

TP du Chemin de Fer

Buts du TP :

- avoir un aperçu de la synchronisation de processus par moniteur
- manipuler les variables conditionnelles de la norme POSIX

Cadre

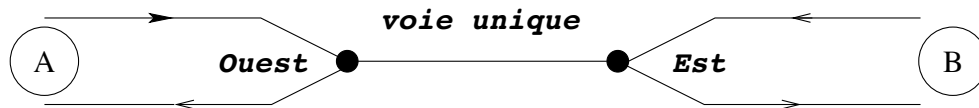


FIGURE 1 – Ligne de chemin de fer à voie unique

Une ligne de chemin de fer reliant deux villes A et B comporte une section à voie unique comme dans la figure FIG. 1. On représente les trains par des processus légers, dont l'algorithme général est décrit ci-après :

<code>/* Processus train A vers B */</code>	<code>/* Processus train B vers A */</code>
<code>moniteur_voie_unique_entree_ouest()</code>	<code>moniteur_voie_unique_entree_est()</code>
<code>...</code>	<code>...</code>
<code><trajet voie unique></code>	<code><trajet voie unique></code>
<code>...</code>	<code>...</code>
<code>moniteur_voie_unique_sortie_est()</code>	<code>moniteur_voie_unique_sortie_ouest()</code>

Le but du TP est de définir un moniteur `voie_unique` qui garantit que tous les trains engagés à un instant donné sur la voie unique circulent dans le même sens.

Téléchargement

- 1) Récupérez tout d'abord l'archive du TP à partir :
 - de ma page *Enseignements* à
<http://www-lium.univ-lemans.fr/~jacob/enseignement.html>
 - du serveur de l'IUP à
[/info/tmp/AnnexesTPM1_177UD002/TP_Chemin_Fer/TP_Chemin_Fer.tar.gz](http://info/tmp/AnnexesTPM1_177UD002/TP_Chemin_Fer/TP_Chemin_Fer.tar.gz)
- 2) Compilez le par `make all`
- 3) Testez éventuellement la distribution par `make tests`

Question 1

Écrire le moniteur `voie_unique` en complétant les fonctions du fichier `moniteur_voie_unique.c`

Testez votre moniteur avec le programme `trafic_1` où les trains circulent sur une ligne avec une seule section à voie unique.

Vous pouvez procéder en deux temps :

Version 1 : considérez qu'un nombre illimité de trains puissent utiliser la voie unique (il suffit de ne pas utiliser le 2^{ieme} paramètre de `trafic_1`)

Version 2 : l'utilisation de la voie unique doit être limitée à un nombre maximum de trains (prise en compte du deuxième paramètre de `trafic_1`)

Remarques :

- l'affichage d'une ligne se fait par la fonction `ligne_print` dans le fichier `ligne_stdio.c`; dans le code fourni, cette fonction n'est pas ré-entrante (ou multi-thread).
- la fonction `ligne_mapper` n'est utilisée qu'à des fins de "debuggage".

Question 2

Testez ensuite votre moniteur avec le programme `trafic_2` où la ligne est composée de plusieurs sections à voies uniques.

Question 3

Testez enfin votre moniteur avec le programme `trafic_3` où l'affichage de la ligne s'effectue sur l'écran avec la bibliothèque `ncurses`. L'intérêt de ce nouveau test est dans la gestion d'une nouvelle ressource critique : l'écran.

Remarques :

- dans le code fourni, l'écran n'est pas programmé pour être utilisé en exclusion mutuelle. Pour corriger cela, on peut transformer sa définition (fichier `ecran.c`) en moniteur.
- l'affichage d'une ligne sur l'écran se fait par la fonction `ligne_wprint` dans le fichier `ligne_ncurses.c`. Dans le code fourni, cette fonction n'est pas ré-entrante (ou multi-thread).

1 Contenu du rapport

Le rapport de ce TP devra comprendre :

- les codes sources du moniteur `moniteur_voie_unique.[ch]`
- les fichiers contenant les fonctions d’affichage de la ligne : `ligne_stdio.c` et `ligne_ncurses.c`
- ainsi que tous les fichiers que vous avez modifiés. Justifiez alors dans les commentaires vos modifications.