CS 385: HW3 – Due March 30, 2004

Jon A. Solworth

March 18, 2004

1. **Semaphore** There are three processes p_1, p_2 and p_3 and three code segments c_1, c_2 and c_3. Design a solution, using semaphores, such that the first code segment to be executed is c_1 and p_1 is the process executing it. This is followed by segment c_2 being executed by process p_2 followed by segment c_3 being executed by process p_3. And this happens cyclically, that is now process p_1 executes segment c_1 and so on.

2. In the *reader–writer* solution given in the notes, the readers can starve out the writers. Present a semaphore based solution to reader–writer that does the operations in order but still supports concurrent readers.

3. **Two Phase Commit** If two processes are waiting for the same location abort the process with the higher process id (pid). Does this method have any problem?

4. **Wait For Graph** Are the following wait-for-graphs deadlock free? If yes give a possible sequence for the allocation of resources and process completion. If not, list some of the cycles in the graph.

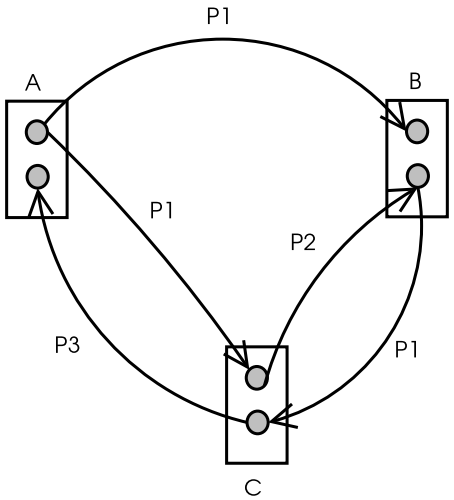

Figure 1: (a)Three resources A, B and C and three processes p1, p2 and p3.

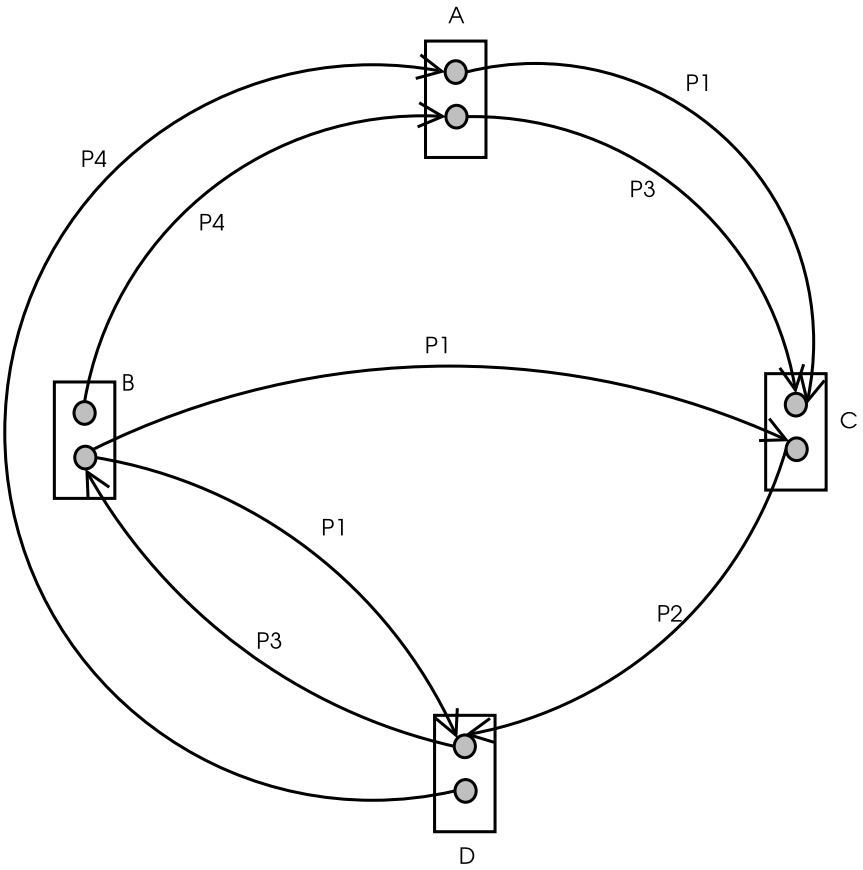

Figure 2: (b)Four resources A, B, C and D and four processes p1, p2, p3 and p4.

5. **Deadlock**

   (a) Consider two arrays of resources A[10] and B[20] (each element in the array is an individual resource). Describe an allocation scheme which avoids deadlock by preventing *circular wait*.

   (b) Consider a system in which a process requests a set of resources while holding another set. If the resources are unavailable it releases all the held resources and starts requesting resources afresh. Can this scheme deadlock. Why?

   (c) How does the operating system prevent deadlocks involving the disk drives?

   (d) Consider four resources A, B, C and D. A process can preempt a resource held by another process if it is holding more resources than the process holding the resource in contention. Is this allocation scheme deadlock-free? Explain.