

## PARTIE I. Application des principes des Sémaphores

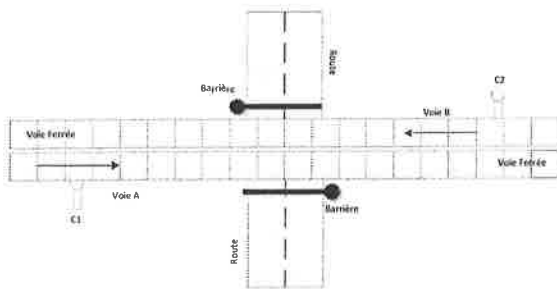
Barème 6 points

N'oubliez pas d'initialiser les variables, et commentez votre *pseudocode*.

On considère un passage à niveau à double voies composé de barrières sur chacun des axes routiers. Des capteurs  $C_1$  et  $C_2$  sont positionnés à une certaine distance en amont sur les voies ferrées unidirectionnelles A et B. Le train active le capteur  $C_x$  relatif à la voie sur laquelle il circule lors de son passage. Cette action provoque la fermeture des barrières. Les barrières restent fermées durant une durée  $D_{C_x}$  ( $D_{C_1}$  pour le capteur  $C_1$  et  $D_{C_2}$  pour le capteur  $C_2$ ). Dans notre cas, on considère que les trains ne peuvent pas s'arrêter à l'intersection de la voie et du passage routier.

On propose de simuler le comportement de ce passage à niveau en s'attachant à montrer précisément la synchronisation des entités qui prennent la forme de processus en activités. Ainsi, les processus seront les suivants :

- Processus Train ( $T_A$  et  $T_B$ ) relatifs aux trains utilisant la voie A et la voie B
- Processus Superviseur qui se charge de commander la fermeture et l'ouverture des barrières.



## Travail à effectuer :

1 - Développez une solution à ce problème en tenant compte des aspects suivants :

- Les barrières se ferment dès lors que le capteur est activé par le passage d'un train.
- Le temps de fermeture des barrières a été évalué de sorte que le passage à niveau soit sécurisé.
- Les barrières s'ouvrent dès lors que la durée  $D_{C_x}$  est dépassée et que les capteurs ne sont plus actifs.
- Les voies sont unidirectionnelles.
- Les trains peuvent se croiser. Attention un train peut en cacher un autre.

2 - Même question en considérant qu'il existe un capteur supplémentaire sur l'axe routier qui détermine l'arrêt ou non d'un véhicule sur le passage à niveau. On considère dans ce cas que ce capteur est actif si  $n$  secondes consécutives se sont écoulées. Si cette information est valide alors le train doit s'arrêter en amont du passage.

## PARTIE II. SGF

Barème 4 points

Soit un système Unix / Linux qui utilise une table d'allocation à 12 entrées. Les 10 premières entrées de la table contiennent l'adresse d'un bloc de données du fichier. La 11<sup>ème</sup> entrée contient l'adresse d'un bloc d'index (INDIRECT\_1) qui pointe sur des blocs de données, la 12<sup>ème</sup> entrée contient l'adresse d'un bloc d'index (INDIRECT\_2) qui pointe sur des blocs d'index qui eux-mêmes pointent sur des blocs de données. Considérons des tailles de blocs de 1024 octets et des adresses de bloc de 4 octets.

- 1- Quel est le nombre d'entrées par bloc d'index ?
- 2- Quel est la taille maximale d'un fichier contenu dans ce SGF ?
- 3- Pour un fichier de 526 Ko, en considérant que les blocs seront écrits de façon contiguë et que le SGF est initialement vide, quel est le nombre de blocs de données pointés par le bloc d'index INDIRECT\_1 et le nombre de bloc de données pointés par le bloc d'index INDIRECT\_2 ? Expliquez précisément la cartographie des blocs.

PARTIE III. Détection de l'Interblocage

Barème 4 points

Pour détecter un interblocage dans un système disposant de plus d'une instance pour chaque type de ressources, on peut utiliser l'algorithme de détection suivant : l'algorithme du banquier.

<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th></th><th>R0</th><th>R1</th><th>R2</th></tr> <tr><th>P0</th><td>1</td><td>1</td><td>0</td></tr> <tr><th>P1</th><td>7</td><td>4</td><td>2</td></tr> <tr><th>P2</th><td>3</td><td>2</td><td>2</td></tr> <tr><th>P3</th><td>4</td><td>4</td><td>1</td></tr> <tr><th>P4</th><td>4</td><td>0</td><td>3</td></tr> </table> <p>Matrice : Allocation</p>		R0	R1	R2	P0	1	1	0	P1	7	4	2	P2	3	2	2	P3	4	4	1	P4	4	0	3	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th></th><th>R0</th><th>R1</th><th>R2</th></tr> <tr><th>P0</th><td>7</td><td>2</td><td>5</td></tr> <tr><th>P1</th><td>8</td><td>4</td><td>4</td></tr> <tr><th>P2</th><td>4</td><td>3</td><td>5</td></tr> <tr><th>P3</th><td>5</td><td>7</td><td>6</td></tr> <tr><th>P4</th><td>4</td><td>2</td><td>3</td></tr> </table> <p>Matrice : Annonce</p>		R0	R1	R2	P0	7	2	5	P1	8	4	4	P2	4	3	5	P3	5	7	6	P4	4	2	3	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th>R0</th><th>R1</th><th>R2</th></tr> <tr><td>25</td><td>15</td><td>14</td></tr> </table> <p>Ressources Totales</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th>R0</th><th>R1</th><th>R2</th></tr> <tr><td>6</td><td>4</td><td>6</td></tr> </table> <p>Ressources Disponibles</p>	R0	R1	R2	25	15	14	R0	R1	R2	6	4	6
	R0	R1	R2																																																											
P0	1	1	0																																																											
P1	7	4	2																																																											
P2	3	2	2																																																											
P3	4	4	1																																																											
P4	4	0	3																																																											
	R0	R1	R2																																																											
P0	7	2	5																																																											
P1	8	4	4																																																											
P2	4	3	5																																																											
P3	5	7	6																																																											
P4	4	2	3																																																											
R0	R1	R2																																																												
25	15	14																																																												
R0	R1	R2																																																												
6	4	6																																																												

**Travail à effectuer :**

- a. Donnez l'ordre des processus exécutés en appliquant l'algorithme du banquier. Précisez chacune des étapes.
- b. Répondre à la question suivante : la suite est-elle fiable ?

PARTIE IV. Implémentation en langage C

Barème 6 points

Construire un programme selon le principe suivant : un processus père écrit une chaîne de caractères ( minimum 16 caractères ) dans une zone de mémoire partagée. Son fils récupère le contenu de la chaîne à partir de la zone de mémoire et écrit ensuite cette chaîne dans un fichier « Toto.txt ». Au moment de l'écriture dans le fichier, le fils insérera à chaque position paire une lettre P à la place de la lettre présente à la position initiale.

Exemple :

Père écrit dans la zone mémoire : « Le sudoku est un jeu en forme de grille »

Le fils écrit dans le fichier « PEPSPDPKP PSP PNPJPUPEP POPMP PEPGPIPLP »

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	
L	E		S	U	D	O	K	U		E	S	T		U	N	J	E	U		E	N		F	O	R	M	E		D	E		G	R	I	L	L	E		
P	E	P	S	P	D	P	K	P		P	S	P		P	N	P	J	P	U	P	E	P		P	O	P	M	P		P	E	P	G	P	I	P	L	P	