



Module Code & Module Title

Level 5 - CT5052NI

Assessment Weightage & Type

Individual Report Writing

Year and Semester

2023-24 Autumn

Student Name: Aman Bahadur Paudel

London Met ID: 23048978

College ID: NP04CP4A230136

Assignment Submission Date: 28/12/2024

Submitted To: Mr. Prashant Adhikari

Declaration: I understand that I am required to submit my assignment under the appropriate module page prior to the specified deadline, in order for it to be considered for marking. I acknowledge that any assignment submitted after the deadline will be deemed as a non-submission and will not be marked, resulting in a score of zero.

Table of Contents

1. Introduction	1
2. Objective.....	1
3. Required Tools	1
4. Practiced contents	2
6. Conclusion	11

Table of Figures

Figure 1: Creating directory	2
Figure 2: Changing directory.....	2
Figure 3: Creating files testa and testb	3
Figure 4: grep ll testa	4
Figure 5: grep -v ll testa	4
Figure 6: grep -n ll testa	4
Figure 7: grep -l ll *	4
Figure 8: grep -i ll *	5
Figure 9: grep -i LL *	5
Figure 10: grep -c ll *	5
Figure 11: grep '^K' testa testb.....	6
Figure 12: grep -n '^' testa.....	6
Figure 13: Defining alias for lsal.....	6
Figure 14: Removing alias lsal.....	7
Figure 15: Reassigning alias to preserve for future session also	7
Figure 16: Alias lsal.....	7
Figure 17: Executing with alias	8
Figure 18: Defining alias nwho.....	8
Figure 19: Executing nwho	8
Figure 20: Running history command	9
Figure 21: Command fc -r	9
Figure 22: Command !-3	10

1. Introduction

Linux is a powerful open-source operating system that can be widely used in various domains, including system administration, cybersecurity, and software development. This lab report focuses on gaining hands-on experience with essential Linux commands. It provides a foundation for managing files, users, and system processes efficiently. After practicing these commands, one can navigate and operate a Linux-based system and enhance the ability to perform administrative tasks.

2. Objective

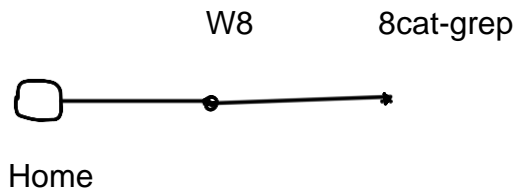
The main objective of this workshop is to be familiar with the Linux commands and the related utilities.

3. Required Tools

Linux or UNIX based operating system. Here, I am using the Terminal Emulator from Kali Linux.

4. Practiced contents

1. Create the directory structure presented in the figure below.



```
(kali@kali)-[~]
$ mkdir w8
(kali@kali)-[~]
$ cd w8
(kali@kali)-[~/w8]
$ mkdir 8cat-grep
(kali@kali)-[~/w8]
$
```

Figure 1: Creating directory

Mkdir command creates the directory, and the command cd change the directory. Here, w8 directory is created and 8cat-grep directory is created within w8.

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

```
(kali@kali)-[~]
$ cd w8/8cat-grep
(kali@kali)-[~/w8/8cat-grep]
$
```

Figure 2: Changing directory

The cd command is used to change the working directory from home to 8cat-grep.

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

File testa	File testb
------------	------------

Kkkll	KKKKK
lllmm	LLLLL
oo--oo	MMMMM
mmmdd	DDDDD
dddkk	

```
(kali@kali)-[~/w8/8cat-grep]
$ cat > testa
Kkkll
lllmm
oo--oo
mmmdd
dddkk

(kali@kali)-[~/w8/8cat-grep]
$ cat > testb
KKKKK
LLLLL
MMMMM
DDDDD

(kali@kali)-[~/w8/8cat-grep]
$ tree
.
├── testa
└── testb

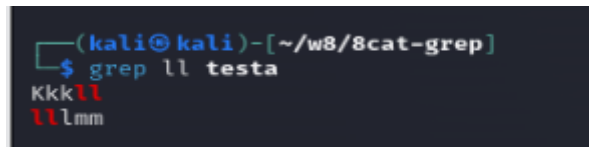
1 directory, 2 files
```

Figure 3: Creating files *testa* and *testb*

Cat command can be used to create the file. Here, two files *testa* and *testb* are created with the corresponding multiline text.

5. Give the following commands and explain the results for yourself

- **grep ll testa**

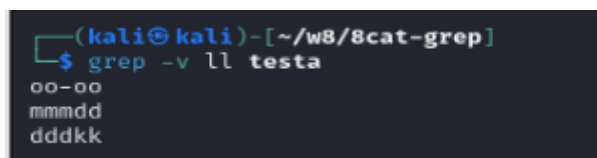


```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep ll testa
Kkkll
llmm
```

Figure 4: *grep ll testa*

“Grep ll testa” command finds the lines containing ‘ll’ in the file testa.

- **grep -v ll testa**

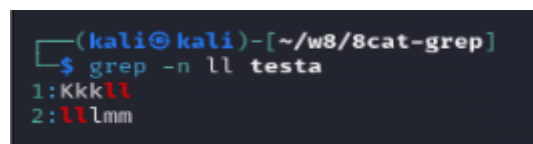


```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -v ll testa
oo-oo
mmdd
ddkk
```

Figure 5: *grep -v ll testa*

“Grep -v ll testa” command finds the lines that doesn’t contain ‘ll’ in the file testa.

- **grep -n ll testa**



```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -n ll testa
1:Kkkll
2:llmm
```

Figure 6: *grep -n ll testa*

“Grep -n ll testa” command shows line numbers matching the text ‘ll’ from the file testa.

- **grep -l ll ***



```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -l ll *
testa
```

Figure 7: *grep -l ll **

“Grep -l ll* ” command lists the filenames that has the text matching to ‘ll’.

- **grep -i ll ***

```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -i ll *
testa:Kkkll
testa:lllmm
testb:LLLLL
```

Figure 8: grep -i ll *

This is the search command with case-insensitive matches for ‘ll’.

- **grep -i LL ***

```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -i LL *
testa:Kkkll
testa:lllmm
testb:LLLLL
```

Figure 9: grep -i LL *

This is the search command with case-insensitive matches for ‘LL’.

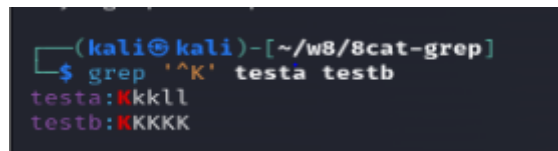
- **grep -c ll ***

```
(kali㉿kali)-[~/w8/8cat-grep]
$ grep -c ll *
testa:2
testb:0
```

Figure 10: grep -c ll *

“Grep -c ll* ” command counts the matching lines in each file: testa and testb.

- **grep '^K' testa testb**

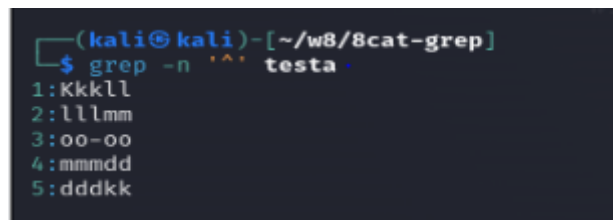


```
(kali@kali)-[~/w8/8cat-grep]
$ grep '^K' testa testb
testa:Kkkll
testb:KKKKK
```

Figure 11: `grep '^K' testa testb`

“Grep '^K' testa testb” command matches the lines starting with 'K' in both files testa and testb.

- **grep -n '^' testa**



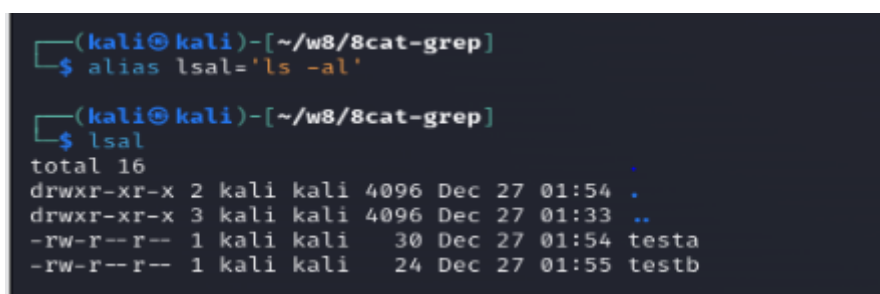
```
(kali@kali)-[~/w8/8cat-grep]
$ grep -n '^' testa
1:Kkkll
2:lllmm
3:oo-oo
4:mmdd
5:dddkk
```

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

“Grep -n '^' testa” command displays all the lines with line number of the mentioned file testa.

6. Define the **lsal** alias for **ls -al** command

Show that your system stores it giving the **alias** command (without arguments).



```
(kali@kali)-[~/w8/8cat-grep]
$ alias lsal='ls -al'

(kali@kali)-[~/w8/8cat-grep]
$ lsal
total 16
drwxr-xr-x 2 kali kali 4096 Dec 27 01:54 .
drwxr-xr-x 3 kali kali 4096 Dec 27 01:33 ..
-rw-r--r-- 1 kali kali 30 Dec 27 01:54 testa
-rw-r--r-- 1 kali kali 24 Dec 27 01:55 testb
```

Figure 13: Defining alias for `lsal`

Here, “ls -al” command is assigned to lsal as alias name. After assigning the alias, we can enter the alias name ‘lsal’ to run the assigned command ‘ls -al’.

7. Remove the alias. Show that your system does not store it.

```
(kali㉿kali)-[~/w8/8cat-grep]
$ unalias lsal

(kali㉿kali)-[~/w8/8cat-grep]
$ lsal
Command 'lsal' not found, did you mean:
  command 'lsar' from debunar
Try: sudo apt install <deb name>

(kali㉿kali)-[~/w8/8cat-grep]
$
```

Figure 14: Removing alias lsal

By using the command “unalias”, we can remove the assigned alias name. Here, we can see that, after removing the alias ‘lsal’, the command ‘lsal’ is not recognized by the system.

8. Define this alias again preserving it for the next session. Show that the system still keep this your alias.

```
(kali㉿kali)-[~/w8/8cat-grep]
$ alias lsal='ls -al' >> ~/.bashrc
```

Figure 15: Reassigning alias to preserve for future session also

Here, the alias name ‘lsal’ is again assigned to ‘ls -al’ and extended to preserve it for the next session also. After the terminal is closed, we can reuse the alias again by following this method as shown in the pictures below.

```
(kali㉿kali)-[~/w8/8cat-grep]
$ alias lsal
lsal='ls -al'
```

Figure 16: Alias lsal

```
(kali㉿kali)-[~/w8/8cat-grep]
$ lsal
total 16
drwxr-xr-x 2 kali kali 4096 Dec 27 01:54 .
drwxr-xr-x 3 kali kali 4096 Dec 27 01:33 ..
-rw-r--r-- 1 kali kali 30 Dec 27 01:54 testa
-rw-r--r-- 1 kali kali 24 Dec 27 01:55 testb
```

Figure 17: Executing with alias

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

alias nwho='getent passwd|wc -l'

```
(kali㉿kali)-[~/w8/8cat-grep]
$ alias nwho='getent passwd|wc -l'

(kali㉿kali)-[~/w8/8cat-grep]
$ alias nwho
nwho='getent passwd|wc -l'
```

Figure 18: Defining alias nwho

The **nwho** alias is defined to count the number of system users (or system files) on a UNIX system. The command “getent passwd” retrieves a list of user accounts from the system's database, and “wc -l” counts the number of lines in the output. The alias “nwho” can be used to quickly check the number of user accounts without typing the whole command manually, each time.

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

```
(kali㉿kali)-[~/w8/8cat-grep]
$ nwho
57
```

Figure 19: Executing nwho

The **nwho** alias counts the number of user accounts on a UNIX-like system.

11. List your last commands executed giving the **history** command.

```

73 cat > testa << EOF\nKkkll\nlllmm\noo-oo\nmmdd\ndddkk\nEOF
74 grep ll testa
75 grep -v ll testa
76 grep -n ll testa
77 grep -l ll*
78 grep -l ll *
79 grep -i ll *
80 grep -i LL *
81 grep -c ll *
82 grep '^K' testa testb
83 grep -n '^' testa
84 ../..
85 alias lsal='ls -al'
86 alias
87 cd lsal
88 unalias lsal
89 alias
90 alias lsal='ls -al'
91 echo "alias lsal='ls -al' " >> ~/.bashrc
92 source ~/.bashrc
93 \n
94 echo "alias lsal='ls -al' " >> ~/.bashrc
95 source ~/.bashrc
96 alias lsal
97 alias lsal='ls -al'
98 alias nwho='getent passwd | wc -l'
99 nano ~/.bashrc\
100 nano ~/.bashrc
101 alias nwho='getent passwd | wc -l'
102 source ~/.bashrc
103 nwho
104 getent passwd
105 nwho
106 alias nwho
107 nwho
108 which getent
109 which wc
110 getent password | wc -l
111 alias nwho='getent password | wc -l'
112 nwho
113 alias nwho='getent password | wc -l'
114 nwho

```

Figure 20: Running history command

The “history” command allows us to view a list of commands we've recently executed. With this command, we can quickly recall and reuse commands without having to retype them again.

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

fc -r

```

(kali@kali)-[~]
$ fc 2
cd

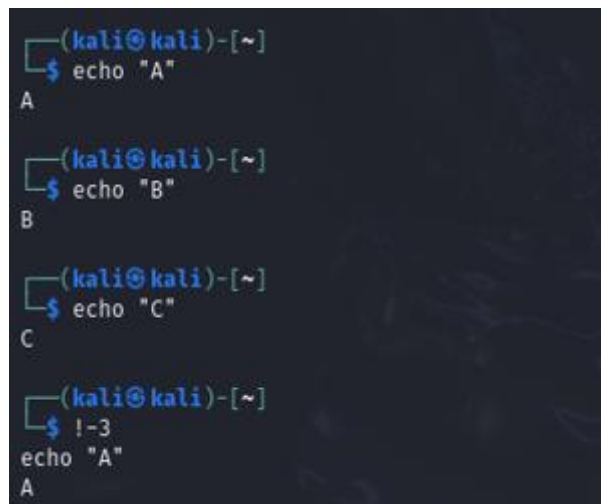
```

Figure 21: Command fc -r

The “fc -r” command re-executes a previous command based on its event number. The ‘-r’ option runs the command in reverse order, allowing us to quickly repeat past commands from the history.

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

!*-3*



```
(kali㉿kali)-[~]  
$ echo "A"  
A  
  
(kali㉿kali)-[~]  
$ echo "B"  
B  
  
(kali㉿kali)-[~]  
$ echo "C"  
C  
  
(kali㉿kali)-[~]  
$ !-3  
echo "A"  
A
```

Figure 22: Command !-3

“!*-3*” command is used to re-execute the command that was run three commands ago.

6. Conclusion

This document has the detailed practical steps and theoretical knowledge required to manage directories and files in a Linux environment effectively. From basic commands like creating and navigating directories to more advanced operations such as modifying file permissions, defining and executing through alias, getting history and re-executing command from history without retyping the same command again, each section provides essential skills.

By practicing these operations, we can gain a deeper understanding of Linux file systems, improve workflow efficiency, and strengthen their command-line proficiency. Whether creating structured directories, or executing commands with various options like using alias and through history, the tasks equip us with the confidence to navigate real-world scenarios. Through these practices, we also can enhance security and accessibility in a multi-user environment.