

# Shell Programming – Set1

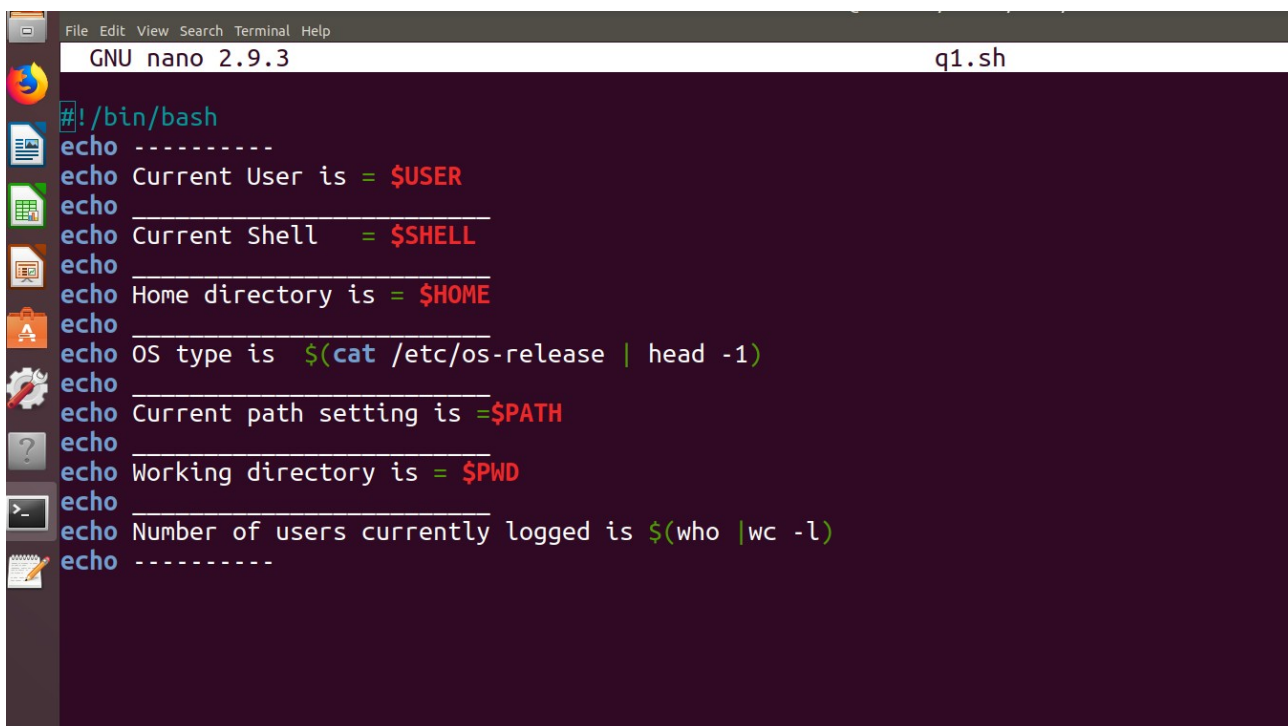
Experiment 4

19/03/2019

Q1. Write a shell script to show various system configuration like

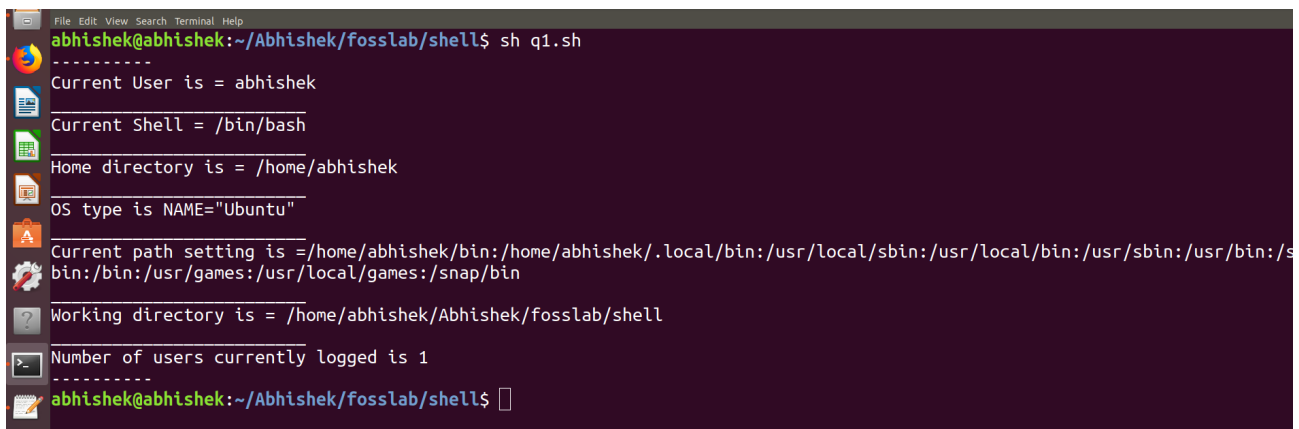
1. Currently logged user and his login name
2. Your current shell
3. Your home directory
4. Your operating system type
5. Your current path setting
6. Your current working directory
7. Number of users currently logged in.

## shell script



```
File Edit View Search Terminal Help
GNU nano 2.9.3 q1.sh
#!/bin/bash
echo -----
echo Current User is = $USER
echo -----
echo Current Shell = $SHELL
echo -----
echo Home directory is = $HOME
echo -----
echo OS type is $(cat /etc/os-release | head -1)
echo -----
echo Current path setting is =$PATH
echo -----
echo Working directory is = $PWD
echo -----
echo Number of users currently logged is $(who | wc -l)
echo -----
```

## Output



```
File Edit View Search Terminal Help
abhishek@abhishek:~/Abhishek/fossilab/shell$ sh q1.sh
-----
Current User is = abhishek
Current Shell = /bin/bash
Home directory is = /home/abhishek
OS type is NAME="Ubuntu"
Current path setting is =/home/abhishek/bin:/home/abhishek/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/s
bin:/bin:/usr/games:/usr/local/games:/snap/bin
Working directory is = /home/abhishek/Abhishek/fossilab/shell
Number of users currently logged is 1
-----
abhishek@abhishek:~/Abhishek/fossilab/shell$
```

2. Write a shell script to show various system configurations like
1. your OS and version, release number, kernel version
  2. all available shells
  3. computer CPU information like processor type, speed etc
  4. memory information
  5. hard disk information like size of hard-disk, cache, model etc
  6. File system (Mounted).

## Shellscript

```
GNU nano 2.9.3 q2.sh
#!/bin/bash
echo -----
echo OS details are
cat /etc/os-release | head -5
echo
echo kernal $(uname -r)
echo
echo release number is $(cat /etc/*release | sed -n 2p)
echo
echo Available shells are
cat /etc/shells
echo
echo computer cpu informations are
cat /proc/cpuinfo | head -7
echo
echo memory informations are
free -m
echo
echo Hard disk informations are
cat /proc/meminfo
df -H
echo
echo File system mounted $(mount | column -t)
echo -----
```

## output

```
File system mounted
sysfs                                on /sys                                type sysfs                                (rw,nosuid,nodev
,noexec,relatime)
proc                                on /proc                                type proc                                (rw,nosuid,nodev
,noexec,relatime)
udev                                on /dev                                type devtmpfs                                (rw,nosuid,relat
ime,size=4016000k,nr_inodes=1004000,mode=755)
devpts                                on /dev/pts                                type devpts                                (rw,nosuid,noexe
c,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs                                on /run                                type tmpfs                                (rw,nosuid,noexe
c,relatime,size=807524k,mode=755)
/dev/sda4                                on /                                type ext4                                (rw,relatime,err
ors=remount-ro,data=ordered)
securityfs                            on /sys/kernel/security                type securityfs                            (rw,nosuid,nodev
,noexec,relatime)
tmpfs                                on /dev/shm                            type tmpfs                                (rw,nosuid,nodev
)
tmpfs                                on /run/lock                            type tmpfs                                (rw,nosuid,nodev
,noexec,relatime,size=5120k)
tmpfs                                on /sys/fs/cgroup                        type tmpfs                                (ro,nosuid,nodev
,noexec,mode=755)
cgroup                                on /sys/fs/cgroup/unified                type cgroup2                                (rw,nosuid,nodev
,noexec,relatime,nsdelegate)
cgroup                                on /sys/fs/cgroup/systemd                type cgroup                                (rw,nosuid,nodev
)
/bin/pasn
/bin/rbash

computer cpu informations are
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 142
model name    : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
stepping      : 10
microcode     : 0x96

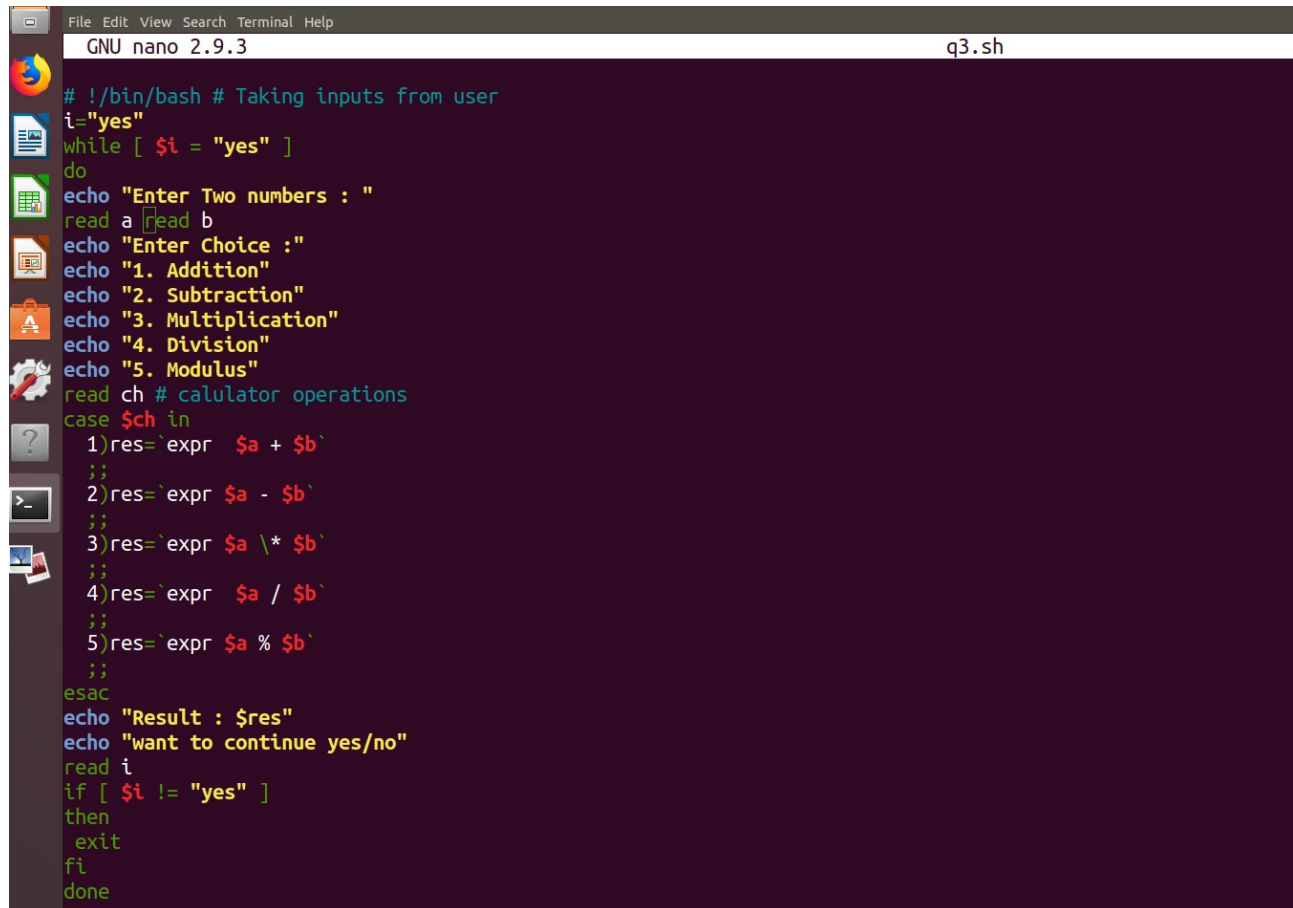
memory informations are
              total      used      free      shared  buff/cache  available
Mem:          7885       1170       5575        338       1140       6122
Swap:          0          0          0

Hard disk informations are
MemTotal:      8075228 kB
MemFree:       5708908 kB
MemAvailable:  6269516 kB
Buffers:        62992 kB
Cached:        1031732 kB
SwapCached:      0 kB
Active:        1368364 kB
Inactive:       719928 kB
Active(anon):   994932 kB
Inactive(anon): 344812 kB
Active(file):   373432 kB
Inactive(file): 375116 kB
Unevictable:    32 kB
Mlocked:       32 kB
SwapTotal:      0 kB
```

3. Write a shell script to implement a menu driven calculator with following functions

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus

## Shellscript



```
File Edit View Search Terminal Help
GNU nano 2.9.3 q3.sh

# !/bin/bash # Taking inputs from user
i="yes"
while [ $i = "yes" ]
do
echo "Enter Two numbers : "
read a read b
echo "Enter Choice : "
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
echo "5. Modulus"
read ch # calculator operations
case $ch in
1)res=`expr $a + $b`
;;
2)res=`expr $a - $b`
;;
3)res=`expr $a \* $b`
;;
4)res=`expr $a / $b`
;;
5)res=`expr $a % $b`
;;
esac
echo "Result : $res"
echo "want to continue yes/no"
read i
if [ $i != "yes" ]
then
exit
fi
done
```

## output

```
s1702@linux-server:~/fossilab/Shell_script$ sh q3.sh
Enter Two numbers :
5
3
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
5
Result : 2
want to continue yes/no
yes
Enter Two numbers :
6
2
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
4
Result : 3
want to continue yes/no
yes
Enter Two numbers :
5
7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
3
Result : 15
```

4. Write a script called addnames that is to be called as follows ./addnames ulist username

Here ulist is the name of the file that contains list of user names and username is a particular student's username. The script should

1. check that the correct number of arguments was received and print a message, in case the number of arguments is incorrect
2. check whether the ulist file exists and print an error message if it does not
3. check whether the username already exists in the file. If the username exists, print a message stating that the name already exists. Otherwise, add the username to the end of the list.

## Shellscript

```
GNU nano 2.9.3 addnames.sh
#!/bin/bash
if [ "$#" -eq 2 ]; then
    echo "You have entered arguments correctly"
    if [ -f "$1" ]
    then
        echo $1 " found."
        if grep -Fxq $2 $1
        then
            echo "The name already Exist"
        else
            echo "The name was absent Dont worry we will add"
            echo $2 >> $1
            echo "New modified content is"
            cat $1
        fi
    else
        echo "error" $1 "not found."
    fi
else
    echo "You must enter two command line arguments"
fi
```

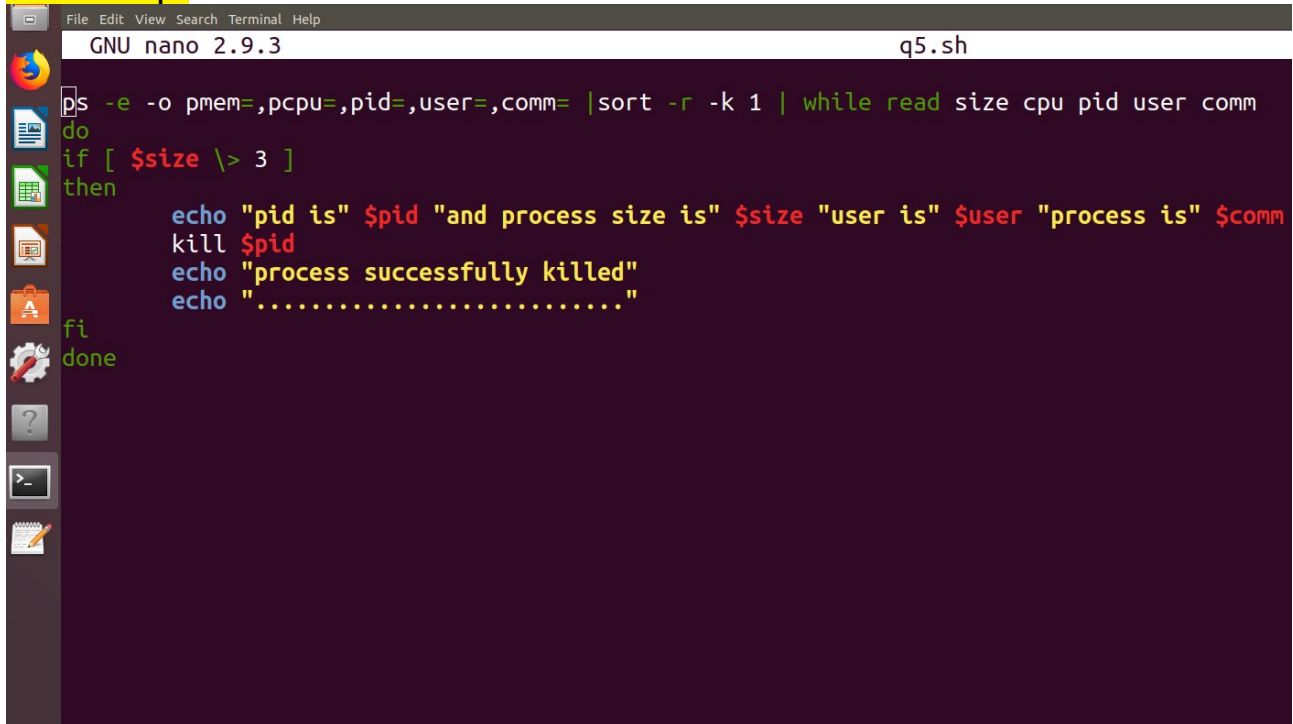
## output

```
File Edit View Search Terminal Help
abhishek@abhishek:~/Abhishek/fosslab/shell$ cat ulist
arther
mary
lianda
abhishek@abhishek:~/Abhishek/fosslab/shell$ sh addnames.sh 1
You must enter two command line arguments
abhishek@abhishek:~/Abhishek/fosslab/shell$ sh addnames.sh wrongfile 2
You have entered arguments correctly
error wrongfile not found.
abhishek@abhishek:~/Abhishek/fosslab/shell$ sh addnames.sh ulist mary
You have entered arguments correctly
ulist found.
The name already Exist
abhishek@abhishek:~/Abhishek/fosslab/shell$ sh addnames.sh ulist abhishek
You have entered arguments correctly
ulist found.
The name was absent Dont worry we will add
New modified content is
arther
mary
lianda
abhishek
abhishek@abhishek:~/Abhishek/fosslab/shell$
```



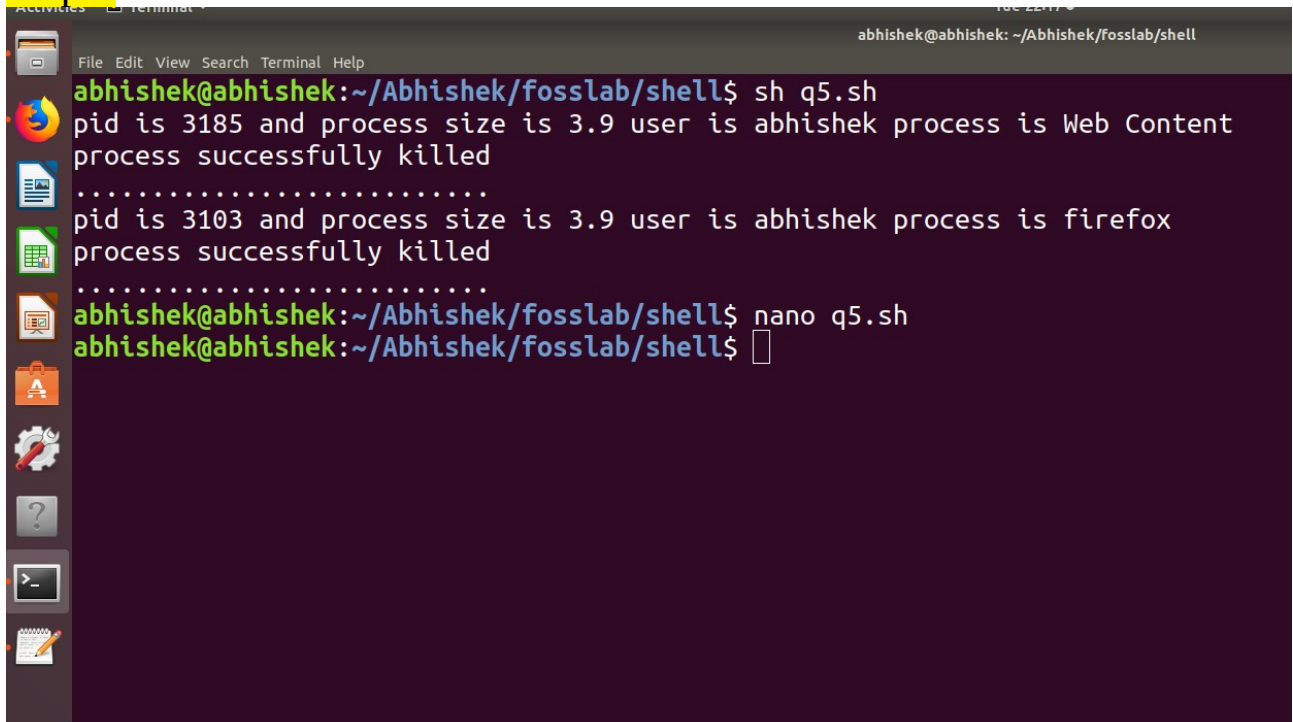
5. Write a Shell script which starts on system boot up and kills every process which uses more than a specified amount of memory or CPU.

### Shellscript



```
GNU nano 2.9.3 q5.sh
ps -e -o pmem=,pcpu=,pid=,user=,comm= |sort -r -k 1 | while read size cpu pid user comm
do
if [ $size \> 3 ]
then
    echo "pid is" $pid "and process size is" $size "user is" $user "process is" $comm
    kill $pid
    echo "process successfully killed"
    echo "....."
fi
done
```

### output



```
abhishek@abhishek: ~/Abhishek/fosslab/shell$ sh q5.sh
pid is 3185 and process size is 3.9 user is abhishek process is Web Content
process successfully killed
.....
pid is 3103 and process size is 3.9 user is abhishek process is firefox
process successfully killed
.....
abhishek@abhishek: ~/Abhishek/fosslab/shell$ nano q5.sh
abhishek@abhishek: ~/Abhishek/fosslab/shell$
```

submitted by  
Abhishek manoharan No 2