
CSC 412 – Operating Systems

Lab Session 11, Spring 2022

Monday, April 25th and Wednesday, April 27th, 2022

What This Lab Is About

In this lab, you are going to set up a shared memory segment between image processing applications. This example is a bit contrived because the task is silly, but you know the operation of all the parts, and can therefore concentrate on the new stuff.

The Handout

The handout for this lab consists of a subpart of our old image processing library.

Task 1: Run the make file

The statements to compile the whole thing is complex, and I don't want to spend the whole session looking at the error messages of a 4-line-long `g++` statement produced by a mis-typed path or missing option.

So, I provided a make file. Simply execute

```
make clean run
```

This is going to build all the executables (even though the source files are incomplete) and try to execute. The result will be an invalid image written in the `Output` folder.

The Task

You will need to edit two files in the `applications` folder: `startMain.cpp` and `flipV_shm.cpp` (and maybe `flipH_shm.cpp` if you have the time).

`startMain` reads the image. What it needs to do is explained in the comments:

- setup a shared memory segment;
- copy the image information needed in that shared memory segment;
- fork and exec to launch the `flipV_shm` executable;
- After the child process has terminated, write the output image.

`flipV` will get the image data from the shared memory segment. Here again the comments give the main steps in the comments:

- map the shared memory segment;
- Map an image to the information stored in that shared memory segment;
- perform the flip;
- copy information back ;
- terminate.

The “new” part is to decide what you need to copy, in what order in the shared memory segment. After that, you’re basically just going to copy blocks of code from the shared memory code seen in class.