CMPSC 400
Operating Systems
Spring 2020

Participation 2: Build A Mini-OS

The little book about OS development

Erik Helin, Adam Renberg

Figure 1: We will be following this book's tutorial. Please see Section 2 of `https://littleosbook.github.io/` to create your *Cafebabe* OS.

Summary

In this participation, you are given Sections to read from the online book, "The Little Book About OS Development" by Erik Helin and Adam Renberg (`https://littleosbook.github.io/`).

From this book, you will learn how to make a very basic OS called *Cafebabe* that secures memory and loops infinitely until it is shut-down. We will be using an emulator to run this OS which will be hosted by a Docker container.

GitHub Starter Link

General Participation Repository
`https://classroom.github.com/a/S8lbI9Z5`

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

  - `git commit <nameOfFile> -m ''Your notes about commit here''`
  - `git push`

  Alternatively, you can use the following commands to add multiple files from your repository:

  - `git add -A`
  - `git commit -m ''Your notes about commit here''`
  - `git push`

What TODO

Your tasks are outlined below.

1. Read through Sections 1 and 2 of the online book, *The Little Book about OS Development*. The link is offered in Figure 1. You will find source code in the reading to place into your source files in `src/` of your `classDocs` lessons directory.
2. Locate your `Dockerfile` the `scr/` of your `classDocs` lessons directory. Build your container which will contain the compiler, linker and emulator software. The commands for this task are found in the comments at the end of the file. Note: building your container may take some time.
3. Run the container using the file `src/run_osDevi.sh`. To run this file, use the following command: `. run_osDevi.sh`. Please note, the builder file will create your OS in a different way than the one specified in the reading. Here, we will use an emulator in Docker to run the OS.
4. From inside your container, use the builder file, `build_cafebabe.sh` to compile your source code and run your OS. Note: to quit the OS emulator, type Esc+2 and then 'q' (twice) to exit. When you close-down your container, please use `exit` rather than clicking to close the terminal window. This will leave the container running in the background and will also use up your battery and memory resources.
5. Once you have a working, bug-free OS and have had a moment to play with it, please complete the below Questions-In-Blue.

Directory Instructions

We will be using a previous participation repository. Please copy the `02part/` directory from `classDocs` and work in the participations repository. Please do not forget to push your work. We will not be using GatorGrader for this work.

Questions-In-Blue

The below questions have been designed to be thought-provoking and may not necessarily have one correct answer. Please do your best to answer them. Each question can be answered in a few sentences.

1. The following questions concern the builder file.
   - (a) In your own words, using clear and meaningful language, please describe what `nasm` does in the build process.
   - (b) In your own words, using clear and meaningful language, please describe what `ld` does in the build process.
   - (c) In your own words, using clear and meaningful language, please describe what `qemu-system-i386` does in the build process.

2. The following questions concern the OS project.
   (a) According to the required reading, what role does the file `loader.s` play in the OS (in a few written lines)?
   (b) According to the required reading, what role does the file `link.ld` play in the OS (in a few written lines)?
   (c) According to the required reading, what are the next steps to add some written text to the output?

Summery of Deliverables

1. Copy the directory called `02part/` from your `classdocs` into your `participations/` repository. Begin editing your files from the participation repository (and not in `classDocs`.) Do not forget to push your work.

2. File `writing/reflection.md`: In this file, all your questions-in-blue from above are to be answered.

3. Files `src/loader.s` and `src/link.ld`: These files are to be *error-free* and serve to create a basic OS when compiled by the file `builld_cafebabe.sh`.