) PID: 4](#_Toc66742335)

[j) CPU and Memory usage. 4](#_Toc66742336)

[k) Current State of Process. 4](#_Toc66742337)

[Question 3: 4](#_Toc66742338)

# Lab 1: Introduction

## Question 1: /proc filesystem.

### Processor and Core:

#### Processor:

A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions. Traditionally, the term "CPU" refers to a processor, more specifically to its processing unit and control unit (CU), distinguishing these core elements of a computer from external components such as main memory and I/O circuitry.

#### Core:

A processor core (or simply “core”) is an individual processor within a CPU. Processing performance of computers is increased by using multi-core processors, which essentially is plugging two or more individual processors (called cores in this sense) into one integrated circuit.

### How many cores?

There are 2 cores per processor/CPU.

### How many processors?

There are 2 processors, 0 and 1.

### Frequency of each Processor.

Frequency of each processor is **2904.004 MHZ.**

### Total Physical Memory.

The total available memory is **4803296 kB**. It was found using command, **cat/proc/meminfo**.

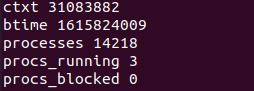
### Free Memory.

The freed memory is **1217500 kB**. It was found using command, **cat/proc/meminfo**.



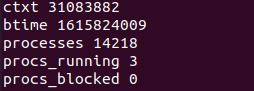
### Number of Forks since boot.

The number of processes that have been created since the last bootup is, processes **14218**. It was found using the command, **cat /proc/stat.**



### Context Switches since bootup.

The number of context switches that the system has performed, since the bootup are, **ctxt 31668818.** It was found using the command, **cat /proc/stat.**



## Question 2:



### PID:

The PID of the process running the cpu command is **14251**.

### CPU and Memory usage.

This process is using **99.7% of CPU** and **0.0% of memory**.

### Current State of Process.

The current state of the process is **Running**.

## Question 3:

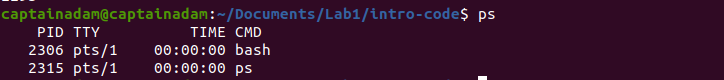
### PID:

The PID of process running cpu-print command is **2293**.



### PID of Shell and ancestors.

PID of Shell process is **2306**.



Ancestral tree is



PID of cpu-print:2293

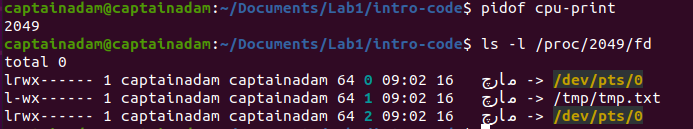
PID of bash:2306

PID of gnome-terminal: 2158

PID of systemd: 1248

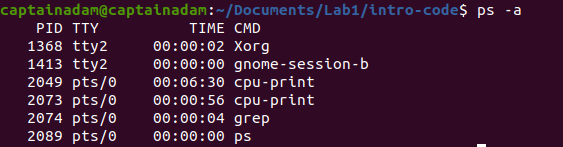
### ./cpu-print > /tmp/tmp.txt &

In this command when we redirect the output to a file, we see this by file descriptor 1 i.e. standard output, pointing to /tmp/tmp.txt. The output is being written to this file.

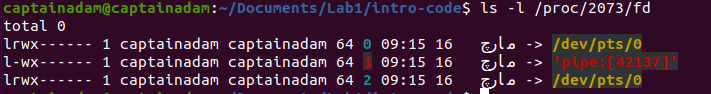


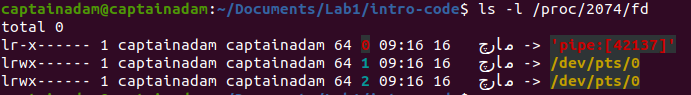
### ./cpu-print | grep hello &

The most recent processes spawned after running this command are cpu-print with pid 2073 and grep with pid 2074.



While looking at the file descriptors of cpu-print(pid 2073) we observer that the output descriptor is pointing to pip:[42137].



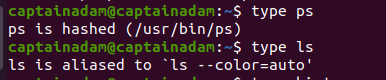
Similarly, looking at the file descriptors of grep(pid 2074) we observe that the input descriptor is pointing to pip:[42137].

Hence, the output of the right command/operand which is cpu-print in our case is being directed to right command/operand which is grep. This is how the pipes are implemented in shell.

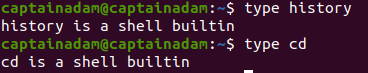
### cd, ls, history, ps

ps and ls already have their executables are simply executed when called. ‘type ps’

and ‘type ls’ can be used to locate executables.

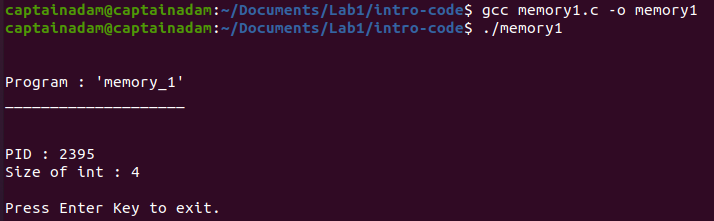


While cd and history are implemented by the bash code itself because these are shell built-in and can be checked by running the ‘type cd’ and ‘type history’ commands.

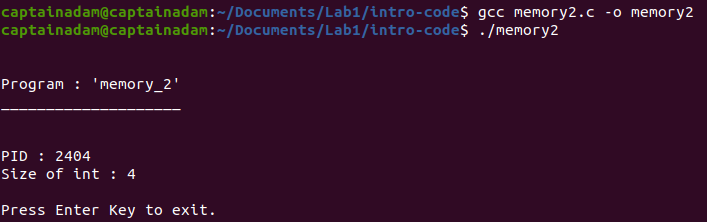


## Question 4:

### memory1.c



### memory2.c



## GitHub Link:

<https://github.com/MuhammadIrfan92/Operating_Systems_lab.git>