Lab Assignment 1

Process and Thread Management

You are to design a OS manager to manage processes and threads within your soon-to-be-developed operating system. The high--level state diagram for the thread states to be implemented (your operating system is multi-threaded). The process control block and thread control blocks need to be implemented.

Constraints on your design:

•Single---core system

Deliverables:

1. Process Manager

Testing Interface:

The following testing interface is based on the assumption that the state machine implemented will contain at least the following queues: Running, Ready, Waiting, New, Terminated. If you have other states than besides these, you might have to add new commands to be able to fully test your implementation. These commands allow you to simulate actions that might be taken by a scheduler (which you are not implementing in this assignment), so in your tests, test all functionality that a scheduler might want to perform. You should use the queue manager from the last project to manipulate the queues for this assignment, but you might have to enhance it to work for this assignment.

Initialization Command

CREATE pid psw reg0 reg1 reg2

The created PIDs must be unique. If this command requires you to perform multiple actions within your PM, then this one directive will cause those actions to happen. The result of this directive will be that there is a new PCB in the Ready queue and that PID will be displayed back to the test interface.

CREATE tid pid tsw reg0 reg1 reg2

The created thread id must be unique

Status Commands

LIST {*queue_name* | *all* | SCHED}

Should display the selected queue, all the process queues, or the PID of the process that is scheduled next, when this command is executed.

Action Commands

1. GO

   Use of this command triggers running of the scheduler module in your manager.

   It should be performed after one or more transition commands (defined in the next paragraph). Transition commands are used to manipulate the contents of your process manager queues.

Transition Commands (used to manipulate the contents of your manager queues)

1. UNWAIT t*id*

   This command moves a thread from the WAIT queue to the ready queue

2. EOQUANTUM

   Moves the current Running thread back to the Ready queue as if its time quantum has expired

3. EOLIFE

   Moves the current Running thread to the Terminated queue as if it's asked to quit

4. WAIT

   Moves the current Running thread to the Waiting queue as if it performed some I/O and needs to wait for it to complete

As the transition command is given, the action should be performed, meaning the queue manipulations should be done.

Expected use of this test interface will be like:

- Multiple "creates" to build up some processes and threads
- Some transitions to move them around the queues
- An occasional "list" to see what thread is where
- Then specific transitions, mostly with ending the time quantum followed by a "GO" to run the scheduler

Run a "list" to see the accounting totals and possibly where the threads have moved around the queues.