

Système M1 — TD 5 : Tubes et signaux

Semaine du 19 octobre 2006

Exercice 1 – Retour sur pipeline

1. Ecrire la commande shell qui affiche le nombre de sous-répertoires d'un répertoire donné.
2. Ecrire le shell qui gère ce genre de commandes (fin du TP 3). On réutilisera la fonction `char ** parse(char *chaine, char *separateurs)` pour extraire des arguments.

Exercice 2 – Invincible

1. Ecrire un programme qui quand on tape le caractère `<intr>` sur son terminal, affiche "Aie!" et continue de s'exécuter.
2. Le modifier pour qu'il prenne un paramètre `p`, affiche autant de ! que de caractères `<intr>` frappés, et meurt quand `p` est atteint ; par exemple :

```
$ ./invincible 3
(^C)Aie!
(^C)Aie!!
(^C)Aie!!! Au revoir...
```

3. Maintenant le programme aura un processus fils chargé de négocier la terminaison. Quand le père aura reçu le signal `n` fois, le fils devra demander la confirmation de la fin à l'utilisateur (il ignorera tous les signaux sauf celui envoyé par son père). Les deux processus vont alors se terminer, ou bien recommencer.

```
Fin?(y/n)
m
Fin?(y/n)
n
(^C) Aie!
```

Exercice 3 – Typespeed

Ecrire un programme `vitesse` qui prend deux paramètres, un nom de fichier et un nombre `n`, affiche le contenu du fichier et lit un texte de la même longueur (au plus) sur l'entrée standard pendant au plus `n` secondes. Ensuite il compare les deux textes, et calcule le nombre de caractères correctement entrés pour obtenir ensuite la vitesse moyenne de frappe (en caractères par seconde). Si le caractère `<susp>` es frappé, le temps ne doit pas s'écouler jusqu'à ce que le programme reprenne son exécution.