Proc Parser

Aim of the proc parser is to report the behavior of the Linux kernel.

The first part of the proc parser gives the following details:

● Processor type

● Kernel version

● The amount of memory configured into this computer

● Amount of time since the system was last booted

The second part of the proc parser gives the following details given the read rate and print rate as inputs

● The percentage of time the processor(s) spend in user mode, system mode, and the percentage of time the processor(s) are idle

● The amount and percentage of available (or free) memory

● The rate (number of sectors per second) of disk read/write in the system

● The rate (number per second) of context switches in the kernel

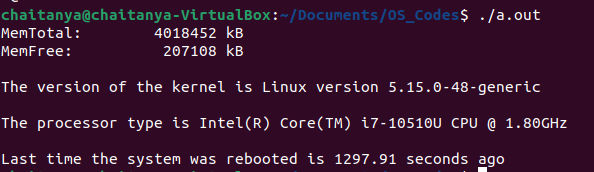
● The rate (number per second) of process creations in the system

To get all the following details, we will be using the /proc command.

The first part of the proc parser was achieved with the following commands:

* /proc/meminfo command was used to find the memory information of the computer.
* /proc/version command was used to find the version of the kernel.
* /proc/cpuinfo command was used to find the type of the processor used by the computer.
* /proc/uptime command was used to find the last reboot time of the system.

The output of the first part:



The second part of the proc parser was implemented by the following commands:

* The code runs an infinite loop until the user stops the program.
* This code returns the percentage of the time of processor spent in user, kernel, idle time of the system, rate of context switches, rate of processes created, the percentage of free memory available and the rate of reads and writes per second.
* To execute this part of the code, the user has to give the print rate and read rate as inputs.

For finding the time of processor spent in user, kernel modes and idle time of the system and rate of context switches and processes creations.

* /proc/stat file was used for the following task.
* To find the rate of context switches and processes, we take two arrays to store the context switches and processes at different times. The rate is given by subtracting the top most element of the array and the second top most element of the array.

Here, top is the pointer to the top most element in the array.

Rate of context switches: Context\_time[top] – Context\_time[top-1]

Rate of proc creation: proc\_time[top] – proc\_time[top-1]

* The percentage of the time spent in user mode, kernel mode and idle time are calculated as follows:

Time (User mode)/Total Time \* 100 = % of the time spent in user mode

Time (kernel mode)/Total Time \* 100 = % of the time spent in user mode

Time (idle)/Total Time \* 100 = % of the time spent in user mode

For finding the amount and percentage of free memory.

* /proc/meminfo file was used for the following task.
* To find the amount of free memory, the file directly gives us the memory available and the total memory.
* With this information, we can find the % of free memory by

Amount of Free memory / Total memory \* 100

For finding the number of disk read/writes.

* /proc/diskstats file was used for the following task.
* The 4th line and 8th line of the file gives the number of reads and writes.
* We use two arrays to find the reads and writes at different time instances.
* We append the read and writes to their respective arrays at different time instances.
* Read\_time[] and write\_time[]
* To find the rate of the reads and writes, we used the top pointer to find the reads and writes.

Rate of reads = modulus(read\_time[top] – read\_time[top-1])

Rate of writes = modulus(write\_time[top] – write\_time[top-1])

The output of the following code:

