

# Lab4

---

## Red Hat

1. List the user commands and redirect the output to /tmp/commands.list

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$head /tmp/commands.list
alert
egrep
fgrep
grep
l
la
ll
ls
if
then
```

2. Count the number of user commands

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$wc -l /tmp/commands.list
4629 /tmp/commands.list
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

3. Get all the users names whose first character in their login is 'g'.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$cut -d: -f1 /etc/passwd | grep
'^g'
games
geoclue
gnome-remote-desktop
gnome-initial-setup
gdm
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

4. Get the logins name and full names (comment) of logins starts with “g”.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$grep '^g' /etc/passwd | cut -d: -f1,5  
games:games  
geoclue:  
gnome-remote-desktop:GNOME Remote Desktop  
gnome-initial-setup:  
gdm:Gnome Display Manager  
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

5. Save the output of the last command sorted by their full names in a file.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$grep '^g' /etc/passwd | cut -d: -f1,5 | sort -t: -k2 > /tmp/g_users_sorted.txt
```

we can use cat for display:

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$cat /tmp/g_users_sorted.txt  
geoclue:  
gnome-initial-setup:  
games:games  
gdm:Gnome Display Manager  
gnome-remote-desktop:GNOME Remote Desktop  
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

6. Write two commands:

**first:** to search for all files on the system that named .bash\_profile.

The .bas\_profile response:

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$find / -name .bash_profile 2>/dev/null
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$ls -R | sort > /tmp/ls_output.txt 2> /tmp/ls_error.txt &
[1] 3303
```

.bashrc response:

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$find / -name .bashrc 2>/dev/null
/snap/core22/2164/etc/skel/.bashrc
/snap/core22/2194/etc/skel/.bashrc
/home/maryam_abdelraheem/.bashrc
/etc/skel/.bashrc
[1]+ Done ls -R | sort > /tmp/ls_output.txt 2> /tmp/ls_error.txt
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

**Second:** sorts the output of ls command on / recursively, Saving their output and error in 2 different files and sending them to the background.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$ls -R | sort > /tmp/ls_output.txt 2> /tmp/ls_error.txt &
[1] 3385
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$ls: cannot open directory '/boot/lost+found
': Permission denied
ls: cannot open directory '/etc/credstore': Permission denied
ls: cannot open directory '/etc/credstore.encrypted': Permission denied
ls: cannot open directory '/etc/cups/ssl': Permission denied
ls: cannot open directory '/etc/multipath': Permission denied
ls: cannot open directory '/etc/polkit-1/rules.d': Permission denied
ls: cannot open directory '/etc/ssl/private': Permission denied
ls: cannot open directory '/etc/sssd': Permission denied
ls: cannot open directory '/home/ftpuser': Permission denied
ls: cannot open directory '/home/islam': Permission denied
ls: cannot open directory '/home/maryam_abdelraheem/testdir': Permission denied
ls: cannot open directory '/lost+found': Permission denied
ls: cannot open directory '/proc/1/fd': Permission denied
ls: cannot open directory '/proc/1/fdinfo': Permission denied
ls: cannot open directory '/proc/1/map_files': Permission denied
ls: cannot open directory '/proc/1/ns': Permission denied
ls: cannot open directory '/proc/1/task/1/fd': Permission denied
```

- `ls -R` → lists all files and directories recursively starting from the current directory (/ if you ran it there).
- `ls | sort` → sorts all the file and folder names alphabetically.
- `> /tmp/ls_output.txt` → saves the sorted output in a file called `ls_output.txt`.
- `2> /tmp/ls_error.txt` → saves any error messages (like “Permission denied”) in `ls_error.txt`.  
number 2 >> stands for error in linux/unix
  - 0 → standard input
  - 1 → standard output
  - 2 → standard error
- `&` → runs the whole command in the background, so you can keep using the terminal while it runs.

7. Display the number of users who is logged now to the system.

```
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$who | wc -l  
2  
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

- who → standard output lists users
- Pipe (|) sends output to wc
- wc -l → counts lines

8. Display lines 7 to line 10 of /etc/passwd file

```
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$sed -n '7,10p' /etc/passwd  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

Or

```
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$head -n 10 /etc/passwd | tail -n 4  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

9. What happens if you execute:

a. cat filename1 | cat filename2

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$cat filename1 | cat filename2
cat: filename2: No such file or directory
cat: filename1: No such file or directory
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$touch filename1 | touch filename2
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$echo 'welcom to file 1'| echo 'welcom to fi
le 2'
welcom to file 2
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$cat filename1 | cat filename2
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

Pipes don't force commands to read stdin unless designed

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$ls /etc/passwd | wc -l
1
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

b. ls | rm

rm needs arguments, does not read stdin → error

c. ls /etc/passwd | wc -l

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$ls | rm
rm: missing operand
Try 'rm --help' for more information.
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

10.Issue the command sleep 100.

```
maryam_abdelraheem@maryamabdelraheem:~/Desktop$sleep 100  
$
```

11.Stop the last command.

```
maryam_abdelraheem@maryamabdelraheem:~/Desktop$sleep 100  
$^Z  
[1]+ Stopped sleep 100  
maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

12.Resume the last command in the background

```
maryam_abdelraheem@maryamabdelraheem:~/Desktop$bg  
[1]+ sleep 100 &  
maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

13.Issue the jobs command and see its output.

```
maryam_abdelraheem@maryamabdelraheem:~/Desktop$jobs  
[1]+ Done sleep 100  
maryam_abdelraheem@maryamabdelraheem:~/Desktop$
```

14. Send the sleep command to the foreground and send it again to the background.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$sleep 100
^Z
[1]+  Stopped                  sleep 100
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$bg %1
[1]+ sleep 100 &
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$fg %1
sleep 100
```

15. Kill the sleep command.

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$kill %1
[1]+  Stopped                  sleep 100
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

```
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$kill <PID>
bash: syntax error near unexpected token `newline'
[1]+ Terminated                sleep 100
:maryam_abdelraheeml@maryamabdelraheem:~/Desktop$
```

16. Display your processes only

```
>> ps -u $USER
```

17. Display all processes except yours

```
>> ps -e | grep -v $USER
```

## 18. Use the pgrep command to list your processes only

```
- 3718 pts/0    00:00:00 grep
:maryam_abdelraheem@maryamabdelraheem:~/Desktop$pgrep -u $USER
1787
1792
1804
1805
1807
1809
1812
1824
1847
1859
1919
1920
1934
1942
1945
1978
1982
1989
2027
2040
2042
2055
2067
2070
2078
2082
2089
2097
2098
2099
2102
2104
2105
2106
```

## 19.Kill your processes only.

```
3499 :maryam_abdelraheem@maryamabdelraheem:~/Desktop$pkkill -u $USER
```

After running the command:

