3) TACC ID:

j3elam

djohn8

4a) Counter Example to a selfish turn variable selection:

|  |  |  |
| --- | --- | --- |
| Instruction #: | Process 0: P0 | Process 1: P1 |
| 1 | wantCS[0] = true; |  |
| 2 |  | wantCS[1] = true; |
| 3 | sets turn = 0; then enters CS |  |
| 4 |  | sets turn = 1; then enters CS |
|  | CS | CS |

* **Both P0 and P1 are in CS, which violates mutual exclusion**

4b) Counter Example to turn variable set before want variable:

|  |  |  |
| --- | --- | --- |
| Instruction #: | Process 0: P0 | Process 1: P1 |
| 1 | P0 sets turn = 1; |  |
| 2 |  | P1 sets turn = 0; |
| 3 | sets wantCS[0] = true; then enters CS |  |
| 4 |  | sets wantCS[1] = true; then enters CS |
|  | CS | CS |

* **Both P0 and P1 are in CS, which violates mutual exclusion**

5) public class PetersonAlgorithm implements Lock {

boolean wantCS[] = {false, false};

int turn0, turn1 = 1;

public void requestCS(int i) {

if(i == 0) {

int j= 1-i;

wantCS[i] = true; turn0 = j;

while (wantCS [j] && (turn1 == i));

}

if(i==1) {

int j = 1-i;

wantCS[i] = true;

turn1 = j;

while (wantCS [j] && (turn0 == i));

}

}

public void releaseCS ( int i ) { wantCS [i] = false; }

}

}

6) Counter example:

* Key
  + selects = number chosen, but not yet stored in number[]

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction #: | Process 0: P0 | Process 1: P1 | Process 2: P2 |
| 1 | selects 1 |  |  |
| 2 |  | selects 1 |  |
| 3 | number[0] = 1 |  |  |
| 4 | enters CS |  |  |
| 5 |  |  | selects 2 |
| 6 |  |  | number[2] = 2 |
| 7 | number[0] = 0, exits CS |  |  |
| 8 |  |  | enters CS |
| 9 |  | number[1] = 1 |  |
| 10 |  | enters CS |  |
|  |  | in CS | in CS |

* both P1 and P2 in CS, which violates safety
* since no one had to wait for choosing[], P2 was allowed to enter the CS before P1 had stored its number and entered the CS