CSI 400/500 Spring 2009
Prof Robert Ekblaw

Homework Assignment #2

Due: Monday, February 23rd 11PM

Description
Create a pseudo-kernel to handle scheduling. Your scheduler kernel will then be used to manage an imitation local user network. The network has six users, any number of whom could be running threads at any time.

There are basically four different types of “threads” your scheduler will manage. Note that these threads are not real, but you are simulating them during the operation of your scheduler:

1. Logging into the network. This thread will start, then wait for two user inputs: the userid and password. This wait is uninterruptible. Once the inputs are received, it will continue running to verify the information. In most cases, the userid and password will succeed, so a new thread will be started. This thread then terminates.
2. Database query. This thread will wait for a response from the database. This wait is interruptible. Once the response is received, the thread is awakened. When the processing is complete, it terminates.
3. Printing a report. This thread runs uninterruptible. It just performs its job and then terminates.
4. Log off. This thread runs very briefly (simulating cleaning up storage) and terminates.

You will design a test run to demonstrate the functionality of the scheduler. Every fifteen seconds your scheduler should report what threads are being managed, what user they are associated with, and what state they are in. Your run should last at least three minutes. You can determine which threads run when and what users are active when. There should always be at least one active thread, and at some point all six users will have threads queued.

Turn in
1. Your entire code, including the scheduler kernel and any included files that you wrote.
2. The plan for a sample of your scheduler. Include the order of threads and which “user” is associated.
3. The output from a sample run.

Turn in your assignment to csi500@cs.albany.edu.