Contents

Lab 0: C Warmup

Overview

The purpose of this assignment is to gain some basic experience with C programming, including such topics as arrays, stack allocation, heap allocation, function calls, and memory safety (or the lack thereof). It is a credit/no credit exercise and should take a short time to finish.

Development Environment

Since we want to make sure that all students have a working (and generally equivalent) development environment, we are providing a Virtual Machine (VM) to use. Your first task, before starting any programming assignment, will be to ensure that your VM is setup correctly as outlined by the instructions page. **The VM we give you will be the only officially supported environment for all programming assignments** and will be used throughout the course.

Acquiring the code

Once your VM has been properly setup, you will now be able to retrieve the provided code for this assignment. To do so, you will have to use the `update-course` script located on the course VM. This script may be run from a command-line terminal or by using the launcher located on the top toolbar or by using the launcher on the desktop. This script will place all necessary files in the "course-materials" sub-directory within your home folder. For convenience, a shortcut to this directory has been placed on the desktop. **It is important to our infrastructure that you do not move/rename any files in the "course-materials" directory.** Also, **you should not change any files in this directory except for those that we note**. You may, however, copy files until your heart's content.

Now that you have acquired the source file, open arrays.c in your favorite text editor. arrays.c file contains a number of TODOs, which you are expected to complete. Most have a proceeding line that says "Answer:", which is where you should leave an answer to the question(s) posed in the TODO. One TODO requires you to write some code, which you should place immediately after the comment block that describes what to do.

As a convenience, here is an archive of the course-materials directory for this lab: lab0.tar.gz.

Compiling and running the code

The source file arrays.c won't do you any good by itself; you need a compiler (specifically the GNU C compiler) to compile it to an executable format. The GNU C compiler is available on the VM and most popular variants of Linux, such as Ubuntu and Fedora. You're free to use whichever machine you like, although **we will only provide support for the VM.**

Using the VM (or your own system if you're adventurous!), open a terminal and execute `gcc -v`. On our machine, here is what we see:

```
$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/4.6/lto-wrapper
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro
4.6.3-1ubuntu5' --with-bugurl=file:///usr/share/doc/gcc-4.6/README.Bu
gs --enable-languages=c,c++,fortran,objc,obj-c++ --prefix=/usr --prog
ram-suffix=-4.6 --enable-shared --enable-linker-build-id --with-syste
m-zlib --libexecdir=/usr/lib --without-included-gettext --enable-thre
ads=posix --with-gxx-include-dir=/usr/include/c++/4.6 --libdir=/usr/l
ib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstd
cxx-debug --enable-libstdcxx-time=yes --enable-gnu-unique-object --en
able-plugin --enable-objc-gc --disable-werror --with-arch-32=i686 --w
ith-tune=generic --enable-checking=release --build=x86_64-linux-gnu -
-host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5)
```

The output tells us a bunch of the configuration options for the our installation of GCC as well as the version number, which is 4.6.3. Navigate to the "~/course-materials/lab0" directory (where ~ denotes your home directory) and then use GCC to compile arrays.c with the following command:

```
gcc -g -Wall -std=gnu99 -o arrays arrays.c
```

It's not that important right now for you to know what all of these options do, but `-g` tells the compiler to include debugging symbols, `-Wall` says to print warnings for all types of potential problems, `-std=gnu99` says to use the C99 standard (now only 14 years old!), `-o arrays` instructs the compiler to output the executable code to a file called arrays, and `arrays.c` is the source file being compiled.

Having executed that command, you should be able to see a file named arrays in the same directory:

```
$ ls
arrays  arrays.c
```

The arrays file is an executable file, which you can run using `./arrays`. On our machine, here is what we see:

```
$ ./arrays
Filling an array at address 0x7fff6b4e3e60 with 10 values
Done!
Filling an array at address 0x7fff6b4e3eac with 1 values
```

```
Done!
Filling an array at address 0x7fff6b4e3e90 with 4 values
Done!
Filling an array at address 0x11ca010 with 5 values
Done!
```

If you look through the source code of the arrays.c file, you should be able to match up each line that is printed with the output that you see in the console.

With that, you should have everything you need to complete the assignment. Note that every time you want to test a modification to the source file, you will need to use the `gcc -g -Wall -std=gnu99 -o arrays arrays.c` command to produce an updated arrays executable file.

Checking your work

When you have finished the exercise, please take the time to ensure that your code both compiles without warnings (the GCC command will output lines that say "warning:" if you have warnings) and is readable. In particular, please try to make the format of your code match that of what was given to you.

Submitting the exercise

You will be able to submit your assignment with the `submit-hw` script that is bundled with the lab0 course-materials update. Using the script should be straight-forward, but it does expect you to not move/rename any files from the "course-materials" directory. Open a terminal and type the following command:

```
submit-hw lab0
```

The script will then prompt you for your Coursera username and then your submission password. **Your submission password is NOT the same as your regular Coursera account password!!!!** You may locate your submission password at the top of the assignments list page. With a working Internet connection, this should submit your code properly. You can go to the assignments list page to double-check that it has been submitted and check your score as well as any feedback the auto-grader gives you. You may also download your submission code here, if you wish.