

Quiz 3 (4H)

Solution

Question 2: [5 Marks] Consider multiple threads executing the following three functions (fun_1, fun_2, fun_3). These threads print a string containing any number of a's, b's and c's in any order. Synchronize the threads (using Semaphores) so that the string becomes a concatenation of the substring "abbc".

Following are few examples:

abbc (correct)

abbc abbc (correct)

bab (incorrect)

abba cbba (incorrect)

```
T1
void func1() {
    wait(S1);
    cout << "a";
    signal(S1);
}
```

```
T2 int counter=0;
void func2() {
    wait(S2);
    counter++;
    cout << "b";
    if (counter % 2 == 0)
    { signal(S3);
    }
    else { signal(S2); }
}
```

```
T3
void func3() {
    wait(S3);
    cout << "c";
    signal(S3);
}
```

Question 3: [marks 5]

Consider the following ticket numbers for given processes and draw the table, how it will be dealt by Bakery algorithm.

Process	Ticket #	(Ticket #, Pid #)
P0	0	(0,0)
P1	5	(5,1)
P2	3	(3,2)
P3	4	(4,3)
P4	3	(3,4)

j	P0	P1	P2	P3	P4
0	number[1]=1	number[1]=1	number[1]=1	number[1]=1	number[1]=1
1	(5,1) < (0,0)	(5,1) < (5,1)	(5,1) < (3,2)	(5,1) < (4,3)	(5,1) < (3,4)
2	(3,2) < (0,0)	(3,2) < (5,1)	(3,2) < (3,2)	(3,2) < (4,3)	(3,2) < (3,4)
3	(4,3) < (0,0)	(4,3) < (5,1)	(4,3) < (3,2)	(4,3) < (4,3)	(4,3) < (3,4)
4	(3,4) < (0,0)	(3,4) < (5,1)	(3,4) < (3,2)	(3,4) < (4,3)	(3,4) < (3,4)

P(0) has no ticket it is not ready to execute its critical section

P1 waits for P2, P3 and P4

P3 waits for P2 and P4

P4 waits for P2 (ticket# is tie but $2 < 4$ the pids)

we can also omit P0 as it is not ready for executing its critical section.