



Indian Institute of Technology Bhilai  
Department of Computer Science  
CSL301: Operating Systems  
Scope: xv6  
Difficulty Level: Beginner

Class Assignment - 2  
August 13, 2025

## Instructions -

- Save your final shell source code with the following naming convention: `<rollnumber>_sh.c`
- After each code addition, run the new commands in xv6 qemu docker to verify the change and capture a screenshot. Name each screenshot `<rollnumber>_q<question-number>.png` (e.g., 21000123-q1.png for question 1).
- Create a plain text file, where for each question you write the question number (e.g., Q1, Q2, ...) and include only the code segments you updated for that question. Do not include full files, just the specific changes made. Save this file using: `<rollnumber>_part2.txt`
- Submit these files as follows: - Place all the files above in a single folder and compress it. Name the compressed archive as `CA2-<rollnumber>_part2.zip`

**Note:** Do not include explanations or code outside the specified text file and annotated source file. Each screenshot should clearly show your qemu terminal running and reflecting the update for each task.

## XV6 : Shell Updates

### Overview

For this exercise, you will extend the **xv6 shell** to add new built-in commands and customizations. All your changes will be made in the file `sh.c`.

### Edit Location in `sh.c`

Most changes will go inside:

```
case EXEC:
    ecmd = (struct execcmd*)cmd;
    if(ecmd->argv[0] == 0)
        exit();
    //Insert your code here
    exec(ecmd->argv[0], ecmd->argv);
    printf(2, "exec %s failed\n", ecmd->argv[0]);
    break;
```

### General Notes

- Order matters: Place each `strcmp(...)` check above the `exec()` call.
- If your built-in runs, call `exit()`; so the shell does not attempt to run `exec()` next.
- Include `fcntl.h` for file access flags.

### Question 1 — Change the Shell Prompt

Replace the default `$` prompt with your roll number (e.g., 21000123\$).

#### Hints:

- Implement this in the function “`int getcmd(char *buf, int nbuf)`”
- The `printf` inside `getcmd()` controls the prompt.

### Question 2: Add a touch Built-in Command

Implement a basic `touch <filename>` command that creates an empty file with the given name.

#### Hints:

- Implement in case `EXEC:` — before `exec()`.
- `ecmd->argv[0]` contains the command name ("touch").
- `ecmd->argv[1]` contains the filename.
- Use `open()` with flags `O_CREATE | O_WRONLY` (from `fcntl.h`).
- Call `close(fd)` after creating the file.
- If successful, call `exit()`; to avoid running `exec()`.

### Question 3 — Update Error Message for Unknown Commands

Change the existing failure message:

```
exec <cmd> failed → Unknown command or exec failed: <command>
```

#### Hints:

- Implement inside case `EXEC:`, after the `exec()` call.
- Modify the `printf()` line.
- `ecmd->argv[0]` contains the command entered.
- This code executes only if `exec()` fails.

### Question 4 — Add a help Built-in Command

Implement a `help` command that displays available built-ins and their descriptions.

#### Hints:

- Implement inside case `EXEC:` — before `exec()`.
- Check if `ecmd->argv[0]` is "help".
- Print help text:

```
help          - show this help
touch <file>   - create new file
press ctrl+A then X - exit xv6 shell
teststrrev    - Prints your name in reverse
```

- Call `exit()`; after printing help.