



Compte rendu  
Projet temps réel

**Synchronisation des trains**

Lundi le 8 Janvier 2018

Noms du binôme :

- Yasmina AIT ALI.
- Saliha OUDJIDANE.

Année universitaire:2017/2018

## Introduction :

Notre projet consiste à réaliser une application de synchronisation des trains, qui circulent tous en même temps chacun sur son trajet sans avoir des collisions. Dans ce qui suit nous allons expliquer en détail les démarches prises, et comment nous avons modélisé ce problème pour arriver au bon fonctionnement de ce système.

### 1. Modélisation du problème :

Afin de réaliser ce projet ,on a du passé par plusieurs modélisations pour ensuite arriver à une modélisation plus optimale et qui approche très bien le problème posé.

**1.1. Les trains :** on a choisi de les représenter par des threads , 3 threads ,chacun d'entre eux exécute son propre trajet un nombre de fois infinie (dans notre cas 50 tours pour chaque train) .notre structure de donnée représentant les threads nous permet d'ajouter ou bien de supprimer des trains facilement,vu que c'est le nombre de trains est déclaré dans un tableau de type pthread\_t .

Pour les trajets exécutés par chaque train ,on les a représentés par des fonctions suivantes :

-train 1 « ELBA » exécute cette fonction pour parcourir son trajet :

void\* train\_de\_A\_B\_C\_B\_A(void\* arg)

-train 2 « ELAO » exécute cette fonction pour parcourir son trajet :

void\* train\_de\_A\_B\_D\_C\_B\_A(void\* arg)

-train 3 « DEBA » exécute cette fonction pour parcourir son trajet :

void\* train\_de\_A\_B\_D\_C\_E\_A(void\* arg)

**1.2. Les Stations :** dans notre cas ,on a 5 stations (villes),on les a défini par des caractères A, B ,C,D,E qui représentent des entier de 0 à 4 afin de faciliter leurs traitements dans la programmation des fonctions.

**1.3. Les voies ferrées:**(considérées comme des sections critiques dans la modélisation du problème). Pour un meilleur affichage on a défini une fonction « void voie(int ville\_depart, int ville\_arrivee) » qui prend comme paramètres les deux villes d'extrémités ,pour former un segment de voie et l'afficher sous le format : « AB » « BC » ,« CB » ,« BA » ,« CE » ,« BD » ,« DC » ,« EA ».

**1.4. Les déplacements :** pour assurer les déplacements des trains dans les deux sens sur les voies qui sont à double sens, et aussi le cas d'un sens unique ,on a défini deux fonctions :

#### **1.4.1. void deplacement\_train\_voieAB(int train,int ville\_depart, int ville\_arrivee) :**

Cette fonction prend en entrée 3 paramètres :

-**int train:** il s'agit du numéro du train qui est représenté le numéro du thread ,mais qu'on affiche sous un nom particulier pour chaque train.

- **int ville\_depart** : l'entier définissant la ville de départ .

- **int ville\_arrivee** : l'entier définissant la ville d'arrivée.

Cette fonction autorise le passages de tous les trains dans un sens particulier,de ville de départ à ville d'arrivée ,par exemple le passage des train de A à B ,d'une manière synchroniser telle que plusieurs trains peuvent se suivre sur le même segment du moment que c'est dans le même sens ,tout en assurant une synchronisation FIFO ,premier train qui demande l'acquisition de la voie l'aura en premiers .ainsi en vérifiant le sens contraire de la voie si elle est double sens ,en faisant appel à la fonction deplacement\_train\_voieBA,qui a le même principe qu'elle mais qui assure le passage de trains dans le sens contraire.

#### **1.4.2. void deplacement\_train\_voieBA(int train,int ville\_depart, int ville\_arrivee) :**

Cette fonction a le même principe que la précédente mais qu'on utilise pour autoriser le passage de trains dans le sens opposé.

### **2. Modèle de synchronisation:**

Pour répondre aux contraintes posées dans ce problème de gestion de trains sur les voies ferrées ,on s'est inspirés de la philosophie de l'égalité entre lecteur/écrivain ,mais en la modifiant en double lecteurs (lecteur/lecteur) où le lecteur1 sont les trains partants dans un sens (AB) et le lecteur2 sont les trains partants dans le sens opposé (BA), afin de répondre à quelques exigences imposées :

1 - Plusieurs trains peuvent être présents,se suivre, sur la même voie.

2 - Le premier train qui demande la voie ,doit l'acquérir en premier dans les deux sens.

3 - Si un train s'engage sur la voie AB on doit bloquer l'accès des trains dans le sens BA.

### 3. Manuel d'utilisation :

Pour la compilation et l'exécution du programme, nous avons deux méthodes de compilation et d'exécution :

**3.1. En utilisant le Makfile :** C'est la plus fiable et recommandée, avec l'utilisation d'un Makfile, qui contient l'ensemble des commandes nécessaires pour l'exécution et la compilation de chacune des parties :

Partie1 :      Compilation : make compiler1  
                  Exécution: make executer1

Partie2 :      Compilation : make compiler2  
                  Exécution: make executer2

Partie3 :      Compilation : make compiler3  
                  Exécution: make executer3

**3.2. En utilisant un script shell :** qui lance la compilation et l'exécution à la fois avec la commande suivante :

bash script.sh

**4. Banc de tests :** pour la réalisation des bancs de tests, nous avons créé un fichier texte pour sauvegarder le temps d'exécution pour chaque train sur l'ensemble de ses trajets, pour les trois parties ( mutex, Sémaphore et Verrou). Pour ce faire, nous avons utilisé un **Timer t1** qui démarre avec le lancement de chacun des trains (threads), et un autre **Timer t2** à l'arrêt du train, On calcule la différence entre ces deux temps (**t2-t1**) pour trouver le temps total en second que le train a pris depuis le début de service jusqu'à la fin.

```

Début de service du Train ELBA : Mon Jan 8 13:16:45 2018

Le train ELBA veut s'engager sur la voie AB
Le train ELBA est dans la voie AB

Début de service du Train ELAO : Mon Jan 8 13:16:45 2018

Le train ELAO veut s'engager sur la voie AB
Le train ELAO est dans la voie AB

Début de service du Train DEBA Mon Jan 8 13:16:45 2018

Le train DEBA veut s'engager sur la voie AB
Le train DEBA est dans la voie AB
Le train ELBA quitte la voie AB
Le train ELBA veut s'engager sur la voie BC
Le train ELBA est dans la voie BC
Le train ELBA quitte la voie BC
Le train ELBA veut s'engager sur la voie CB
Le train ELBA est dans la voie CB
Le train ELAO quitte la voie AB
Le train ELAO veut s'engager sur la voie BD
Le train ELAO est dans la voie BD
Le train DEBA quitte la voie AB
Le train DEBA veut s'engager sur la voie BD
Le train DEBA est dans la voie BD
Le train ELBA quitte la voie CB
Le train ELBA veut s'engager sur la voie BA
Le train ELBA est dans la voie BA
Le train ELAO quitte la voie BD
Le train ELAO veut s'engager sur la voie DC

```

**Figure 1. Capture d'écran pour le début d'exécution**

```

Le train ELAO est dans la voie BD
Le train DEBA quitte la voie DC
Le train DEBA veut s'engager sur la voie CE
Le train DEBA est dans la voie CE
Le train ELAO quitte la voie BD
Le train ELAO veut s'engager sur la voie DC
Le train ELAO est dans la voie DC
Le train DEBA quitte la voie CE
Le train DEBA veut s'engager sur la voie EA
Le train DEBA est dans la voie EA
Le train ELAO quitte la voie DC
Le train ELAO veut s'engager sur la voie CB
Le train ELAO est dans la voie CB
Le train ELAO quitte la voie CB
Le train ELAO veut s'engager sur la voie BA
Le train ELAO est dans la voie BA
Le train DEBA quitte la voie EA

Le temps consommé (s) par le train DEBA sur l'ensemble de ses trajets est : 0.007159 s

Fin de service du Train DEBA : Mon Jan 8 13:16:45 2018

Le train ELAO quitte la voie BA

Le temps consommé (s) par le train ELAO sur l'ensemble de ses trajets est : 0.007666 s

Fin de service du Train ELAO : Mon Jan 8 13:16:45 2018

```

**Figure 2. Capture d'écran pour la fin d'exécution**

**Remarque :** pour le devancement des trains, nous avons pensé à l'utilisation des variables conditions tel que, on met un wait(condition) pour les trains présents sur une voie jusqu'à ce que le premier train qui a emprunté la voie arrive à la ville d'arriver et fait un signal pour le train qui le précède pour le réveiller et ainsi de suite,

## Conclusion :

pour conclure, ce projet nous a permis de mener un travail de groupe, on a commencé par discuter ensemble la modélisation du problème jusqu'à l'obtention d'une conception adéquate, puis nous avons entamé la partie programmation dont on a travaillé à deux sur chacune des trois parties. Cela nous a permis de nous familiariser davantage avec la programmation système en général et le langage C en particulier.

## Annexe

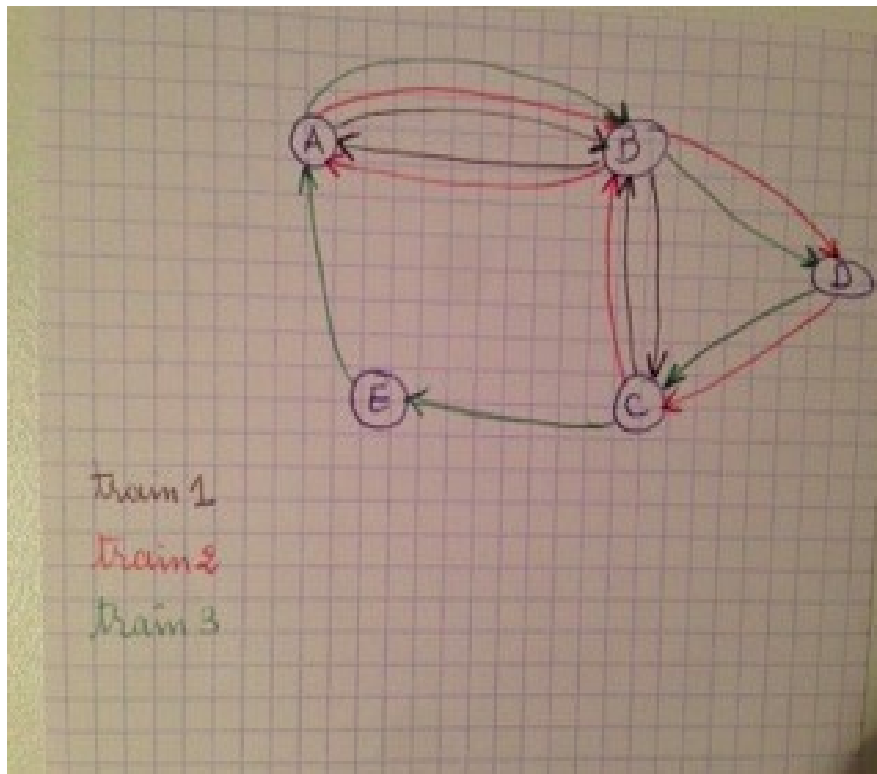


Figure 3. Schéma du réseau ferré modélisé.