

INFO 038, « *Architecture d'un système d'exploitation* », R. DEVILLERS
Examen de seconde session, partie écrite**Thème : grille de machines**

Nous vous demandons de modéliser le logiciel nécessaire à la gestion d'une grille définie de la façon suivante.

Une grille est un ensemble d'ordinateurs reliés entre eux par un réseau selon une certaine topologie. Nous supposons que cette topologie est donnée par un graphe connexe non-orienté, dont chaque sommet a au plus 256 voisins. Les machines peuvent communiquer entre elles mais, pour des raisons de coût et de performance, uniquement entre nœuds adjacents.

Le but de la grille est de pouvoir répartir les calculs sur l'ensemble des machines de manière à utiliser au mieux les ressources disponibles. Les utilisateurs des machines composant la grille ont la possibilité de soumettre des *requêtes*, c.-à-d. des demandes d'exécution de processus. La première action qui devra s'exécuter lors de la réception d'une telle requête sera le *routage*, c.-à-d. l'acheminement de la requête vers la machine qui l'exécutera. Une fois une requête arrivée à sa machine de destination, elle s'exécute entièrement sur cette machine. À tout moment, une machine peut donc être en train d'exécuter zéro, une, ou plusieurs requêtes. Nous pouvons donc définir la *charge* d'une machine comme étant le nombre de requêtes s'exécutant sur celle-ci. Pour permettre une distribution équitable des requêtes, chaque machine envoie sa charge courante à chacun de ses voisins à intervalles réguliers.

Le routage se déroule de la façon suivante. L'idée de base est d'essayer d'envoyer la requête vers une machine ayant une petite charge. Lorsqu'une machine reçoit une nouvelle requête, elle vérifie si elle a une plus petite charge que celle de ses voisins (elle a une idée de la charge de ses voisins, étant donné que ceux-ci la lui envoient périodiquement). Si c'est le cas, elle accepte la requête et l'exécute, sinon elle la transmet à son voisin ayant la plus petite charge. Le voisin, une fois la requête reçue, recommence le même processus, et ainsi de suite, jusqu'à ce que la requête aboutisse finalement sur une machine qui la gardera. Nous supposons qu'une requête trouvera toujours une machine où s'exécuter (mais essayez d'imaginer une situation où ce ne sera pas le cas).

Le logiciel que nous vous demandons de modéliser, et qui devra donc s'exécuter sur chaque machine, devra pouvoir gérer :

- l'envoi d'une nouvelle requête ;
- le routage des requêtes ;
- l'exécution à proprement parler de ces requêtes ;
- l'envoi périodique des charges aux voisins.

Notes :

- Nous supposons que les requêtes contiennent au minimum deux parties : le contenu du fichier exécutable du programme, et les données d'input au programme.
- On ne vous demande pas de gérer le renvoi de l'output des processus vers les utilisateurs.
- Nous ne demandons pas que les envois périodiques des charges soient synchronisés entre les différentes machines.

Modalités de réalisation

Nous vous demandons de réaliser l'analyse correspondant à la mise en œuvre d'un tel système. Votre analyse comprendra :

1. Un diagramme des entités (sur au moins deux niveaux) ;
2. Un diagramme des processus commenté et justifié ;
3. Un diagramme des IPC commenté et justifié ;
4. Un dictionnaire complet des traitements et des structures ;
5. Un dictionnaire d'implantation des traitements pour les 5 primitives les plus significatives ;
6. Un dictionnaire d'implantation des structures de données pour les 5 primitives choisies ;
7. Un diagramme et une découpe en modules (suivant la relation *utilise*) ;
8. Les commentaires que vous jugerez utiles ou nécessaires.

Remarques très importantes : structurez votre réponse selon le schéma fourni ci-dessus et utilisez impérativement les formalismes vus durant les travaux pratiques.

Bon travail !