

# Operating Systems

CMSC 422 and MSCS 515

## Final Project

Starting with and including all functionality of *iProject* four, add new features as specified below.

### OS Functional Requirements

- Add a shell command to allow the user to select a CPU scheduling algorithm. [required]
  - `set cpu scheduling [rr, fcfs, priority]`
- Add a shell command to return the currently selected cpu scheduling algorithm. [required]
- Mount / Unmount different File System Device Drivers [optional]

### Source Code Requirements

- Add first-come, first-served (FCFS) and non-preemptive priority CPU scheduling algorithms to your CPU scheduler, executing whichever one the user has selected. Default to RR. [required]
- Implement swapped virtual memory
  - Allocate enough memory for three concurrent user processes. [required]
  - Allow the OS to execute four concurrent user process by writing roll-out and roll-in routines to . . .
    - Take an unused process and store it to the disk via your FSDD. [required]
    - Load a swapped-out process and place it on the ready queue. [required]
  - . . . as necessary.
- Add System Call handlers to allow the user program to perform disk I/O. [required]
  - Read the first block of a file and place the content in memory
    - X: 03
    - Y: start address of the file name
    - A: start address of the location in memory to save the read bytes
  - Write the first block of a file
    - X: 04
    - Y: start address of the file name
    - A: start address of the (null-terminated) string to write to file
- When the currently executing process makes a disk I/O request
  - Move the current process to an I/O queue to wait while its I/O request is being serviced. [required]
  - Invoke the scheduler to get the next process on the Ready Queue (RQ) [required]
  - Once the I/O request is complete, take the waiting process off the I/O Queue and put it back on the Ready Queue. (Be sure to keep track of any data returned from the I/O operation.) [required]
- Detect and gracefully handle errors in the user machine language code. [required]
  - Invalid op codes [required]
  - Missing operands [required]
  - Memory out of bounds access (attempts) [required]

### Hints

- Continue to (constantly) refactor. Your code should improve as your understanding does.

# Operating Systems

CMSC 422 and MSCS 515

## Final Project

### **Style, Art, and Science Requirements**

- Your code must separate structure from presentation, be professionally formatted, and use and demonstrate best practices
- Remember the value of comments and expressive identifier names and how much their presence and quality affect my grading.