CISC 181 Lab 7 The command line interface

This lab will require some independent research on your part. You may already know some of the answers, but even if you do, in some cases I want you to find another authority online that supports your answer and tell us, by way of their web address, who they are.

The first one should be easy for you.

1. (1 mark) What is the name of your computer's operating system (e.g., Ubuntu)?

MacOS

1. (3 marks) Your operating system (OS) will come with at least one text-based, command line interface (CLI) program. What is the name of one such program on your computer and what steps must you take to start it? For this problem, provide a web source for the information. Also, there will likely be more than one way to start the CLI, so just pick one that you can find a web source for. The table cells should expand as needed to hold the information you type or paste into them.

|  |  |
| --- | --- |
| CLI program's name: | Terminal |
| Directions for starting: | Open the Terminal app |
| Web source for this information: | https://support.apple.com/en-ca/guide/terminal/apdb66b5242-0d18-49fc-9c47-a2498b7c91d5/mac |

Start your computer's CLI and leave its window open for the rest of this lab. Enlarge it if you like, but do not make it any smaller.

1. (1 mark) We are all used to graphical user interface (GUI) programs that show us the contents of folders on our computers. On Windows, the program that shows us what is in a folder is called *File Explorer* (sometimes referred to as simply *Explorer*, not to be confused with Microsoft's old web browser, Internet Explorer). *Finder* provides similar functionality on macOS. The Linux distro Ubuntu has its own GUI file explorer, called *Files*, that looks like a cross between File Explorer and Finder. All three programs offer high-level views of the secondary storage *file system* on their respective computers. These file systems are all hierarchical: For each file device (drive) there is a so-called *root folder*, and this root folder contains other folders and (probably) files. The folders in the root folder likely contain more folders and files, and so on, with the whole file system arranged logically in a tree structure that grows outward from the root.

The pre-GUI name for a folder was *directory* and this term still appears in the context of your OS's CLI. It is the term I will use here.

Just as your GUI shows (typically) the content of one directory at a time, your CLI keeps track of where on your file system's hierarchy of directories it is currently focused. This location is variously called the "working directory" or "current directory." It is obvious in the Windows CLI what the working directory is at any moment because it shows on the line where anything you type appears. Here is what that line looks like in my own PC's CLI:



This tells me that my working directory is on my first hard disk (which Windows calls C:, pronounced "C drive") and is called \Users\richard. The root directory on a Windows secondary storage device is called \ which is the **backslash** character. My C drive's root directory contains, among other things, a directory called Users and this in turn contains the directory called richard (my username on my PC). Note that backslashes are used to separate directories from their contents.

On non-Windows machines, the **slash** (or **forward slash**) character (/) is used instead of the backslash, so the root directory on a Mac, for example, is called /. By default, the Unix-like OS CLIs don't show what the working directory is, but this can be revealed by typing

pwd

into the CLI and pressing Return (or Enter). "pwd" is short for "print working directory." Pressing **Return or Enter is how you tell a CLI that you have finished typing a command and want to execute it.**

What is the current/working directory in your CLI when you first open it?

/Users/petergelgor

1. (2 marks) The CLI you are working with no doubt has a way to list the contents (the names of files and directories, possibly with additional information) of the current/working directory. Even if you know what the appropriate command is for your computer's operating system, give it below and the address of a web page that contains the information.

|  |  |
| --- | --- |
| Directory listing command: | ls |
| Web source for this information: | https://www.cnet.com/tech/computing/terminal-fun-options-for-printing-folder-and-subfolder-contents/ |

1. (2 marks) Some CLI commands are the same in Windows and the Unix-like OSs. The command to create a directory in Windows, macOS, and in Linux, for example, is

mkdir *target*

where *target* is the name of the directory you would like to create. (Windows users can use md instead of mkdir if they like. Both commands do the same thing.) Similarly, to change the working directory, users of Windows, macOS and Linux can all use

cd *target*

A reasonable place to experiment with these commands and others is a new directory on your desktop, since anything you put there will be instantly seen in your OS's GUI. So, the next thing for you to discover with a web search (whether you need to perform one or not!) is the location of your desktop directory. Once you have found it, copy the **full path**[[1]](#footnote-1) to it into the table box below along with the address of your web source. By "full path", I mean something like \Users\richard, but indicating the location of your desktop directory instead. The full path starts with the \ or / (depending on your OS) and ends with the name of the desktop directory.

|  |  |
| --- | --- |
| Location of YOUR desktop directory: | /Users/petergelgor/Desktop |
| Web source for this information: | https://stackoverflow.com/questions/39649875/os-x-how-can-i-get-path-to-desktop-directory-on-macos |

1. (6 marks) Now that you know how to change the working directory in your CLI and you know the path to your desktop, use the CLI to change your working directory to your desktop. Windows users will be able to see that they are in the right place because the text on the input line in their CLI will change to show the new working directory. macOS or Linux users should confirm they are in the right place by using the pwd command again.

Next, create a new directory called **lab07**. It will be a sub-directory of your desktop and show up as a directory there in your GUI. To do this, execute the command

mkdir lab07

All being well, if you check your desktop in your GUI, you should see a new folder has appeared with the name "lab07".

Back in your CLI window, make the new lab07 directory your working directory with cd. This time, you do not need to include the whole path to the new directory. Since your working directory is your desktop, and you are "there" already, just cd into it like this:

cd lab07

And now that we have got some space to experiment in, let us make a file! There are many ways to do this, including the usual GUI ways you are used to (using an app like Word for example) but we just want to make a simple text file using the CLI itself. The way to do this varies a bit between OSs, so we will tackle macOS and Linux first.

Mac or Linux users: Type this, exactly as shown, into your CLI and press Return/Enter:

cat > test.txt

This will leave your CLI in a state where it is prepared to capture characters from the keyboard to write into a new file called test.txt. So, let us do that. Type this, exactly as shown, **then press Return/Enter**:

This is a test.

To stop capturing characters for the new file, signal that you want to stop by holding down your keyboard's control key while you press the D key (control-D). This should cause the file input to cease and the file to be created.

All right, Windows users: Type this, exactly as shown, into your CLI and press Enter:

copy con: test.txt

The copy command is usually used for copying files (cp is the Mac and Linux equivalent) but when used with the special con: device it takes input directly from the keyboard. "con" is short for "console," by the way, which is an antiquated term meaning a keyboard paired with a monitor or printer. Now type this, exactly as shown, and **do not press Enter** when you are done.

This is a test.

The Windows end-of-file key sequence is Ctrl-Z (hold down the Ctrl key while pressing the Z key), so apply that now. You should see a "^Z" appear at the end of "This is a test." like this:

This is a test.^Z

but be aware that typing the "^" character and then a "Z" will not get you the result we want; you must use Ctrl-Z.

Press Enter now and the file will be saved and appear on your Windows desktop.

All right, **macOS and Linux users**, execute the following command:

cat test.txt

That should dump the content of your test.txt file to your display. (Note that "cat" is short for "concatenate" because this command can be used to send multiple files to one long concatenated stream of output.)

**Windows users**, to get the same effect, execute this command:

type test.txt

Finally:

* take a screen shot of your CLI window (NOT your whole screen – do a search for information on how to capture a single window if you don't know how)
* save your screen shot as either a .jpg file or a .png file (again, research this if you don't know how) using the file name **Lab 07.jpg** or **Lab 07.png**
* Place your completed Lab 07.docx file and your screen shot file into a zip file called **Lab 07.zip** and submit it by the lab deadline.

To tidy up your machine after you've completed the lab, you can either figure out how to remove the lab07 folder from your desktop using the CLI, or just drag it to your Trash/Recycle Bin in your GUI. You can close your CLI window using its close button, or by using the

exit

command.

1. Another term for "full path" is *absolute path*. There are also *relative paths*, one of which is used with the instruction cd lab07 in problem 6 of this lab. [↑](#footnote-ref-1)