CMPT 214: Programming Principles and Practice
Term 1 2016-17
Lab 7 - Software builds and more UNIX

At the beginning of your lab period, your lab instructor will give a brief demonstration of specifying a `-l` option to `gcc(1)` or `g++(1)` in order compile a C or C++ program that uses library functions that are not in the standard C or C++ (respectively) libraries. He will also explain how to use the `-c` option to have the compiler only produce an object module from a C source file.

Answer each of the questions below using the LINUX facilities provided by **tuxworld**. For question 1, you will need to download the auxilary file `L6Q4.with_cr.txt.tar`; look for the link on the moodle pages for this lab. For all questions involving the use of LINUX/UNIX commands, place the command or pipeline you used along with the resulting output (i.e. copy-and-paste from your terminal window) in a file called `lab7.txt`. However, do not include extraneous or superfluous commands or output; only include content relevant and essential to the specified task. Then, with a text editor, add to `lab7.txt` identifying information to clearly distinguish which commands/output/code/description correspond to each task/question. This lab is out of a total of 9 marks, with each question (1a, 1b, 3, 4, etc.) being worth one mark except for 2(a), which is worth 2 marks. Marks may be docked for extraneous, irrelevant, or superfluous content or for not following directions. Your submission is due at 11:55 p.m. on Thursday, October 27.

You can use either **tuxworld** or one of the **ismac** machines for this lab. However record in `file7.txt` which you used as the answers to the questions vary slightly between machines.

1. UNIX/LINUX commands typically expect lines to be terminated by newline characters rather than carriage return characters. Lines terminated with carriage return characters can cause unusual and unexpected results.

   (a) Download the file `L6Q4.with_cr.txt.tar` from the lab index page. Use the `tar(1)` command to un-archive the file (to `L6Q4.with_cr.txt`).

   (b) Use the `pr(1)` command in the same way as you did in question 3 of Lab 6 to output `L6Q4.with_cr.txt` (from part (a)) with line numbers.

   (c) Use a UNIX pipeline involving `hexdump -c` and `grep` to show that `L6Q4.with_cr.txt` contains carriage return characters rather than, or in addition to, newline characters. The logic of the pipeline is that if `grep(1)` outputs anything (any lines), then `L6Q4.with_cr.txt` contains carriage returns. Note that `hexdump -c` represents a carriage return character by "\r" and newline by "\n" Make sure to have `grep` look for "\r" and not "r".

2. (a) Create a C (not C++) program called `test_math.c` that:
   - includes the system-wide `math.h` header file for the math library;
   - reads a floating-point number (type `float`) from the standard input;
   - calls the `sinf()` function (in the math library) with the input value as argument;

- prints, to the standard output, the result from calling the `sinf()` function.

Keep your `test_math.c` program as simple as possible. Submit your `test_math.c` program as part of your lab solution.

(b) Compile (with `gcc`), link, and run your finished program, testing it with a few input cases. Extensive testing is not necessary. Make sure that your program compiles, links, and runs without error. However, compiler warnings about the parameters to `main()` not being used can be tolerated.

(c) Compile (with `gcc`) your `test_math.c` file from part (a) to an object module rather than producing an executable file. You should end up with an object module named `test_math.o`. Confirm that it is an object module using the `file(1)` command.

3. The behaviour of some commands can be modified by setting or changing environment variables. Determine which environment variable you would modify to make `grep` case-insensitive. (Don't submit a log of this.) Then change this environment variable appropriately using the syntax "`export VARIABLE=value`" (in `bash`). Finally, execute the command "`grep a <<< A`" to show that `grep` now ignores case. Note: do not change locale settings.

4. The command "`ls -l ~`" will show the permissions and ownership, among other information, for each file in your home directory. However, suppose that you wish to show these attributes for your home directory itself (rather than the contents of your home directory). Use a UNIX command to do this. Like the `ls` command above, your command must use "~" as the argument.