Mantri Krishna Sri Ipsit
180070032
180070032@iitb.ac.in

Assignment 1
CS 347: Operating Systems

2022/01/12

Part A: Understanding Linux Processes

1. (a) My machine has **6** CPU cores. Each core has **2** threads. Hence the command `more /proc/cpuinfo` shows **12** different processors.
   (b) Frequency of each CPU is **2**.
   (c) My system has a total of **16335724 kB**. Out of this, **6286200 kB** is free. This is found using the `more /proc/meminfo` command.
   (d) The total number of forks since bootup is **24541**. And the total number of context switches since bootup is **87459116**. This is found using the `more /proc/stat` command.

2. (a) The PID of the process running the `cpu` command is **24831**.
   (b) This process is consuming **100%** of CPU and **0%** of memory.
   (c) The current state of the process is **R**. It is in **running** state.

3. (a) The PID of the process spawned by the shell to run the `cpu-print` executable is **27038**. This is found using the `ps -C cpu-print` command.
   (b) The PID of the parent of the `cpu-print` process is **11168**. This is found using the `ps -f -C cpu-print` command. The PIDs of all ancestors are:

   ```
   systemd(1)---systemd(6069)---gnome-terminal-(11158)---bash(11168)---cpu-print(27038)
   ```

   (c) Using the `ls -l /proc/32077/fd` (here 32077 is the PID of the output redirection process) command, we get the following:

   ```
   total 0
   lrwx------ 1 ipsit ipsit 64 Jan 12 13:54 0 -> /dev/pts/1
   l-wx------ 1 ipsit ipsit 64 Jan 12 13:54 1 -> /tmp/tmp.txt
   lrwx------ 1 ipsit ipsit 64 Jan 12 13:54 2 -> /dev/pts/1
   ```

   Here we can see that the file descriptor for standard output is being pointed to `/tmp/tmp.txt` while the other descriptors are pointing towards a pseudo-terminal. Hence we can say that the I/O redirection happens in the following way: ***First, based on the redirection type (i, ¿, or 2¿), a file is opened (if it already exists then that will be used) with the given filename and then in the process, the file descriptor will point to the new file opened. In that way, whenever a system call is made to print to a screen or throw an error or take input, the data is sent to the file to which the file descriptor points to.***

4.