Process Manager Test Interface Requirements

CSE325, February 2, 2012

For your process manager, I am providing the specifications of a test interface. Your project should be able to accept commands, as defined in this document, as a way to exercise your process manager. As other managers are added, additional commands may be introduced to enhance testing of the new functionality.

The commands listed are not necessarily all the commands you might need to test your design. I have only designed commands to test state machines that contain the following queues: Running, Ready, Waiting, New, Terminated. If you have other process states than these, you might have to add new commands to be able to fully test your implementation.

You should use the queue manager from the last project to manipulate the queues for this assignment.

Initialization Commands

1. CREATE *pid psw page-table 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.

Status Commands

1. 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 process 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 process manager queues)

1. UNWAIT *pid*

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

2. EOQUANTUM
   Moves the current Running process back to the Ready queue as if its time quantum has expired

3. EOLIFE

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

4. WAIT

   Moves the current Running process 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
- Some transitions to move them around the queues
- An occasional "list" to see what process 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 processes have moved around the queues.