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 with at most 15 nodes. Each node has a random integer value (which should be less than 30). 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 30. 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 30. When the buffer is full, both should generate a message and wait.

c. Consumer #1 will delete the first node whose value is odd from the head of the linked list. Consumer #2 will delete the first node whose value is even from the head of the linked list. 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 finish the assignment, i.e. JAVA is forbidden in this assignment. (If your program got stuck, analyze the reason from the current output.)

Date Due: before the end of the class on Monday, April 14, 2003 (i.e., before 8:50am, April 14, 2003). Only hand in the source code and output. Do not hand in diskette unless requested. Any late assignment will lose 3 marks for each late day.