Linux file paths

- (Nearly?) anyplace you can specify a file or directory you can also include the path to that file or directory
- Paths can be expressed as absolute paths or relative paths
- Relative paths are evaluated using your current directory as the starting point
- Absolute paths are evaluated using the root of the system as the starting point

Absolute paths

- The root of the overall file system is indicated simply as /

  e.g. to switch from your current directory to the root use: **cd /**
- Within the root directory on our system you'll find subdirectories like bin, dev, lib, home, usr, etc
- Absolute paths have the full specification of how to get from the root to the desired file or directory. For example your own home directory is probably located along a path something like **/home/student/yourusername**
- If you enter the command **pwd** it will show you the

Bash and the ~ (tilde)

- When you enter paths that start with the tilde, ~, bash assumes what comes next is a username and expands/replaces the ~ with the absolute path to their home directory, e.g. ~davestu/foo becomes /home/student/davestu/foo
- If there isn't a username, e.g. **cd ~** or **cd ~/bin** then bash assumes you mean your own home directory

Relative paths

- Paths that do not begin with a / or ~ are treated as relative paths
- Relative paths are evaluated starting from your current directory.
- For example, if you are in directory **/home/student/davestu**

  and you use the command **cd foo**

  then it is the equivalent of **cd /home/student/davestu/foo**

The use of . and ..

- When used as a pathname, the **.** represents the current directory
- When used on its own, the **..** represents the parent of the current directory
- When used within a path, the **..** represents the parent of the directory represented to that point in the path, thus the command **cd ../..** means change to the parent of the parent of the current directory

Symbolic links

- in addition to files and directories, you may also encounter symbolic links
- these are links from one directory to either a file or another directory
- e.g. suppose we have two subdirectories: test and other, and the test directory contains a file named foo
  - inside other we can create a link to foo using
    - ln -s ../test/foo
  - if we run ls when in other we'll see foo listed, but this is actually just a link to the real foo, not an entirely separate copy