Experiment 5

20th March 2019

Aim: To be able to write bash scripts.

---

1. 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

```
  GNU nano 2.8.7                                    File: bash_1_1.sh

echo "The logged user is:" `env logname`
echo "The current shell is:" $SHELL
echo "The home directory is: " $HOME
echo "The OS type is: " `uname -o`
echo "The current path setting is: " $PATH
echo "The current working directory is: " $PWD
echo "The number of users logged in are: " `users | wc -l`
```

Most of the information can be got from the system environment variables. The variable names can be seen in the bash command env and can be seen using $<variable-name>

‘users’ returns all the logged on devices

The output is as follows:

```
protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_1.sh
The logged user is: protonegative
The current shell is: /bin/zsh
The home directory is:  /home/protonegative
The OS type is:  GNU/Linux
The current path setting is:  /usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/home/protonegative/bin
The current working directory is:  /home/protonegative/work/BashScr
The number of users logged in are:  1
protonegative@fedora  ~/work/BashScr  master ●
```

---

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 memory, model etc
    6. File system (Mounted)

```
GNU nano 2.8.7                                File: bash_1_2.sh

echo -e "\e[1mThe current OS is:\e[0m " `uname -o`
echo
echo -e "\e[1mThe OS version is:\e[0m " `uname -a`
echo
echo -e "\e[1mThe available shells are as follows:\e[0m "
cat /etc/shells/
echo
echo -e "\e[1mThe CPU information is as follows:\e[0m "
lscpu
echo
echo -e "\e[1mThe memory information is as follows:\e[0m "
cat /proc/meminfo
echo
echo -e "\e[1mThe hard disk information is as follows:\e[0m "
lsblk
echo
echo -e "\e[1mMounted file system information:\e[0m "
df -Th
```

The OS information can be found using the command ‘uname’.
The file ‘shells’ in /etc contains all the available shells
The CPU information can be seen using lscpu
The memory information can be seen in the file ‘meminfo’ in /proc
The hard-disk info can be found using lsblk
The mounted file system information can be found using command df

The output is as follows:

```
protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_2.sh
The current OS is:  GNU/Linux

The OS version is:  Linux fedora 4.15.8-300.fc27.x86_64 #1 SMP Fri Mar 9 18:11:36 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

The available shells are as follows:
cat: /etc/shells/: Not a directory

The CPU information is as follows:
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  2
Core(s) per socket:  2
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               61
Model name:          Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz
Stepping:            4
CPU MHz:             1888.124
CPU max MHz:         3000.0000
CPU min MHz:         500.0000
BogoMIPS:            4789.15
Virtualization:      VT-x
L1d cache:           32K
L1i cache:           32K
L2 cache:            256K
L3 cache:            4096K
NUMA node0 CPU(s):   0-3
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht t
```

```
q dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave a
vx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust b
mi1 avx2 smep bmi2 erms invpcid rdseed adx smap intel_pt xsaveopt dtherm ida arat pln pts

The memory information is as follows:
MemTotal:        8075844 kB
MemFree:         2136452 kB
MemAvailable:    4614664 kB
Buffers:          248780 kB
Cached:          2532480 kB
SwapCached:            0 kB
Active:          3455872 kB
Inactive:        2034876 kB
Active(anon):    2404048 kB
Inactive(anon):   489828 kB
Active(file):    1051824 kB
Inactive(file):  1545048 kB
Unevictable:         184 kB
Mlocked:             184 kB
SwapTotal:       8216572 kB
SwapFree:        8216572 kB
Dirty:               460 kB
Writeback:             0 kB
AnonPages:       2709692 kB
Mapped:           682460 kB
Shmem:            491808 kB
Slab:             245928 kB
SReclaimable:     185116 kB
SUnreclaim:        60812 kB
KernelStack:       14876 kB
PageTables:        84184 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    12254492 kB
```

```
VmallocTotal:   34359738367 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:      277332 kB
DirectMap2M:     8019968 kB
DirectMap1G:     1048576 kB

The hard disk information is as follows:
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda               8:0    0 931.5G  0 disk
├─sda1            8:1    0   500M  0 part /boot/efi
├─sda2            8:2    0    40M  0 part
├─sda3            8:3    0   128M  0 part
├─sda4            8:4    0     2G  0 part
├─sda5            8:5    0 492.4G  0 part
├─sda6            8:6    0 109.1G  0 part
├─sda7            8:7    0 146.5G  0 part
├─sda8            8:8    0   450M  0 part
├─sda9            8:9    0   9.5G  0 part
├─sda10           8:10   0     1G  0 part /boot
└─sda11           8:11   0 169.9G  0 part
  ├─fedora-root 253:0    0    50G  0 lvm  /
  ├─fedora-swap 253:1    0   7.9G  0 lvm  [SWAP]
  └─fedora-home 253:2    0 112.1G  0 lvm  /home
```

```
DirectMap1G:     1048576 kB

The hard disk information is as follows:
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda               8:0    0 931.5G  0 disk
├─sda1            8:1    0   500M  0 part /boot/efi
├─sda2            8:2    0    40M  0 part
├─sda3            8:3    0   128M  0 part
├─sda4            8:4    0     2G  0 part
├─sda5            8:5    0 492.4G  0 part
├─sda6            8:6    0 109.1G  0 part
├─sda7            8:7    0 146.5G  0 part
├─sda8            8:8    0   450M  0 part
├─sda9            8:9    0   9.5G  0 part
├─sda10           8:10   0     1G  0 part /boot
└─sda11           8:11   0 169.9G  0 part
  ├─fedora-root 253:0    0    50G  0 lvm  /
  ├─fedora-swap 253:1    0   7.9G  0 lvm  [SWAP]
  └─fedora-home 253:2    0 112.1G  0 lvm  /home
sr0              11:0    1  1024M  0 rom

Mounted file system information:
Filesystem              Type      Size  Used Avail Use% Mounted on
devtmpfs                devtmpfs  3.9G     0  3.9G   0% /dev
tmpfs                   tmpfs     3.9G   11M  3.9G   1% /dev/shm
tmpfs                   tmpfs     3.9G  2.0M  3.9G   1% /run
tmpfs                   tmpfs     3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/mapper/fedora-root ext4       49G   14G   34G  29% /
tmpfs                   tmpfs     3.9G  540K  3.9G   1% /tmp
/dev/mapper/fedora-home ext4      110G   65G   40G  63% /home
/dev/sda10              ext4      976M  152M  758M  17% /boot
/dev/sda1               vfat      496M   68M  429M  14% /boot/efi
tmpfs                   tmpfs     789M   16K  789M   1% /run/user/42
tmpfs                   tmpfs     789M  7.1M  782M   1% /run/user/1000
 protonegative@fedora  ~/work/BashScr  master ●
```

---

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

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

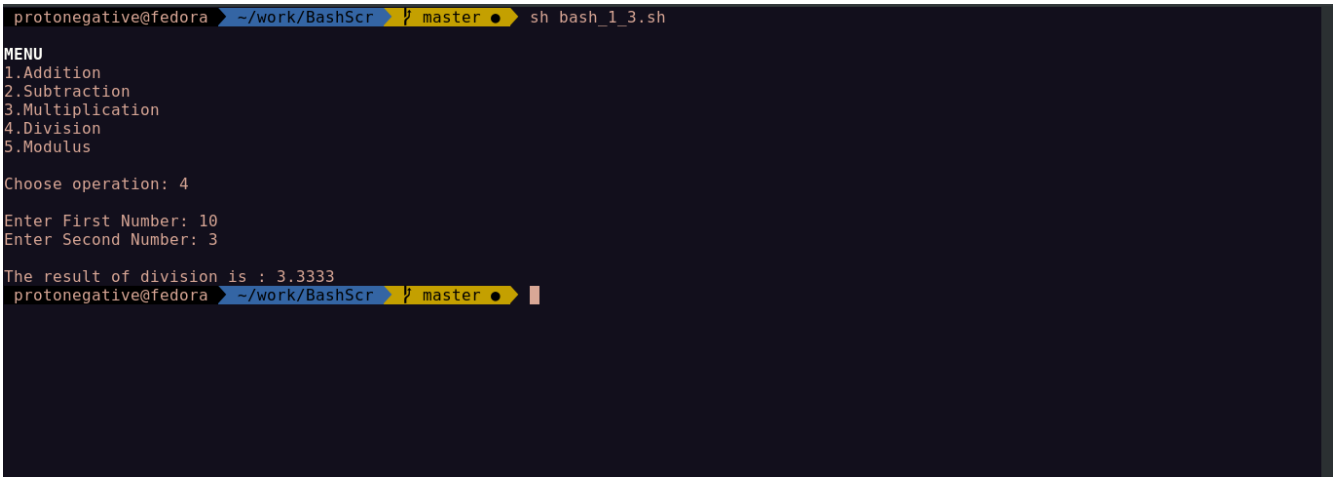
```
GNU nano 2.8.7                                File: bash_1_3.sh

bold=$(tput bold)
normal=$(tput sgr0)
echo
echo -e "${bold}MENU${normal}\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n5.Modulus"
echo
printf "Choose operation: "
read  opt
echo
if [ $opt -gt 5 ]
then
echo "Invalid option"
exit
fi
printf "Enter First Number: "
read no1
printf "Enter Second Number: "
read no2
echo
case $opt in

1) printf "The sum is: "
   expr $no1 + $no2
   ;;
2) printf "The difference is: "
  expr $no1 - $no2
   ;;
3) printf "The result of multiplication is: "
   expr $no1 \* $no2
   ;;
4) printf "The result of division is : "
   bc <<< "scale=4; $no1 / $no2"
   ;;
5) printf "The result of modulo is: "
   expr $no1 % $no2
```

The read command can be used to accept an input and store in a variable.
‘expr’ can be used to evaluate a numerical expression.
‘bc’ or basic calculator can be used to evaluate an expression but the arguments are piped or from a file

The output is as follows:



---

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.

```
  GNU nano 2.8.7                                    File: bash_1_4.sh

if [[ $# -ne 2 ]]
then
echo "Invalid number of arguments"
exit
fi

if [[ ! (-a $1) ]]
then
echo "Not a valid file location or file dosent exist"
exit
fi
NO=$(grep -c -e $2 $1)
if [[ $NO -eq 0 ]]
then
echo $2 >> $1
echo "Username is added"
exit
else
echo "Username already exists"
exit
fi
```

The number of arguments($#) are checked if equal to zero and corresponding error message is shown.
The first argument is checked to see if it is an existing file(-a)
The main operation is done using simple bash commands.

The output is as follows:

```
 protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_4.sh ulist sashi
Username is added
 protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_4.sh ulist sashi
Username already exists
 protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_4.sh ulist sashi 123
Invalid number of arguments
 protonegative@fedora  ~/work/BashScr  master ●  sh bash_1_4.sh ulist2 sashi 123
Invalid number of arguments
 protonegative@fedora  ~/work/BashScr  master ●
```

---

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.

```
  GNU nano 2.8.7                                    File: bash_1_5.sh

if [[ $# -ne 1 ]]
then
echo "Max memory usage not entered"
exit
fi
ME=`pwd`/$0
CHECK=$(grep -c -e "sh $ME" ~/.bashrc)
if [[ CHECK == 0 ]]
then
echo "sh $ME" >> ~/.bashrc
fi
MEM=$1
kill `ps -o pid,%mem ax | sort -b -k2 -r | awk -F " " '$2>$MEM' | awk -F "  " '{print $1}'`
```

‘ps’ shows running processes and their memory usage in percentages and hence the corresponding PID’s can be obtained and can be killed.

<u>Result:</u> Understood how to write bash scripts.

---

eof
