

### #Example 1:

□ For Peterson's problem below conditions will applied.

- Each statement will take 1ms to complete.
- For process 0: i=0, j=1; and for process 1: i=1, j=0.
- Context switching will occur after 3ms.
- In critical section area carried only 1 statement.
- In remainder section area carried only 1 statement.
- Initial information common to both processes:
- turn=0;
- flag [0] =FALSE;
- flag [1] =FALSE;

```
bool flag [2] = {false, false};
int turn;
P0 Process:
flag [0] = true;
turn = 1;
while (flag [1] == true && turn == 1)
{
    // busy wait
}
critical section
flag [0] = false;
```

**Solution:** Information common to both processes:

flag [0] = false;

flag [1] = false;

turn=0;

Process 0	Process 1
i=0, j=1	i=1, j=0
1.flag [0] = true; 2.turn = 1; 3.while (flag [1] == true && turn == 1); //check the condition is false because flag [1] = false // since condition is false, no waiting in while loop. // Enter the critical section and process 0 happens to lose the processor.	
	1.flag [1] = true; 2.turn = 0; 3.while (flag [0] == true && turn == 0); // check the condition which is true, it keeps busy waiting until it loses the processor.
// process 0 resume until it finishes in the critical section. 4. Execute line #1 of C.S; // leave C.S 5. flag [0] = false; // start executing the remainder section of the process 6. Execute line #1 of R.S; // process 0 happens to lose the processor.	

	<p>4.while (flag [0] == true &amp;&amp; turn == 0); // check the condition is false because flag [0] = false. // no busier waiting and enter the critical section.</p> <p>5. Execute line #1 of C.S;</p> <p>6. flag [1] = false;</p>
<p>7.flag [0] = true;</p> <p>8.turn = 1;</p> <p>9.while (flag [1] == true &amp;&amp; turn == 1); //check the condition is false because flag [1] = false // since condition is false, no waiting in while loop. // Enter the critical section and process 0 happens to lose the processor.</p>	
	<p>7. Execute line #1 of R.S;</p> <p>8. flag [1] = true;</p> <p>9.turn = 0;</p>