Distributed Systems (CS304)

Assignment - 3

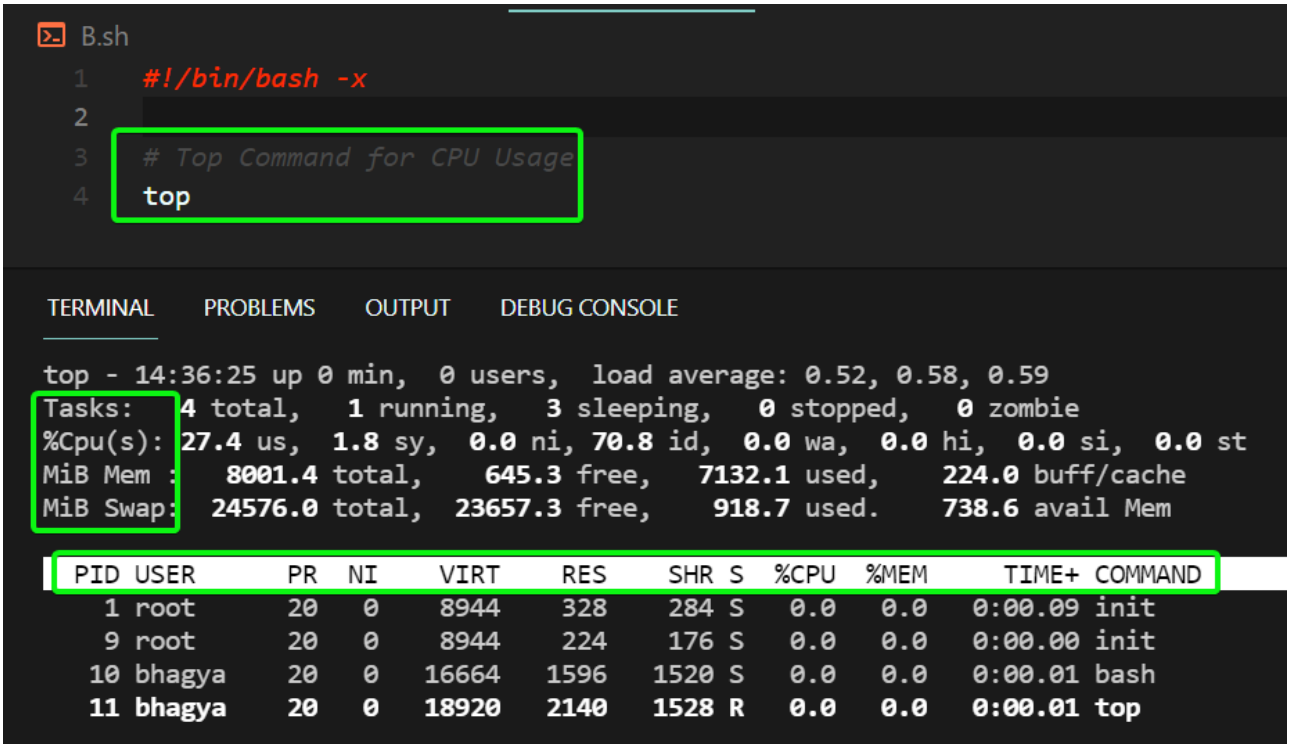
U19CS012

1. Calculate the **CPU Load** for your machine and **Identify the State** (under loaded or overloaded) of your machine. You have to find out the CPU usage of your computer using some Unix command.

Hint: With the help of **grep** Unix command, you can extract CPU usage. If CPU load is greater than 70% than it is overloaded, if it is between the range of 30% to 70% than it is moderately loaded and if it is less than 30% than it is lightly-loaded.

Unix Command for CPU Utilization - **top**

- ✓ The **top** program provides a dynamic real-time view of a running system.
- ✓ It can display **system summary information** as well as a list of tasks currently being managed by the Linux kernel.
- ✓ The top command monitors CPU utilization, process statistics, and memory utilization.



We will run it in Batch Mode and Read Single Byte.

B.sh

```
#!/bin/bash

# Top Command for CPU Usage => Batch Mode & Single Byte Reading
# We need Screenshot of Particular Timestmap
top -bn1
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

Got for Particular Timestamp

```
PS C:\Users\Admin\Desktop\DS_A3> bash B.sh
top - 14:44:11 up 0 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 29.1 us,  5.5 sy,  0.0 ni, 65.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   8001.4 total,    762.0 free,   7015.4 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23529.9 free,   1046.1 used.    855.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0    8944    332    284 S   0.0   0.0   0:00.09 init
    9 root      20   0    8944    228    180 S   0.0   0.0   0:00.00 init
   10 bhagya    20   0   16664   1600   1528 S   0.0   0.0   0:00.01 bash
   11 bhagya    20   0   18788   2004   1448 R   0.0   0.0   0:00.00 top
PS C:\Users\Admin\Desktop\DS_A3>
```

Now, For CPU Load, We require the Line with **%Cpu(s)**, So we use **grep** to filter our results.

B.sh

```
#!/bin/bash

# Top Command for CPU Usage => Batch Mode & Single Byte Reading
# We need Screenshot of Particular Timestmap
# Filter the Line with %Cpu(s)
top -bn1 | grep "%Cpu(s)"
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
PS C:\Users\Admin\Desktop\DS_A3> bash B.sh
%Cpu(s): 26.0 us,  1.6 sy,  0.0 ni, 72.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
PS C:\Users\Admin\Desktop\DS_A3>
```

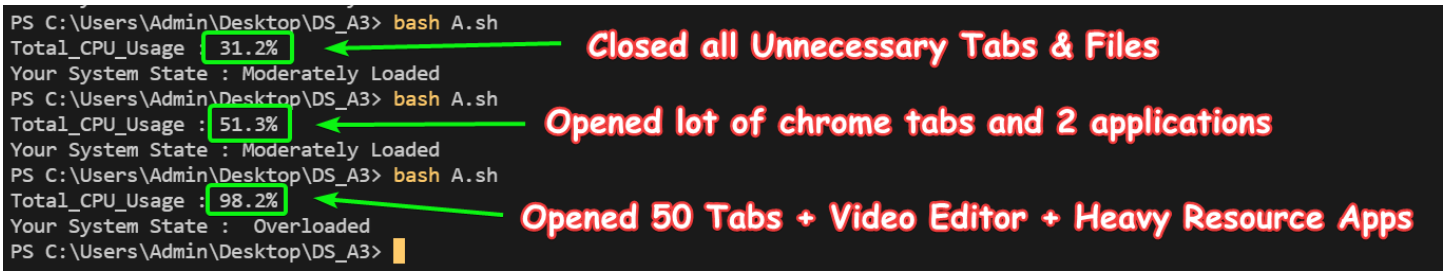
```
us: user cpu time (or) % CPU time spent in user space
sy: system cpu time (or) % CPU time spent in kernel space
ni: user nice cpu time (or) % CPU time spent on low priority processes
id: idle cpu time (or) % CPU time spent idle
wa: io wait cpu time (or) % CPU time spent in wait (on disk)
hi: hardware irq (or) % CPU time spent servicing/handling hardware interrupts
si: software irq (or) % CPU time spent servicing/handling software interrupts
st: steal time - - % CPU time in involuntary wait by virtual cpu while hypervisor
```

We need to store Output in **Array** and add the User and System CPU Time.

Code

```
#!/bin/bash
# [U19CS012 BHAGYA VINOD RANA]

# read -a => Array which will have the Output of top.. command
read -a arr <<< `top -bn1 | grep "%Cpu(s)"`

User_CPU_Time=${arr[1]}
# arr[2] => us, Which is of no use
System_CPU_Time=${arr[3]}

# bc is used for inline calculator in linux
Total_CPU_Time=`echo $User_CPU_Time + $System_CPU_Time | bc`

echo "Total_CPU_Usage : $Total_CPU_Time%"

# Set the Limits for Demarcation of States
upper_limit=70.0
lower_limit=30.0

# If the Total Usage is Greater than 70%, then its Overloaded
if [ $(echo "$Total_CPU_Time > $upper_limit" | bc) -eq 1 ]; then
    echo "Your System State :  Overloaded"
# Else if Total Usage is Less than 30%, then its Lightly Loaded
elif [ $(echo "$Total_CPU_Time < $lower_limit" | bc ) -eq 1 ]; then
    echo "Your System State :  Lightly-Loaded"
# Else Moderately Loaded
else
    echo "Your System State : Moderately Loaded"
fi
```

Output



SUBMITTED BY: U19CS012

BHAGYA VINOD RANA