

Projets SR70

Projet 1 : Distributeur de tickets de spectacles :

On veut construire un distributeur de tickets de spectacles. Pour chaque spectacle, on a une file de messages qui contient la liste des places disponibles restantes. On suppose qu'on a plusieurs catégories de places. Si la catégorie demandée par l'utilisateur est encore disponible, on lui fournit la place demandée. Sinon, on propose la catégorie la moins chère parmi celles qui sont plus chères que la catégorie demandée. S'il n'y en a pas non plus, on propose la catégorie la plus chère parmi celles qui sont moins chères que la catégorie demandée.

Projet 2 : Serveur de licences

Il s'agit de simuler le fonctionnement d'un serveur de licences logicielles selon un mode client/serveur. Un serveur met à disposition de ses clients potentiels un nombre limité de licences flottantes (les jetons) tel que $Nb_Licence < Nb_ClientTotal$. Ce mode de gestion est restrictif puisqu'il est probable qu'un client soit en attente d'un jeton pour pouvoir exécuter le programme associé. Plusieurs clients peuvent être en concurrence pour accéder aux jetons. D'autre part pour faciliter l'accès au jeton, le serveur garde momentanément un historique puisqu'il donne la priorité aux derniers utilisateurs. Chaque numéro de licence est unique. En cas de redistribution du jeton à un client, le serveur propose par défaut le numéro de licence initial. Le serveur est garant du temps d'utilisation du logiciel. Au-delà d'un temps limite, le serveur interroge le client pour vérifier son utilisation.

Projet 3 : Le cinéma

L'accès aux salles de cinéma est souvent problématique. Bien souvent les files aux guichets sont trop longues et la gestion du cinéma peu efficace. Pour résoudre ce problème, nous vous proposons d'implanter une solution selon les règles de fonctionnement suivante :

- Un client doit être en mesure de connaître en temps réel le nombre de place disponible dans la salle ;
- Les tickets peuvent être achetés auprès d'une hôtesse ou à un guichet automatique
- Les places sont numérotées ;
- Au-delà d'une certaine limite, les salles peuvent modifier leur projection en fonction du nombre de clients ;
- Un client peut échanger son billet auprès d'une hôtesse, celle-ci devra lui proposer un autre film ou une projection ultérieure ;
- Certains films sont soumis à autorisation. En cas de doute, l'hôtesse peut interdire l'accès à une salle et donc échanger le billet ;
- Les personnes munies de réservations sont prioritaires ;

Sujet 4 : Agenda

On désire gérer un agenda pour la planification de réunions dans un service constitué de 6 équipes. Les effectifs des équipes varient de 5 à 10 personnes. Les membres d'une équipe seront considérés comme des processus, ayant des disponibilités aléatoires et étant joignable en permanence par un moyen de communication que vous définirez. Le service dispose de 2 petites salles de réunion : l'une contient 8 places et l'autre 15 places. L'idée d'utilisation prioritaire est acceptée (urgence, direction, ...), il faudra alors éventuellement replanifier une réunion.

Vous lancerez des planifications aléatoire ou paramétrées de réunion au sein de chaque équipe, tenant compte des disponibilités de chaque personne et des salles de réunion.

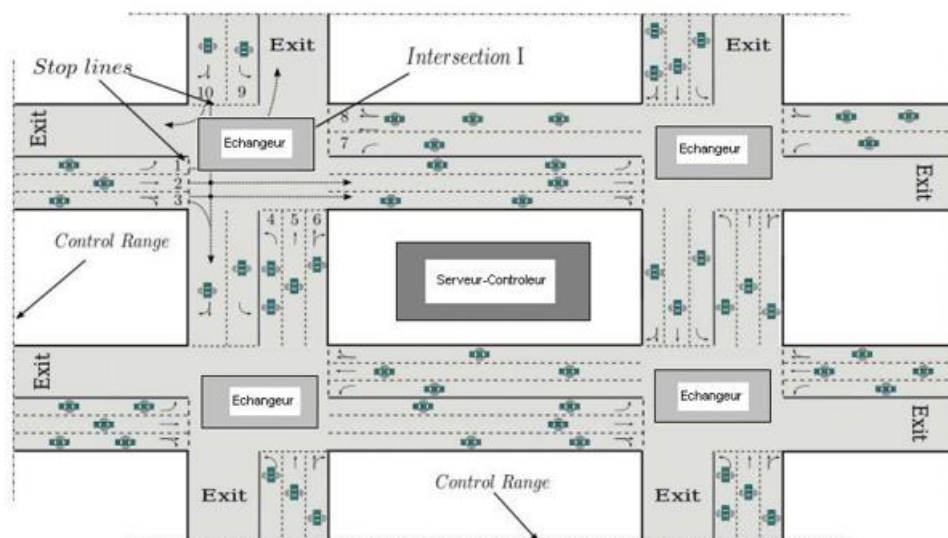
Dans la démonstration, on devra pouvoir de façon interactive programmer une réunion prioritaire et suivre l'ordonnancement de l'ensemble des réunions.

Sujet 5 : Gestion d'un multi-carrefours

On veut simuler un ensemble de 4 carrefours (voir figure ci-dessous). Tous les carrefours sont dépourvus de feux de signalisation. On suppose que tous les véhicules sont équipés de dispositifs embarqués leur permettant la communication avec l'infrastructure routière.

A chaque carrefour, on dispose d'un échangeur communiquant avec les véhicules arrivant au carrefour. Cet échangeur permet de donner le droit de passage aux véhicules souhaitant se déplacer d'un endroit vers un autre.

Exemple, un véhicule arrivant dans un carrefour transmet à l'échangeur ses données (type de véhicule, vitesse,) ensuite lui envoie une requête indiquant son itinéraire. L'échangeur identifie le véhicule et envoie ensuite une requête au serveur-contrôleur. Ce dernier analyse l'état du trafic sur l'ensemble des 4 carrefours (nombre total des véhicules à chaque carrefour, type des véhicules...) et envoie une réponse à l'échangeur qui à son tour la transmet au véhicule demandeur permettant ainsi un chemin optimal. Certains véhicules sont prioritaires (Ambulances, pompiers, police...) et doivent donc emprunter le chemin le plus court quel que soit l'engorgement du trafic



Projet 6 : Gestion d'un ascenseur

Soit un immeuble de bureaux de N étages (prendre $N = 10$ par exemple). Les étages sont accessibles par un ascenseur. On simulera un processus pour ce dernier en fonction d'un scénario de fréquentation des usagers de l'immeuble, à définir. Réserver au moins $1/3$ de passages pour le niveau 0 (entrée et sortie de l'immeuble).

L'immeuble est géré ensuite par 2 ascenseurs. Proposer un algorithme qui gère les parcours de ceux-ci en fonction des demandes, en optimisant les trajets. Comparer le nombre de trajets effectués dans les 2 cas pour votre scénario.