Lab 1: Linux Commands to Practice

Important Note: You are required to practice all the commands. Questions will be asked during lab evaluation.

File Commands		
0	clear	Clear screen
1	ls ls -l	Directory listing // listing in long listing format
2	ls -al	Formatted listing with hidden files
3	ls -lt	Sorting the Formatted listing by time modification
4	cd dir	Change directory to dir
5	cd	Change to home directory
6	pwd	Show current working directory
7	mkdir dir	Creating a directory dir
8	cat >file	Places the standard input into the file
9	more file	Output the contents of the file
10	head file	Output the first 10 lines of the file
11	tail file	Output the last 10 lines of the file
12	tail -f file	Output the contents of file as it grows,starting withthe last 10 lines
13	touch file	Create or update file
14	rm file	Deleting the file
15	rm -r dir	Deleting the directory
16	rm -f file	Force to remove the file
17	rm -rf dir	Force to remove the directory dir
18	cp file1 file2	Copy the contents of file1 to file2
19	cp -r dir1 dir2	Copy dir1 to dir2;create dir2 if not present
20	mv file1 file2	Rename or move file1 to file2,if file2 is an existingdirectory
21	wget [url]	Download the url contents (if downloadable)

22	man [command]	read the online manual page for a command
23	whatis [command]	Give brief description of a command
24	unzip file unzip file -d dest	Unzip the contents of the zipped file
25	zip -r dest.zip source	Zip source contents recursively in the destination

Searching		
1	grep pattern file	Search for pattern in file
2	grep -r pattern dir	Search recursively for pattern in dir
3	locate file	Find all instances of file
4	find . -name filename	Searches in the current directory (represented by a period) and below it, for files and directories with names starting with filename

Compression		
1	tar cf file.tar file	Create tar named file.tar containing file
Shortcuts		
1	ctrl+c	Halts the current command
2	ctrl+z	Stops the current command, resume with fg in theforeground or bg in the background
3	ctrl+d	Logout the current session, similar to exit
4	ctrl+w	Erases one word in the current line
5	ctrl+u	Erases the whole line
6	ctrl+r	Type to bring up a recent command
7	!!	Repeats the last command
8	exit	Logout the current session
9	Ctrl+_ then ctrl+v	In nano text edited these commands are used to move to the end of the file

Run C Files using GCC Compiler

Step 1:
If your computer has no installation of gcc, install it first with following commands.

```
sudo apt-get update
sudo apt-get install gcc
```

Step 2:
Write your C code.

```
nano your_program.c
```

if file is not created already, create it using appropriate command (Mentioned above).

Step 3:
Compile your C Code

```
gcc -o output_executable your_program.c
```

Step 4:
Run the executable

```
./output_executable
```

Reading Material

Command Line arguments:

Command-line arguments are a way to pass data to the program. Command-line arguments are passed to the main function. Suppose we want to pass two integer numbers to the main function of an executable program called a.out. On the terminal write the following line:
./a.out 1 22

./a.out is the usual method of running an executable via the terminal. Here 1 and 22 are the numbers we passed as command-line arguments to the program. These arguments are passed to the primary function. In order for the main function to be able to accept the arguments, we have to change the signature of the primary function as follows:

int main(int argc, char *arg[]);

argc is the counter. It tells how many arguments have been passed. arg is the character pointer to our arguments. argc in this case will not be equal to 2, but it will be equal to 3. This is because the name ./a.out is also passed as command line argument. At index 0 of arg we have ./a.out; at index 1 we have 1; and at index 2 we have 22. Here 1 and 22 are in the form of character string, we have to convert them to integers by using a function atoi.

Suppose we want to add the passed numbers and print the sum on the screen:
cout << atoi(arg[1]) + atoi(arg[2]);