

## Intitulé : Collecte des déchets urbains

### Présentation

Le but de ce TP est la mise en application des concepts liés à la synchronisation et la communication des processus à travers l'utilisation d'outils offerts par le système d'exploitation UNIX. On se propose d'implémenter un système de gestion de la collecte de déchets urbains selon une méthode qui sera décrite par la suite.

Il s'agit de la planification des tournées des camions de transport des déchets urbains comme suit :

- Chaque camion a pour rôle, entre autres, d'attendre la réception de l'affectation pour se déplacer à l'adresse de la poubelle concernée. Il se charge de son contenu et se déplace vers la décharge afin de vider cette charge pour, par la suite, se rendre disponible pour une prochaine chose à faire qui peut être en général une de trois possibilités : soit prendre une charge, soit aller à la centrale pour se reposer, soit aller s'approvisionner du carburant. A la fin de chacune de ces opérations, un compte rendu est communiqué au contrôleur.
- Le contrôleur affecte les missions de décharges, les autorisations de repos et l'approvisionnement en carburant selon la consommation du carburant.

### Modélisation

Les éléments qui interviennent dans le problème sont :

- Le contrôleur, implémenté par un processus *controleur* cyclique, reçoit des avis de fin d'opérations (fins de décharge, retour de repos, fin d'approvisionnements au carburant) des camions dans un tampon nommé *Tfmissions* de taille *R*. Ce processus décide de l'opération à affecter à un camion disponible selon la méthode indiquée ci-dessous selon l'état actuel du système en l'envoyant dans une file de messages commune *Faffect*.
- Les camions ont les mêmes capacités de charge et les mêmes capacités de réservoir de carburant (*CP*) et sont implémentés par *N* processus cycliques *camion[i]*,  $i=0, \dots, N-1$ . Chaque processus a pour rôle de simuler le comportement d'un camion réel. Lorsqu'un processus *camion [i]* reçoit un ordre pour une opération donnée dans la file de messages *Faffect*, il simule la réalisation du service et communique la fin de l'opération au contrôleur en lui remontant la consommation éventuelle en carburant. On admet que la capacité d'un camion lui permet de vider deux poubelles dans chaque mission.
- Le nombre de poubelles est *M* ( $\geq N$ ) et chaque poubelle *j* est définie par un état (connu du contrôleur) *etatp[j]* qui peut être : *vide*, *pleine* ou *affectée*. *etatp[j]* passe de *vide* à *pleine* selon une fonction aléatoire.
- La distance du trajet entre chaque poubelle *j* et la décharge est *dist\_decharge [j]*. La distance entre deux poubelle *j* et *k* est *dist\_poubelles [j, k]*. Toutes ces distances sont connues par chaque camion et sont fixées au départ. On admet que la décharge, le parc et la station de carburant sont situés dans le même endroit.
- Le nombre de missions d'un camion *i* entre deux pauses ne doit pas dépasser en consommation du carburant  $\frac{1}{3}$  de sa capacité. Le seuil minimal du réservoir avant approvisionnement en carburant est *Min*. La consommation du carburant est proportionnelle à la distance parcourue (, elle est donnée par  $C \cdot \text{distance parcourue}$ ). Après *Max* missions de décharge, le camion signale au coordinateur son retrait du travail.

### Travail demandé

Il s'agit de synchroniser tous ces processus à l'aide de sémaphores dans le but de réaliser la fonction décrite ci-dessus. Le système doit comprendre une interface (textuelle seulement, sous forme d'impressions) permettant de suivre et de comprendre le fonctionnement des différentes entités du service objet de ce TP. On prendra  $M=20$  ;  $N=5$ ,  $R=3$ ,  $CP=300$ ,  $\text{dist\_decharge [j]} \leq 20$ ,  $\text{dist\_poubelles [j, k]} \leq 20$ ,  $\text{Min}=70$ ,  $C=2$ ,  $\text{Max}=12$ .

### Connaissances requises sous Unix

- Création de processus
- Partage de mémoire
- Synchronisation par sémaphores
- Communication par files de messages

### Echéance

03 Décembre 2024.