***Lab 2: C Programming on Linux, Mac OS X***

***3 Exercises***

*3.1 Questions*

1. Compiling a program in the first time usually takes a longer time in comparison with the next re-compiling. What is the reason?

A: Because the compiling process of C programming language often has 4 phases to make C program to become an executable: Pre-processing, compilation, assembly, and linking. These phases proceed a lot of works to gather the information and make the program understandable for computers, the output for each phases is stored in different files (.i, .s, .o for the first three phases respectively). After being through such phases in the first time, the second time we compile our program, it is notice that the results for each phase has already been there (if they weren’t deleted), so compiler only has to check the validity of these files rather than redo the whole the process of compiling. Thus, this makes compiling a program the second time faster than how it was in the initial time. This statement, however, is arguable for other programming languages.

2. Can we use Makefile for other programming languages?

A: I think that we can use Makefile for a lot of other programming languages nowadays (like Java, C++ or C#,…) as long as you call the correct compiler and linker following the correct target chosen. For some rare programming languages or self-developed languages, a unique compiler often required in order to make the program executable, hence Makefile is useless here.

3. In case of source code files located in different places, how can we write a Makefile?

A: I think we can create a make file in each subdirectory to build them independently and then write a top-level Makefile to call the Makefiles in the subdirectories to build everything.