CSE 7343, Operating Systems, Spring 2017
Course Project

Course project is an individual effort and will be 25% of your semester grade. You will deliver your project in the following two phases.

- Phase1: Process management.
- Phase2: Memory management.

A running demo of both Phase 1 and Phase 2 will be required.

Grading:

Phase1 accounts for 25 points, and Phase2 accounts for 25 points (total 50 points)

Grading Considerations:

- Even if your program is not working, you should submit your design/code/documentation/etc for grading. In this case, the maximal points you can get is 50%.
- If your program works exactly as specified in the assignment, you will get at least 70%, the rest will depend on design/documentation/demonstration etc.
- Try to use good CS/SE/UI practices, such as modular design, consistency, good documentation, etc.

Phase1 – Process Management

Design and implement a program (a simulated OS) for general process management, with the following specifications:

- ~~Each process will be represented by its Process Control Block (PCB) and each process will be given a priority.~~
- ~~Each process info can be entered manually by user input or loaded from text file. Proper Error Handling is expected.~~
- ~~Each PCB should have a uniform format.~~
- ~~The collections of PCBs form single or doubly linked lists for all queues implemented in the system.~~
- ~~You need to implement at least two queues for this part of the project (and probably more for the later part of the project): a ready queue and a waiting queue (or several waiting queues for different resources).~~
- The operations on a queue include:

- adding a PCB to a given position in the queue,
  ~~the default position is the end (tail) of the queue.~~
- ~~deleting a PCB with a given PID.~~
  ~~The default is the beginning (head) of the queue.~~

- ~~The control program will call on different functions/methods to add/delete PCBs.~~
- ~~You also need some utility functions/methods to show the contents of different queues.~~
- Write a scheduler that can implement:
  - Shortest Job First (SJF) scheduling, FCFS Scheduling, Non-preemptive Priority scheduling (Use only 1,2,3,4 for priorities and assume 1 is the highest priority) and Round-Robin scheduling with a given Q parameter (input to the program). Compare these scheduling algorithms in terms of average waiting time on a given sample of processes.
  - The set of processes are input to the program via a text file that may have the following format:

    Process_id, arrival_time, burst_time, priority

    Example:
    2760, 1, 16, 1
    2750, 1, 9, 2
    ……
    ……

- Some important design considerations should include:
  - program structure and modularity.
  - program clarity and comprehensibility (how easy to understand).
  - program maintainability (modification to work for assignment).

- Also, some important user interface considerations:
  - you do need to consider both the efficiency and the usability issues.
  - consistency with your user interface?

<u>What to turn in:</u>

- The program code (written in your favorite language).

- Documentation that includes execution trace, showing different PCBs being added/deleted, and the contents of the queues.
- Documentation on how to install (compile/link) your program on SoE/UNIX machines or on PC running Windows OS.
- All the above three parts, as well as overall code/design clarity/modularity/maintainability and documentation, will be considered in grading your project.
- You can use existing library to represent the data structure needed for this assignments. You can also use third party API to handle some of the tasks including: file processing, process management. But you need to understand how that works and quote that in your document.

Due date: March 30.

Phase2 – Memory Management

Building upon the facilities you implemented for project Phase1, implement memory management for the processes, with the following specification:

- Each process will include some information about the amount of memory needed and when (implicit: when it is created) and how long it will be needed for.
- Your program (simulated OS) will need to keep track of all the memory space, in connection with information kept at each process.
- You are required to implement first-fit, best-fit, and worst-fit algorithms.
- You need to report possible memory fragmentations and continue running the algorithm. However, if you don't encounter the problem, run your program until all processes are finished or until an explicit exit command is issued by user and/or OS-admin.
- Some standard test cases will be posted close to the project due date so that you can test your program before the demo and handing in the deliverables. The program will be given the following information:
    - PID,
    - arrival time (in case of a tie, process with a lower PID will be handled first),
    - duration of the process (when some process are finishing and others are starting at the same time, the one who release resources will be handled first).
    - size of memory request

- Design considerations and utility programs to report the status are similar to Phase I of this project.

What to turn in:

- The program code (written in your favorite language).
- Some execution trace.
- Documentation on how to install (compile/link) your program on SEAS/UNIX machines or on PC running Windows OS with screenshots
- All the above parts, as well as overall code/design clarity and documentation, will be considered in grading your project.

Due date: April 27

A demo and defense of each Phase will be scheduled with your grader. Demo days will be announced.

Where: To be announced.

- The demo will involve the following steps:
    1. setting up the computer (your own or a standard PC)
    2. compile/link your program to produce an executable version
    3. run with standard test data
    4. run with new test data from your grader/instructor
    5. answering questions, if there are any

- For distance students: If you are in the DFW area, please try to come in and have your live demo with the grader. If you live outside DFW area, please report to the instructor for different arrangements.

- Grader Info: TBA