Assignment 0

Note 1: It turns out there is a size limit that prevents pushing the whole /usr/src directory. For now, only add, commit, push the README file from the /usr/src directory. This will be sufficient for the /usr/src item of the assignment.
Note 2: After completing your assignment, your repository should have two folders: (1) folder /usr: Inside it is folder src that contains the README file. (2) folder asgn0 that contains the c file, design document, and makefile.
Note 3: A git branch is NOT a folder. Look up what a git branch is for this assignment.
Note 4: As written in the assignment, you have to use "open(), close(), read(), and write() system calls (see the man pages for them) as well as perror()". However, other than opening, closing, reading, and writing the file and standard input, you may use C library calls.
Note 5: The following is removed from the assignment: "If no files are specified on the command line, mytail should just copy standard input to standard output until it runs out of output, just like the installed version of tail does." as it is not applicable.

Assignment 0: tail

Goals

There are two goals for Assignment 0. The first is to get your FreeBSD environment set up properly. The second is to write a simple program does the same thing as the Unix tail command. This includes writing both a design document for the program and the program itself.

Setting up FreeBSD

There's a separate page on installing FreeBSD (see in canvas pages).

After installing FreeBSD, you need to commit and push the content of /usr/src to your git repository. See the details in the git and GitLab help pages.

Design document

Before writing code for this assignment, as with every other assignment, you should write up a design document. Your design document should be called design.txt (if in plain text), or design.pdf (if in Adobe PDF) and should reside in the asgn0 directory with the rest of your code. Formats other than plain text or PDF are not acceptable; please convert other formats (Word, LaTeX, HTML, ...) to PDF. **Don't submit a Word file—it's not a text file.** 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. Please see the sample design document in Canvas Files for a simple example of what a design document looks like.

Yes, the design for mytail will be short (albeit a bit longer than the example we provided, 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, of course, but we can't debug code without a design any more than you can.

tl;dr: Write your design document *first*, then write your code.

The tail program

The only code you have to write for this assignment is to implement the basic tail program, *without support for any flags*. That means your code needs to read a set of files and copy data from the last 10 lines of each of the files specified in the command line to standard output. For example,

```
mytail file1 file2 file3
```

will copy all the last 10 lines from file1, file2, and file3 to standard output, in that order. Your program is called mytail, to ensure that when you run it, you don't accidentally run the installed version of tail.

~~If no files are specified on the command line, mytail should just copy standard input to standard output until it runs out of output, just like the installed version of tail does.~~

Note that the data might be binary; your code should work in that case. Process files one at a time; if tail can't open a file to read it, the program should stop and print an error message to standard error. This may mean that some of the files were copied, and others not—this is fine, and is what the “real” tail program does. Your program must be written in C (not C++, not Java), and must be compiled by a Makefile that creates an executable called mytail. You should do your work in the asgn0 branch of the git repository, which you can get to by running the command:

```
git checkout asgn0
```

Your code, design document, and Makefile should be in a directory asgn0 at the top level of your repository in the asgn0 branch. This directory is created for you when you check out asgn0; feel free to add a README.txt file.

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 some time (a few hours).

The mytail program is very short—fewer than 60 lines of code, most of which are error handling. You'll need to use the open(), close(), read(), and write() system calls (see the man pages for them) as well as perror(). You may not use C library calls such as fread(), printf(), and getchar().

Read the git page in canvas to become familiar with git and GitLab. You'll be using them a lot.

Get in the habit of committing to git **frequently**. Your commits will be stored locally until you push them to the server, which you can do any time you want. Note that, if you commit your changes but *don't* push them to the server, your grader can't see them. Again, please read the git and GitLab page in canvas for more details.

Submitting your assignment

Please see the git help page in canvas for information on how to submit your assignment.

You need the following in the asgn0 branch:

1) The /usr/src directory from your FreeBSD installation. **(see note 1 above)**
2) A folder asgn0 with the design document and code (including Makefile) of the mytail program.

Grading

Unlike later assignments, this assignment is pass / no pass. *You must pass it in order to pass the class.* It won't otherwise impact your final grade, though you'll have a *lot* of difficulty with later assignments if you have difficulty with this one.