

# Homework 10

## Libraries

EECS 201 Winter 2020

### Submission Instructions

This homework will be submitted as a repository on the UMich GitLab server. The repository you submit will be a **private** project called `eeecs201-hw10`.

### Preface

In this homework you'll be provided yet another zipped archive containing some starter files.

```
curl -O https://www.eecs.umich.edu/courses/eeecs201/files/assignments/hw10-files.tar.gz
tar xzf hw10-files.tar.gz
```

Initialize a Git repository **in the extracted** `hw10-files` **directory**. Create a file called `REPORT` in this directory. Add all of the present files and commit them.

Create a **private** project named `eeecs201-hw10` on the UMich GitLab ([gitlab.umich.edu](https://gitlab.umich.edu)) and add the instructor `brng` as a **Reporter**. Set this UMich GitLab project as your remote: you'll be pushing to it in order to submit.

**For this assignment you're going to need to be in a Linux environment, be it WSL or your Ubuntu VM or whatever else you have.**

**This homework assignment assumes that you have a basic understanding of Makefiles. If you did not do HW7 or ADV7, you may want to get brushed up on Makefiles.**

## 1 Linking against libraries

In this section you'll be linking against a static library and its dynamic version with two different targets.

1. `cd` into the `link` directory.
2. Open the `Makefile`. Note the two TODOs.
3. Address the two TODOs and modify the compilation commands so that the `dynamic` target links against `libthingy.so` and the `static` target links against `libthingy.a`.  
When there's a naming conflict, which takes precedence?
4. You can use the `run-dynamic` target to see if you successfully linked against the dynamic library.

## 2 Creating libraries

In this section you'll be performing the steps to build both a static library and a dynamic library.

1. `cd` into the `create` directory.
2. Take a look at the directory structure.
  - The `inc` directory contains library header(s).
  - The `libsrc` directory contains the source code for the library.
  - The `lib` directory (created by the Makefile) contains produced libraries.
3. Open the `Makefile`. Note the four TODOs.

4. Add the necessary commands to the appropriate recipes to build the dynamic and static libraries and, like in #1, update the `app-dynamic` and `app-static` targets to link against the appropriate libraries.
  - Note that dynamic/shared library objects need to be compiled as position-independent code. You might notice that since both the dynamic and static libraries need object code the names of the object code files will conflict. Feel free to call the object code files whatever you want or put them where you want..

### 3 Conclusion

1. Add and commit the files you created.
2. Fill out the `REPORT` file in the following steps:
3. On the first line provide an integer time in minutes of how long it took for you to complete this assignment. It should just be an integer: no letters or words.
4. On the second line and beyond, write down what you learned while doing this assignment. If you already knew how to do all of this, put down "N/A".
5. Commit your `REPORT` file and push your commits to your remote.