Name: Abdullah Khan

Roll no: 122A1002

Branch: CE

Batch: C1

Experiment #2

Aim: Write shell scripts to do the following: Display top 10 processes in descending order Display processes with highest memory usage. Display current logged in user and logname. Display current shell, home directory, operating system type, current path setting, current working directory. Display OS version, release number, kernel version. Illustrate the use of sort, grep, awk, etc

1) Display top 10 processes in descending order

Program:

```
echo " Top 10 processes in decending order is:";
ps axl | head -n 10
```

2) Display processes with highest memory usage

Program:

```
echo " Display processes with highest memory usage";
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
```

3) Display current logged user and login name

Program:

```
echo "LOGGED IN USERS ARE:-\n";
who -u
echo "NUMBER OF LOGGED IN USERS ARE :-\n";
who -u| wc -l
```

4) Display current shell, home directory, operating system type, current path setting, and current working directory.

Program:

```
echo " Current home directory is:";
whoami
echo " Current operating system type is:";
uname
echo " Current current working directory is:";
pwd
```

5) Display OS name, release number, kernel version

Program:
echo " OS Name is:"
uname;
echo " Release number is:"
uname -a;
echo " Release number is:"
uname -r;

6) Illustrate the use of sort, grep, awk, etc.

Sort- sorts the contents of a text file, line by line

1. *sort syntax*

sort [*OPTION*]... [*FILE*]...

sort [*OPTION*]... --files0-from=*F*
Let's say you have a file, **data.txt**, which contains the following ASCII text:
apples
oranges
pears
kiwis
bananas

To sort the lines in this file alphabetically, use the following command:

sort data.txt

...which will produce the following output:
apples
bananas
kiwis
oranges
pears

Note that this command does not actually change the input file, **data.txt**. If you want to write the output to a new file, **output.txt**, redirect the output like this:

sort data.txt > output.txt

which will not display any output, but will create the file **output.txt** with the same sorted data from the previous command. To check the output, use the **cat** command:

cat output.txt

2. **grep**- print lines matching a pattern , **grep** searches the named input *FILE*s (or standard input if no files are named, or if a single hyphen-minus (**-**) is given as file name) for lines containing a match to the given *PATTERN.*

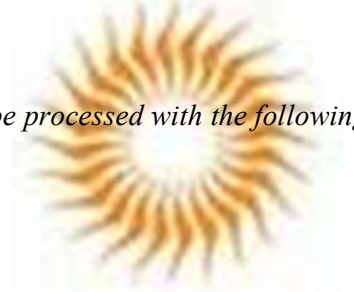
Use grep to search words only

When you search for boo, grep will match fooboo, boo123, barfoo35 and more. You can force the grep command to select only those lines containing matches that form whole words i.e. match only boo word:
$ grep -w "boo" file

Use grep to search 2 different words

Use the egrep command as follows:
$ egrep -w 'word1|word2' /path/to/file

3. ***AWK*** *is an interpreted programming language. It is very powerful and specially designed for text processing.*



Consider a text file ***marks.txt*** *to be processed with the following content −*
1) Root Physics 80
2) Rahul Maths 90
3) Shyam Biology 87
4) Kedar English 85
5) Hari History 89

Printing Column or field

You can instruct AWK to print only certain columns from the input field. The following example demonstrates this −

$ awk '{print $3 "\t" $4}' marks.txt
Physics 80
Maths 90
Biology 87
English 85
History 89

Writing A shell Script:

1. Open the terminal. Go to the directory where you want to create your **script**.
2. Create a file with . **sh** extension.
3. Write the **script** in the file using an editor.

4. Make the **script** executable with command chmod +x <fileName>.
5. **Run** the **script** using ./<fileName>.

Program:

```
echo -e"Display Top 10 PROCESS running"
ps axl | head -n 10

echo -e"\nDisplay processes with highest memory usage"
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head

echo -e"\n\nDisplay current logged user and login name"
echo -e"LOGGED IN USERS ARE:-\n";
who -u
echo -e"\n\nNUMBER OF LOGGED IN USERS ARE :-\n";
who -u| wc -l

echo -e"\n\nDisplay current shell, home directory, operating system type, current path setting, and current working directory"
echo " \nCurrent home directory is:";
whoami
echo -e" \n\nCurrent operating system type is:";
uname
echo -e" \n\nCurrent current working directory is:";
pwd

echo -e"\n\nDisplay OS name, release number, kernel version"
echo -e" \n\nOS Name is:"
uname;
echo -e" \n\nRelease number is:"
uname -a
echo -e" \n\nRelease number is:"
uname -r

echo -e"\n\nIllustrate the use of sort, grep, awk, etc"
sort data.txt>output.txt
cat output.txt

grep -w "sies" JAT.txt

awk '{print "\t" $1}' marks.txt
```

Command Execution:

Output:

```
siesgst@siesgst-OptiPlex-3020:~$ ./temp.sh


Display Top 10 PROCESS running
F   UID     PID    PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4     0       1       0  20   0 168488 11932 -      Ss   ?          0:01 /sbin/init splash
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       5       2   0 -20      0     0 -      I<   ?          0:00 [slub_flushwq]
1     0       6       2   0 -20      0     0 -      I<   ?          0:00 [netns]
1     0       8       2   0 -20      0     0 -      I<   ?          0:00 [kworker/0:0H-events_highpri]
1     0      10       2   0 -20      0     0 -      I<   ?          0:00 [mm_percpu_wq]
1     0      11       2  20   0      0     0 -      S    ?          0:00 [rcu_tasks_rude_]


Display processes with highest memory usage
    PID    PPID CMD                         %MEM %CPU
   3720    3048 /usr/lib/firefox/firefox -c  7.4  1.5
   3755    3048 /usr/lib/firefox/firefox -c  6.5  2.6
   3048    1341 /usr/lib/firefox/firefox -n  6.0  4.2
   5795    5763 /usr/lib/libreoffice/progra  3.1  0.1
   1571    1341 /usr/bin/gnome-shell         2.9  3.4
   4147    3048 /usr/lib/firefox/firefox -c  2.8  1.6
   4926    3048 /usr/lib/firefox/firefox -c  2.1  1.6
   4892    3048 /usr/lib/firefox/firefox -c  2.0  0.0
   3935    3048 /usr/lib/firefox/firefox -c  1.9  0.1


Display current logged user and login name
LOGGED IN USERS ARE:-
siesgst  :0           2024-01-30 09:00   ?          1433 (:0)
NUMBER OF LOGGED IN USERS ARE :-

1
```

```
Display current shell, home directory, operating system type, current path setting, and current working directory
 Current home directory is:
siesgst
 Current operating system type is:
Linux
 Current current working directory is:
/home/siesgst


Display OS name, release number, kernel version
 OS Name is:
Linux
 Release number is:
Linux siesgst-OptiPlex-3020 5.15.0-91-generic #101~20.04.1-Ubuntu SMP Thu Nov 16 14:22:28 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 Release number is:
5.15.0-91-generic
```

```
Illustrate the use of sort, grep, awk, etc


australia
brazil
china
denmark
egypt
france
greece
hungary
india
japan
kenya
lebanon
morocco
norway
oman
portugal
qatar
russia
suriname
tanzania
ukraine
venezuala
wales
xing province
yemen
zimbabwe
```

```
sies sion
love for sies
sies tml 2024
sies technical team


        Physics
        Maths
        Biology
        English
        History
siesgst@siesgst-OptiPlex-3020:~$
```

Conclusion: The shell scripts for the above commands are executed.