

# Operating Systems - Monsoon 2020

Arani Bhattacharya, Sambuddho Chakravarty

November 30, 2020

## Assignment 5 (Total points:80)

### 1 System Bootup and Switching to Protected Mode (Total points: 45)

You need to implement a simple bootloader, which would be booted via a legacy BIOS. The program needs to be written in assembly language and a bootable binary needs to be created. This bootable binary should basically represent the boot sector of a disk and should be recognized by the BIOS as a bootable image. Your program should first switch to the protected mode (32-bits) and thereafter print “Hello world!” and the contents of the CR0 register (the one used to turn on protected mode).

You need to boot your bootable image using `qemu` emulator.

#### What To Submit

- Source code of the program which you have submitted and the binary, which you booted with `qemu`. Inside each file, your name and roll number must be present as a comment on top. Naming convention for the files is *< filename > \_RollNo*.
- `Makefile` to compile your code.
- Description of your code along with the logical and implementation details, along with instructions to boot your image. Submit the document in PDF only, using the naming convention *WriteUp1\_RollNo.pdf*
- Use the given repository as template  
<https://github.com/mayank18049/OS-Assignment-5.git>

#### Grading Rubric

- Source code of bootloader with `Makefile` (written in assembly) – 20 points.
- Correct outcomes, viz. “Hello, world” printed on the terminal and the CR0 register contents – 20 points.
- Descriptive write-up (code description, logic, `Makefile` and booting instructions) – 5 points.

## 2 Advisory file locks (Total points: 35)

The second exercise deals with advisor file locks (e.g. POSIX `flock()` and the likes). You need to create a rudimentary text editor, all that allows for merely open files for editing and saving text. You could use the C library functions in `termios.h` for the same. You need to also use advisory locks such that two simultaneous accesses to the same file using the editor should throw an advisory warning on the one who opens the file that already is advisory locked. However, the access thereafter need not be blocked. The user who receives the warning should be able to edit the file (regardless of race conditions).

### What To Submit

- Source code of the text editor that handles simultaneous access to files using advisory locks. Inside each file, your name and roll number must be present as a comment on top. Naming convention for the files is `<filename>_RollNo.`
- Makefile to compile your code.
- Shell script to show the case where two different processes open use the editor to simultaneously edit (or one edits and the other reads simultaneously).
- Description of your code along with the logical and implementation details. Submit the document in PDF only, using the naming convention `WriteUp_RollNo.pdf`
- Use the given repository as a template  
<https://github.com/mayank18049/OS-Assignment-5.git>

### Grading Rubric

- Full compiling source with Makefile – 15 points.
- Correct usage of the advisory lock function while handling the error cases – 15 points.
- Descriptive write-up (code description, logic and Makefile) – 5 points.

### Final Submission

- Add the two zipped folders made for Q1 and Q2 to a final zipped folder named `RollNo.zip` and upload the same on the Google Classroom.

### Late Submission Policy

- No late submission policy is available for this assignment. The deadline is December 20, 2020 (23:59 hrs)