Process Manager Test Interface Requirements

CSE325, due March 1, 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 and scheduler. 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 the 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.

Initialization Commands

1. Init_Sched [*general_purpose* | *interactive*]

   This directive should clear all processes in the process manager and initialize the process manager to use the specified scheduling algorithm.

2. Create_Proc *psw page-table reg0 reg1 reg2*

   The arguments allow you to set certain values within a PCB. Created PIDs must be unique.

   The directive “create_proc” will exercise your functions to fill in a new PCB, and put it on the Ready queue. If that requires you to perform multiple actions within your process manager, then this one directive will cause all 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

   The resulting output will be either the ordered list of PID’s for all processes in that queue, or the PID’s of all processes in the system, ordered within each queue, or if the SCHED argument is used, then the resulting information will be the PID of the next process that is to be moved to “Running” when the scheduler runs next time. You may want to display more than just the PID with this command since your scheduler will have to show it is scheduling multiple classes fairly.

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

Transition commands allow you to emulate events that might have been performed by either some piece of hardware (an interrupt), or software that has not been implemented yet.

1. UNWAIT PID

   This command moves the named 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 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 the I/O 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 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.