CS450 Assignment 4: OSP CPU Scheduler
Due October 14th, 2002

September 30, 2002

This assignment will allow you to create a simple CPU scheduler for OSP. It should be a combinator of round robin and priority scheduling. This assignment will take 2 weeks. Make use of your time wisely.

1. **Getting Set Up**

   Log in to cs450.cs.iit.edu and create a new directory off your home directory called "as4" using `mkdir as4` when in your home directory.

   Enter that directory and copy the following files:

   ```
   ~osp/asg4.sparc64/Makefile
   ~osp/asg4.sparc64/cpu.c
   ~osp/asg4.sparc64/dialog.c
   ~osp/asg4.sparc64/par.trace
   ~osp/asg4.sparc64/par.low
   ~osp/asg4.sparc64/par.high
   ```

   In addition, if you want to compare your results with that of the standard scheduler you can copy the following files:

   ```
   ~osp/asg4.sparc64/statistics.low
   ~osp/asg4.sparc64/statistics.high
   ```

2. **Creating Your Scheduler**

   You have a lot of freedom in how you desire to implement your scheduling algorithm. The basic premise is that processes should be allowed to run and then interrupted when their quantum expires. Your ready queue should not be first-come-first-served, rather use a priority based algorithm that takes into account factors such as age of process, cost of prepaging, amount of time process has been waiting since last CPU quantum, and other parameters of your own creation.

   You should make your algorithm flexible enough so you can easily modify the parameters because you will have to modify the algorithm to obtain lower turnaround time, lower waiting time, and higher CPU utilization. Some of those might be mutually exclusive, so you'll have to make design choices.

   Please submit three runs for the program, based off of `par.trace`, `par.low`, and `par.high`, respectively (short trace, low process intensity, high process intensity). You can compare your results with the OSP scheduler from `statistics.low` and `statistics.high`.

   All code must be in C and must be in your `cpu.c` file.

3. **Testing Your Work**

   The first step to testing your work is to compile it. Make sure that the program `make` is in your path. You can check this by typing `make` at the command prompt. If the program is not found, you may need to add it to your path. You can set this up by adding the following line to your `.bash_profile`:

```
export PATH="$PATH:/usr/ccs/bin"
```

Then logging out and logging back in again. This will update your path so make will execute.

Now, just run Make in the directory you are working in. This will create a file called `OSP`, which is your executable for the simulator.

4. **Handing In Your Assignment**

   Everything for this assignment is done electronically. There should be no paper material submitted in class.

   Before you hand in your assignment, make sure that your name is at the top of your source code for `cpu.c`. Here is an example:

```
/*********************************
 * Patrick Wagstrom
 * wagspat@iit.edu
 * CS450-251 - Internet Section
 * Student ID: 123-456-7890
 *********************************/
```

   Then, prepare a 1 page (approximately) description of how you implemented your solution and why your statistics are better/worse or the same as the default scheduler. Attach this at the end of your `cpu.c` file as a C source code comment.

   In addition, prepare a 1/2 page document with an explanation of your statistics from program runs. Attach this to the end of `cpu.c`. All of this text should in a C comment block. Here is an example:

```
/*****************************************************************
 Implementation Design and Analysis
 *****************************************************************

 I believe that I created a well rounded design that performs
 adequately, but could use improvement in the following areas...

 *****************************************************************
 Statistic Explanation
 *****************************************************************
 It's clear that my CPU utilization was below where it should be in
 all situations, this is shown by the value of 0.45...

 *****************************************************************/
```

   After you have prepared all of this, you can then run the command:

```
 ~osp/asg4.sparc64/hand_in
```

   Follow the on-screen prompts for what to do next. You may submit as many times as you wish, but only the final submission will be graded. All submissions are timestamped, any submission after the deadline will be penalized accordingly.

NOTICE:
UNAUTHORIZED COLLABORATION OR ASSISTANCE IS PROHIBITED. IF YOU HAVE A QUESTION, ASK THE TEACHING ASSSITANTS. DO NOT COPY CODE FROM A NEIGHBOR. STUDENTS WHO PARTICPATE IN COPYING WILL RECEIVE AN 'E' GRADE FOR THE COURSE. THE SAME PENALTY APPLIES FOR STUDENTS WHO FAIL TO ADEQAUTELY PROTECT THEIR ACCOUNTS.