Module Code & Module Title

Level 5-CT5052NP & Network Operating System

Assessment Type

Logbook _8

Semester

2023/24 Spring/Autumn

Student Name: Arij Lamichhane

London Met ID: 22085885

Assignment Due Date: 27 December 2024

Assignment Submission Date: 28 December 2024

Submitted To: Mr. Prashant Adhikari

I confirm that I understand my proposal needs to be submitted online via College's MST PORTAL under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of contents

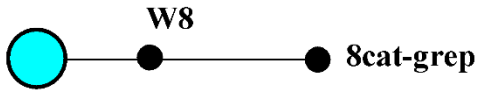
Table of Figures:

Introduction

This log focuses on practicing UNIX utilities and commands in a Linux-based Networking Operating System (NOS) environment. The primary goals are to enhance proficiency in directory navigation, file creation, and text processing, as well as defining and using aliases and executing commands with history. Task included exploring commands like mkdir, cat, grep, alias , and history These tasks are essential for mastering the UNIX environment, which is widely used in system administration, programming, and software development.

1.Creating Directory for structured figure given

Create the directory structured presented in the figure below:



Your
home
directory

```
vboxuser@Ubuntu:~/W8/8cat-grep$ mkdir -p W8/8cat-grep
vboxuser@Ubuntu:~/W8/8cat-grep$ ls
W8
vboxuser@Ubuntu:~/W8/8cat-grep$ tree W8
W8
└── 8cat-grep

2 directories, 0 files
```

Figure 1:Creating Directory for structured figure given

2.Changing to 8 cat-grep directory by one step using relative pathname

Change to the 8 cat-grep directory by one step using a relative pathname.

```
vboxuser@Ubuntu:~$ cd W8/8cat-grep/
vboxuser@Ubuntu:~/W8/8cat-grep$ ls
```

Figure 2: changing to 8cat-grep directory by one step using relative pathname

3. Using the cat utility for creating two files

Using the **cat** utility, create two files

File testa	**File testb**
Kkkll	KKKKK
lllmm	LLLLL
oo-oo	MMMMM
mmmdd	DDDDD
dddkk	

```
vboxuser@Ubuntu:~$ cat > testa
Kkkll
lllmm
oo-oo
mmmmdd
dddkk
vboxuser@Ubuntu:~$ cat > testb
KKKKk
LLLLL
MMMMM
DDDDD
vboxuser@Ubuntu:~$
```

Figure 3: Creating two files using cat utility

4. Giving command to view the result.

Give the following commands and explain the results for yourself

I. Command: grep || testa

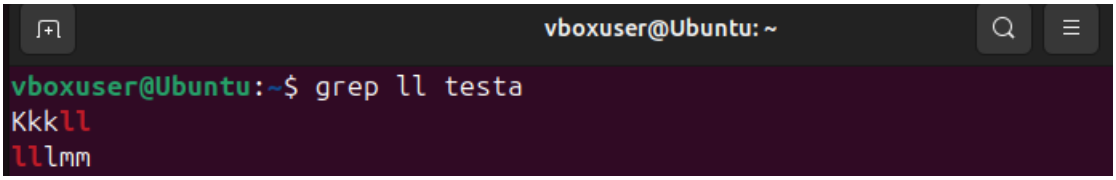


Figure 4: Command grep || testa

II.Command: grep -v || testa

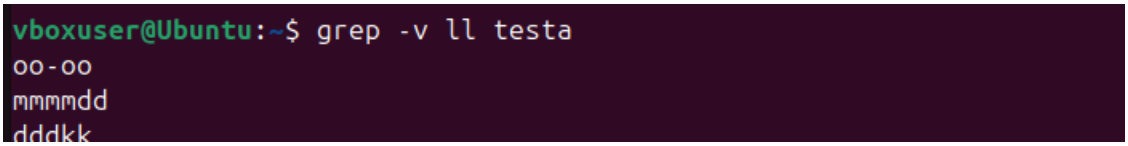


Figure 5: Command: grep -v || testa

III. Command : grep -n || testa

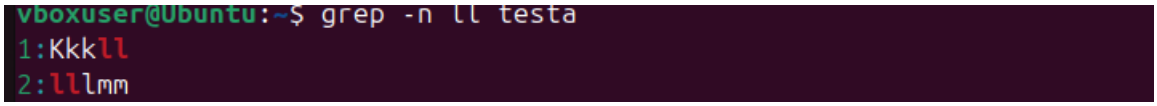


Figure 6: grep -n || testa

IV. Command : grep -l || *

*Figure 7: Command: grep -l || **

V. Command: grep -i || *

```
testa:Kkkll
testa:lllmm
testb:LLLLL
```

*Figure 8 :Command : grep -i || **

VI. Command grep -I LL *

```
testa:Kkkll
testa:lllmm
testb:LLLLL
```

*Figure 9: Command grep -I LL **

VII. Command: grep -c || *

```
testa:2
testb:0
```

*Figure 10: Command: grep -c || **

VIII.Command: grep '^K' testa testb

```
vboxuser@Ubuntu:~$ grep 'k' testa testb
testa:Kkkll
testa:dddkk
testb:KKKKk
```

Figure 11: Command: grep '^k' testa testb

IX.Command: grep -n '^' testa

```
vboxuser@Ubuntu:~$ grep -n '^' testa
1:Kkkll
2:lllmm
3:oo-oo
4:mmmmdd
5:dddkk
```

Figure 12: Command: grep -n '^' testa

5. Defining lsal for ls -al Command

Define the **lsal** alias for ls -al command. Show that your system stores it giving the alias command (without arguments)

```
vboxuser@Ubuntu:~$ alias lsal="ls -al"
vboxuser@Ubuntu:~$ lsal
total 144
drwxr-x--- 25 vboxuser vboxuser 4096 Dec 26 14:37 .
drwxr-xr-x  3 root     root     4096 Dec  8 06:42 ..
-rw-rw-r--  1 vboxuser vboxuser 5875 Dec 12 05:38 a1script
-rw-rw-r--  1 vboxuser vboxuser    0 Dec  8 07:01 alscript
```

Figure 13: Defining lsal command without arguments

6.Removing the alias so that the system does not store it.

Remove the alias. so that your system does not store it.

```
vboxuser@Ubuntu:~$ unalias lsal
bash: unalias: lsal: not found
vboxuser@Ubuntu:~$
```

Figure 14: Removing the alias so that the system does not store it.

7.Defining alias again for preserving for next sessions

Define this alias again, preserving it for the next sessions. Shows that the system still keeps this your alias

```
al
ias lsal="ls
 -al"


File Name to Write: .bashrc
^G Help          M-D DOS Format      M-A Append          M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend         ^T Browse
```

Figure 15: Defining alias again for preserving for next sessions

8. Defining the nwho alias for the number of system file at UNIX computers.

Define the **nwho** alias for the number of system files at UNIX computers.

Alias nwho ‘getent passws|wc-l’

```
vboxuser@Ubuntu:~$ alias nwho="getent passwd | wc -l"
vboxuser@Ubuntu:~$ nwho
```

Figure 16: Defining the nwho alias for the number of system file

9. Comparing the figure displayed with ones got by your UNIX -mates by giving command nwho

Give the command nwho. Compare the figure displayed with ones got by your UNIX -mates.

```
vboxuser@Ubuntu:~$ alias nwho="getent passwd | wc -l"
vboxuser@Ubuntu:~$ nwho
49
```

Figure 17: Command nwho

```
vboxuser@Ubuntu:~$ uname -a
Linux Ubuntu 6.8.0-49-generic #49-Ubuntu SMP PREEMPT_DYNAMIC Mon Nov  4 02:06:24
 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
vboxuser@Ubuntu:~$
```

Figure 18: Figure displayed with one got by UNIX -mates

10. Listing last command executed by history command

List your last commands executed giving the history command.

```
vboxuser@Ubuntu:~$ history
    1  script alscript
    2  whoami
    3  who
    4  finger linuxnnn
    5  date
    6  is
    7  Ls
```

Figure 19: Listing the command executed by history command

11. Re-executing last one command using redo(r) command

Re-execute the last but one command using the **redo (r)** command and the number of the event. Fc-r

```
vboxuser@Ubuntu:~$ fc 225
script alscript1
```

Figure 20: Re-executing last one command using redo(r) command

12. Re-executing the command given three commands ago using the negative integer! -3

. Re-execute the command given three commands ago using the negative integer.

! -3

```
vboxuser@Ubuntu:~$ echo "A"
A
vboxuser@Ubuntu:~$ echo "B"
B
vboxuser@Ubuntu:~$ echo "C"
C
vboxuser@Ubuntu:~$ !-3
echo "A"
A
```

Figure 21: Re-executing command given three commands ago using the negative integer

13. Re-executing the last command which name begins with 'l'.

Re-execute the last command which name begins with 'l'.

Fc -e-1

```
vboxuser@Ubuntu:~$ fc -s e
echo hello
hello
vboxuser@Ubuntu:~$
```

Figure 22: Re-executing the last command which name begins with 'l'

Conclusion:

The tasks in this workshop provided valuable insights into using Unix/Linux utilities to perform core system operations. By creating directories, working with files, and using commands like grep, participants learned to process text effectively. Defining and managing aliases increase the ability to customize the shell environment for improved productivity. Additionally, exploring command history demonstrated ways to efficiently re-execute previously executed commands. Overall, these exercises have strengthened the participants' confidence and proficiency in navigating and managing the Unix/Linux environment, laying a strong foundation for further exploration of advanced system administration tasks.

References:

1. Nemeth, E., Snyder, G., Hein, T., Whaley, B., & Mackin, D. (2017). UNIX and Linux System Administration Handbook (5th ed.). Addison-Wesley Professional.

2. Linux Manual Pages. Available at: https://man7.org/linux/man-pages/

3. Linux.org. (n.d.). Tutorials and Guides. Available at: https://www.linux.org/

4. The Linux Command Line. (2023). Available at: https://linuxcommand.org/