Computer Science 338 - Homework/Lab 4

due: 9:55 AM, Thursday, October 5, 2000

1. *The Dining Savages.* A tribe of savages eats communal dinners from a large pot that can hold M servings of stewed missionary. When a savage wants to eat, he helps himself from the pot, unless it is empty. If the pot is empty, the savage wakes up the cook and then waits until the cook has refilled the pot. The behavior of the savages and cook is defined by the following processes:

   $Savage_i$: while (1) { get serving from pot; eat; }
   $Cook$: while (1) { sleep; put M servings in pot; }

   Develop psuedocode for the actions of the savages and the cook. Use semaphores for synchronization. Your solution should avoid deadlock and awaken the cook only when the pot is empty.

2. Recall the example from class on September 26 which showed an implementation of a producer/consumer interaction though a buffer of one integer using pthreads to create the producer and consumer, and used unix semaphores for synchronization. Modify this code (i) to allow the buffer to be of any size by a second command-line argument, and (ii) to allow the number of producer threads and consumer threads to be controlled by additional command-line arguments. Run it on the FreeBSD systems in the Williams CSLab and on the Origin 2000 at NCSA. Compare the performance for different numbers of threads and different buffer sizes in each case. What conclusions can you draw from these comparisons?