Table des matières

[INTRODUCTION 2](#_Toc58455373)

[I. PHENOMENE D’INTERBLOCAGE (deadlock) 3](#_Toc58455374)

[a) CONDITION NECESSAIRE 3](#_Toc58455375)

[I.1 Cas particulier de 2 processus 3](#_Toc58455376)

[a) Evitement 4](#_Toc58455377)

[b) Algorithme de Résolution de l’interblocage (cas de 2 processus) 5](#_Toc58455378)

[I.2 GENERALISATION 6](#_Toc58455379)

[a) Evitement 6](#_Toc58455380)

[II. PRESENTATION DU MODELE LECTEUR/REDACTEUR 7](#_Toc58455381)

[III. ETUDE DES ELEMENTS DE LA SIMULATION 8](#_Toc58455382)

[a) Processus (voiture) 8](#_Toc58455383)

[b) Feu 8](#_Toc58455384)

[C) Route 9](#_Toc58455385)

[CONCLUSION 9](#_Toc58455386)

[BIBLIOGRAPHIE 9](#_Toc58455387)

INTRODUCTION

Le sujet de ce projet touche à la simulation du fonctionnement d’un Système d’exploitation en s’inspirant de celui d’un système routier. Un système routier est l’ensemble des voies de circulation terrestres permettant le transport des biens et personnes ceci grâce à un ensemble de moyens de locomotion et sous la supervision d’ordonnanceurs ce fonctionnement s’apparente donc à celui d’un système d’exploitation, qui est charge de la gestion des processus. Ainsi il convient de se demander quelle sont les opérations ayant lieu dans d’un système routier permettant de mettre en scène l’execution de certain processus d’un système d’exploitation ? comment adapter les principaux modèles de gestion des données à notre vision des différents transports routier ? Pour répondre à toute ces interrogations ils sera question dans ce devoir d’expliquer et simuler le problème d’interblocage qui est une conséquence de l’utilisation des sémaphores et un moyen de résolution grâce aux concepts de WAIT/LEASH et WAIT/RUN effectue par deux processus pour l’accès à une ressource, et d’en faire une généralisation, d’expliquer ce que nous entendons par la méthode F.I.F.O associer à notre simulation ,d’expliquer le modèle LECTEUR/REDACTEUR mis en exergue et enfin d’expliquer le fonctionnement des éléments entrant en jeu dans notre simulation.

1. PHENOMENE D’INTERBLOCAGE (deadlock)

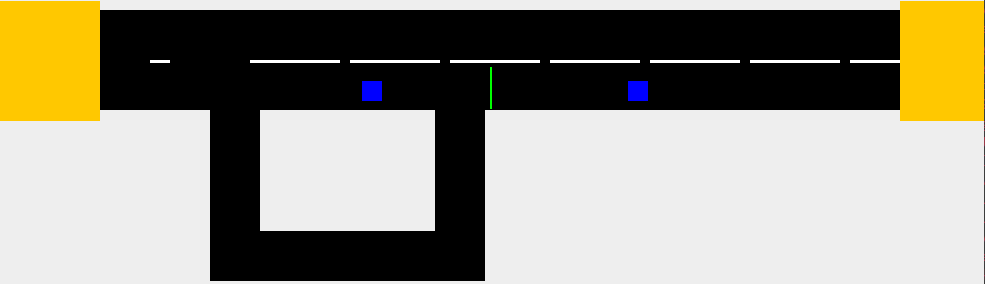
# CONDITION NECESSAIRE

C’est un état qui se produit lorsque au moins deux processus s’attente mutuellement (l’un détenant la ressource que l’autre a besoin). Pour qu’elle puisse avoir lieu, quatre conditions doivent être réunir à savoir :

* Les ressources doivent être utiliser en mode non partageable c’est-à-dire seul un processus ne peux utiliser la ressource en une instante donne
* La ressource ne peut être libérer que volontairement par le processus
* Chaque processus doit détenir au moins une ressource et demander des ressources supplémentaires détenus par d’autres processus
* Chaque processus doit attendre une ressource détenue par un autre processus qui, également attend que le premier processus libère sa ressource.

## Cas particulier de 2 processus

Dans le cadre de notre simulation nous avons utilisé deux processus (les processus assimiler ici a des voitures) et les ressources (assimiler a des portions de la route), tout en respectant les conditions d’un interblocage. Ici chaque processus dispose d’une ressource (portion de la route) ces ressources sont dans un ordre oppose selon le schéma ci-dessous :



Ressource détenue processus P1 processus P2 Ressource détenue

par P1 par P2

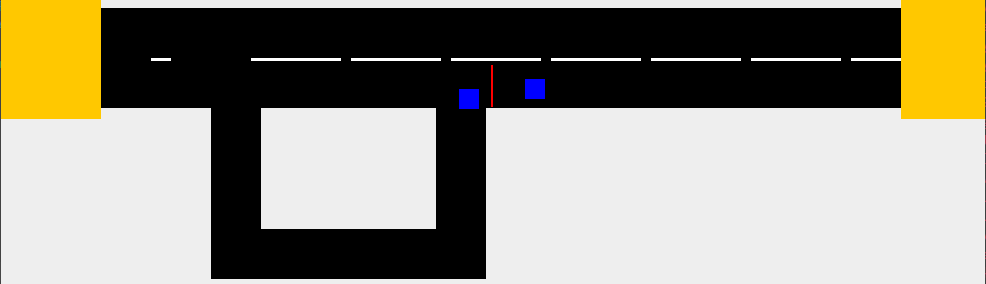
# Evitement

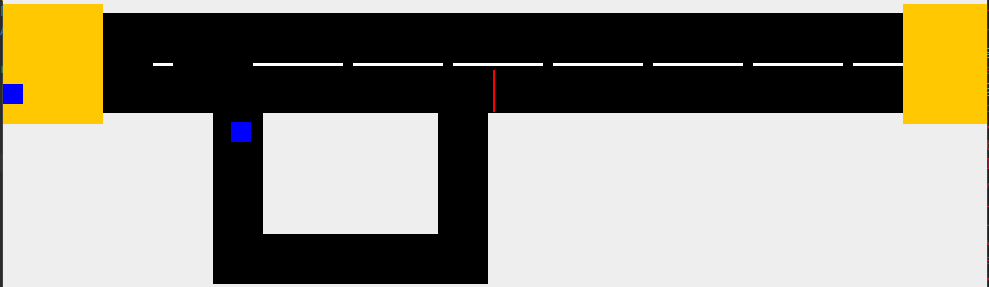
Les algorithmes WAIT/LEASH et WAIT/RUN sont deux méthodes d’évitement de l’état d’interblocage, elles consistent à mettre l’un des processus P1 qui était en état de WAITING en état de RUNNING et pendant ce temps l’autre processus P2 qui était en état de WAITING en état de LEASH.

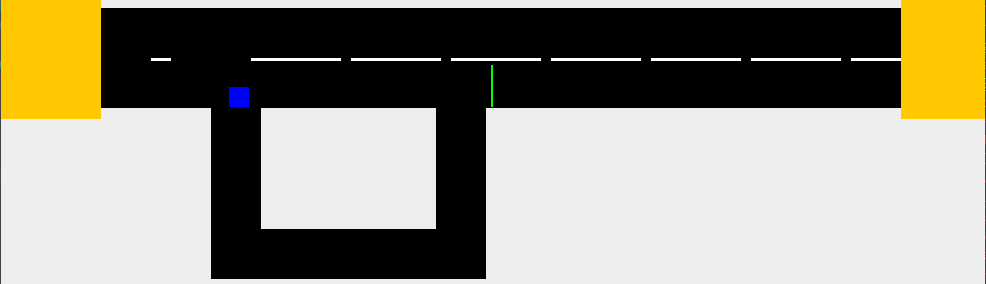
|  |  |  |
| --- | --- | --- |
|  | WAIT/LEAST | WAIT/RUN |
| P1 a besoin de la ressource détenue par P2 | P1 Attend | P2 Passe |
| P2 a besoin de la ressource détenue par P1 | P1 Passe | P2 Attend |

A chaque fois l’un des processus doit aller en attente (libérer volontairement sa ressource) pour que l’autre processus puisse avoir accès à la ressource dont il avait besoin (utiliser la ressource qui a été libérée), ici nous avons considéré que le processus P2 est prioritaire au processus au P1.

Dans le cadre de notre Simulation nous avons utilisé comme ordonnanceur un Feu de signalisation ce feu aura pour but de changer de couleur après un certain temps de manière à ce qu’il deviennent rouge lorsque les 2 processus P1 et P2 sont sur le point de se rencontrer à la position du Feu ,lorsque , une fois devenu rouge le processus P1 va en LEASH et libère sa ressource R1 laissant ainsi le temps au processus P2 de laisser sa ressource R2 et d’occupe entièrement la ressource R1 laissée par P1 et une fois que le processus P1 sort de l’état de LEASH il réoccupe sa ressource R1 (cette dernière ayant déjà été occupée par P2) et ensuite R1.







# Algorithme de Résolution de l’interblocage (cas de 2 processus)

Algo : ResolutionInterblocage

Variable : Processus : P1, P2 ;

Feu :f3 ;

Debut

Si((positionX(P1)>positionX(f3)-25) &&(positionX(P1)<positionX(f3))

&&(positionY(P1)==positionY(f3))&&(Color(f3)==red)) Alors :

/\*positionX, positionY des fonctions qui prennent en paramètre un processus ou un feu et renvoi leur abscisse et ordonnées respectivement ;Color est une fonction qui prend en paramètre un feu et renvoi comme résultat sa couleur\*/

LEAST(P1) ;

/\*LEAST est une fonction qui prend en paramètre un processus et veille à ce que ce processus libère volontairement sa ressource\* /

RUN(P2) ;

/\*prend en paramètre un processus et fais accéder ce processus a la ressource dont il a besoin \*/

Finsi

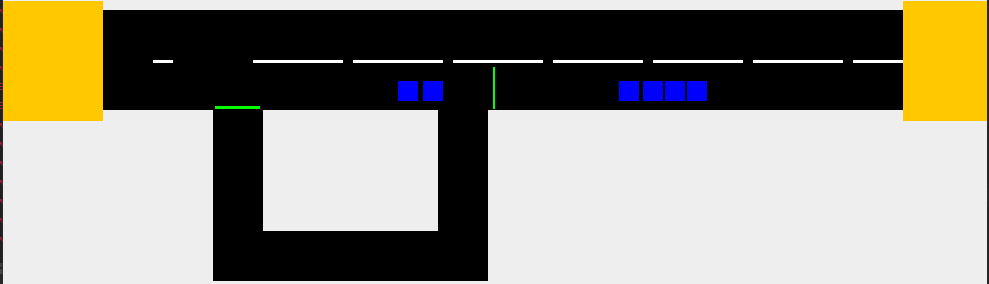
FinAlgo

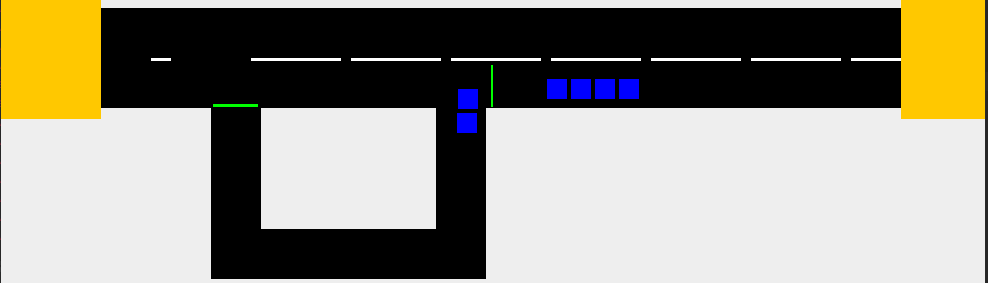
## GENERALISATION

Dans cette partie nous avons fait une généralisation du processus d’interblocage fait précédemment ceci en utilisant le même principe que précédemment, au lieu de deux processus nous ferons notre simulation avec plusieurs processus, et cela nous permettra de mettre en évidence la méthode du F.I.F.O (First In First Out) dans un premier temps et dans un second temps le modèle de LECTEUR/REDACTEUR.

# Evitement

Tout comme précédemment les processus de droite seront priorité a ceux de droite et ceux de gauche seront misent en attente selon l’ordre d’accès à la ressource et le premier à entrer en attente sera le premier à accéder à la ressource supplémentaire dont elle a besoin, ainsi de suite. Une fois que tous les processus de gauche seront en attente ceux de droite pourront accéder la ressource aisément dont ils avaient besoin mais cette fois-là les processus de gauche y accèderont de façon séquentielle c’est-à-dire si nous appelons l’ensemble des processus prioritaire de droite une fois les processus de gauche en attente accède la ressource R1 une fois terminer accède ensuite à la ressource R1 ainsi de suite jusqu’à ce que tous les aient accéder la ressource R1 (la raison pour laquelle ces processus accèdent de manière séquentielle a la ressource R1 sera expliqué dans la seconde partie de ce rapport), cet accès séquentielle sera contrôler par un verrou ce verrou ce verrouillera lorsqu’un élément de accèdera a R1, les autres éléments resteront en retrait attendant que l’élément en R1 termine son exécution une fois la ressource accédée par tous les les éléments maintenu en attente par le Feu Horizontale auront deux ressources à savoir R1 et R2 à leur disposition mais ceux-ci y accèderons seront la méthode du F.I.F.O vue plus haut.





1. PRESENTATION DU MODELE LECTEUR/REDACTEUR

Dans le cadre de la modélisation de ce modèle nous avons assimile la base de données ici est assimiler à la ressource R1, les rédacteurs aux processus de droite et les lecteurs à ceux de gauche dans un système routier les voitures de gauches seraient des éléments qui viennent modifier certaine caractéristiques de la route notamment :la température, l’épaisseur, la forme, la résistance aux intempéries,….chacun des processus de modifie l’une de ces caractéristiques, plus haut nous définit nos processus de droite comme des processus ne pouvant accéder à la ressource dont ils besoin que de façon séquentielle ceci dans le souci de respecter de la condition primordiale du modèle LECTEUR/REDACTEUR qui est qu’une base de données ne peux subir plusieurs modification simultanément ou de consulter une base de données au même moment qu’elle est modifiée.

Une fois tous les processus exécutés c’est-à-dire une fois tous les modification apporte a la route par les processus les processeurs de gauche qui sont des LECTEURS viennent les consultées, ceci de manière séquentielle ou simultanée.

1. ETUDE DES ELEMENTS DE LA SIMULATION

## Processus (voiture)

Comme nous l’avons écrit plus haut nos processus ont été assimiles à des voitures, ceux-ci sont caractérisées par :

* Un Sens : il vaut 1 si le processus va de la gauche vers la droite ,2 si le processus va de la droite vers la gauche, 3 si le processus va du haut vers le bas et 4 si ce dernier va du bas vers le haut.
* Des Positions : ces positions sont par rapport à l’horizontale et la verticale. Pour quitter d’une position X1 à une position X2 nous avons ajouté a X1 la valeur algébrique de la vitesse de ce processus, ceci à chaque fois qu’un processus change de sens.
* Une vitesse : elle est définie comme un entier et est utiliser pour quitter d’une position à une autre.

## Feu

Dans notre simulation nous ne disposons que d’un Feu celui-ci joue le rôle d’ordonnanceur, il s’assure que lorsque l’un des processus libère volontairement sa ressource lorsqu’il est en attente au niveau du Feu avec un autre processus. Comme éléments caractéristiques nous avons :

* Sa position
* Sa couleur : lorsqu’il est rouge tous les processus s’arrêtent (ceci lorsqu’ils sont situés à une certaine distance du feu), lorsqu’il est vert le processus passe.
* Un numéro

## C) Route

Elle contient le feu de signalisation et nos processus

CONCLUSION

Parvenu au terme de notre travail ou il a été question pour nous de simuler le fonctionnement d’un système d’exploitation, il en ressort que la résolution de l’interblocage se fait en éliminant l’une des conditions y aboutissant, mais généralement le système d’exploitation avant de lancer certains processus a des instants donnes s’assurent que le cela ne conduira pas à un interblocage ou à une inversion de priorité mais cela souvent très couteux en ressource, il est également a noter que les processus qui sont bloqués le sont indéfiniment , dans notre modèle LECTEUR/REDACTEUR nous avons considéré les processus de droite(rédacteurs) comme prioritaire et ceux de gauche (LECTEUR) comme moins prioritaire cela peut entrainer la famine chez des rédacteurs or cela doit être prise en compte par un système d’exploitation.

BIBLIOGRAPHIE

<https://fr.wikipedia.org/wiki/Interblocage>

Livre CSI 3201\_FRIntroduction aux systèmes d’exploitation de Diomede Nzisabira PP51-60

<https://fr.wikipedia.org/wiki/Semaphore>

<https://fr.wikipedia.org/wiki/Diner> des philosophes

Cour sur Les Systèmes d’exploitation du Professeur THOMAS NDJOTIO

Livre Systèmes d’exploitation : Communication entre processus de D. BECHET PP33-52.