OS Lab – 1

Madhav Bhardwaj (CS23B035)

August 2025

1 Problems Faced During Installation and Coding

Cloning xv6 Inside or Outside Docker

Initially, I was unsure whether to clone the `xv6` repository inside the Docker container or outside it. After considering how Docker containers work, I realized that cloning it outside the container was better. This way, the files and changes would not be lost even after the container was closed. Apart from this, the installation went smoothly.

Using spike Instead of qemu

At first, I used `spike` to simulate the code instead of `qemu`. Even though the program compiled and ran successfully, the main issue was in my code: I had mistakenly placed the instructions under the `.section .data` segment, which caused unintended behavior.

2 Assembly Code

The code printing Helloworld is:

```
.section .text
.global main

main:
    li a0, 1
    la a1, msg
    li a2, 34
    li a7, 16
    ecall

    li a7, 2
    ecall

.section .data
msg: .asciz "Hello CS23B035! Welcome to CS3500\n”
```

Explanation of the Code

- `.section .text` – this is the section declaration of code.
- `.global main` – Entry point of the code as main. makes this label accessible
- `li a0, 1` – Loads the value `1` into register `a0` (file descriptor for stdout).
- `la a1, msg` – Loads the address of the string `msg` into `a1`.
- `li a2, 34` – Loads the length of the string into `a2`.
- `li a7, 16` – System call number for `write` It can be seen in the file syscall.h inside the kernel folder of xv6.
- `ecall` – Executes the system call.
- `li a7, 2` – System call number for `exit`.
- `.section .data` – Declares the data section for variables and strings.
- `msg: .asciz "Hello ..."` – null-terminated string in memory.

Steps to run my code

- First, after writing the code, we need to compile it and run it.
- Place the assembly code file inside the `xv6-riscv/user` directory.
- Edit the `Makefile` in `xv6-riscv` by adding the filename (without extension) to the list of `UPROGS` so that xv6 will build it into the file system.
- Rebuild xv6 by running:

  ```
  make qemu
  ```

  in the `xv6-riscv` directory.
- Once xv6 boots inside QEMU, run the program by typing its name at the shell prompt (same as the filename without the `.S` extension).