Introduction to Unix

Unix is a multi-user, multi-tasking operating system originally designed circa 1970. It is one of the most, if not the most, influential operating systems ever developed. Unix is the basis for Linux, macOS, FreeBSD, OpenBSD, and many other operating systems.

Unix must have a way of organizing users files and of allowing multiple programs to run simultaneously without interfering with each other. Files are organized hierarchically in what is known as a 'tree' structure. The root of the tree is called `/` (slash). Each 'branch' of the tree is a subdirectory. Each subdirectory has its own name (subdirectories are equivalent to folders in a Windows or Macintosh environment).

Every user is assigned their own subdirectory whose name is the same as the user's login. Thus, the user `maryt`, keeps all of her files in a subdirectory called `maryt` or in subdirectories within the `maryt` subdirectory. Each subdirectory is protected so that only certain users are allowed access to it. In addition, there are several levels of access, for example you might be allowed to read the files in a certain subdirectory, but not to modify those files. (This is common in subdirectories containing programs that every user wants to use, such as the C++ program, but that users shouldn't be allowed to change.)

There are a number of basic commands that are used to negotiate Unix's file structure:

- **ls** Short for LiSt, this command lists the files and subdirectories in the current subdirectory. It does not list 'hidden' files; ones whose names starts with a '.' (dot).
- **ls -a** This command lists all of the files and subdirectories in the current subdirectory including the hidden files.
- **ls -l** This command lists all of the files and subdirectories in the current directory in the 'long' format - it lists information other than just the names.
- **pwd** Short for Print Working Directory, this command tells the used where they are in the files structure. A typical output would be:
  `/users/faculty/maryt/` showing that the user is currently in the subdirectory `maryt`, which is in the subdirectory `faculty`, which is in the subdirectory `users`, which is in the root directory `/`.
- **cd *directoryname*** Short for Change Directory. The command changes which directory the user is currently in.
- **cp *filename1 filename2*** Short for CoPy, this command copies filename1 into filename2. If filename2 already exists, it is replaced with the new file. The file names can include directories.

- **mv *filename1 filename2*** Short for MoVe, this command copies filename1 into filename2 and removes filenmae1. If filename2 already exists, it is replaced with the new file. The file names can include directories.

- **mkdir *directoryname*** Short for MaKe DIRectory, this command creates a new subdirectory.

- **rm *filename*** Short for ReMove, this command removes (deletes) a file.

- **rmdir *directoryname*** Short for ReMove DIRectory, this command removes (deletes) a directory. A directory must be empty before you can remove it.

- **script *filename*** The script command makes a record of everything printed on the screen by writing it to the file *filename*. (If no filename is given it is written to a file called *typescript*.) To use the script command type script *filename*, run the program, **then type exit**. Typing *exit* stops the script. The script file can be printed and turned in as the sample output of a program.

  There are two errors to avoid when using the script command. First, if you fail to type exit everything you do will continue to be dumped into the script file. In particular, if you attempt to open the script file before exiting the script command you will set up an infinite loop. Second, make sure that the *filename* you use is not the same as the name of a program you wrote; otherwise the script file will overwrite (erase) the program and you will have to rewrite the program.

- **cat *filename1 filename2* > *outfilename*** Short for conCATenate, this command will write to the file *outfilename* the contents of *filename1* followed by the contents of *filename2*.

- **man *commandname*** Short for MANual, this command presents help information on the given command.

All Unix directories include two special subdirectories called `.` and `..` The directory called `.` (dot) is the current directory. The directory called `..` (dot dot) refers to the directory above the current directory.