HW 1

3. TACC UserID: vsridhar

4.

1. With 2 processes (P0 and P1), if P0 runs first and sets its turn to itself, it will skip the while loop (since turn is 0 and not 1) and go into the critical section. If P1 starts running when P0 is already in the critical section, and also sets turn to itself, it will skip the while loop (since turn is 1 and not 0) and then enter the critical section, so both processes will be in the critical section at the same time.
2. If we set turn before setting wantCS, there could be a scenario where P0 and P1 are both in the critical section at the same time. If P0 sets its turn to 1, and then is interrupted by P1, then turn will become 0 and wantCS[1] will be true. Since P0 didn’t get to set wantCS[0] yet, the while loop condition is false, so P1 will enter the critical section. If P1 is interrupted here, P0 can also enter the critical section. Turn is still 0, so the while loop condition for P0 is false, and both P0 and P1 will then be in the critical section.

5.

|  |  |
| --- | --- |
| P0 | P1 |
| wantCS[0] = true;  turn0 = 1;  while((turn1 = 1) ^ wantCS[1])  no-op();  CS  wantCS[0] = false; | wantCS[1] = true;  turn1 = 1;  while((turn0 = 1) ^ wantCS[0])  no-op();  CS  wantCS[1] = false; |

Proof by contradiction:

Assume P0 and P1 are both in the critical section.