1. ShowPeterson’s solution for the given scenario.

* There are two processes: P0 and P1.
* Each Statement takes 4 ms to execute.
* Context Switch will occur after 8 ms.
* Critical section contains 2 statements.
* Remainder section contains 4 statements.
* For P0: i=0 and j=1
* For P1: i=1 and j=0
* turn=0
* flag[0] = TRUE, flag[1] = FALSE

**The structure of process Pi in Peterson’s solution:**

|  |
| --- |
| do{  flag[i] = true;  turn = j;  while(flag[j] == true && turn == j){  //busy wait  }  //critical section  flag[i] = false;  //remainder section  }while(true); |

Complete the table given below for processes P0 and P1 using Peterson’s solution.

|  |  |
| --- | --- |
| Process 0: i=0, j=1 | Process 1: i=1, j=0 |
| Flag [0]=True  Turn=1 |  |
|  | Flag[1]=True  Turn=0 |
| While {turn false)  CS1 |  |
|  | While True |
| CS2  Flag[0]=False |  |

|  |  |
| --- | --- |
|  | While False  CS1 |
| Rs1  Rs2 |  |
|  | CS2  Flag[1]=False |
| Rs3  Rs4 |  |
|  | Rs1  Rs2 |
|  | Rs3  Rs4 |

1. ShowPeterson’s solution for the given scenario.

* There are two processes: P1 and P2.
* Each Statement takes 3 ms to execute.
* Context Switch will occur after 15 ms.
* Critical section contains 6 statements.
* Remainder section contains 10 statements.
* For P1: i=1 and j=0
* For P2: i=0 and j=1
* turn=0
* flag[0] = FALSE, flag[1] = TRUE

**The structure of process Pi in Peterson’s solution:**

|  |
| --- |
| do{  flag[i] = true;  turn = j;  while(flag[j] == true && turn == j){  //busy wait  }  //critical section  flag[i] = false;  //remainder section  }while(true); |

|  |  |
| --- | --- |
| Process 1: i=1, j=0 | Process 2: i=0, j=1 |
| Flag[1]=True  Turn=0  While False  CS1  CS2 |  |
|  | Flag[0]=True  Turn=1  While True  Busy wait |
| CS 3, 4 ,5 ,6  Flag[1]=False |  |

|  |  |
| --- | --- |
|  | While False  CS 1-4 |
| RS 1-5 |  |
|  | CS5,6  Flag[0]=False  RS 1,2 |
| RS 5-10 |  |
|  | RS 3-10 |