Threads II

CS314 Operating Systems

NOTE: Use the POSIX pthread and semaphore (sem_post/sem_wait) libraries to solve the following thread problems.

PROBLEM A:

Little rascals, Spanky, Alfalfa, and Darla are sitting around a table waiting to eat an ice cream sundae. To eat a sundae, you need three things:jimmies, ice cream, chocolate sauce.

- Spanky has ice cream and chocolate sauce.
- Alfalfa has jimmies and chocolate sauce, and
- Darla has jimmies and ice cream

Initially all three kids are "blocked."

Their lovely teacher, Miss Crabtree,   has an infinite supply of each item. Miss Crabtree randomly places one item on the table, unblocking the kid who needs it, and then Miss Crabtree blocks (waits). The kid who needs that item takes it and eats.   Only one item may be placed on a table at any time. Each rascal waits for the one item he/she needs then consumes some ice cream for a random amount of time unblocking Miss Crabtree when he/she is done eating. Miss Crabtree then puts another item on the table and the scenario continues.

Write a program that synchronizes the threads such that deadlock does not occur and no little rascal starves. The output should specify which rascal is eating.

THANKS TO RALPH FOR THIS PROBLEM

PROBLEM B:

Enhance the reader/writer problem we discussed in class using N readers and 1 writer so that the writer will not suffer from starvation.

Simulate readDB() and writeDB() using sleep().

Ensure that the delay for readDB() is much shorter than writeDB().

PROBLEM C:

Consider a doctor's office with a very overworked doctor.

The doctor has an examining room and a waiting room with a number of chairs in it. When the doctor finishes with a patient, the patient leaves and the doctor goes to the waiting room to see if there are other patients waiting. If there are, she brings one of them back to the examining room and examines the patient. If there are no patients waiting, she returns to her examining room, lies down on the examining table and goes to sleep.

Each patient (or I should say impatient), when he arrives, knocks on the examining room door to see what the doctor is doing. If the doctor is sleeping, then the patient wakes her up and enters the

examining room. If the doctor is performing an examination, then the patient goes to the waiting room. If there is a free chair in the waiting room, the patient sits in it and waits his turn. If there is no free chair, then the patient leaves.

Assume there are 10 chairs in the waiting room. The doctor must be a thread. Each patient must be a thread, assume 13 patients, each running in an infinite loop to simulate arriving at the office, being examined (or not), then leaving… then repeat.