CS314-001 Suggested Project #2 Milestones

Milestone #1: Write a short program (i.e., a *.cpp source code file) that performs the followings:

(1) Create (define and attach) a block of shared memory, which consists of:

   ```
   int x; // for testing purpose (can be eliminated later)
   char message [256]; // the shared text message
   ```

(2) Create a semaphore

(3) Initialize the new semaphore by ‘1’

(4) Assign any arbitrary integer to ‘x’ in the shared memory

(5) Assign any arbitrary text (a character string using “strcpy”).

(6) Sleep for 10 seconds

(7) Delete (detach) the shared memory from the memory

(8) Delete the semaphore

Review the sample source code: “with_semaphore.cpp” (posted to the CS314 course home)
Expected time needed to complete this milestone: 30 minutes
Suggested time window: within 3 to 4 days (after you receive this handout)

Milestone #2: Write a short program (i.e., “CS314_P2_999.cpp”) that performs the followings:

(1) Create a new child process (“1st child process”), using “fork” system call and let the child process to sleep (as soon as it is created) for 10 seconds.

(2) Create another new child process (“2nd child process”), using “fork” system call and let the child process to sleep (as soon as it is created) for 10 seconds.

(3) Create another new child process (“3rd child process”), using “fork” system call and let the child process to sleep (as soon as it is created) for 10 seconds.

(4) Create another new child process (“4th child process”), using “fork” system call and let the child process to sleep (as soon as it is created) for 10 seconds.

(5) Create another new child process (“5th child process”), using “fork” system call and let the child process to sleep (as soon as it is created) for 10 seconds.

(6) Let the “parent process” sleep for 10 seconds.

(7) Let all the five processes terminate.

Expected time needed to complete this milestone: 2 to 3 hours
Suggested time window: within 3 to 4 days (after you receive this handout)
Milestone #3: Integrate the above two programs to implement the following using "spin wait" for necessary process synchronizations (the parts in rectangles) as shown below:

Expected time needed to complete this milestone: 5 to 6 hours
Suggested time window: within 7 days
Milestone #4: Integrate the above two programs to implement the following without using “spin wait” for necessary process synchronizations (the parts in rectangles) as shown below:

- Where a “spin wait” is NOT allowed

START
create a shared memory
create semaphores
create a process
create a process
create a process
create a process
create a process
wait until all five processes are active
wait until all five processes are active
wait until all five processes are active
post a message NUM_REPESAT times
post a message NUM_REPESAT times
terminate
terminate
terminate
terminate
terminate
read the posted message as long as W1 and W2 are active
read the posted message as long as W1 and W2 are active
read the posted message as long as W1 and W2 are active
read the posted message as long as W1 and W2 are active
read the posted message as long as W1 and W2 are active
wait until all five processes are active
wait until all five processes are active
wait until all five processes are active
wait until all five processes are active
wait until all five processes are active
terminate
terminate
terminate
terminate
terminate
wait until all four children terminate
wait until all four children terminate
wait until all four children terminate
wait until all four children terminate
wait until all four children terminate
delete the shared memory
delete the shared memory
delete the shared memory
delete the shared memory
delete the shared memory

Expected time needed to complete this milestone: 5 to 6 hours

Suggested time window: To be completed by 9:30 a.m., on April 3rd.