

Tutorial Session N°1: Mutual Exclusion

Objective: Learn how to verify a software solution for mutual exclusion against Dijkstra's four conditions.

Study the algorithmic solutions presented below to the mutual exclusion problem. Determine whether each solution is correct, and justify your answer with reference to Dijkstra's four conditions:

1.

Common context:

c : boolean; {initialized to false}

Process P_i; {i = 1, 2}

Begin

...;

Repeat

 while c do;

 c := true;

 CRITICAL SECTION;

 c := false;

 ...;

Until false;

End;

2.

Common context:

c : boolean; {initialized to false}

Turn : 1..2; {initialized to 1 or 2}

Process P_i; {i = 1, 2}

Begin

...;

Repeat

 Turn := j;

 while (Turn ≠ i) and (c) do;

 c := true;

 CRITICAL SECTION;

 c := false;

 ...;

Until false;

End;

3.

Common context:

Flag: array [0,1] of boolean; {initialized to false}

Turn: 1..2; {Turn initialized to 1 or 2}

Process P_i;

Begin

 Repeat

 :::

 Flag[i] := true

 while (Flag[j]) {

 if (Turn ≠ i) {

 Flag[i] := false;

 while (Turn ≠ i) ;

 Flag[i] := true ;

 }

 }

 CRITICAL SECTION;

 Flag[i] := false;

 Turn := j;

 :::

Until false;

End;

4.

Common context:

flag: array [1..2] of boolean; {initialized to false}

turn: 1..2; {initialized to 1 or 2}

Process P_i; {i = 1, 2}

Begin

 ...

 flag[i] := true;

 while (turn ≠ i) {

 while (flag[j]) do ;

 turn := i;

 }

 CRITICAL SECTION;

 flag[i] := false;

 ...

End;