

Basic C Programming

CS 350: Computer Organization & Assembler Language Programming

Lab 0, Not for handing in

A. Why?

- You'll be writing your programs for CS 350 in C and you'll definitely be writing programs for CS 351 in C.
- One of our later topics will be seeing how high-level programs in C are implemented as lower-level programs in machine code (the instructions that the hardware understands).

B. Outcomes

After this lab, you should be able to:

- Log into the `fourier.cs.iit.edu` machine and compile and run a simple C program.

C. Discussion

- C is a “lower-level” language than Java: its constructs more easily map to the data and operations found on typical hardware.
- You will be getting accounts on `fourier.cs.iit.edu` — I'll let you know when they have been created.
- As part of the zip file that makes up this lab, you should find `Lab0.c`.

D. Logging Into `fourier` and Compiling

- The `fourier` machine runs Linux; if you don't already know how to use Linux, it'll be good for you to learn how to. The `linux-account.pdf` file that's part of this lab will show you the basics of Linux.
- If you already had an account on `fourier`, just continue using it. If you didn't already have an account on `fourier`, you should receive an email from it/Dr. Beckman telling you about it.

- If you need help, your Lab TAs can show you how to log into `fourier` from the using a secure shell session (`ssh`) via PuTTY (on Windows), Terminal or iTerm2 (Mac OS X) or `ssh` (Linux).
- To transfer files to `fourier`, you'll probably want an SFTP (secure file transfer protocol) program; FileZilla seems popular.
- For this lab, practice logging into the `fourier` machine and compiling and running the `Lab0.c` program. Once you have a copy of the program in your current directory, the Linux command to compile the program is

```
gcc -Wall -std=c99 -lm Lab0.c
```

“gcc” means “GNU [pronounced Guh-Noo] C compiler,” the standard compiler for Linux environments¹. The option `-Wall` says to print all error messages; `-std=c99` says to use the ISO C99 standard; the `-lm` (ell em) says to include the math library (so you can use `sqrt`). Depending on your setup, you may not need the `-lm`; if you get a complaint about a missing `sqrt` routine when you compile your program, then you need the `-lm`. It may also be possible to put the `-lm` after the *filename.c*: `gcc -Wall -std=c99 -lm Lab0.c -lm`

- If the compile succeeds, it produces an executable file named `a.out`. To run your program, execute that file with the command `./a.out`
- Optional: If at some point during the semester, you get tired of typing in all the `gcc` compile options, use a text editor to edit (or create) your `~/.bashrc` file, which contains initializations used by the `bash` “shell” program that you type your Linux commands into. Add the line

```
alias gcc="gcc -Wall -std=c99 -lm"
```

to the `~/.bashrc` file. Close the file and log out and log back in. Now you can just type `gcc filename.c` when you want to compile, and the `bash` shell will substitute the `gcc` with options for the `gcc` in your typed-in input.

¹ “GNU” stands for “GNU’s Not Unix”, a reference to GNU being different from the versions of Unix that existed when the GNU project was started

E. The Sample Program

- Read through the sample program `Lab0.c`. You'll find much of C is similar to Java, but there are some fairly large differences too. [Ignore the problems for now.]
- The program contains a number of constructs, including:
 - Declarations of variables of basic types (`int`, `double`, `char`) and arrays of basic types of values.
 - The `printf` (print formatted) function for printing out values to the screen. Some basic formats (`%d`, `%f`, `%c`, and `%s`) are used.
 - String constants and strings stored as character arrays.
 - The `scanf` (scan formatted) function for reading values from the keyboard.
 - The `sscanf` (string scan formatted) function for reading values from a string.
 - The type `long int` (long integer), which is like regular `int` but can store larger values.
 - Hunt down some reference material on basic C programming as necessary to understand how the program works. The syllabus has some links to some free e-books in the school library.

F. Problems

- There are problem descriptions in the comments of the `Lab0.c` program. Write out answers to the problems and check them with the answers I'll post next week.