Johnny Li

COP4600

HW 2.3

23. Does the busy waiting solution using the turn variable (Fig. 2-23) work when the two processes are running on a shared-memory multiprocessor, that is, two CPUs sharing a common memory?

The busy waiting solution using the turn variable (Fig. 2-23) would work when the two processes are running on a shared-memory multiprocessor, but still busy waiting occurs.

24. Does Peterson’s solution to the mutual-exclusion problem shown in Fig. 2-24 work when process scheduling is preemptive? How about when it is nonpreemptive?The Peterson’s solution to the mutual-exclusion problem shown in Fig. 2-24 work when process scheduling is preemptive but fails when it is nonpreemptive.

30. Consider the following solution to the mutual-exclusion problem involving two processes P0 and P1. Assume that the variable turn is initialized to 0. Process P0’s code is presented below.

/\* Other code \*/

while (turn != 0) { } /\* Do nothing and wait. \*/

Critical Section /\* . . . \*/

turn = 0;

/\* Other code \*/

For process P1, replace 0 by 1 in above code. Determine if the solution meets all the required conditions for a correct mutual-exclusion solution.

For a mutual-exclusion solution to correct, it must meet the safety, concurrency, and liveness requirements.

Safety: A variable is used to “turn”, ensure that processes P0 and P1 do not conflict. The processes are not inside their critical regions are the same time. Requirement met.

Concurrency: No assumptions on the speed and number of CPUs. Requirement met.

Liveness: In this case, P1 not being in its critical section is preventing P0 from entering its critical section, P0 is being blocked by P1 even if it’s not in its critical region. Requirement fails.

Therefore, for process P1, replace 0 by 1 in above code, the solution does not meet all the required conditions for a correct mutual-exclusion solution.

36. A fast-food restaurant has four kinds of employees: (1) order takers, who take customers’ orders; (2) cooks, who prepare the food; (3) packaging specialists, who stuff the food into bags; and (4) cashiers, who give the bags to customers and take their money. Each employee can be regarded as a communicating sequential process. What form of interprocess communication do they use? Relate this model to processes in UNIX.

Each employee can be thought as a process where the form of interprocess communication is message passing. Direct communication model of IPC message is sent by the process through the fast-food restaurant. The order is taken, and a message is passed to the cook to prepare the food. When the food is done, a message is sent to the packaging specialists and then passed to the cashier. In UNIX, the pipes share the memory between the different processes.