Experiment No.3
Write shell scripts programming.
Date of Performance:
Date of Submission:

Aim: Write Shell Scripts to do the following:

1. Perform the basic arithmetic operations
2. Display top 10 processes in Ascending order.
3. Display processes with highest memory usage.
4. Display current logged in user and log name.
5. Display current shell, home directory, kernel version.

Objective: The shell is the operating system's command-line interface (CLI) and interpreter for the set of commands that are used to communicate with the system. A shell script is usually created for command sequences in which a user has a need to use repeatedly in order to save time.

Theory:

Shell is a user program or its environment is provided for user interaction. It is a command prompt within Linux where you can type commands. It is a program that takes your commands from the keyboard and gives them to the OS to perform. Shell is not part of system KERNAL but it uses system KERNAL to execute programs, create files,etc. A Shell Script is a text file that contains a sequence of commands for a UNIX based OS. It is called a Shell Script because it combines into a "Script" in a single file a sequence of commands, that would otherwise have to be presented to the system from a keyboard one at a time. A Shell Script is usually created for command sequences for which a user has a repeated need. You initiate the sequence of commands in Shell Script by simply entering the name of the Shell Script on a command line.

Types of Shell Script :-

1. **sh** - Simple Shell
2. **bash** - Bourne Again Shell
3. **ksh** - Korne Shell
4. **csh** - C Shell
5. **ssh** - Secure Shell

To use a particular Shell type the Shell name at the command prompt. Eg:- $csh - It will switch the current Shell to C Shell. To view the current Shell that is being used, type echo $ SHELL at the command prompt.

Result:

1. Perform the basic arithmetic operations

```
student@ubuntu:~$ expr 100 + 100
200
student@ubuntu:~$ expr 100 - 100
0
student@ubuntu:~$ expr 100 / 10
10
student@ubuntu:~$ expr 100 \* 100
10000
```

2. Display top 10 processes in Ascending order.

```
student@ubuntu:~$ ps aux --sort=%mem | head -n 11
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           2  0.0  0.0      0     0 ?        S    00:35   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   00:35   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   00:35   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   00:35   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   00:35   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   00:35   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   00:35   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    00:35   0:00 [rcu_tasks_rude_]
root          12  0.0  0.0      0     0 ?        S    00:35   0:00 [rcu_tasks_trace]
root          13  0.0  0.0      0     0 ?        S    00:35   0:00 [ksoftirqd/0]
```

3. Display processes with highest memory usage.

```
student@ubuntu:~$ top

top - 01:08:39 up 33 min,  1 user,  load average: 0.12, 0.05, 0.01
Tasks: 273 total,   1 running, 272 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.7 us,  1.2 sy,  0.0 ni, 96.0 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   1928.3 total,    316.8 free,    900.5 used,    711.0 buff/cache
MiB Swap:    923.3 total,    923.3 free,      0.0 used.    863.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1530 student   20   0  292624  71040  48564 S   2.7   3.6   0:08.70 Xorg
   1676 student   20   0 3757732 254336 103232 S   2.0  12.9   0:16.02 gnome-shell
   2041 student   20   0  814424  50500  37944 S   1.3   2.6   0:01.75 gnome-terminal-
   1860 student   20   0  144668  41296  29532 S   0.7   2.1   0:05.72 vmtoolsd
    758 root      20   0  239460   7348   6336 S   0.3   0.4   0:05.57 vmtoolsd
      1 root      20   0  102696  11484   8268 S   0.0   0.6   0:03.27 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.07 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.26 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 I   0.0   0.0   0:00.70 kworker/0:1-cgroup_destroy
```

4. Display current logged in user and log name.

```
student@ubuntu:~$ who
student  :0           2024-02-24 00:35 (:0)
```

5. Display current shell, home directory, kernel version.

```
student@ubuntu:~$ echo $SHELL
/bin/bash
```

```
student@ubuntu:~$ echo $HOME
/home/student
```

```
student@ubuntu:~$ uname -r
5.15.0-94-generic
```

Conclusion In conclusion, the shell scripts provided above demonstrate the versatility and power of shell scripting in Unix-like operating systems. They allow users (us) to perform arithmetic operations, manage processes, and retrieve system information with ease.