

Communication et synchronisation : les signaux et les files de messages

Exercice 1

- 1) Écrire un programme qui pendant un temps t donné, compte le nombre de <INTR> et <QUIT> frappés au clavier et qui affiche ces nombres avant de se terminer.
- 2) Écrire une fonction utilisant l'appel système `alarm` et ayant le même fonctionnement que la fonction `sleep`.
- 3) Un processus se duplique. Le père doit rester vivant tant que le fils est vivant. Si le fils se termine, le père doit se terminer aussitôt. Un <CTRL C> envoyé au père doit arrêter les deux processus.
- 4) Un processus crée un fils. Le père ignore les interruptions mais pas le fils. Lorsque le fils se termine, le père n'ignore plus les interruptions.
- 5) On veut créer un protocole d'échange de signaux entre deux processus. Un père et son fils s'échangent 50 signaux `SIGUSR` avant de se terminer. Le fils émet un signal vers son père et se met en attente de l'accusé de réception. Écrire le programme avec les primitives du TD. Expliquer pourquoi ce programme peut ne pas fonctionner correctement. Quel mécanisme supplémentaire doit-on posséder pour réaliser ceci ?

Exercice 2 On souhaite réaliser un serveur d'information du système. Un client émet une requête via une file de messages pour demander :

1. le nom d'un utilisateur en fonction de son **uid**
2. le nom d'un groupe en fonction de son **gid**
3. les informations sur un processus en fonction de son **pid**

Le fonctionnement pourrait être le suivant : On utilise une file de message pour la question et une pour la réponse. Le serveur est en attente sur la file question. Lorsqu'une question arrive (on utilise ici le type du message pour préciser la question), le serveur se duplique. Le père reprend l'attente alors que le fils traite la question et envoie sa réponse dans la file dédiée.

Le fils pourra utiliser les primitives **getpwuid**, **getgrgid**, et les informations récupérées dans **/proc** (voir TD ordonnancement).

Lors de la terminaison du serveur, toutes les files devront être supprimées. On captera les signaux **SIGINT** et **SIGQUIT**

- 1) Ecrire les algorithmes pour le serveur puis pour le client.
- 2) Ecrire les programmes.