

# COMP2700 LAB 1: VIRTUAL MACHINE SETUP AND BASIC LINUX COMMANDS

This lab aims to introduce students to the virtual lab environment that we will use for the first half of this course and Assignment 1. Upon the completion of this lab, the students will be able to:

- Connect to a virtual machine (VM) hosted on Microsoft Azure Labs cloud service, containing an installation of the operating system Ubuntu 20.04, that will be used in subsequent labs and the assignment related to operating system and software security.
- Perform simple tasks related to file system navigation, querying system and users information, and switching users, in the bash command line interpreter in Linux.

Discussions outside the lab sessions will be hosted on the Piazza forum for this course.

## TASK 1: CONNECTING TO THE LAB VIRTUAL MACHINE

For the labs in Week 1 to Week 6, with the exception of Week 3, and for Assignment 1, we will use a virtual machine (VM) containing an installation of Ubuntu 20.04 LTS operating system. The VM has been customised specifically for this course and is meant to be used to demonstrate basic concepts surrounding access control in operating systems, software security and the interplay between the two.

## IMPORTANT NOTES ON THE SECURITY OF THE LAB VIRTUAL MACHINES

**You should assume that the lab VM is insecure**, so make sure you **do not put any confidential data in the VM**. Make sure you **use a secure password**, but **do not use your ANU password** or any passwords that you use for your other online accounts.

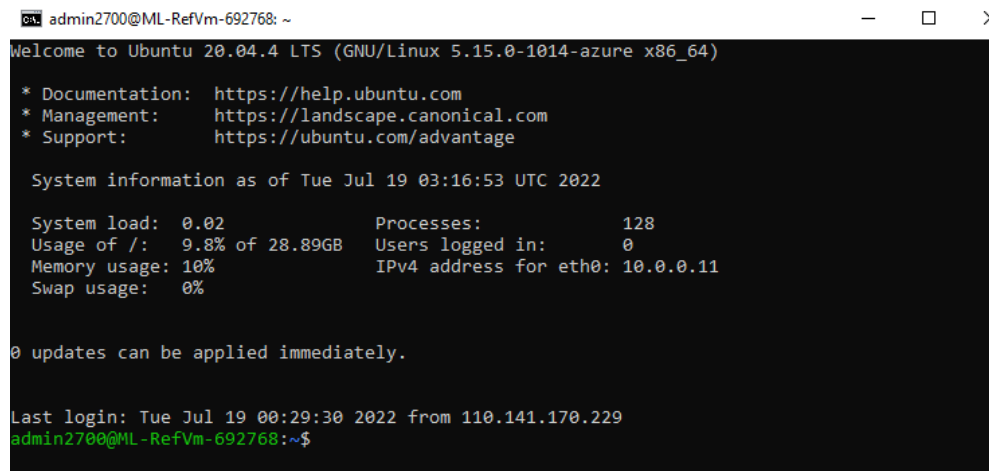
## CONNECTING TO THE VM

1. Launch an SSH client. Here are some suggestions of SSH client for different operating systems.
  - o **Windows 10** (latest update). The SSH client is already built in in the "Command Prompt" app in Windows. If your Windows machine does not already have SSH client built-in, you can use a third party tool such as [putty](#).
  - o **Linux**. Launch the terminal application. For example, if you use Ubuntu desktop, the Terminal application can be accessed from the applications menu, or using the shortcut Ctrl-Alt-T.
  - o **Mac**. Launch the Terminal app (from Applications > Utilities > Terminal.app).
2. Enter an SSH command of the following format:

```
ssh -p PORT USER@HOSTNAME
```

For the lab VMs the hostname is **localhost**, the username is **admin2700**, and the port number is **8888**.

3. Press 'Enter' to connect to the VM. If this is the first time you connect to your VM, you may see a warning about the authenticity of the host. Type 'yes' to continue with the connection.<sup>1</sup>
4. Type in the password for the admin2700 account. If this is your first time logging into this VM, the password will be **ANU\_comp2700**. You may be prompted to change the password. If you reset your password, you will need to login to the VM again using your new password. GOTO step 2).
5. Once you successfully log in, you will see a screen like the following:



```
admin2700@ML-RefVm-692768: ~  
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1014-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue Jul 19 03:16:53 UTC 2022  
  
System load:  0.02          Processes:            128  
Usage of /:   9.8% of 28.89GB Users logged in:          0  
Memory usage: 10%          IPv4 address for eth0: 10.0.0.11  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
Last login: Tue Jul 19 00:29:30 2022 from 110.141.170.229  
admin2700@ML-RefVm-692768:~$
```

You are now logged in to your lab VM. It is running a bash shell. Try to run the following command in the shell:

```
cat welcome.txt
```

This should print a welcome message.

## DISCONNECTING FROM AND TURNING OFF THE LAB VM

When you are done with your tasks (after your lab session, for example), disconnect from the VM by typing the command 'exit' (you may need to execute this command more than once if you run nested login sessions). To log in back to the VM, you can type in the SSH command you used previously (see step 2-3 above).

Note that disconnecting from the VM does not turn the VM off. The VM is still running in the background, so make sure that you turn off the VM when you are not using it. To do so, close the window in which your VM is running. A dialog box will appear offering you three choices. You should select “Send the shutdown signal” to safely terminate the process.

---

<sup>1</sup> Note that it is generally a poor security practice to proceed connecting to an SSH server if you are presented with this warning of the authenticity of the host, unless you have another way of verifying its authenticity.

## TASK 2: TRANSFERRING FILES TO AND FROM THE LAB VM

To transfer files in and out of the VM, you can use the Secure FTP protocol. This file transfer protocol is supported natively in many operating systems, such as Windows, Linux and Mac OS. The command you need is “sftp”. The command to connect to the lab VM using sftp is very similar to ssh:

```
sftp -P PORT USER@HOSTNAME
```

where PORT, USER and HOSTNAME are the same parameters you used previously in Task 1. Note that the option for specifying the port number uses an upper-case ‘P’, instead of the lower-case ‘p’ in ssh.

Once you are connected to the lab VM, you will be presented with a “sftp>” prompt, in which you can type commands. Two basic useful commands are get (to download files from the VM) and put (to upload files to the VM). Here are some examples of sftp commands, assuming you are connected to the VM as “admin2700” users:

```
sftp> get welcome.txt
```

This command will retrieve the file /home/admin2700/welcome.txt. By default, sftp uses the home directory of the logged in user (admin2700) as the current directory in the VM.

```
sftp> put myfile.txt
```

This command will upload the file “myfile.txt” in the current directory of your computer to the lab VM. The file “myfile.txt” will be copied to /home/admin2700/myfile.txt.

To exit sftp, use the command “bye”.

You can use the “help” command to display information about other useful sftp commands.

**Exercise 1.** Try the commands above to copy the 'welcome.txt' file from the lab VM to your local machine, and try add a new file to the lab VM.

get  
put

## TASK 3: BASIC LINUX COMMANDS

In this task, we will be looking at some basic commands in Linux. This is meant to provide a minimum background necessary to follow more advanced lectures and labs related to Unix/Linux security later in the course; it is not meant to be a crash course on Linux administration. If you are already familiar with the command line interface of Linux, you can safely skip this part.

Students are assumed to have some knowledge of files and directory concepts in operating systems, and some experience in programming languages, so commands and language constructs will not be given in great detail, and some self-study will be required to learn more advanced features.<sup>2</sup>

On the completion of this task, students are expected to execute shell commands for file system navigation, basic file operations such as listing, deleting, moving, renaming files and executing programs from shell.

More advanced commands will be covered in Lab 2.

### TASK 3.1: SWITCHING USERS

The lab VM comes with some pre-configured users. Their usernames and passwords are as follows:

- User admin2700: this is the administrative account, which means that it can leverage its permission to the 'root' user, which will have unlimited access to the lab VM (including altering system settings). The lab VM is configured so that only admin2700 can log in remotely – all the other users below will be blocked from log in remotely for security reasons. The password to this account is the one you set up when you connect to the lab VM for the first time.
- The remaining users are non-administrative users (but as we shall see later, user 'alice' holds a special privilege; we will come back to this in the lab on Unix security), and their user names are as follows: alice, bob, charlie, dennis, eve and felix. The passwords for these users are their usernames followed by the string '123'. For example, alice's password is alice123.

When you connect to the lab VM, you log in as user 'admin2700'. You can switch to another user using the bash command 'su'. For example, to login as user alice, run

```
su -l alice
```

You will be prompted for a password; enter alice's password (alice123) to log in as user alice. The command 'whoami' can be used to check the currently logged in user. The command 'exit' will terminate the current bash session. If you used su -l to login previously, the effect of 'exit' is to go back to the bash session of 'admin2700'.

**Exercise 2.** Use the 'su -l' command to log in as other non-administrative users. Verify that you have logged in correctly. Then exit back to the 'admin2700' user.

---

<sup>2</sup> See for example, <https://www.tutorialspoint.com/unix/index.htm>, for a more comprehensive tutorial on Linux.

## TASK 3.2: BASIC LINUX COMMANDS

For this task, we assume you have already setup the lab VM and connected to it as user admin2700. There are a number of exercises for this task. **Exercises marked with (\*) indicate extension exercises and will not be covered during this lab session.** However, students are encouraged to try them and discuss with the tutor if needed.

This lab comes with some files and directory that we will use to demonstrate the commands below. You will first need to install these files in the lab VM; this is done as follows:

- Login to the lab VM you set up in Task 1. By default this will log you in as the user 'admin2700'.
- Switch to user 'alice' using the 'su -l' command as in Exercise 2:

```
su -l alice
```

- Download the file **lab1.tar.gz** and extract it using the following commands:

```
wget http://users.cecs.anu.edu.au/~tiu/comp2700/lab1.tar.gz
extract-lab lab1.tar.gz
```

- This will extract the files for Lab 1 in a folder called **lab1** in Alice's home directory. We will use this directory to illustrate some basic linux commands in the following exercises.

All the commands discussed here are part of the 'shell' commands. In Unix systems, there are a family of shell interpreters; for Linux, the most common one is the bash shell, which we will use to run the commands in this lab.

In the following, we assume that you have successfully logged in as user 'alice' and have extracted the lab1 folder, which will be located in Alice's home directory /home/alice/.

## GETTING HELP

You can obtain more information about a particular command by using the 'man' command.

**man *command-name***

For example, typing

**man ls**

will bring up the manual page for the command 'ls' (see below).

## FILES AND DIRECTORIES

In Linux, everything starts from the root directory (or folder). It is denoted by a single slash '/'. You can usually see the directories name under the root folder as in Table 1. The user directories are located under the folder /home.

Table 1. Some standard directories in Ubuntu

/bin/	essential user command binaries
/boot/	static files of the boot loader
/dev/	device files
/etc/	host-specific system configuration
/home/	user home directories
/lib/	essential shared libraries and kernel modules
/media/	mount point for removable media
/mnt/	mount point for a temporarily mounted file systems
/opt/	add-on application software packages
/sbin/	system binaries
/tmp/	temporary files
/usr/	user utilities and applications
/var/	variable files
/root/	home directory for the root user
/proc/	virtual file system documenting kernel and process status as files

## GETTING THE CURRENT WORKING DIRECTORY

### Syntax:

**pwd**

**print working directory:** writes the full pathname of the current working directory to the standard output.

In this case, running this command should give you **/home/alice** as the output.

## LISTING FILES

### Syntax:

**ls [-options] [regular expression of filenames/directories]**  
list files in the current or specified working directory (or directories).

Some useful shortcuts to directories:

- . (dot): This denotes the current directory.
- .. (two dots): This denotes the parent directory of the current directory.
- ~ (tilde): this denotes the home directory of the current user.

The command `ls` by itself (without any arguments)

**ls**

lists the file in the current directory, i.e., it is the same as "`ls ./`" (without the quotes).

The option "`-a`" instructs "`ls`" to list all file types, including the hidden files, i.e., those files/directories whose name start with a "." (dot). For example:

**ls -a**

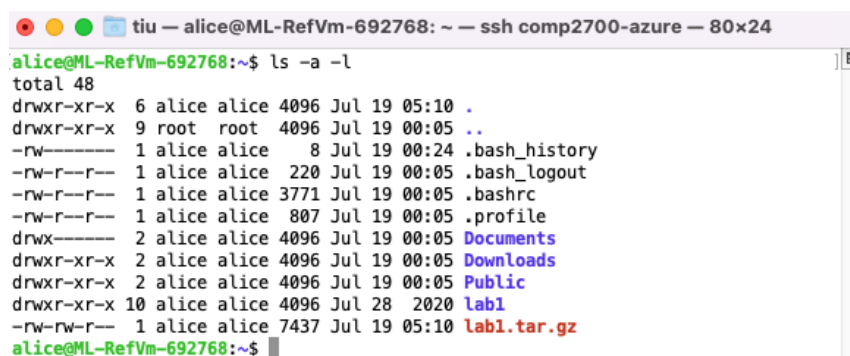
lists all files/folders (including hidden ones) in the current directory.

The option "`-l`" outputs the list of file in a long listing format. For example:

**ls -l**

lists all files/directories in the current directory, with various details such as file size, creation dates, permissions, etc shown in the output.

We can combine two or more options by listing them next to each other. For example: to list all files in the current directory, including hidden files/directories, in a long listing format, we can use `ls -a -l`



```
tiu - alice@ML-RefVm-692768: ~ - ssh comp2700-azure - 80x24
alice@ML-RefVm-692768:~$ ls -a -l
total 48
drwxr-xr-x 6 alice alice 4096 Jul 19 05:10 .
drwxr-xr-x 9 root root 4096 Jul 19 00:05 ..
-rw-r--r-- 1 alice alice 8 Jul 19 00:24 .bash_history
-rw-r--r-- 1 alice alice 220 Jul 19 00:05 .bash_logout
-rw-r--r-- 1 alice alice 3771 Jul 19 00:05 .bashrc
-rw-r--r-- 1 alice alice 807 Jul 19 00:05 .profile
drwxr-xr-x 2 alice alice 4096 Jul 19 00:05 Documents
drwxr-xr-x 2 alice alice 4096 Jul 19 00:05 Downloads
drwxr-xr-x 2 alice alice 4096 Jul 19 00:05 Public
drwxr-xr-x 10 alice alice 4096 Jul 28 2020 lab1
-rw-r--r-- 1 alice alice 7437 Jul 19 05:10 lab1.tar.gz
alice@ML-RefVm-692768:~$
```

Multiple options can also be combined using a single '-', instead of listing them separately. For example, the command "`ls -a -l`" (without the quotes) can also be shortened to "`ls -al`" (without the quotes).

### Using wildcards with ls

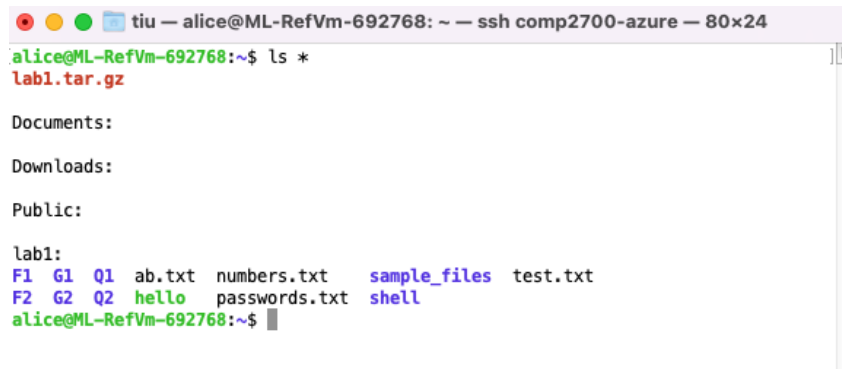
The `ls` command accepts a simple form of "wildcard" expressions to specify patterns of files or directories we want to list. This is usually called "[globbing](#)". The asterisk character "\*" in an expression can be matched with

*almost* any strings (including the empty string) – one exception is that it cannot be matched with a '.' (dot) that appears in the beginning of a file name. The question mark character "?" in a pattern that can be match with exactly one character.

For example:

```
ls *
```

will list all the files and directories, and the files/directories in all the subdirectories of the current directory. Note that it does not however list the hidden files/directories (whose names start with a dot).



```
alice@ML-RefVm-692768: ~$ ls *
lab1.tar.gz

Documents:

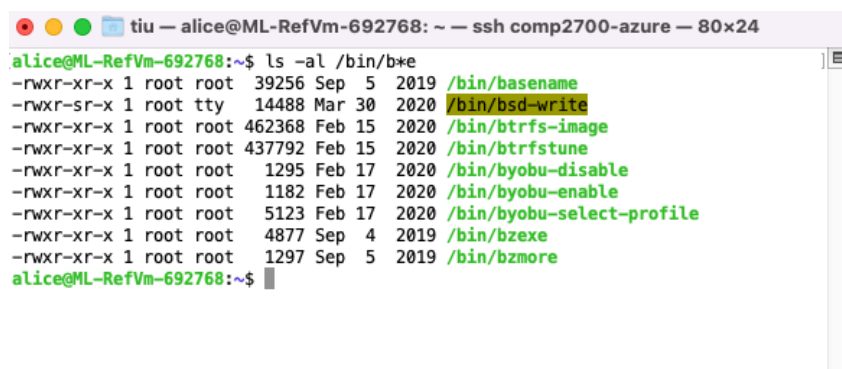
Downloads:

Public:

lab1:
F1 G1 Q1 ab.txt numbers.txt sample_files test.txt
F2 G2 Q2 hello passwords.txt shell
alice@ML-RefVm-692768:~$
```

Here's another example:

```
ls -al /bin/b*e
```



```
alice@ML-RefVm-692768:~$ ls -al /bin/b*e
-rwxr-xr-x 1 root root 39256 Sep  5 2019 /bin/basename
-rwxr-sr-x 1 root tty 14488 Mar 30 2020 /bin/bsd-write
-rwxr-xr-x 1 root root 462368 Feb 15 2020 /bin/btrfs-image
-rwxr-xr-x 1 root root 437792 Feb 15 2020 /bin/btrfs-tune
-rwxr-xr-x 1 root root 1295 Feb 17 2020 /bin/byobu-disable
-rwxr-xr-x 1 root root 1182 Feb 17 2020 /bin/byobu-enable
-rwxr-xr-x 1 root root 5123 Feb 17 2020 /bin/byobu-select-profile
-rwxr-xr-x 1 root root 4877 Sep  4 2019 /bin/bzexe
-rwxr-xr-x 1 root root 1297 Sep  5 2019 /bin/bzmore
alice@ML-RefVm-692768:~$
```

It lists all files and directories in the directory /bin/ whose names start with 'b' and ends with 'e'.

We can also list multiple directories in one "ls" command, e.g.,

```
ls -al /bin/ /etc/
```

will list all files/directories in /bin and /etc in a long format.

### Exercise 3.

- List all the files in /bin directory that has at least 3 characters in their names.
- List all the files in /bin directory that has at most 3 characters in their names.



## CHANGING THE CURRENT WORKING DIRECTORY

### Syntax:

#### ***cd Destination-Directory***

Change **directory** is used to change the current working directory.

Examples: assuming you are currently in the home directory of alice, i.e., /home/alice. Then

#### **cd Documents**

will change the current directory to /home/alice/Documents.

Here are some shortcuts that are useful for navigating directories:

. ('dot'): The character '.' (dot) represents the current directory.

.. (two dots): Two dot characters in sequence represent the parent directory.

~ (tilde): This represents the home directory of the user.

For example, the following command will navigate to the parent of the parent directory of the current directory. This operation is equivalent to doing **cd ..** twice.

**cd ../../**

The following command lead change the current directory to /home/alice/lab1.

**cd ~/lab1**

**cd ~/ Documents/**

**Exercise 4.** Change your current directory to the subdirectory ~/lab1/G1/. Assuming you are logged in as user alice, what is a shortest command you can use to navigate to /home/alice/Documents/ ?

*Hint: You may also use wildcards with the cd command; it will work if the pattern matches exactly one directory name.*

## CREATING NEW DIRECTORIES

### Syntax:

#### ***mkdir directory-name***

**Make directory** will create a new directory under the current working directory. E.g.

Example:

**mkdir testdir**

This will create a new directory called 'testdir'.

## COPYING, MOVING AND RENAMING FILES AND DIRECTORIES

Syntax:

***mv [options] source destination***

move files or directories from one place to another. If the **destination** file/directory does not exist, the **source** file/directory will be **renamed** to the name of the **destination**.

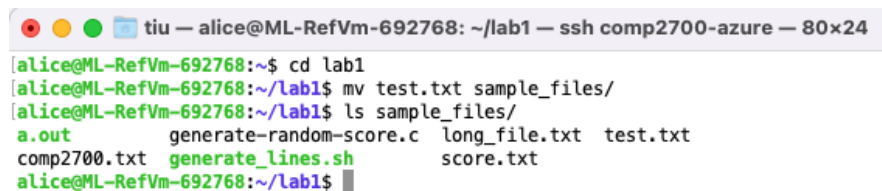
***cp [options] source destination***

copy files or directories from one place to another.

Example: Change to the /home/alice/lab1 directory and run the following:

```
mv test.txt sample_files/
```

This will move the test.txt file from the current working directory to sample\_files/ folder.



```
tiu — alice@ML-RefVm-692768: ~/lab1 — ssh comp2700-azure — 80x24
[alice@ML-RefVm-692768:~$ cd lab1
[alice@ML-RefVm-692768:~/lab1$ mv test.txt sample_files/
[alice@ML-RefVm-692768:~/lab1$ ls sample_files/
a.out      generate-random-score.c  long_file.txt  test.txt
comp2700.txt generate_lines.sh        score.txt
```

To rename “file1” to “file2”, we can use the command:

```
mv file1 file2
```

Note that if file2 exists, its content will be overwritten by the content of file1.

Here are a couple of examples for copying files/directories:

```
cp sample_files/test.txt ./
cp -r Q1 sample_files/
```

This first command will copy the test.txt file located under sample\_files to the current working directory. The second command will copy the directory Q1 to sample\_files/. The option -r specifies that the cp command should copy the source directory recursively (including all files and subdirectories).

**Exercise 5.** Using the cp command, create an exact copy of the directory lab1 and call it lab1a. Then rename lab1a to lab1b using the mv command.

```
cp -r lab1 lab1a
mv lab1a lab1b
```

**Exercise 6 (\*).** The file copy command 'cp' does not issue a warning when the target file exists. For example, "cp file1 file2" will override the content of file2 with the content of file1 without warning. However, the cp command comes with a variety of options that allow one to implement the following behaviour:

- Suppose we want to copy "file1" to "file2". First, check whether "file2" already exists. If it does, rename "file2" to "file2.bak". This ensures that the old "file2" can still be recovered (i.e., it is now called "file2.bak") if needed. Then perform the copying of "file1" to "file2".

Try to implement this behaviour as a one-line command using cp with the appropriate option(s). Your command has to work for arbitrary files.

Use the 'man' page to find the suitable option(s) to use with the cp command.

## DELETING FILES AND DIRECTORIES

**Syntax:**

***rm [options] filename-or-directory***

Remove file or directories.

Examples:

**rm text.txt**

This will remove the file named text.txt under the current working directory.

**rm -rf testdir**

This command simply forces to remove the directories we just created recursively. We have the -f option here such that we don't have to type 'yes'/'no' whenever Linux asks us for confirming the deletion.

**Note: be cautious when using the command 'rm -rf'. For example, in some linux systems, running 'rm -rf /' as a root user (administrator) will delete your entire system without warning!**

The argument [filename/directory] can be patterns as well, just as in the ls command.

**Exercise 7.** Remove all the subdirectories in ~/lab1b/ (created in Exercise 5) whose names start with 'F', using a single command. What command would you use?

## DISPLAYING THE CONTENT OF A FILE

### Syntax:

#### ***cat [option] filename***

**cat** command can be used for creating files, viewing content of files (mostly), concatenating files, etc.

#### ***head [option] filename***

**head** command outputs the first part of files. By default, head returns the first ten lines

#### ***tail [option] filename***

**tail** command outputs the last few lines of files.

#### ***less***

**less** command is used to view the files progressively. It is useful when the file or the output is long.

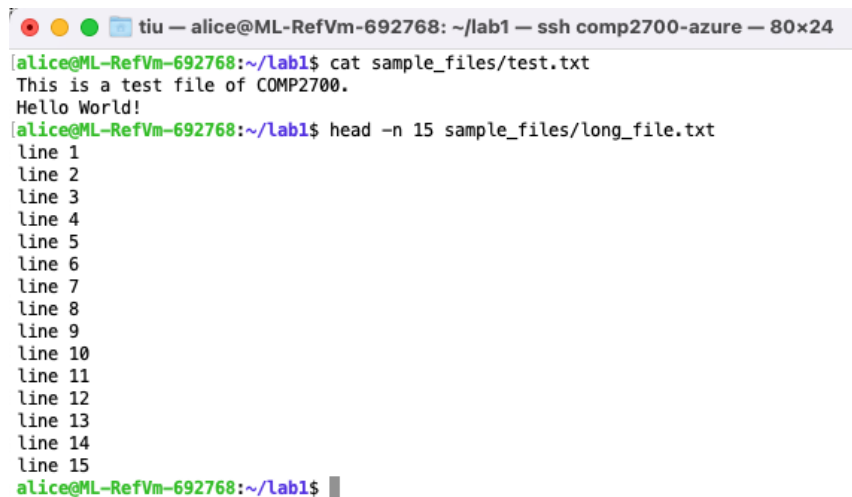
### Examples:

```
cat sample_files/test.txt
head -n 15 sample_files/long_file.txt
tail sample_files/long_file.txt
```

Running 'cat' with several file arguments will display the content of those files to the standard output (screen). For example,

```
cat ~/lab1/*.txt
```

will display the contents of all files that end with the extension '.txt' in the lab1 folder.

A terminal window titled 'tiu — alice@ML-RefVm-692768: ~/lab1 — ssh comp2700-azure — 80x24'. The prompt is 'alice@ML-RefVm-692768:~/lab1\$'. The first command is 'cat sample\_files/test.txt', which outputs 'This is a test file of COMP2700.' and 'Hello World!'. The second command is 'head -n 15 sample\_files/long\_file.txt', which outputs lines 1 through 15 of the file, each labeled 'line 1' through 'line 15'. The prompt returns to 'alice@ML-RefVm-692768:~/lab1\$'.

**Exercise 8.** Display the contents of all the files in the subdirectories F1 and F2 in lab1 using one command.