

1 Présentation générale

On souhaite réaliser la simulation d'un ordonnanceur en utilisant un protocole client/serveur. Cet ordonnanceur décidera du processus à élire, s'occupera de la gestion des signaux et stockera l'ensemble des informations concernant chacun des processus dont il a la charge.

2 Ordonnanceur et structure de données

L'ordonnanceur gèrera une ou plusieurs files contenant des informations pour chacun des processus. Cette structure de données pourra être assimilée aux Blocs de Contrôle de Processus (BCP). Un BCP contiendra au minimum

- identifiant et identifiant du père (propre au serveur),
- pid réel,
- nom de la commande lancée,
- tableau de gestion pour quelques signaux,
- état,
- temps CPU consommé (calculé par le serveur)
- priorité

L'ordonnanceur sera appelé avec en paramètre la politique d'ordonnement qui pourra être

- premier arrivé, premier servi (FCFS),
- tourniquet sans priorité (RR),
- tourniquet avec priorité.

Le serveur ou ordonnanceur élira et fera un changement de contexte grâce aux signaux. On utilisera une alarme pour simuler le quantum de temps. L'ordonnanceur gèrera l'envoi de signaux, mettra les processus dans la bonne file lorsqu'ils feront une Entrée/Sortie, les arrêtera et mettra à jour sa table contenant les structures de données. Lors de la terminaison de l'ordonnanceur, tous les programmes dont il a la charge devront être interrompus proprement. L'ordonnanceur devra également remplir un fichier de log dans lequel il décrira les différentes actions qu'il effectue. Le degré de granularité des actions décrites sera passé en paramètre lors du lancement de l'ordonnanceur.

3 Les outils pour la consultation et la manipulation

Vous devrez écrire également un ensemble de commandes pour la manipulation de votre ordonnanceur.

- un lanceur de programmes se comportant comme un `shell`. Plusieurs questions se posent. Ce shell sera-t-il ordonnancer par le serveur ou non ? Sera-t-il un descendant du serveur ou non ?
- un visualiseur de processus se comportant comme `ps`. Il s'agira juste de faire une consultation des tables du serveur en utilisant l'un des mécanismes de communication possible.

4 Les appels-système

Il faudra réécrire un ensemble d'appels-système dont le prototype (obligatoire) suit :

- `int mfork()` duplique le processus. Le serveur crée une entrée dans la table des processus. La fonction renvoie deux valeurs (0 dans le fils, l'identifiant relatif du fils dans le père).
- `int mexec(const char *, const char **)` qui permettra de faire un recouvrement du processus courant avec celui passé en paramètre. La ligne de commande devra être récupérée afin de renseigner le bloc de contrôle de processus.
- `int mwait(int identifiant)`. Le processus courant s'endort jusqu'à la fin du processus dont le numéro est passé en paramètre.
- `int mprintf(const char *, /* variables */)` permet de faire un affichage. Le processus s'endormira un temps aléatoire avant d'être réveillé et remis dans une file d'attente adéquate.
- `int mscanf(const char *, /*variables */)` permet de faire une lecture de ligne. Comme pour `mprintf`, le processus s'endormira un temps aléatoire avant d'être replacé dans une file d'attente.
- `int mkill(int sig, int process)` permet d'envoyer le signal `sig` (défini par le serveur) à un processus (défini par son identifiant).
- `int mgetpid()` et `int mgetppid()` qui permet de récupérer les identifiants du processus et de son père dans les tables de l'ordonnanceur.

Ces différents appels-systèmes serviront aussi bien pour les outils demandés que pour fournir quelques **jeux d'essais** en écrivant vos propres programmes.

5 Outils Unix à utiliser

Pour la liaison entre clients et serveur, vous pourrez utiliser plusieurs mécanismes comme les I.P.C. (segments de mémoire partagée, files de messages) ainsi que les signaux.

6 Commentaires sur le travail à effectuer

6.1 Spécification

Vous devrez spécifier correctement toutes vos fonctions ainsi que votre serveur. Par exemple pour la fonction `mfork()` vous devrez dire si le père continue, si c'est le fils, ou si l'ordonnanceur choisit un nouveau processus parmi les processus prêts.

Vous devrez veiller à la cohérence de votre système. Que faire si un processus ayant des fils se termine ?

6.2 Commentaires généraux

Le projet est à réaliser en langage C. Ce projet est à réaliser par binôme pour le Vendredi 27 janvier 2006. Vous devrez fournir un rapport détaillé expliquant le fonctionnement ainsi que vos choix de programmation. Une soutenance aura lieu durant le mois de février.