

## Computer Science 432

Spring 2022

Lab 0: A Simple C program, due Monday before lab.

**Objective.** To write a simple program, beautifully.

**Discussion.** An important aspect of this course will be programming in the C language. C is a thin language that allows you to control the underlying hardware in a rather direct way. Some have described C as a high-level assembler. We will certainly think of it that way.

Using C, however, requires that you develop a model for how C is translated into assembly. You have begun the development of that model in *Computer Organization*, but here we want that model to be as exact as possible.

**Assignment.** Before lab on Monday, I would like you to write a small C program that does the following:

Write a program, `prime`, that reads an integer from the input and prints out, on a single line, the 1 or 2 primes that are *closest* to it. Think about this very carefully. You might use this in a course on data structures, if you're looking to size a hash table.

Assume that input is of the correct form. Be aware there are no primes less than 2. Do not use floating point values and do not call `math` library functions. You will probably want to make use of `scanf(3)` and `printf(3)`.

1. We will use `git` to keep track of your work. For this assignment, clone the following repository:

```
$ git clone ssh://22xyz@lohani.cs.williams.edu/~cs432/22xyz/lab0.git
```

where the `$` is the unix prompt and `22xyz` is your CS username.

2. Within the `lab0` folder, you'll find an empty C file, `prime.c`; place your code here.
3. Compile your code. If you're on our Ubuntu machines<sup>1</sup>, you can accomplish this with the command:

```
$ gcc -Wall -g -o prime prime.c
```

There is a `Makefile` in this repository, as well, so you can type:

```
$ make prime
```

4. Run and test your program.
5. When you're finished, `add`, `commit`, and `push` changes to the repository for review and grading.

### Thought questions.

1. One of the remarkable features of unix is the fact that users can directly extend the set of commands that are available for use in the shell. This is accomplished through the environment variable `PATH`. How does this work?
2. When you're running the `prime` command, you make use of shared resources managed by the operating system. What are some examples of these shared resources?
3. How is your program beautiful? How might you improve its utility?
4. While I'm not particularly worried about the performance of your code, a beautiful implementation is likely to be pretty efficient. How would you describe the time and space efficiency of your program?

**What to expect.** Each week we will gather in small groups to review your various efforts. We will begin by performing a *code walk-through* for each member of the group. We will discuss what efforts are strong and identify ways that it could be improved. I will also ask questions to press your understanding of the assignment. That may or may not involve discussion of the thought questions. You will also receive a grade that reflects your lab effort and engagement in the code review.

★

---

<sup>1</sup>Secure shell to any of the CS unix lab machines: `amerifax`, `bagual`, `barzona`, `brava`, `charolais`, `devon`, `galloway`, `guernsey`, `kuri`, `lidia`, `niata`, `panda`, `rathi`, `reina`, `sharabi`, `sind`, `siri`, `sykia`, `tundaca`, and `zebu`. Off campus, use: `deoni`, `limia`, or `lohani`.