15-440 Spring 2022
Project 1

These are the samples and starter code for Project 1. Because this project uses precompiled libraries and binaries, it will only run on 64-bit x86 Linux machines. (e.g., Andrew unix servers; it won't run on Windows or Mac, unless you run 64-bit Linux in a VM).

The tcp-sample directory has a sample code for a simple server and client.

The interpose directory has starter code for creating a interposition library. Only a simple "pass-through" open() function is defined. Use this as a template for your checkpoint 1 code. You will need to add other file operators, and networking code to log the operations to the remote server (which you will have to write).

Run "make" in tcp-sample and interpose directories to build the programs.

The tools directory has a few programs we will be using to test your code. These are binary-only tools that operate on the local filesystem. You will make them operate across the network by interposing on their C library calls. Run any of these tools without arguments for a brief message on how to use it. These binaries should work on x86 64-bit Linux systems (e.g., unix.andrew.cmu.edu servers).

To use the interposing library, try (if using BASH shell):

```
        LD_PRELOAD=./interpose/mylib.so ./tools/440read README
```

or (if using CSH, TCSH, ...):

```
        env LD_PRELOAD=./interpose/mylib.so ./tools/440read README
```

You should see a message indicating the open function in the interpositon library was called.

Note that the 440tree tool uses the getdirtree function implemented in libdirtree.so in the lib directory. Please add the absolute path of this directory to LD_LIBRARY_PATH to make sure that the system can find the library, e.g. on BASH:

```
        export LD_LIBRARY_PATH="$LD_LIBRARY_PATH;$PWD/lib"
```

or if using CSH, TCSH, ...:

```
        setenv LD_LIBRARY_PATH "$LD_LIBRARY_PATH;$PWD/lib"
```

Autolab guidelines

Please adhere to the following guidelines to make sure your code is compatible with our autgrading scripts.

Untar this archive into a private directory in your AFS space (e.g. ~/private). cd into the 15440-p1 directory. Create your working directory here, i.e. in the same directory that holds this README. From your working directory, the provided library should be avaialble at ../lib, and incude file at ../include.

Write your code and Makefile in your working directory. You may need to add "-I../include" and "-L../lib" to the compiler and linker flags in your Makefile. Ensure that by simply running "make" in your working directory, both your library and server are built. Please name your library "mylib.so" and your server "server" and make sure both are in your working directory (i.e., not in a subdirectory). This naming convention and locations are critical for the grading system to build and run your programs.

To handin your code, from your working directory, create a gzipped tar file that contains your make file and sources. E.g.,

```
        tar cvzf ../mysolution.tgz Makefile mylib.c server.c
```

Of course, replace these with your actual files, and add everything you need to compile your code. If you use subdirectories and/or multiple sources, add these. Do not add any intermediate files (e.g., .o files) generated during compilation -- just the clean sources. Also, do not add the headers, .so files, or binary tools that we have provided -- these will be installed automatically when grading. Your makefile should expect the header in ../include, and the .so in ../lib.

You can then log in to https://autolab.andrew.cmu.edu using your Andrew credentials. Submit your tarball (mysolution.tgz in the example above) to the autloab site. Note, each of the checkpoints show up as a separate assessment on the Autolab course page. For your final submission, include your writeup as a PDF document in your tarball.