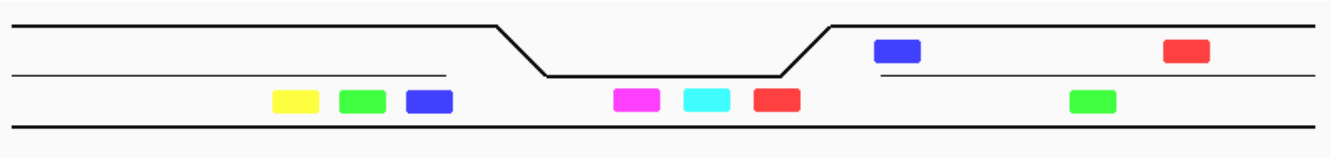
On the road once again.



Le répertoire de ce TP dans votre **dépôt GIT** s'appellera **"TPSynchronisations".**

Comme le nom du TP l'indique, vous allez utiliser les verrous (**mutex**) et les sémaphores pour synchroniser des threads en charge de faire avancer des voitures. Le but est que les voitures se croisent sans collision.

Attendu

Vous avez **deux séances** pour ce TP.

Fichiers attendus:

- `OS_Nom1_Nom2/`
  - `TPSynchronisations/`
    - `.gitignore` (avec tout ce qui n'est pas attendu)
    - `Makefile`
    - `stillOnTheRoadQ1.c`
    - `stillOnTheRoadQ1.1.c`
    - `stillOnTheRoadQ2.c`

Mise en place - Compilation

Compilation

Comme pour le TP précédent, la bibliothèque `libroad.so` et un programme principal fonctionnel vous sont fournis dans cette archive.

Ce code est aussi disponible dans le répertoire :
`/users/dut/info/Public/Systeme/stillOnTheRoad`
vous pouvez donc laisser la bibliothèque et `road.h` à cet endroit et ajuster les commandes de compilation en conséquence. Vous devez en revanche récupérer le fichier `stillOnTheRoad.c`.

Les **flags** :

- `-pthread`
- `-lallegro` et

- `-lallegro_primitives`

sont toujours nécessaires.

Ordinateur personnel

Pour ceux qui voudraient travailler sur un ordinateur personnel, la bibliothèque Allegro 5 s'installe normalement facilement. Sous debian, la commande suivante permet d'installer tous les paquets nécessaires:

```
apt-get install liballegro5-dev
```

À distance par ssh

Pour ceux qui voudraient travailler à distance via connexion ssh à gigondas, voir la section [à-distance].

Exécution

Pour rappel, comme la bibliothèque **libroad.so** est dans un répertoire non standard, vous devrez **ajouter** à votre variable d'environnement **"LD_LIBRARY_PATH"** le répertoire où elle se trouve pour que bash (en fait `ld-linux`) soit en mesure de lancer votre programme. La documentation sur cette variable d'environnement se trouve dans le man de "ld-linux". N'hésitez pas à ajouter la commande nécessaire dans votre fichier ".bashrc" afin de ne pas avoir à l'exécuter dans chaque nouveau terminal.

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/chemin/vers/le/bon/repertoire/"
```

Un programme principal fonctionnel vous est fourni dans `stillOnTheRoad.c`.

Le programme fourni crée des voitures dans les deux sens et les fait avancer. Vous remarquerez que les collisions entre voitures sont signalées en rouge.

1. Passage d'une seule voiture à la fois

La première étape va consister à faire en sorte qu'une seule voiture ne passe dans la voie réduite à la fois. Pour cela vous allez utiliser un **mutex**.

1.1. Création d'un mutex partagé

Vous passerez à un fichier ***stillOnTheRoadQ1.1.c*** pour cette question.

Lorsque l'on utilise les threads POSIX, un mutex est de type `pthread_mutex_t` et on l'initialise à l'aide de la macro `PTHREAD_MUTEX_INITIALIZER`. Ce qui donne :

```
pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER;
```

En utilisant un mutex, la fonction `road_distToCross` et la consante `road_minDist`, faites en sorte qu'une voiture ne tente de passer le croisement que si elle parvient à obtenir le mutex.

> *Attention : un thread qui tente de faire un "lock" sur un mutex qu'il a déjà "locké" sera bloqué indéfiniment.*

Il vous faudra donc prendre soin de :

- ne prendre le mutex qu'une fois par passage de la voiture et
- bien penser à libérer le mutex en sortie de croisement.

Les voitures ne doivent plus entrer en collision dans le croisement. En revanche, pour cette question, il est normal qu'elles entrent en collision dans la file d'attente.

1.2. Éviter les collisions dans la file

> Vous passerez à un fichier ***stillOnTheRoadQ1.2.c*** pour cette question.

Utilisez la fonction `road_distNextCar` pour faire en sorte que les voitures n'avancent pas si elles sont trop près de la voiture qui les précède.

2. Passage de plusieurs voitures dans le même sens

> Vous passerez à un fichier ***stillOnTheRoadQ2.c*** pour cette question.

Afin de permettre à plusieurs voitures d'être sur la voie rétrécie simultanément, vous allez utiliser deux sémaphores, un par voie. Chacun de ces sémaphores peut être vu comme un feu de circulation.

Pour simplifier la synchronisation de ces feux, celle-ci sera effectuée par un thread à part qui s'occupera de bloquer un des sémaphores de voie, puis l'autre à intervalle régulier.

Vous aurez besoin des fonctions vues en cours :

- `sem_init` pour initialiser les variables partagées de type `sem_t` qui seront les sémaphores de voie;
- `sem_wait` et
- `sem_post`.

Il est suggéré de faire une fonction `switchLanes` qui se charge de bloquer le sémaphore de la voie qui est libre puis de libérer le sémaphore de celle qui bloquée. Cette fonction sera appelée à intervalle régulier dans un thread autonome.

Remarque: vous pouvez modifier les paramètres de génération de voitures pour obtenir des résultats plus intéressants.

2.1. S'il vous reste du temps

S'il vous reste du temps, faites en sorte que le changement de voie se fasse sur réception d'un signal et non plus de manière automatique. Vous utiliserez SIGALRM et/ou SIGINT.

La réponse éventuelle à cette question se fera dans un nouveau fichier `stillOnTheRoadQ4.c`.

Vous avez deux séances pour réaliser ce TP.

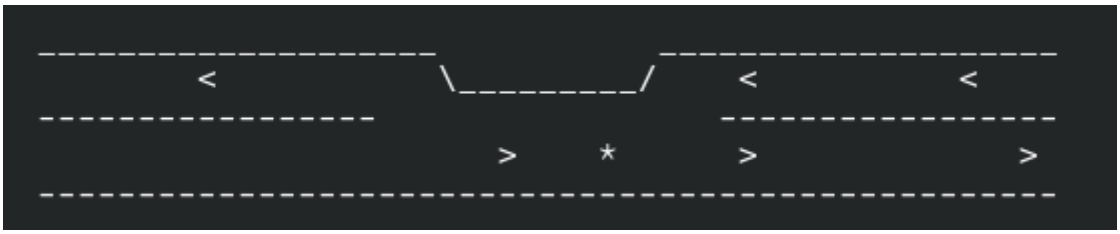
À distance

Si vous utilisez un terminal connecté en ssh sur gigondas et que vous n'avez pas d'environnement graphique sous Linux, vous pouvez utiliser une version "ascii-art" de la route en utilisant la base de travail contenue dans cette archive que vous trouverez également dans le répertoire `/users/dut/info/Public/Systeme/`

Vous aurez éventuellement besoin d'installer la bibliothèque NCurses:

```
$ sudo apt-get install ncurses-dev
```

Cette version devrait fonctionner dans un terminal et donner un résultat similaire à cela :



Pour debuguer

Vous pouvez utiliser la fonction `mvprintw` (Un exemple est inclus dans la création des voitures)

NCurses n'étant pas **thread safe**, vous verrez des artefacts dans l'affichage. Vous pouvez réinitialiser l'affichage avec la fonction `clear`.

Il peut être nécessaire d'exécuter la commande `reset` dans le terminal après avoir quitté l'affichage de NCurses si le terminal ne se comporte pas correctement.

Enjoy!