Unix : concepts et outils avancés
TD no 8 : mini-shell (suite)

Olivier Aumage
LIP, bureau 302
email : oaumage@ens-lyon.fr

22 novembre 2000

1 Introduction

L'objectif de cette séance de TP est d'introduire le support des tubes (c'est-à-dire des commandes pipelinées) dans votre mini-shell. Pour cela, une bonne idée est de commencer par expérimenter l'utilisation des tubes sur de petits exemples simples illustrant les mécanismes que vous devrez mettre en place dans votre shell.

2 `ls | wc -w`

La première étape est de comprendre comment exécuter deux processus communiquant via un tube : écrivez un petit programme réalisant l'équivalent de la commande shell `ls | wc -w` en utilisant entre autres les primitives `fork`, `exec*` et `pipe`.

Attention : écrivez votre programme de façon à ce qu'il se termine bien (*i.e.* qu'il rende la main au shell) **après** l'affichage du résultat.

3 `ls | head | wc`

Les choses se compliquent un peu dès que l'on tente de généraliser ce schéma à plus de deux commandes. En particulier, il faut apporter un soin particulier à la manipulation des tubes : création, héritage, fermeture des descripteurs inutilisés, etc.

Modifiez le programme précédent pour réaliser la cascade de commandes suivante : `ls | head | wc`. Généralisez à un nombre arbitraire de commandes (sans argument pour simplifier) lues dans un tableau global. Exemple :

```
      char *commandes[] = {
        "ls",
        "head",
        "wc",
        NULL
      };
```

Plusieurs stratégies sont possibles en matière de création des différents tubes. Peut-on trouver une stratégie n'utilisant qu'un petit nombre fixé de variables pour manipuler les descripteurs de fichiers associés aux tubes (au lieu d'un inesthétique tableau statique) ?

4 Application au mini-shell

Il ne vous reste plus qu'à adapter le programme précédent afin de l'introduire dans votre mini-shell... Le plus difficile est fait !

— * —