

# PERFORMANCE, DATA STRUCTURES AND ALGORITHMS

Exercise 03

Exercise 03 Answer

CORRECTING A SIMPLE C PROGRAM AND USING “MAKE”

# Name: Minakov Nikita csse1709r

PURPOSE

The purpose of this assignment is to give you concrete practice correcting errors in a C program. You will also be introduced to “make” files.

The “make” command in Unix is used to build products based on dependencies. This can be very powerful, especially for products that are comprised of multiple parts. This exercise will introduce you to a very simple “make” file for building C programs. Chapter 19 of the text C in a Nutshell contains very detailed information about all that the “make” command and associated “make” files can do.

By default, the file that the “make” command uses for instructions is called “makefile” or “Makefile”. This file contains *dependencies* and *targets* with instructions on how to build the targets based on the dependencies. For example, a C executable (e.g. a.out) needs to be built if the source file used to create the executable has been modified more recently than the executable. In other words, if you have changed your source code, you need to rebuild the program.

“Make” relies on the modification times that Unix associates with files. You have already used the “ls” command to list the contents of a directory. The “ls” command also can be made to display the time that a file was last modified (as well as additional information). The command “ls -l” (the -l refers to a “long” listing) will display information similar to the following:

-rw-r--r-- 33 user staff 1122 Mar 19 12:18 exercise03.c

The modification time in this example is “Mar 19 12:18”.

There are two ways to change the modification time of a file. When you edit a file and then save it, the modification time is updated to reflect the time that you saved the file. You can also use the “touch” command. Typing “touch file” at the command prompt (where file is the name of an existing file) will set the modification time of the file to the current system time. If the file you specify does not exist, it is created as an empty file with the current system time as the last modification time. The touch command is useful when you want to force “make” to build a program without editing the program.

By convention, most “make” files contain the targets “all” and “clean”. The “all” target is made using “make all” and generally makes all products that can be made by the “make” file. For example, the simple program for this exercise is built using “make all” at the command prompt. It can also be built using “make” (with no specified target) because the first target in the “make” file is “all”.

The clean “target” is used to remove products of “make.” Thus, before you submit your work, you can type “make clean” at the command prompt, and “make” will remove the executable and the intermediate build file so that when you archive your work, the archive will not contain these files.

ACTIVITIES

Perform each of the following activities. If you have questions, issues, or doubts, please ask for help and do not just guess.

1. Download and unzip (or unarchive) Exercise 03 from the LMS as you did for Exercise 02.
2. Change directory to the src directory, where you will find source code for a C program and a file called “makefile”. The “make” file for this exercise will compile a source file named exercise03.c into an executable called exercise03.
3. Edit the C source file and change the author as you did for the Hello World program in Exercise 02.
4. Compile the C program in the src directory using the “make” command by typing “make” at the command prompt. By default, “make” uses the “make” file in the current directory.
5. Copy and paste the output from the “make” command below:
6. Correct the error(s) in the program. Save your changes and run “make” again. Continue until the program builds with no errors or warnings.
7. Run the program and copy and paste the output below:

|  |
| --- |
| 0  1 3 6  10 15 21  28 36 45  55  final sum 55 |

1. When you are ready to submit your work, first remove all intermediate files from your src directory. This includes exercise03.o and the exercise03 executable. You can use the “make” command to do this by typing “make clean” at the command prompt.
2. Save and archive your work, including this document and your correct program, and upload it to the LMS as you did for Exercise 02.

