

Shell Scripting

Set 1

Rwithik Manoj

March 18, 2019

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

- (a) Currently logged user and his login name
- (b) Your current shell
- (c) Your home directory
- (d) Your operating system type
- (e) Your current path setting
- (f) Your current working directory
- (g) Number of users currently logged in

Script:

```
#!/bin/bash

echo -e "Current User           : $USER (Login Name: $LOGNAME)"
echo -e "Current Shell             : $SHELL"
echo -e "Home Directory             : $HOME"
echo -e "OS Type                    : $OSTYPE"
echo -e "Path                       : $PATH"
echo -e "Current Working Directory   : $PWD"
echo -e "Number of Users Logged in   : `who | wc -l`"
```

Output:

```
rwthik@Zeus foss-lab/Scripts (master *)
» ./Expt4.sh
Current User           : rwthik (Login Name: rwthik)
Current Shell          : /usr/bin/zsh
Home Directory         : /home/rwthik
OS Type               : linux-gnu
Path                  : /home/rwthik/.gem/ruby/2.5.0/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/opt/android-sdk/platform-tools:/opt/android-sdk/tools:/opt/android-sdk/tools/bin:/usr/lib/jvm/default/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl
Current Working Directory : /home/rwthik/Projects/foss-lab/Scripts
Number of Users Logged in : 1
rwthik@Zeus foss-lab/Scripts (master *)
» █
```

2. Write a shell script to show various system configurations like
 - Your OS and version, release number, kernel version
 - All available shells
 - Computer CPU information like processor type, speed etc
 - Memory information
 - Hard disk information like size of hard-disk, cache memory, model etc
 - File system (Mounted)

Script:

```
#!/bin/bash

echo -e "OS Details: `uname -a`\n"
echo -e "Available Shells"
echo -e "-----"
cat /etc/shells | grep -e "^/"
echo -e "\nCPU Info"
echo -e "-----"
lscpu | grep "Architecture\|Model name\|MHz"
echo -e "\nMemory Info"
echo -e "-----"
free -mh
echo -e "\nHard Disk Info"
echo -e "-----"
df
echo -e "\nMounted Filesystems"
echo -e "-----"
lsblk
```

Output:

```
rwithik@Zeus foss-lab/Scripts (master *)
» ./Expt5.sh
OS Details: Linux Zeus 4.14.102-1-MANJARO #1 SMP PREEMPT Wed Feb 20 22:55:55 UTC 2019 x86_64 GNU/Linux

Available Shells
-----
/bin/sh
/bin/bash
/bin/zsh
/usr/bin/zsh
/usr/bin/git-shell
/usr/bin/fish

CPU Info
-----
Architecture:      x86_64
Model name:         Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz
CPU MHz:            666.149
CPU max MHz:        2400.0000
CPU min MHz:        400.0000

Memory Info
-----
              total        used        free      shared  buff/cache   available
Mem:          3.6Gi         1.3Gi         1.3Gi         84Mi         1.1Gi         2.0Gi
Swap:          7.6Gi         3.0Mi         7.6Gi

Hard Disk Info
-----
Filesystem      1K-blocks      Used Available Use% Mounted on
dev              1904660           0    1904660   0% /dev
run              1912548       1252    1911296   1% /run
/dev/sda2       78218924  14127908  60074632  20% /
```

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

- (a) Addition
- (b) Subtraction
- (c) Multiplication
- (d) Division
- (e) Modulus

Script:

```
#!/bin/bash

echo -e "Enter the operator:\n\
      + for addition\n\
      - for subtraction\n\
      * for multiplication\n\
      / for division\n\
      % for modulus"

read CHOICE
if [[ $CHOICE == '+' ]] || [[ $CHOICE == '-' ]] \
    || [[ $CHOICE == '*' ]] || [[ $CHOICE == '/' ]] \
    || [[ $CHOICE == '%' ]]
then
    echo -e "Enter two numbers:"
    read A
    read B
    echo -e "Result: ${A}${CHOICE}${B}"
else
    echo -e "Invalid Choice"
fi
```

Output:

```
rwithik@Zeus foss-lab/Scripts (master *)
» ./Expt6.sh
Enter the operator:
    + for addition
    - for subtraction
    * for multiplication
    / for division
    % for modulus
+
Enter two numbers:
12
23
Result: 35
rwithik@Zeus foss-lab/Scripts (master *)
» █
```

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
- (a) Check that the correct number of arguments was received and print a message, in case the number of arguments is incorrect
 - (b) Check whether the `ulist` file exists and print an error message if it does not
 - (c) 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.

Script:

```
#!/bin/bash

if [[ $# -ne 2 ]]; then
    echo -e "Wrong usage!"
    exit
fi

if [[ -f $1 ]]; then
    for name in `cat $1`; do
        if [[ $name == $2 ]]; then
            echo -e "Name already exists in list!"
            exit
        fi
    done
    echo $2 >> $1
    exit
fi

echo -e "File $1 not found!"
```

Output:

```
rwthik@Zeus foss-lab/Scripts (master *)
» ./Expt7.sh
Wrong usage!
rwthik@Zeus foss-lab/Scripts (master *)
» ./Expt7.sh ulist rwthik
rwthik@Zeus foss-lab/Scripts (master *)
» ./Expt7.sh ulist rwthik
Name already exists in list!
rwthik@Zeus foss-lab/Scripts (master *)
» ./Expt7.sh ulist rahul
rwthik@Zeus foss-lab/Scripts (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.

Script:

```
#!/bin/bash

CHECK_MEM_VAL=10.0
CHECK_CPU_VAL=10.0

while true
do
    ps -e -o pmem=,pcpu=,pid=,user=,comm= --sort=-pmem |
    while read SIZE CPU PID USER COMM
    do
        KILL_MEM=0
        KILL_CPU=0
        if [ "$USER" != "root" ]
        then
            if [ `echo "$SIZE>$CHECK_MEM_VAL" | bc` = "1" ]
            then
                echo "kill $COMM due to mem usage"
                kill $pid
            elif [ `echo "$CPU>$CHECK_CPU_VAL" | bc` = "1" ]
            then
                echo "kill $COMM due to cpu usage"
                kill $pid
            fi
        fi
    done
    sleep 10
done
```

Output:


```
rwthik@Zeus foss-lab/Scripts (master *%)  
» ./Expt8.sh  
kill Web Content due to mem usage  
kill firefox due to mem usage  
█
```