Lab 1 Report

Submitted by:
2017-EE-155 Fatima Nadeem
2017-EE-160 Seemara Ejaz
2017-EE-186 Azhak Anwar
2017-EE-189 Madni Sadiq

Submitted to: Dr. Nauman Ahmad

Question 1:
(a) A **Processor** is the main Integrated circuit in a computer, responsible for fetching executing and processing different processes, it has mounted memories, calculation units and buses on it. Whereas, a **Core** is the main brain, responsible for all the process executions and calculations. By running the lscpu command, we can easily verify the following information:
(b) Number of cores in device = 2
(c) Number of processors in device = 4
(d) Frequency of each processor:
CPU MHz: 892.362

Physical and Free memory can be looked up using grep MemTotal /proc/meminfo command.
(e) MemTotal: 8067204 kB
(f) MemFree: 3471664 kB

(g) Number of Forks can be found using vmstat -f command that gives the result:
12182 forks

(h) Number of context switches can also be found similiarly, using vmstat 1 command. It would give the context switches happening per second. Which, on average, were:
Context Switches: 1-2k per second

Question 2:
(a) To find the pid of the running process, top command was used that resulted in:
PID of CPU process: 12897

(b) The result of top command also displays info such as the process takes CPU= 98.7% mem= 0.0% while running

(c) The Status of the process is also visible in the resultant data from top command that says Status = R(Running)

Question 3:
(a) To get the PID of the running process, we run the ps aux | grep -i cpu-print command to get the following output:
fatima 13030 54.7 0.0 2496 720 pts/1 S+ 13:30 2:18 ./cpu-print
thus we find the PID = 13030

(b) Knowing the PID of current process, we run the following commands and get the respective results to find the PPID and iterate through generations:
fatima@fatima:~/Desktop/OSLab1$ ps -f 13030
UID PID PPID C STIME TTY STAT TIME CMD
fatima 13030 5239 55 13:30 pts/1 R+ 5:23 ./cpu-print

```
fatima@fatima:~/Desktop/OSLab1$ ps -f 5239
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
fatima      5239    4433  0 11:31 pts/1    Ss     0:00 bash
fatima@fatima:~/Desktop/OSLab1$ ps -f 4433
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
fatima      4433     945 14 11:18 ?        Rsl   20:56 /usr/libexec/gnome-terminal-server
fatima@fatima:~/Desktop/OSLab1$ ps -f 945
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
fatima       945       1  0 10:51 ?        Ss     0:01 /lib/systemd/systemd --user
fatima@fatima:~/Desktop/OSLab1$ ps -f 1
UID          PID    PPID  C STIME TTY      STAT   TIME CMD
root           1       0  0 10:50 ?        Ss     0:04 /sbin/init splash
```

(c) The command
./cpu-print > /tmp/tmp.txt &
pushes the output of cpu-print process, into the tmp.txt file, because of the “>”. The & symbol in the end of statement , produces a pointer to the push process and provides the free terminal while outputs gets pushed into tmp.txt file in background. Thus the Output from cpu-print is redirected to tmp.txt as Input.

(d) The command
./cpu-print | grep hello &
Creates a pipe implementation in the shell, it takes the output from cpu-print, links it to the grep command, such that the string “hello” is being searched in the outputs, if it is ever found, the line containing this string is returned.

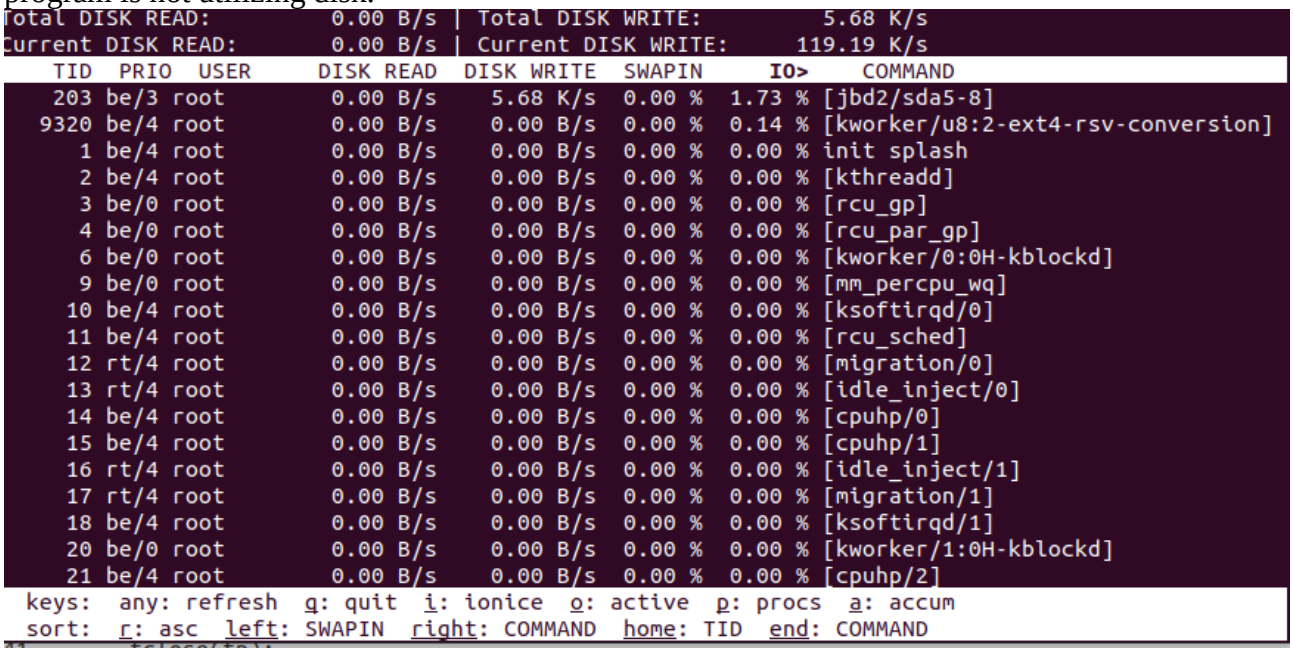
(e) To check if a command is built in bach or not, one can simply use the
type <command-name>
command. This command says a function is built-in if it is, or returns the parent directory if it is implemented by the bash code
cd and history are built-in commands while ls and ps are not.

Question 4:
We have used “ps –aux” command to find the virtual memory and physical memory for both memory1 and memory2 programs.
For memory1 VSZ = 6284
RSS = 4936
For memory2 VSZ = 6280
RSS = 4904
In both cases we see that Virtual memory required by the programs is greater while physical memory allotted is less than the Virtual memory required.
The difference between memory1 and memory2 is that in memory1 we are declaring an array in the memory while in memory2 we are updating elements of the array in addition to declaring it. We see that memory2 requires less memory than memory1 program.

Question 5:
For this task we have used “iotop” command to check disk utilization by both programs disk.c and disk1.c
disk.c

```
  TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN     IO>    COMMAND
 9308 be/4 madni    1347.80 K/s    0.00 B/s  0.00 % 92.35 % ./disk
```

disk1.c
This program is not appearing in the list shown by iotop command after running it. This means this program is not utilizing disk.



We have used following command to clear disk buffer cache
sudo sh –c “sync ; echo 3 > /proc/sys/vm/drop_caches