Exercice 4 :

Soit l'algorithme suivant pour les processus P0 et P1 :

Var D_i, D_j, Use_i, Use_j : booléen := faux ;
Processus P_i
While (true){
D_i = Vrai ;
While (Use_j);
Use_i = vrai ;
if (D_j = vrai & i > j) :{While (Use_j)} **;**

< Section critique>

Use_i = faux ;
D_i :=faux ;
}.

1- Vérifier les quatre conditions de l'exclusion mutuelle.
2- Cet algorithme peut-il être retenu comme solution pour protéger la SC?

Solution :

EM :

Var D_i, D_j, Use_i, Use_j : booléen := faux ;

P0

Processus P_0
While (true){
D_0 = Vrai ;
While (Use_1); Use_1 = false donc il continue
Use_0 = vrai ;
if (D_1 = vrai & 0 > 1) :{While (Use_j)} **; non il rentre en SC**

< Section critique>

Processus P_1 arrive
While (true){
D_1 = Vrai ;
While (Use_0); oui il reste dans cette boucle

Le cas d'une interruption

P1

While (true){
D_1 = Vrai ;
While (Use_0);
←----------Interuption

P0 arrive

While (true){

D_0 = Vrai ;
While (Use_1); non car Use_1 = false
Use_0 = vrai ;
if (D_1 = vrai & 0 > 1) :{While (Use_1)} **;** non il rentre en SC

P1 reprend l'exécution dans ce temps ←----------< Section critique>
Use_1 = vrai ;
if (D_0 = vrai & 1 > 0) :{While (Use_0)} **; condition vérifié il reste dans la boucle**

----------**Cas 2**----------

P0

While (true){
D_0 = Vrai ;
While (Use_1); Non
←----------Interruption---------- P1 arrive

While (true){
D_1 = Vrai ;
While (Use_0); non car Use0=false
Use_1 = vrai ;
if (D_0 = vrai & 1 > 0) oui :{While (Use_1)} **; il reste dans la boucle il ne rentre pas en SC**

P1 reprend l'execution dans ce temps ←----------
Use_1 = vrai ;
if (D_1 = vrai & 0 > 1) :{While (Use_0)} **; condition vérifié il reste dans la boucle**

Donc la première condition est vérifiée, un seul processus au plus dans la section critique

2/ Absence d'interblocage

Processus P_0
While (true){
D_0 = Vrai ;
While (Use_1); non
Use_0 = vrai ;
if (D_1 = vrai & 0 > 1) :{While (Use_1)} **;**
la condition n'est pas vérifié il rentre en SC
< Section critique>

Processus P_1
While (true){
D_1 = Vrai ;
While (Use_0); non
Use_1 = vrai ;
if (D_0 = vrai & 1 > 0) :{While (Use_0)} **;**

< Section critique>

La condition d'absence d'interblocage est vérifier on a au moins un processus qui a rentré en SC qui est le processus P0 dans ce cas

3/ La progression est vérifiée (faite en TD)
4/ L'attente borné est vérifiée (faite en TD)

2 - Oui, cet algorithme peut être retenu comme solution pour protéger la SC car toutes les conditions sont vérifiées.