

## BASH PROGRAMS

1. Write a script that will monitor the ram usage if the usage is more than 80% it should print an error message and if it is less than 80% it is normal.

```
ubuntu@ubuntu-VirtualBox:~$ vi ramusage.sh
ubuntu@ubuntu-VirtualBox:~$ bash ramusage.sh
The Ram usage is 28% and is normal.
ubuntu@ubuntu-VirtualBox:~$ cat ramusage.sh
#!/bin/bash

#for ram usage

total_memory=$( grep MemTotal /proc/meminfo | awk '{print $2}')
free_memory=$(grep MemAvailable /proc/meminfo | awk '{print $2}')

available_memory=$((total_memory - free_memory))

available_percentage=$(( 100 * available_memory / total_memory ))

if [ "$available_percentage" -gt 80 ];
then
    echo "Error, The Ram usage is ${available_percentage}%."
else
    echo "The Ram usage is ${available_percentage}% and is normal."
fi

#total_memory=$(free | grep Mem | awk '{print $1}')
#free_memory=$(free | grep Mem | awk '{print $3}')
ubuntu@ubuntu-VirtualBox:~$
```

2. Write a script that will get the process name as input if the process is running it should print the process id and the current memory usage of the current process in human readable format. If the process is not in running state print a message that the process is not running.

```
root@ubuntu-VirtualBox:/home/ubuntu# vi second.sh
root@ubuntu-VirtualBox:/home/ubuntu# vi second.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash second.sh
Enter the process name:
sudo
The process is running
Process ID: 2806
2807
13489
13490
Current memory usage: 0.1
0.0
0.1
0.0%
root@ubuntu-VirtualBox:/home/ubuntu# cat second.sh

#!/bin/bash

echo "Enter the process name:"
read process

if [ $(ps -e | grep "$process" | wc -l) -eq 0 ]; then
    echo "The process is not running"
else
    echo "The process is running"
    process_id=$(ps -e | grep "$process" | awk '{print $1}')
    memory_usage=$(ps -eo pid,%mem | grep "$process_id" | awk '{print $2}')
    echo "Process ID: $process_id"
    echo "Current memory usage: $memory_usage%"
fi
```

3. Write a script that will print the cpu load in the last 1min 5 min 15 min ,current CPU utilization in % current memoryin %,disk utilization in %.

```
root@ubuntu-VirtualBox:/home/ubuntu# vi cpuload.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash cpuload.sh
CPU load calculation for 1 min, 5 min, 15 min :
0.00 0.01 0.00
Memory utilization
846Mi used ,3.8Gi total and percentage used is (22263.2 %)
Disk Utilization
1.7M used ,390M total , and used percentage is 1%
11G used ,17G total , and used percentage is 40%
0 used ,2.0G total , and used percentage is 0%
4.0K used ,5.0M total , and used percentage is 1%
6.1M used ,506M total , and used percentage is 2%
120K used ,391M total , and used percentage is 1%
62M used ,0 total , and used percentage is 100%
62M used ,0 total , and used percentage is 100%
Total CPU utilization: 2.7%
root@ubuntu-VirtualBox:/home/ubuntu# cat cpuload.sh
#!/bin/bash

#cpu load for 1,5,15 mins

echo "CPU load calculation for 1 min, 5 min, 15 min : "

awk '{print $1, $2, $3}' /proc/loadavg

# memory utilization
echo "Memory utilization"
free -h | awk '/Mem/ {print $3 " used ," $2 " total " "and percentage used is (" $3/$2*100 " %)" }'

# disk utilization
echo "Disk Utilization"

df -h | grep / | awk '{print $3 " used ," $4 " total ," " and used percentage is " $5}'

echo "Total CPU utilization: $(top -b -n 1 | awk '/Cpu/ {print $2}')%"
```

4. Script should get the user name as the input it should create the user if the user is not exist on the system, if the user is already exist check whether password is already set for the particular user.

if the user exists and password is not set go ahead and display a message "set the password".

```
root@ubuntu-VirtualBox:/home/ubuntu
root@ubuntu-VirtualBox:/home/ubuntu# vi password.sh
root@ubuntu-VirtualBox:/home/ubuntu# bash password.sh
Enter the username:giduturi
User 'giduturi' has been created.
Please set the password for the new user 'giduturi'.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
root@ubuntu-VirtualBox:/home/ubuntu# cat password.sh
#!/bin/bash

#taking username from the user
#read -p " Please Enter your username : " username
#if grep -q "$username" /etc/passwd; then

    #echo "User '$username' already exists."

    #password=$(grep "^$username:" /etc/shadow | awk -F: '{print $2}')

    #if [ "$password" = "!" ]; then
        #echo "Password is not set for the '$username'. Set password."
    #else
        #echo "Password is set for the user '$username'."
    #fi
#else
```

```
        #else
            #echo "Password is set for the user '$username'."
        #fi
    #else
        #echo "User '$username' doesnt exist."
        #useradd "$username"
        #echo "User '$username' successfully created."
    #fi

#!/bin/bash
read -p "Enter the username:" username
if id "$username" &>/dev/null; then
    echo "User '$username' already exists."
    if passwd -S "$username" | grep -q "NP"; then
        echo "Password is not set for the user '$username'. Please set the password."

    else
        echo "Password for the user '$username' is already set."

    fi
else
    useradd "$username"
    echo "User '$username' has been created."
    echo "Please set the password for the new user '$username'."
    passwd "$username"
fi

root@ubuntu-VirtualBox:/home/ubuntu#
```

5. write a script to got to a directory, in the directory it should find specific file, if the file size is more than 100mb it should tar file del file and create empty file with same name.

```
root@ubuntu-VirtualBox: /home/ubuntu
root@ubuntu-VirtualBox:/home/ubuntu# bash directory.sh
Enter the path of the directory
/home/ubuntu
Enter the file name
test.txt
directory.sh: line 11: size: command not found
directory.sh: line 15: [: -gt: unary operator expected
The file size is less than 100mb
root@ubuntu-VirtualBox:/home/ubuntu# cat test.txt
testing permission
root@ubuntu-VirtualBox:/home/ubuntu# cat directory.sh
#!/bin/bash

echo "Enter the path of the directory"

read path

echo "Enter the file name"

read filename

size = $(du -s $path | awk '{print $1}')

threshold=10000

if [ $size -gt $threshold ]; then
    echo "The file size is greater than 100mb"
    tar -cvzf $filename.tar.gz $filename
    rm $filename
    touch $filename
else
    echo "The file size is less than 100mb"
fi
root@ubuntu-VirtualBox:/home/ubuntu#
```