

Systèmes et réseaux informatiques – Projet de C

But pédagogique :

- Le but de ce projet est de vous familiariser avec le langage C, la programmation concurrente ainsi qu'à la production d'un programme communiquant. **Le projet comporte deux parties : une version sans communication réseau (20 points) et une version avec communication réseau (20 points). La note de chaque partie est considéré comme note projet aux ECs 'Programmation système et langage C' et 'Réseaux informatiques'.**
- Temps de travail estimé: 20h par étudiant
- **Le projet est individuel**

1 Description du projet

Une entreprise de livraison souhaite utiliser une flotte de drones pour réaliser ses livraisons à domicile. La carte de la ville est représenté par une forme rectangulaire et elle contient des obstacles. Pour déplacer les drones d'un point à un autre sur cette carte, les ingénieurs de l'entreprise planifient un chemin à suivre par chaque drone. Chaque jour, les ingénieurs envoient aux drones des chemins codifiés pour effectuer leurs livraisons.

La position actuelle d'un drone est représentée par des coordonnées x et y de la carte ainsi qu'une lettre pour indiquer son orientation. La position 0 0 N indique que le drone est situé en haut à gauche du carte et orienté vers le nord. Nous considérons qu'un drone est orienté vers le nord lorsqu'il se déplace sur plateau de (x,y) à $(x,y+1)$.

Le chemin envoyé pour chaque drone est codifié sous la forme d'une chaîne de caractères avec chaque caractère représente le mouvement à effectuer par le drone. Les caractères utilisés sont les suivants :

- L : tourner de 90° sur place à gauche.
- R : tourner de 90° sur place à droite.
- M : avancer d'une case dans la même direction.
- B : reculer d'une case dans la même direction.

Par exemple suite à la réception de la chaîne LMLM, un drone positionné initialement à 2 5 N, effectue les manœuvres suivantes : tourner à gauche, avancer d'une case, tourner à gauche, avancer d'une case. La position finale du drone est 3 4 S. Le drone ne traverse pas les obstacles et ne les contourne pas. Il exécute seulement la chaîne des commandes. Si la case suivante vers laquelle il doit avancer (commande M) ou reculer (commande B) est occupée par un obstacle, alors il reste sur place et il continue à exécuter les commandes suivantes, jusqu'à ce qu'une commande lui permet de se déplacer vers une case libre. Le drone se déplacer seulement vers les cases sans obstacles. En revanche à la sortie de la carte, le drone continue son chemin dans la direction opposée.

Le but de ce projet est de programmer en C un simulateur des déplacements d'un ensemble de drones sur un plateau avec des obstacles. Pour réaliser ce projet, vous allez développer 3 versions du simulateur de vol :

- Version séquentielle (10 points) : dans cette version les drones se déplacent vers leurs destinations l'un après l'autre sans aucune synchronisation. Les positions des drones et leurs commandes sont fournis dans un fichier texte.
- Version concurrente (10 points) : dans cette nouvelle version les drones se déplacent vers leurs destinations d'une façon concurrente. Les positions des drones et leurs commandes sont fournis dans un fichier texte.
- Version communicante (20 points) : les drones se déplacent d'une façon concurrente et ils reçoivent leurs position initiales et les commandes (chaîne de caractères) depuis le réseau via un programme client (station de commande). Dans cette version, vous avez besoin également

Systèmes et réseaux informatiques – Projet de C

de développer le programme client qui envoie les informations au programme serveur (simulateur).

Les 3 versions sont programmées dans le même programme simulateur et à son lancement, l'utilisateur choisi le mode à exécuter : basique (version 1) , concurrente (version 2), serveur (version 3). Vous pouvez programmer un menu pour le choix du mode.

Les deux premières versions utilisent un fichier texte qui contient dans sa première ligne la hauteur et la largeur de la carte. Les lignes suivantes contiennent les coordonnées initiales de chaque drone et sa chaîne de mouvements. Un exemple de ce fichier est le suivant pour 2 drones et un plateau de taille (20,20):

20 20
0 5 N LMLMLMM
5 0 S MMRMMRMRM

Pour la version 3, ces informations sont communiquées par le programme client au simulateur.

2 Partie 1 : version basique du simulateur (10 points)

Dans cette version il est attendu de programmer les fonctionnalités suivantes :

- Lecture du fichier contenant taille du plateau, position initiale et les commandes de chaque drone (autant de lignes que de drones). (3 points)
- Création et affichage de la carte avec les drones et des obstacles générés aléatoirement. Les obstacles sont désignés par le caractère #. Vous affichez l'orientation actuelle du drone avec les caractères suivants : > (Est), < (Ouest), ^ (Sud), v (Nord). (2 points). Il faut éviter de placer un obstacle sur la position initiale du drone.
- Déplacement des drones séquentiellement (chaque drone termine ses commandes, ensuite on traite le suivant) et affichage en continu de leurs positions suite à l'exécution de chaque commande. (4 points)
- Affichage des positions finales de chaque drone. (1 point)

[illegible]

3 Partie 1 : version concurrente du simulateur (10 points)

Dans cette version, vous garder les mêmes fonctionnalités que la version précédente, mais les drones exécutent leurs commandes en concurrence (simultanément). Les mouvements de chaque drone s'exécutent dans un thread séparé. Un thread est également dédié à l'affichage en continu de la carte avec les drones et les obstacles (une seule carte est affichée, vous pouvez utiliser la fonction `clearScreen` du TP). A chaque fois qu'un drone termine l'exécution d'un mouvement, il se synchronise avec le thread

Systèmes et réseaux informatiques – Projet de C

de l’affichage pour le mettre à jour. Indication : vous pouvez utiliser `pthread_cond_signal` et `pthread_cond_wait` pour synchroniser les threads drone et le thread d’affichage.

Le barème de cette version est le suivant :

- Threads d’exécution des commandes des drones avec un thread par drone : 5 points
- Thread d’affichage **en continu** des positions et des orientations actuelles des drones : 5 points

4 Partie 2 : version communicante du simulateur : (20 points)

Cette version utilise les mêmes fonctions que la version précédente (version concurrente), mais la taille de la carte, les positions initiales les commandes de chaque drone sont fournis par un programme client en réseau avec le simulateur. Vous utilisez les sockets pour faire la communication entre le simulateur et le client. A début, on lance le simulateur avec l’option serveur et il reste en attente des informations, ensuite on lance le client qui communique au simulateur les informations nécessaires. Ce dernier , à la réception des informations, il affiche et il exécute les mouvements du drone. Enfin, il communique au client les positions et les orientations finales des drones.

Le barème de cette version est le suivant :

- Programmation d’un client pour envoyer des informations (taille carte, positions et orientations des drones) au serveur : 10 points
- Réception des information par le serveur et déroulement de la simulation : 5 points
- Envoie des positions et orientations des drones par le serveur, réception par le client de ces informations et affichage en continu sur la carte des déplacements des drones et de leurs positions finales : 5 points

Bon courage à tous !