UNIVERSITI PUTRA MALAYSIA

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT

Name:	YUE CHENGHAO	Matric:	227154

The lab was conducted on Windows 10 using WSL2 (Ubuntu 22.04 LTS) as the Linux environment.
All commands were executed in the Bash shell, and outputs were captured from the Ubuntu terminal.

LAB 2 – Individual

OBJECTIVE:
The main objective of this lab is to introduce the students to the Linux/Unix operating system. Students will also be introduced to the Linux Command Line Shell (Bash@ Bourne Shell).

DURATION:
1 week / 3 hours

DUE DATE OF SUBMISSION: WEEK 3
Submit your assignment to your demonstrator. Make sure you sign the submission form after submitting.

PROGRAMME OBJECTIVE (PO):
PO2: Use the available software package with appropriate computer components.

RUBRIC:

CRITERIA	LEVEL 1 Very Poor	LEVEL 2 Poor	LEVEL 3 Good	LEVEL 4 Very Good	LEVEL 5 Excellent
The ability to perform operating system (OS) tasks.	Unable to perform OS tasks or able to perform OS tasks with very limited success (< 40%).	Able to perform OS tasks with limited success (≥ 40%,< 60%).	Able to perform OS tasks with some success (≥ 60%, < 80%).	Able to perform OS tasks with considerable success (≥ 80%, < 100%).	Able to perform OS tasks with outstanding success (100%).

QUESTIONS:

Objectives:

- use command line’s instructions
  - to view, navigate, create and delete directories
  - to copy, rename, and delete files
  - to view information regarding process, switching processes

Bourne shell provides a collection of commands which can be used to control some of the important disk operations and operating system’s process management. All Unix and Linux versions support execution of the commands from the command line.

Instruction for the instructor

Explain about the following:

- Bourne Shell and the general purpose of the shell - purpose, example command, case-sensitive character restriction etc.
- the general structure of Unix/Linux file-system – file/directory, permission/access, important folders
- process management

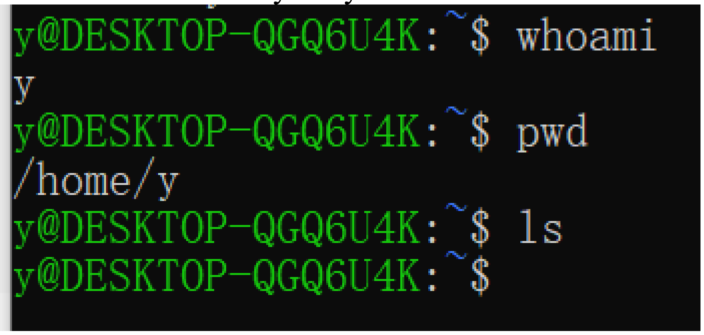
This exercise requires Linux operating system. If you are not familiar with installing operating system, it is safer to install in the VMWare or VirtualBox (**and don’t overwrite the existing OS in your computer**). You can use Knoppix ‘Linux on CD-ROM’ distro which does not require installation. Download and use the bootable .iso file.

Lab Exercises

1. Task: Open a command line shell’s window in Linux GUI (KDE) interface:

Find the ‘terminal emulator’ (computer monitor) icon and double-click. The terminal emulator (command prompt window) will appear. Type the command in the provided prompt (something like ~$).

Who are you in the system? Student(type **whoami** to identify)
What directory are you in? /home/student



```
y@DESKTOP-QGQ6U4K:~$ whoami
y
y@DESKTOP-QGQ6U4K:~$ pwd
/home/y
y@DESKTOP-QGQ6U4K:~$ ls
y@DESKTOP-QGQ6U4K:~$
```

What does the prompt look like? student@ubuntu:~$

2. Task : Working with files

touch command allows creating an empty file without editing.
Create two files named ‘a.txt’ and ‘b.txt’ in the current directory.

- type: ***touch a.txt***
- type: ***touch b.txt***

echo command is used to send (print) a text on the screen. Use **echo “sometext”**. Print out the text ‘this is a text’ to the computer screen.

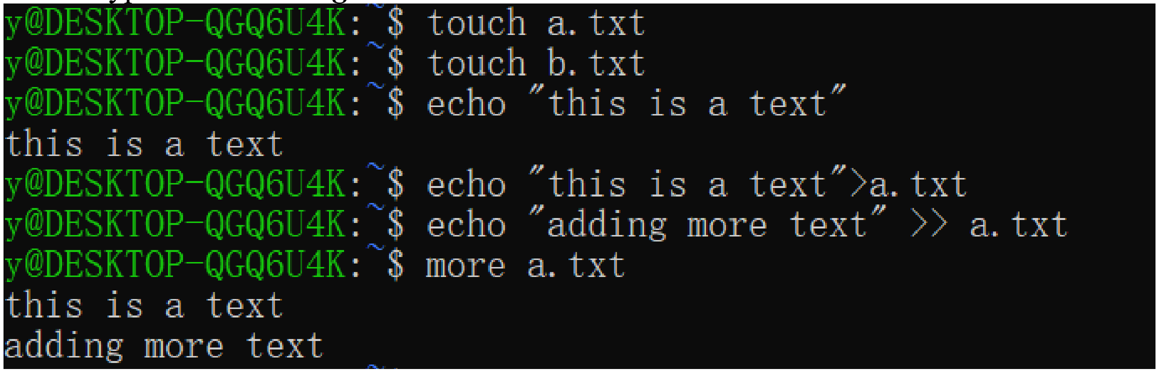
- type: ***echo “this is a text”***

echo command can be used to send (print) a text to file using redirection operator (>). Send the text ‘this is a text’ to the file a.txt

- type**:** ***echo “this is a text” > a.txt***

You can add more content to the file using **echo** with redirection operator (>>)

- type**:** ***echo “adding more text” >> a.txt***



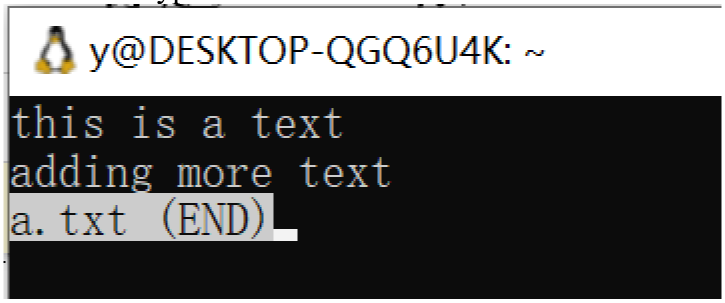
```
y@DESKTOP-QGQ6U4K:~$ touch a.txt
y@DESKTOP-QGQ6U4K:~$ touch b.txt
y@DESKTOP-QGQ6U4K:~$ echo "this is a text"
this is a text
y@DESKTOP-QGQ6U4K:~$ echo "this is a text">a.txt
y@DESKTOP-QGQ6U4K:~$ echo "adding more text" >> a.txt
y@DESKTOP-QGQ6U4K:~$ more a.txt
this is a text
adding more text
```

Why don’t we use ‘>’ to add content to the non-empty file? Try it and observe. What happened to the content of the file? ________________________________

Because doing it this way will overwrite the file; all existing content will be deleted. To add new content, please use >>>.

more and **less** command can be used to view the content of a file. View the content of the file a.txt using **more** or **less**

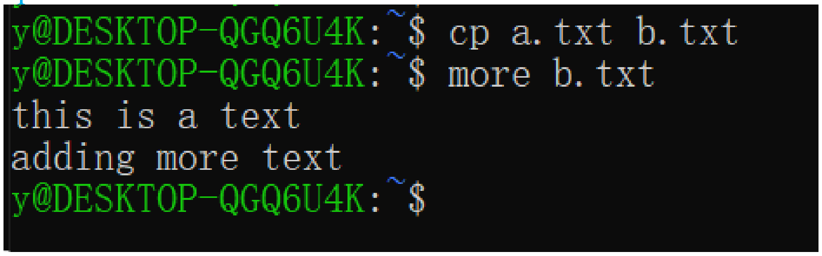
- type: ***more a.txt***
- type: ***less a.txt***



```
this is a text
adding more text
a.txt (END)
```

What is the difference between the two commands (**more** and **less**)?
More: One-way reading, can only turn pages downward.
Less: Can scroll up and down. Supports search, page turning, and exiting the keyboard (press 'q' to exit).

How to copy the content of a.txt into b.txt using **more** command?
cp a.txt b.txt



```
y@DESKTOP-QGQ6U4K:~$ cp a.txt b.txt
y@DESKTOP-QGQ6U4K:~$ more b.txt
this is a text
adding more text
y@DESKTOP-QGQ6U4K:~$
```

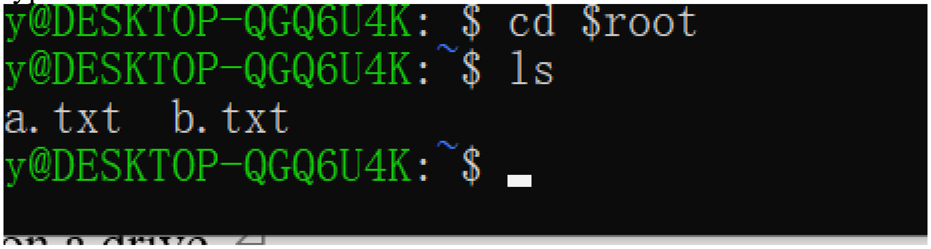
3. Task : Displaying directory's contents

ls command will list directories and files in a directory

(Tips: use **ls –l** to show attributes. Entries with directory types are specified with 'd' in the beginning of a line e.g. 'drwxr-x--- …'. The command displays the related information for the files and directories, such as the permission, owner, timestamp of the file/directory created etc.)

Show a list of the directories located on a drive.

- type***: cd $root*** change to the root directory
- type: ***ls***

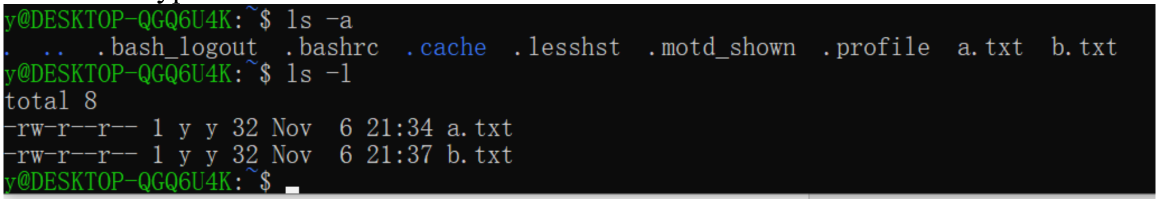


```
y@DESKTOP-QGQ6U4K: $ cd $root
y@DESKTOP-QGQ6U4K:~$ ls
a.txt  b.txt
y@DESKTOP-QGQ6U4K:~$ _
```

What happened? ______________________________________

It listed all the visible directories and files in the root (/) directory, such as bin, etc, home, usr, var, etc.

- type: ***ls -a***
- type: ***ls -l***



```
y@DESKTOP-QGQ6U4K:~$ ls -a
.  ..  .bash_logout  .bashrc  .cache  .lesshst  .motd_shown  .profile  a.txt  b.txt
y@DESKTOP-QGQ6U4K:~$ ls -l
total 8
-rw-r--r-- 1 y y 32 Nov  6 21:34 a.txt
-rw-r--r-- 1 y y 32 Nov  6 21:37 b.txt
y@DESKTOP-QGQ6U4K:~$ _
```

Based on your observation, describe the differences between –**a** and –**l** switches

____

-a shows all files including hidden ones (those starting with a dot), while -l shows detailed information such as file type, permissions, owner, group, size, and modification time.

What does the -l switch do? _______________

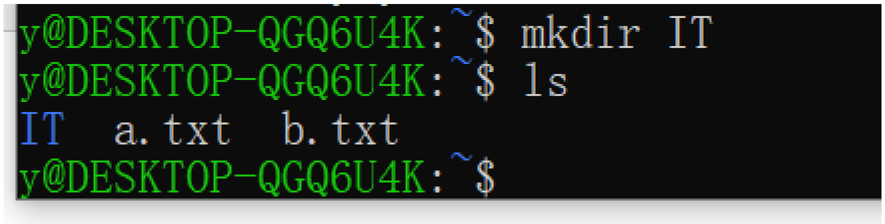
It lists files in **long format**, displaying attributes (permissions, owner, size, date).

4. Task : Create and change a directory

To create directories, use the **mkdir <directory_name>** command. To change directories, use the **cd <directory_name>** command. To move up one level in the directory structure, use the command **cd ..**

Create an **IT** directory in the root directory

- type*: cd* (or *cd $root*) to change to the root directory
- type: ***mkdir IT***



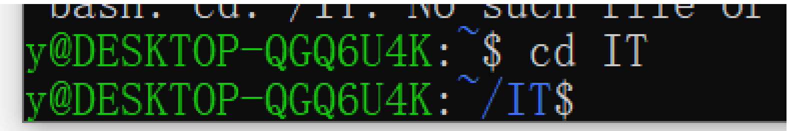
```
y@DESKTOP-QGQ6U4K:~$ mkdir IT
y@DESKTOP-QGQ6U4K:~$ ls
IT   a.txt   b.txt
y@DESKTOP-QGQ6U4K:~$
```

Verify the creation of the directory

- type: ***ls***

Is the **IT** directory in the list? Yes

How to go into the **IT** directory? Cd IT



```
bash: cd: /IT: No such file or
y@DESKTOP-QGQ6U4K:~$ cd IT
y@DESKTOP-QGQ6U4K:~/IT$
```

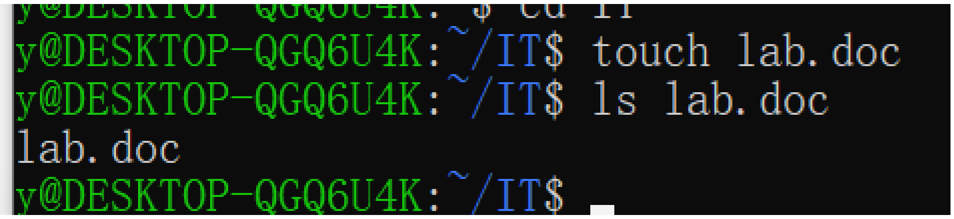
5. Task : Managing Files using cp, mv, rm, rmdir

You can use the **cp** command to make a copy of a file, with a new name.

Create a dummy file to IT directory (***touch lab.doc***).

How to verify whether ***lab.doc*** file is in the **IT** directory?___

Ls lab.doc



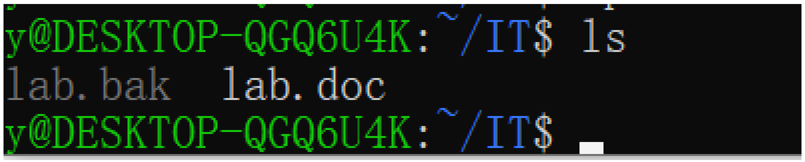
```
y@DESKTOP-QGQ6U4K:~$ cd IT
y@DESKTOP-QGQ6U4K:~/IT$ touch lab.doc
y@DESKTOP-QGQ6U4K:~/IT$ ls lab.doc
lab.doc
y@DESKTOP-QGQ6U4K:~/IT$ _
```

Make a backup of the ***lab.doc*** file with other name ***lab.bak***

- type: ***cp lab.doc lab.bak***

How to verify the new backup file and the original file are there?

Ls



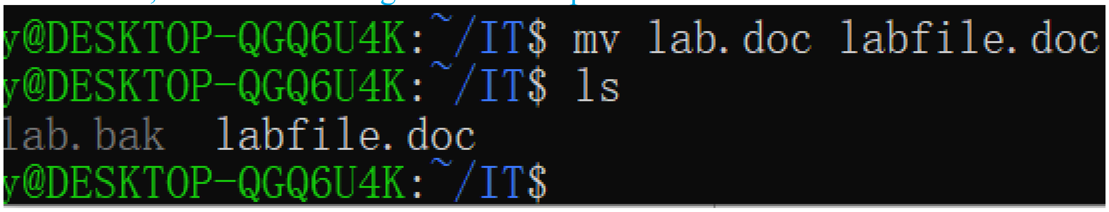
```
y@DESKTOP-QGQ6U4K:~/IT$ ls
lab.bak   lab.doc
y@DESKTOP-QGQ6U4K:~/IT$
```

To rename a file, you can use the **mv** command. Rename ***lab.doc*** to ***labfile.doc***

- type: ***mv lab.doc labfile.doc***

Did the file renamed? How do you know? ___________

Yes, listed name changed in   ls   output.



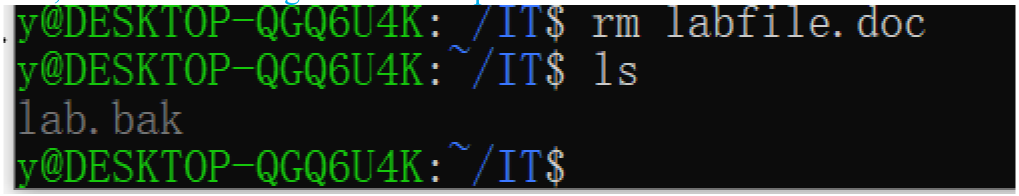
```
y@DESKTOP-QGQ6U4K:~/IT$ mv lab.doc labfile.doc
y@DESKTOP-QGQ6U4K:~/IT$ ls
lab.bak   labfile.doc
y@DESKTOP-QGQ6U4K:~/IT$
```

Delete files by using the **rm** command.

- type: ***rm labfile.doc***

Is the file deleted? How do you know? _____________

Yes, listed name changed in   ls   output.



```
y@DESKTOP-QGQ6U4K:~/IT$ rm labfile.doc
y@DESKTOP-QGQ6U4K:~/IT$ ls
lab.bak
y@DESKTOP-QGQ6U4K:~/IT$
```

6. Task : Delete a directory

To delete a directory in Linux, use the **rmdir directory_name** command.

Delete the IT directory

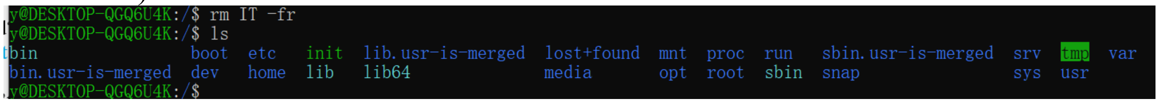
- type: ***rmdir it***

Is the directory deleted? NO
Why? Because it is not empty

If your directories can't be deleted, it might be because the directory is not empty. To remove non-empty directory (and also for empty directory), use **rm -r**

- type: ***rm IT -r***
- type Y to confirm the removal of the it directory and all its subdirectories

(Tips: The deletion process will skip the question if using **–f** switch (force) e.g. **rm IT -fr**)



```
y@DESKTOP-QGQ6U4K:/$ rm IT -fr
y@DESKTOP-QGQ6U4K:/$ ls
bin                boot  etc   init  lib.usr-is-merged  lost+found  mnt  proc  run   sbin.usr-is-merged  srv  tmp  var
bin.usr-is-merged  dev   home  lib   lib64              media       opt  root  sbin  snap                sys  usr
y@DESKTOP-QGQ6U4K:/$
```

Verify directory removal. Is the directory still exists? How do you know?

NO , listed name changed in   ls   output.

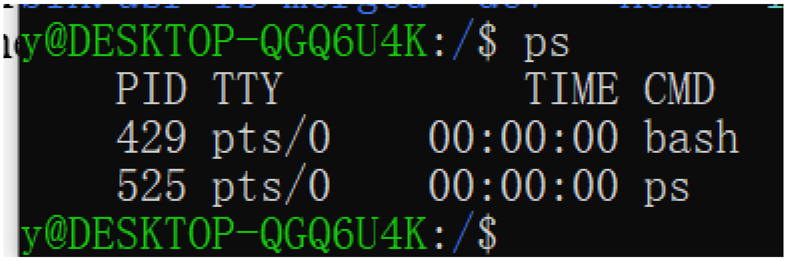
7. Task : View currently running process

To view currently running process(es) in user space, use the **ps** command.

- type: **ps**

How many process(es) is/are currently running?

2



```
y@DESKTOP-QGQ6U4K:/$ ps
    PID TTY          TIME CMD
    429 pts/0    00:00:00 bash
    525 pts/0    00:00:00 ps
y@DESKTOP-QGQ6U4K:/$
```

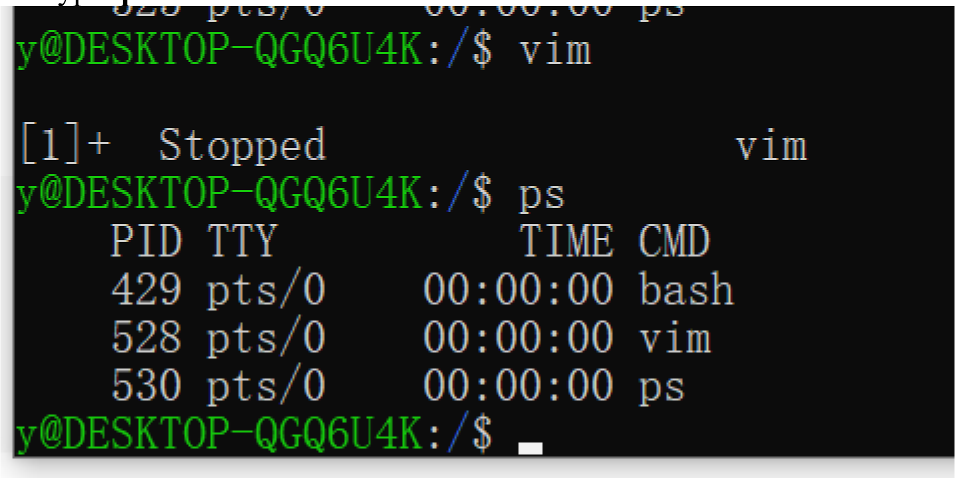
A current process can be suspended by pressing **Ctrl+Z** (e.g. open an application and then pressing Ctrl+Z will cause the application to stop and display the shell).

- type: **vim**

this will run the vim text editor. Press Ctrl+Z.

To view the stopped process,

- type: **ps**



```
y@DESKTOP-QGQ6U4K:/$ vim

[1]+  Stopped                 vim
y@DESKTOP-QGQ6U4K:/$ ps
    PID TTY          TIME CMD
    429 pts/0    00:00:00 bash
    528 pts/0    00:00:00 vim
    530 pts/0    00:00:00 ps
y@DESKTOP-QGQ6U4K:/$ _
```

When user suspend a process, the process can be brought back by using the command **fg**. To bring back the Vim editor,

- type: **fg**

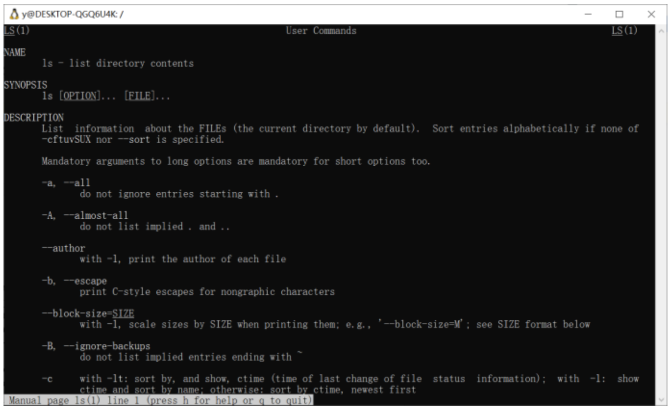
(to exit vim - press Esc, then type :q and press Enter)

8. Task : Exploring other commands

Students are encouraged to search and practice other commands that are provided by the command prompt shell to have more exposure to the shell (search from the documentation, Internet, book etc), or study further the commands already learned. Use **man <command>** to get help about a specific command e.g. **man ls** or use --help switch e.g. **`ls -help`** , and answer the following question (answer either one):

Explain two new options/switches for any two commands that you have learned by showing examples how you used the commands with the switches.

Man ls" shows like this



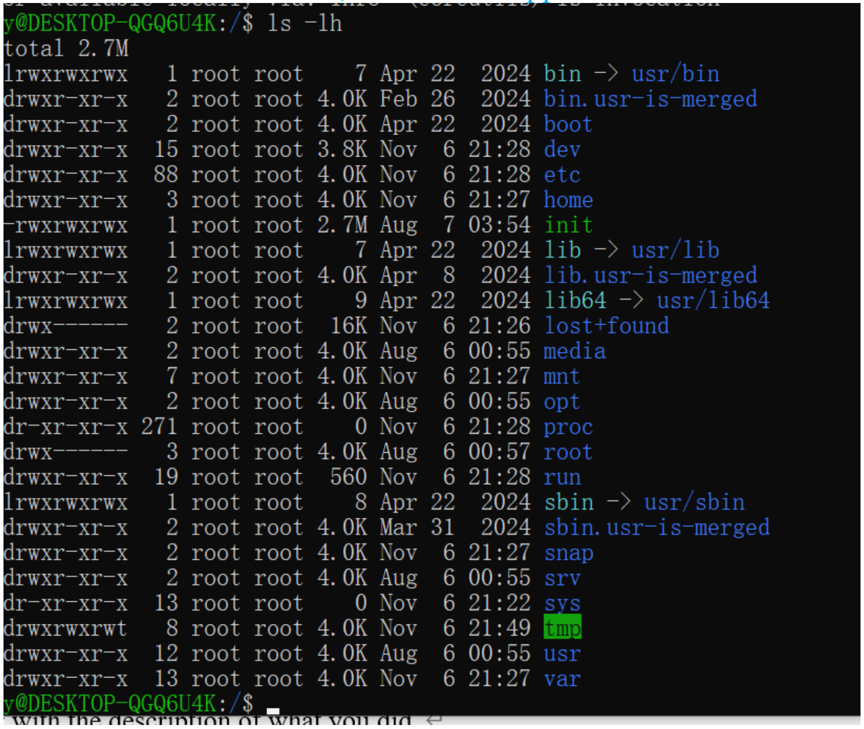
y@DESKTOP-QGQ6U4K: /

```
LS(1)                                  User Commands                                  LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically if none of
       -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last change of file  status  information);  with  -l:  show
              ctime and sort by name; otherwise: sort by ctime, newest first
 Manual page ls(1) line 1 (press h for help or q to quit)
```

ls -lh

-l shows files in a detailed (long) list format.

-h means *human-readable*, so file sizes appear as KB/MB/GB instead of raw bytes



```
y@DESKTOP-QGQ6U4K:/$ ls -lh
total 2.7M
lrwxrwxrwx   1 root root    7 Apr 22  2024 bin -> usr/bin
drwxr-xr-x   2 root root 4.0K Feb 26  2024 bin.usr-is-merged
drwxr-xr-x   2 root root 4.0K Apr 22  2024 boot
drwxr-xr-x  15 root root 3.8K Nov  6 21:28 dev
drwxr-xr-x  88 root root 4.0K Nov  6 21:28 etc
drwxr-xr-x   3 root root 4.0K Nov  6 21:27 home
-rwxrwxrwx   1 root root 2.7M Aug  7 03:54 init
lrwxrwxrwx   1 root root    7 Apr 22  2024 lib -> usr/lib
drwxr-xr-x   2 root root 4.0K Apr  8  2024 lib.usr-is-merged
lrwxrwxrwx   1 root root    9 Apr 22  2024 lib64 -> usr/lib64
drwx------   2 root root  16K Nov  6 21:26 lost+found
drwxr-xr-x   2 root root 4.0K Aug  6 00:55 media
drwxr-xr-x   7 root root 4.0K Nov  6 21:27 mnt
drwxr-xr-x   2 root root 4.0K Aug  6 00:55 opt
dr-xr-xr-x 271 root root    0 Nov  6 21:28 proc
drwx------   3 root root 4.0K Aug  6 00:57 root
drwxr-xr-x  19 root root  560 Nov  6 21:28 run
lrwxrwxrwx   1 root root    8 Apr 22  2024 sbin -> usr/sbin
drwxr-xr-x   2 root root 4.0K Mar 31  2024 sbin.usr-is-merged
drwxr-xr-x   2 root root 4.0K Nov  6 21:27 snap
drwxr-xr-x   2 root root 4.0K Aug  6 00:55 srv
dr-xr-xr-x  13 root root    0 Nov  6 21:22 sys
drwxrwxrwt   8 root root 4.0K Nov  6 21:49 tmp
drwxr-xr-x  12 root root 4.0K Aug  6 00:55 usr
drwxr-xr-x  13 root root 4.0K Nov  6 21:27 var
y@DESKTOP-QGQ6U4K:/$
```

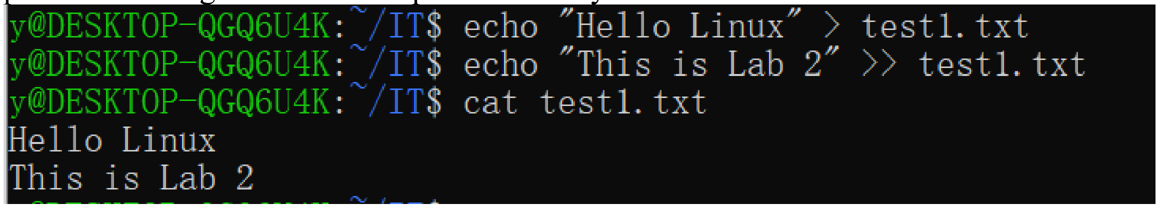
with the description of what you did

The -lh switch combination helps users view detailed file info with readable file sizes. This makes it easier to check which files occupy more space.

cp -i	ask before overwriting files
mkdir -p	create parent directories automatically
ps -u	show processes for a specific user
grep -n	show line numbers for matches

OR

Explain two new commands that you learned by showing the list of operations that you performed along with the description of what you did.

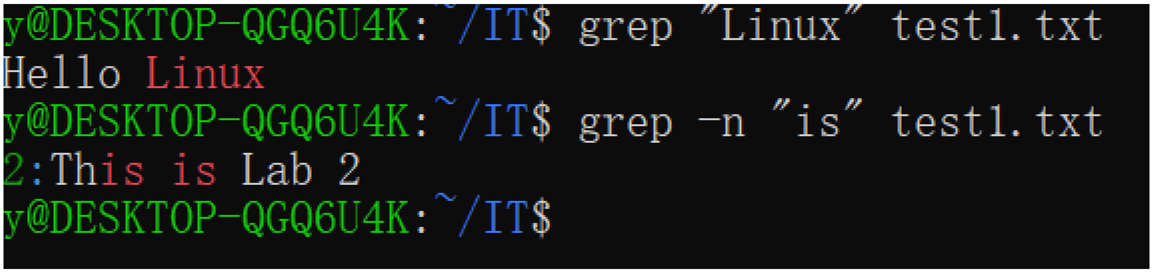


```
y@DESKTOP-QGQ6U4K:~/IT$ echo "Hello Linux" > test1.txt
y@DESKTOP-QGQ6U4K:~/IT$ echo "This is Lab 2" >> test1.txt
y@DESKTOP-QGQ6U4K:~/IT$ cat test1.txt
Hello Linux
This is Lab 2
```

The ***cat*** command is used to display the contents of a file directly in the terminal, combine multiple files, or redirect output into a new file.
In this example:

echo created and added text into test1.txt.
cat test1.txt printed the file content line by line on the screen.

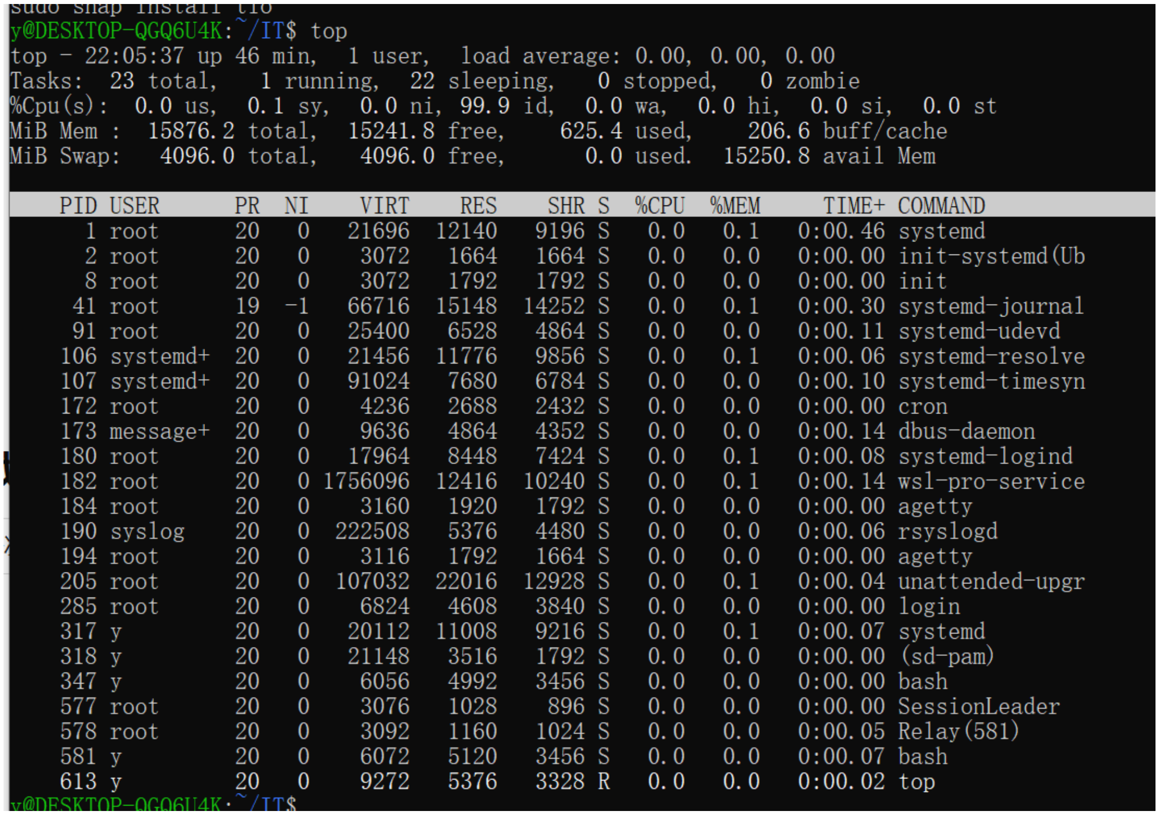
It is helpful to quickly check what's inside a file without opening a text editor like Vim or Nano.



```
y@DESKTOP-QGQ6U4K:~/IT$ grep "Linux" test1.txt
Hello Linux
y@DESKTOP-QGQ6U4K:~/IT$ grep -n "is" test1.txt
2:This is Lab 2
y@DESKTOP-QGQ6U4K:~/IT$
```

The ***grep*** command searches for specific words or text patterns inside a file.
The first command finds all lines containing the word "Linux."
The second command (-n) also shows the **line number** where the word appears.

It's useful when searching through configuration files or long text logs to find specific keywords quickly.



```
sudo snap install tio
y@DESKTOP-QGQ6U4K:~/IT$ top
top - 22:05:37 up 46 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  23 total,   1 running,  22 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.1 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15876.2 total,  15241.8 free,    625.4 used,    206.6 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  15250.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0   21696  12140   9196 S   0.0   0.1   0:00.46 systemd
      2 root      20   0    3072   1664   1664 S   0.0   0.0   0:00.00 init-systemd(Ub
      8 root      20   0    3072   1792   1792 S   0.0   0.0   0:00.00 init
     41 root      19  -1   66716  15148  14252 S   0.0   0.1   0:00.30 systemd-journal
     91 root      20   0   25400   6528   4864 S   0.0   0.0   0:00.11 systemd-udevd
    106 systemd+  20   0   21456  11776   9856 S   0.0   0.1   0:00.06 systemd-resolve
    107 systemd+  20   0   91024   7680   6784 S   0.0   0.0   0:00.10 systemd-timesyn
    172 root      20   0    4236   2688   2432 S   0.0   0.0   0:00.00 cron
    173 message+  20   0    9636   4864   4352 S   0.0   0.0   0:00.14 dbus-daemon
    180 root      20   0   17964   8448   7424 S   0.0   0.1   0:00.08 systemd-logind
    182 root      20   0 1756096  12416  10240 S   0.0   0.1   0:00.14 wsl-pro-service
    184 root      20   0    3160   1920   1792 S   0.0   0.0   0:00.00 agetty
    190 syslog    20   0  222508   5376   4480 S   0.0   0.0   0:00.06 rsyslogd
    194 root      20   0    3116   1792   1664 S   0.0   0.0   0:00.00 agetty
    205 root      20   0  107032  22016  12928 S   0.0   0.1   0:00.04 unattended-upgr
    285 root      20   0    6824   4608   3840 S   0.0   0.0   0:00.00 login
    317 y         20   0   20112  11008   9216 S   0.0   0.1   0:00.07 systemd
    318 y         20   0   21148   3516   1792 S   0.0   0.0   0:00.00 (sd-pam)
    347 y         20   0    6056   4992   3456 S   0.0   0.0   0:00.00 bash
    577 root      20   0    3076   1028    896 S   0.0   0.0   0:00.00 SessionLeader
    578 root      20   0    3092   1160   1024 S   0.0   0.0   0:00.05 Relay(581)
    581 y         20   0    6072   5120   3456 S   0.0   0.0   0:00.07 bash
    613 y         20   0    9272   5376   3328 R   0.0   0.0   0:00.02 top
y@DESKTOP-QGQ6U4K:~/IT$
```

The ***top*** command shows a **dynamic, real-time view of all running processes** on the system.

It displays information such as:Process ID (PID), CPU usage, Memory usage, User running the process, Command name