CS418: Operating Systems

Assignment 2 (10 marks)

This assignment is on process/thread synchronization. You will implement a variation of the Producer/Consumer problem. Assume the following:

a. The buffer is a doubly linked list (for this assignment, let us assume that it contains at most 16 nodes). Each node has a random integer value (which should be less than 40). Initially the linked list contains 3 nodes.

b. Producer #1 will generate a node and add it at the end of the linked list, the value of the new node is a random odd integer less than 40. Producer #2 will generate a node and add it at the end of the linked list, the value of the new node is a random even integer less than 40. When the buffer is full, both should generate a message and wait.

c. Consumer #1 will delete, from the head of the list, the first node whose value is odd. Consumer #2 will delete, from the head of the list, the first node whose value is even. When the buffer is empty, both should generate a message and wait.

Each of the 4 processes/threads prints the contents of the linked list before and after it gets access to the linked list. Your output should contain the running result from every process/thread. **You must call <pthread.h> to use pthread_create to create a thread, etc and you must implement your own protected linked list. Using an existing library/language which supports protected linked list operations, e.g. JAVA, is forbidden.** (If your program got stuck, analyze the reason from the current output.)

Date Due: before the end of the class on **Wednesday, November 16, 2005 (i.e., before 4:50pm, Nov 16, 2005).** Only hand in the source code and output. Do not hand in diskette unless requested. Any late assignment will lose 2 marks for each late day.