Assignment 0
CMPE 105: Principles of Computer System Design, Winter 2019

Due: January 18 at 5:00PM

Goals

There are two goals for Assignment 0. The first is to get your programming environment set up properly, which will require that you install an Ubuntu 18.04 virtual machine on your personal computer. The second is to learn the lab format for CMPE 105 by writing a simple program that does the same thing as the UNIX `cat` command. As with all programming assignments, this lab will require that you submit a design document, README file, and lab writeup along with your code in your `git` repository.

Setting up Ubuntu

Instructions on how to set up an Ubuntu 18.04 VM are available on Canvas.

Programming assignment: `mycat`

Design document

Before writing code for this assignment, as with every other assignment, you must write up a design document. Your design document must be called `DESIGN.pdf`, and must be in PDF (you can easily convert other document formats, including plain text, to PDF).

Your design document should describe the design of your code in sufficient detail that a knowledgeable programmer could duplicate your work. This includes descriptions of the data structures you use, all non-trivial algorithms and formulas, and a description of each function including its purpose, inputs, outputs, and assumptions it makes about the inputs or outputs. A sample design document is available from the assignment page on Canvas.

Yes, the design for `mycat` will be short, but we want you to get experience writing one up on a simple program before tackling more difficult programs. **Your design document is a significant fraction of the grade for each assignment**, so get in the habit of writing a good design document *before* you start writing code. It'll make writing code a lot easier. Also, if you want help with your code, the first thing we're going to ask for is your design document. We're happy to help you with the design, but we can't debug code without a design any more than you can.

Program functionality

The only code you have to write for this assignment is to implement the basic `cat` program, *without support for any flags*. That means your code needs to copy data from each of the files specified on the command line to standard output. For example, `mycat file1 file2 file3` will copy all of the data from `file1`, `file2`, and `file3` to standard output, in that order. Your program is called `mycat` to ensure that when you run it, you don't accidentally run the installed version of `cat`.

If no files are specified on the command line, `mycat` should just copy standard input to standard output until it runs out of output, just like the installed version of `cat` does. Note that the data might be binary; your code must work in that case. Process files one at a time; if `mycat` can't open a file to read it, the program should print an error message to standard error and skip the file, handling the remainder of them.

Your program may not use any of the C library `FILE *` functions such as `fread()` and `printf()` for user data. You may use `printf()` or `perror()` for error messages, but you may not use `fopen()`. Your code must use fixed-size buffers, and may allocate no more than 64 KiB of memory for them (either via `malloc()` or as a direct variable declaration).

Your code may be in either C or C++, though it's unlikely you'll need any C++ features for this assignment.

README and Writeup

Your repository must also include a README file (`README.md`) writeup (`WRITEUP.pdf`). The README may be in either plain text or have MarkDown annotations for things like bold, italics, and section headers. **The file must always be called `README.md`**; plain text will look "normal" if considered as a MarkDown document. You can find more information about Markdown at `https://www.markdownguide.org`.

The `README.md` file should be short, and contain any instructions necessary for running your code. You should also list limitations or issues in `README.md`, telling a user if there are any known issues with your code.

Your `WRITEUP.pdf` is where you'll describe the testing you did on your program and answer any short questions the assignment might ask. The testing can be unit testing (testing of individual functions or smaller pieces of the program) or whole-system testing, which involves running your code in particular scenarios.

For Assignment 0, please answer the following question:

- How does the code for handling a file differ from that for handling standard input? What concept is this an example of?

Submitting your assignment

All of your files for Assignment 0 must be in the `asgn0` directory in your `git` repository. When you push your repository to GitLab@UCSC, the server will run a program to check the following:

- There are no "bad" files in the `asgn0` directory (*i.e.*, object files)
- Your assignment builds properly in `asgn0` using `make` to produce a `mycat` binary
- All required files (`DESIGN.pdf`, `README.md`, `WRITEUP.pdf`) are present in `asgn0`

If the repository meets these minimum requirements for Assignment 0, there will be a green check next to your commit ID in the GitLab@UCSC Web GUI. If it doesn't, there will be a red X. **It's OK to commit and push a repository that doesn't meet minimum requirements for grading.** However, we will only *grade* a commit that meets these minimum requirements.

Note that the *minimum* requirements say nothing about correct functionality—the green check only means that the system successfully ran `make` and that all of the required documents were present, with the correct names.

You must submit the commit ID you want us to grade via Google Form, linked to the assignment page on Canvas. This must be done before the assignment deadline.

Hints

This is a straightforward assignment, designed to get you familiar with the tools you'll need for the remaining assignments. Most of your time will be spent setting up the VM, which isn't difficult, but can take a few hours.

Become familiar with the Coding Guide for this class, available on Canvas.

Grading

As with all of the assignments in this class, we will be grading you on *all* of the material you turn in, with the approximate distribution of points as follows: design document (35%); coding practices (20%); functionality (35%); writeup (10%).

If you submit a commit ID without a green checkmark next to it or modify `.gitlab-ci.yml` in any way, your maximum grade is 5%. Make sure you submit a commit ID with a green checkmark.

