**PRACTICAL-1**

* **Aim:**  Collect the following basic information about your machine using proc.

a. How many CPU cores does the machine have?

b. How much memory, and what fraction of it is free?

c. How many context switches has the system performed since bootup?

d. How many processes has it forked since bootup?

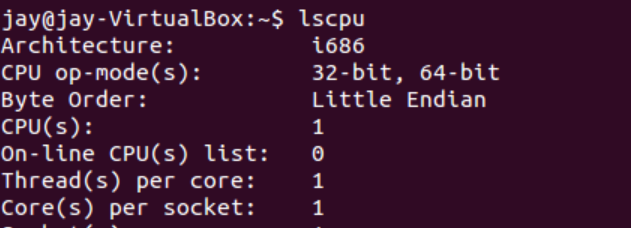
e. How many processors does your machine have?

f. What is the frequency of each processor?

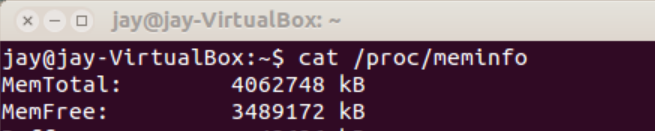
g. Find out various states of process at time of observation.

* **Program Code:**

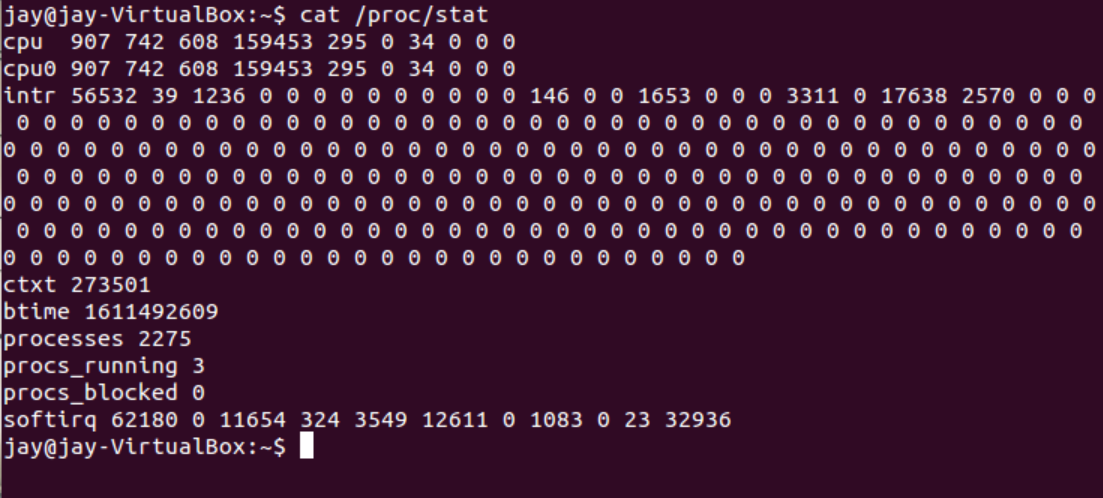
**a. How many CPU cores does the machine have?**

****

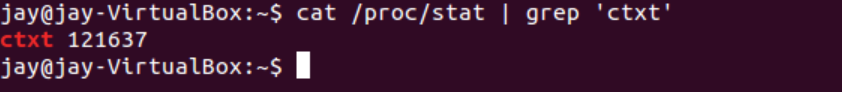
**b. How much memory, and what fraction of it is free?**

****

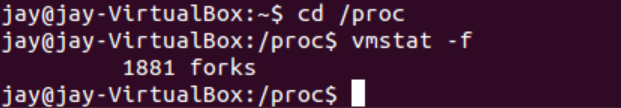
**c. How many context switches has the system performed since bootup?**

****

* Also you can use **‘grep’** command to fetch information.



**d. How many processes has it forked since bootup?**

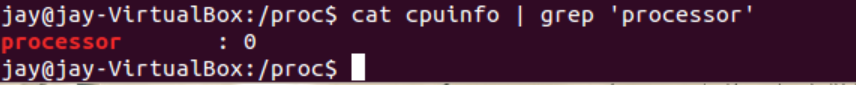


**e. How many processors does your machine have?**

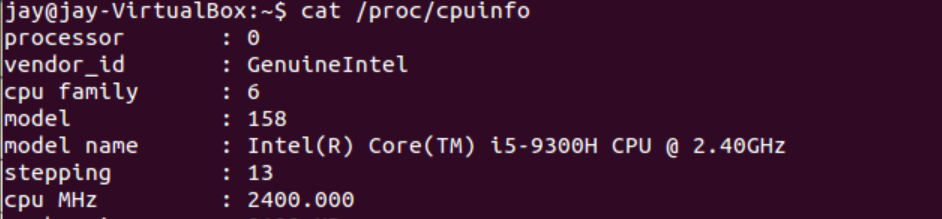
* You can check processor information in proc/cpuinfo directory.

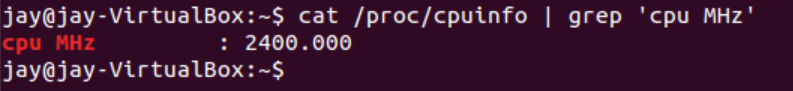
****

* Also you can use **‘grep’** command to fetch information.

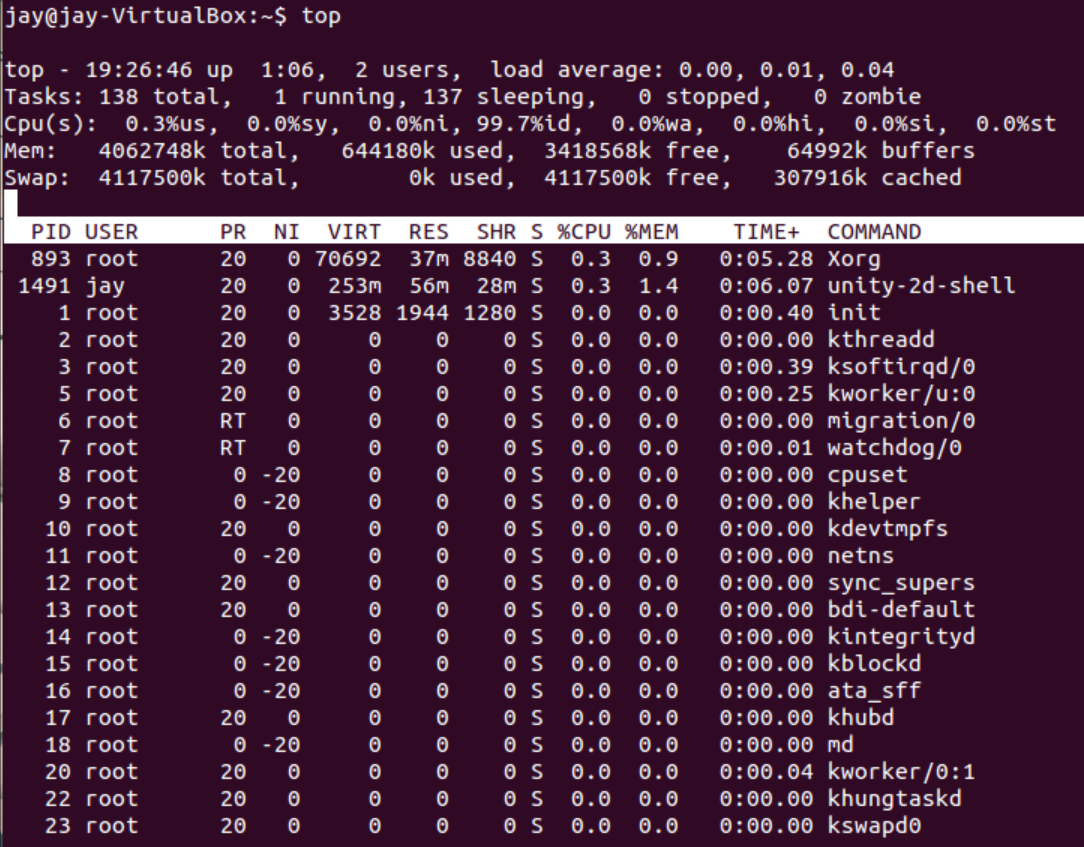


**f. What is the frequency of each processor?**

****



**g. Find out various states of process at time of observation.**



**Learning from practical:**

* In this practical we have studied about proc directory in linux.
* Also we have explore proc information using man command.

**Assignment-1**

1. **What is the time spent by a process in user mode and Kernel mode?**

**Ans :**

User Mode :

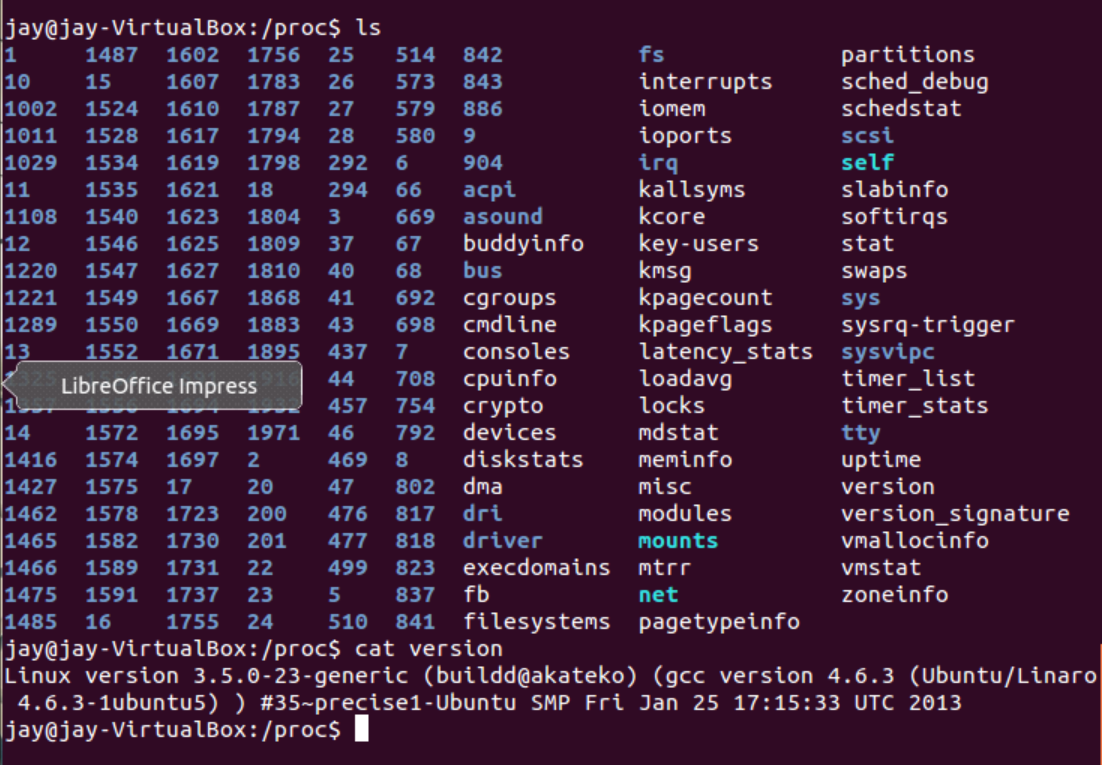
* When an application and program is executed , its initial state and operation mode are loaded on stack. At this point CPU starts executing the program in user mode.

Kernel Mode :

* When an interrupts or traps occurs at that time OS takes the control means stop the execution of program and serves the routine to interrupts or traps by it self , at this point CPU stats the execution in kernel mode.

1. **Check the version of linux kernel from proc directory.**

**Ans :**



**3. Mention the difference between MemFree and MemAvailable fields of the file**

**/proc/meminfo.**

Ans:-

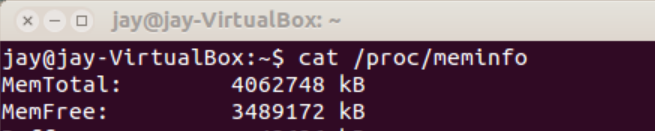
* **Memfree: Number of free memory**

Represents memory that is not yet in use by the system. Memused=memtotal-memfree is the memory that has been used.

* **Memavailable: The number of available memory**

The number of available memory for the application. Some of the memory in the system is used but can be recycled, such as cache/buffer, Slab have a part can be recycled, so memfree can not represent all available memory, this part of the recoverable memory plus Memfree is the system available memory, namely: **Memavailable = memfree+buffers+cached**, which is computed by the kernel using a specific algorithm, is an estimate**. The key difference between it and Memfree is that Memfree is said to be the system level, memavailable is said to be the application level.**

* **We can check Fee and Available memory in linux using belove command.**

****

**4. Explain the content of /proc/key-users**

**Ans :**

* This file lists various information for each user ID that has at least one key on the system. An example of the data that one might see in this file is the following:

**Fields:**  Uid usage nkeys/nikeys qnkeys/makeys *qnbytes*/*maxbytes*

**Values:**  0: 4 3/3 0/200 0/20000

* The fields shown in each line are as follows:
* **uid :**

The user ID.

* **usage :**

This is a kernel-internal usage count for the kernel structure used to record key users.

* **nkeys/nikeys :**

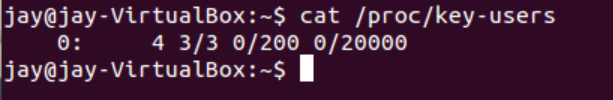
The total number of keys owned by the user, and the number of those keys that have been instantiated.

* **qnkeys/maxkeys :**

The number of keys owned by the user, and the maximum number of keys that the user may own.

* **qnbytes/maxbytes :**

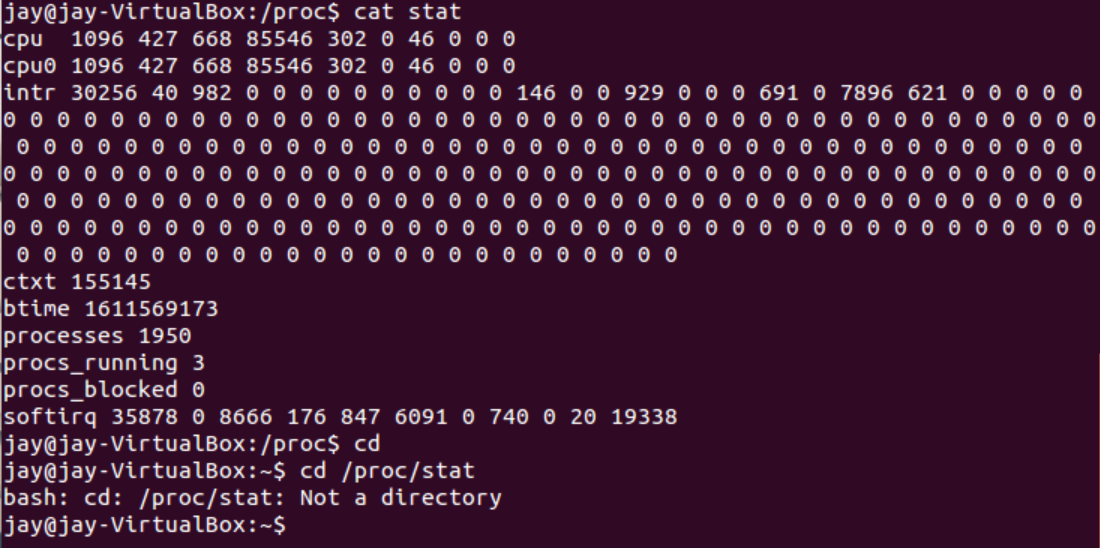
The number of bytes consumed in payloads of the keys owned by this user, and the upper limit on the number of bytes in key payloads for that user.



**5. How many processes are running & how many are blocked?**

**Ans :**

* You can check running and blocked process in proc directory.
* In proc you have to move in stat file where running and blocked process number is available.



* Also you can use **grep** command to fetch following information like below.

