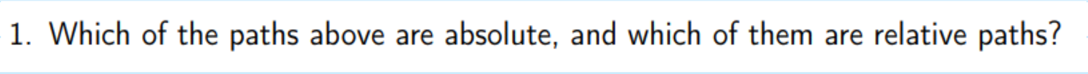
Lab 1

Thursday, 25 July 2024 2:32 PM



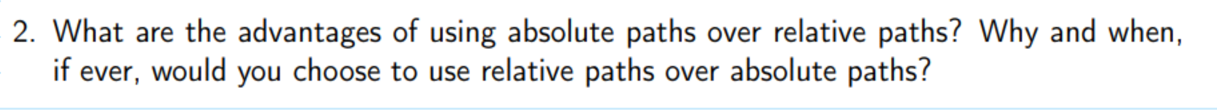
1. Which of the paths above are absolute, and which of them are relative paths?

Relative:

- FIT2100
- .
- ..
- ../../.././home/student

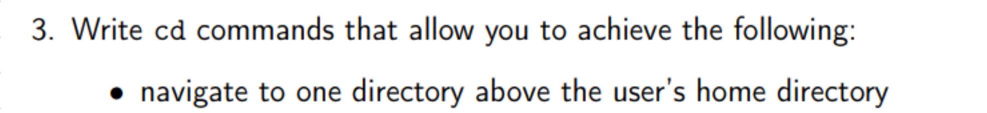
Absolute:

- /home/student/Documents
- ~/Documents/FIT2100
- /



2. What are the advantages of using absolute paths over relative paths? Why and when, if ever, would you choose to use relative paths over absolute paths?

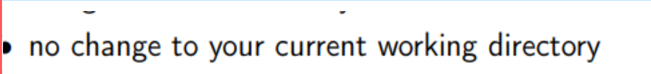
Relative paths are good if you are trying to get to a file in a folder that you are in and especially useful if you are very deeply nested in a folder and you just want to open a file in there. They are also good if you are programming on a repo that will sit on other computers as you can just send the whole repo to the other person and the application will work as it does not matter what their username or machine name is. If you chose an absolute path in this situation the application may not work. Absolute paths are good for if you want to get something from system files or you are deeply nested in the folder and need something from a completely different top level system folder, they are also more readable when the folder structure gets big.



3. Write cd commands that allow you to achieve the following:

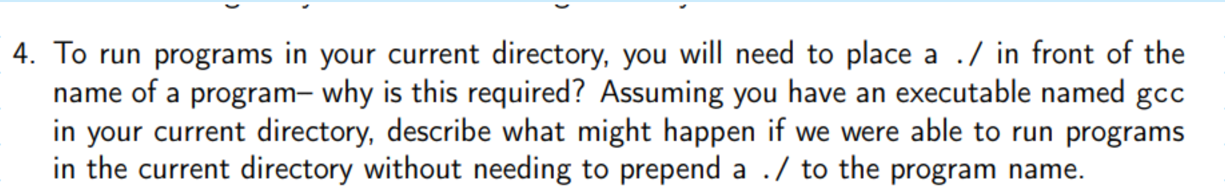
- navigate to one directory above the user's home directory

cd /home
Or cd ~
Which home directory depends



- no change to your current working directory

Cd .



4. To run programs in your current directory, you will need to place a ./ in front of the name of a program– why is this required? Assuming you have an executable named gcc in your current directory, describe what might happen if we were able to run programs in the current directory without needing to prepend a ./ to the program name.

It could run other programs with the same name but that are in a different directory as a file name does not have to be unique throughout the whole system. This could run the wrong program which is bad.