Seminar 4: Assignment 2

Reminder

- Assignment 2 Due: 2017/02/27 @ 23:55

- Still plenty of time

- This assignment does **NOT** require actually running any processes or threads. It's just a simulation.

Required Entities

- A global clock that represents the current elapsed time
- A Ready Queue: a FIFO queue where threads that are awaiting execution are stored
- An Event Queue: a priority queue/sorted queue where events that need to take place are stored

Global Clock

- The clock is not going to be using “wall-time” (actual seconds)

- Just an unsigned integer (we can’t have negative time)

- An example: global clock is at 2 → Event A is popped off the Event Queue, and it takes place at 20 seconds → set the clock to 20 seconds

Organizing the Data - Process

- The following are just examples, you may need to modify them depending on your implementation

- Process Struct containing:

  - ID

  - Array of Threads

Organizing the Data - Thread

- Thread Struct containing:
  - ID
  - Array of CPU Burst Cycles
  - Array of IO Burst Cycles
  - Status (New, Ready, Running, Blocking, Terminated)
  - Arrival Time

Organizing the Data - Event

- Event Struct containing:
  - Type of Event
    - The main ones will be ARRIVAL, CPU_BURST, IO_BURST and TERMINATION
    - You will need more, since there are more events
  - Time the event takes place
  - The thread associated with an event

Organizing the Data - Event Queue

- I would recommend using a sorted queue, which is sorted by the time that the event takes place

- This lets you pop the head element from the queue, and have it guaranteed that it will be the correct event

File Format

- Is a bit convoluted, lots of numbers
- In the commented example in the Assignment PDF, number_of_CPU means number of CPU Bursts

File Format

```
1  4
    Process Struct Information

    1  0  3
        Thread Struct Information

        1  15  400
            Index - CPU_Burst Time- IO_Burst Time
        2  18  200

        3  15
            Note that the final CPU Burst doesn't have a corresponding IO Burst
```