Experiment No 3
Class: SE Comp
Year: 2020-21
Performed by: Danyl Fernandes, 72

Aim: Write shell scripts to do the following:

a) Display OS Version, release number, kernel version
b) Display top 10 processes in descending order
c) Display processes with highest memory usage
d) Display current logged in user & log name
e) Display current shell, home directory, operating system type current path setting, current working directory

Theory:

- Shell script is a regular text file that contains shell or UNIX commands.
    - Before running it, it must have execute permission: chmod + x filename

Scripting Vs C Programming:

- Advantages:
    - Easy to work with other programs.
    - Easy to work with files.
    - Easy to work with strings
    - Great for prototyping. No compilation
- Disadvantages:
    - Slower
    - Not well suited for algorithms & data structures

Shell Variables:

- **To set:** name = value
- **Read:** $var
- i.e. A = 5, B = 6, C = $A + $B, echo C

Environmental Variables:

- $HOME: Absolute pathname of your home directory
- $PATH: A list of directories to search for
- $MAIL: Absolute pathname to mailbox
- $USER: Your login name
- $SHELL: Absolute pathname of login shell
- $TERM: Type of your terminal

Positional parameters:

- The arguments to a shell script
  - $1, $2, $3
- The arguments to a shell function
- $0 Name of the current shell script
- Arguments to the set build-in command
  - set this is a test
  - $1 = this, $2 = is, $3 = a, $4 = test

Special Parameters:

- $#: Number of positional parameters
- $: Options currently in effect
- $?: Exit value of last
- $$: Process number of current process
- $!: Process number of background process
- $*: All arguments on command line
- "$@": All arguments on command line individually quoted "$1", "$2", …

Control Structures:

```
if expression
then
     command 1
else
     command 2
fi
```

For loops:

```
for var in list
do
     command
done
```

Case Statement:

```
case expression in
     Pattern_n)
          Statements
          ;;
     *)
          Statements
          ;;
esac
```

Conclusion:

We were successfully able to implement shell scripting in Linux

1) Display OS Version, Release number, Kernel version:

```
1 echo "OS version"
2 uname -o
```

```
OS version
GNU/Linux[?2004l
```

```
1 echo "Kernel name"
2 uname -s
```

```
Kernel name
Linux04h
```

```
1 echo "Kernel version"
2 uname -v
```

```
Kernel version
#1 SMP PREEMPT Fri Jan 15 21:11:34 UTC 2021
```

```
1 echo "Kernel release"
2 uname -r
```

```
Kernel release
5.10.7-3-MANJARO
```

```
1 echo "Processor name"
2 uname -p
```

```
Processor name
unknownh
```

2) Display top 10 processes in descending order:

```
1 echo "top 10 process"
2 ps axl|head -n 10
```

```
top 10 process
F   UID     PID    PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4     0       1       0  20   0 105496  8816 -      Ss   ?          0:01 /sbin/init
1     0       2       0  20   0      0     0 -      S    ?          0:00 [kthreadd]
1     0       3       2   0 -20      0     0 -      I<   ?          0:00 [rcu_gp]
1     0       4       2   0 -20      0     0 -      I<   ?          0:00 [rcu_par_gp]
1     0       6       2   0 -20      0     0 -      I<   ?          0:00 [kworker/0:0H-kblockd]
1     0       8       2   0 -20      0     0 -      I<   ?          0:00 [mm_percpu_wq]
1     0       9       2  20   0      0     0 -      S    ?          0:00 [ksoftirqd/0]
1     0      10       2  -2   -      0     0 -      S    ?          0:00 [rcuc/0]
1     0      11       2  -2   -      0     0 -      I    ?          0:00 [rcu_preempt]
```

3) Display processes with highest memory usage:

```
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem|head
```

```
  PID    PPID CMD                         %MEM %CPU
 2042    1513 /usr/lib/firefox-developer-  7.7  8.1
 1723    1513 /usr/lib/firefox-developer-  6.6 14.6
 1513    1103 /usr/lib/firefox-developer-  4.6 17.7
  381       1 /usr/bin/dotnet /usr/lib/je  3.8  8.0
 1103       1 /usr/bin/plasmashell         3.2  2.4
 1644    1513 /usr/lib/firefox-developer-  2.4  2.7
 2404    1513 /usr/lib/firefox-developer-  2.4  2.9
 2335    1513 /usr/lib/firefox-developer-  1.9  2.5
 1583    1513 /usr/lib/firefox-developer-  1.6  0.8
```

4) Display current logged in user and log name:

```
echo "display current logged in"
who -u

echo "display count of logged in"
who -u|wc -l

echo "whoami"
who i am
```

```
display current logged in
dan      tty1         2021-04-03 10:23  old          856 (:0)
dan      pts/0        2021-04-03 10:23  old          937 (:0)
display count of logged in
2[?2004h
whoami4h
```

5) Display current shell, home directory, operating system type, current path setting, current working directory:

```
echo "Current shell"
echo $0

echo "Home directory"
echo $HOME

echo "Current path setting"
echo $PATH

echo "Current working directory"
pwd

echo "Operating system type"
uname -o
```

```
Current shell
/usr/bin/bash04l
Home directory4l
/home/dan[?2004l
Current path setting004l
/home/dan/.gem/ruby/2.7.0/bin:/home/dan/.gem/ruby/2.7.0/bin:/home/dan/.local/bin:/usr/local/sbin:/usr/local/bin:/us
r/bin:/usr/lib/jvm/default/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/var/lib/snapd/snap/bin
Current working directory
/home/dan[?2004l
Operating system type04l
GNU/Linux[?2004l
```