## Exercice 1 : Ordonnancement

1. FP préemptif

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | P(ResB) | P(ResA) |  |  |  |  |  |  |  |  |  | V(ResA) | V(ResB) |  |  |
| τH |  |  |  |  |  |  |  |  |  |  |  | 1ms |  |  | 4ms |  |  |  |
|  |  | P(ResA) |  | P(ResB) |  |  |  |  |  |  | V(ResB) | V(ResA) |  |  |  |  |  | t |
| τL |  |  |  | 1ms |  |  |  |  |  | 7ms |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | t |

1. Quel problème y-a-t-il? Comment appelle-t-on cette situation?

La tâche prioritaire (τH) est retardée par la tâche moins prioritaire (τL).

On arrive ainsi à un problème d’inversion de priorité : « Une inversion de priorité correspond à une situation où l’exécution d’une tâche est retardée par une autre tâche moins prioritaire en attendant que celle-ci libère une ressource. »

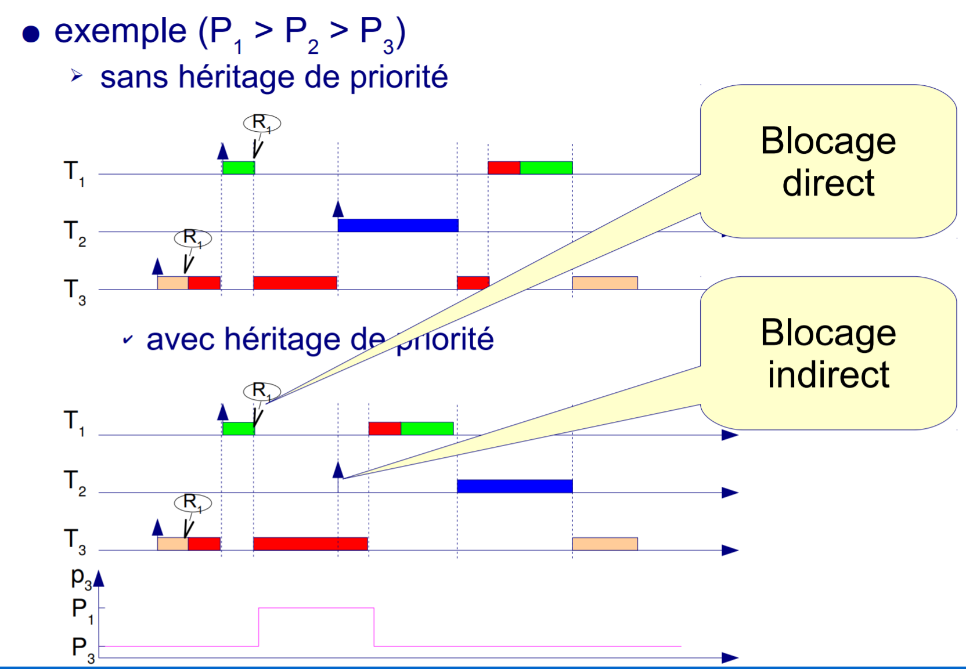
1. Quelles pourraient être les priorités de plafond associées aux deux ressources ResA et ResB?

Les priorités associées pourraient être 9 et 8 respectivement.

1. Plafond de priorité

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ResA |  |  |  |  |  |  |  |  |  |  | V(ResA) | 1ms |  |  | 4ms | V(ResA) |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ResB |  |  |  | 1ms |  |  |  |  |  | 7ms | V(ResB) |  |  |  |  |  | V(ResB) |  |
|  |  |  |  | P(ResB) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| τH |  | P(ResA) |  |  |  | P(ResA) |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | P(ResB) |  |  |  |  |  |  |  |  |  |  |  |  |  | t |
| τL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | t |

1. Intérêt du plafond de priorité

Le protocole d'héritage de priorité minimise le retard dû à des effets d'inversion de priorité ; en effet, les tâches liées aux ressources sont toujours prioritaires, ce qui peut réduire les délais d’attente de la ressource dans certains cas.

## Exercice 2 : Philosophes

Les philosophes ne pourront pas tous manger en même temps ; des sémaphores vont devoir être utilisés.

Chaque philosophe pense et demande à prendre les fourchettes de part et d’autre de son assiette. S’il y arrive, il mange et les repose, sinon il attend qu’elles soient disponibles. Cette solution empêche tout deadlock car seul un philosophe peut manger à tout moment.

philosophe(int i, int nbphilosophes){

gauche = i

droite = (i+1) % nbphilosophes

while(1){

penser(i)

if(gauche < droite){

P(fourchette[gauche])

P(fourchette[droite])

}else{

P(fourchette[droite])

P(fourchette[gauche])

}

manger(i)

V(fourchette[gauche])

V(fourchette[droite])

}

}