

TD 1 d'Environnement Logiciel: le corrigé

Lundi 13 septembre 2004

1 Fichiers et commandes de base

Répertoires

- `ls` affiche les fichiers du répertoire courant en omettant ceux dont le nom commence par un point. Les options `a`, `i` et `l`, 3 options parmi tant d'autres de `ls` permettent respectivement d'afficher tous les fichiers, d'afficher aussi leur numéro d'*inode* et de les afficher un fichier par liste.
- Les répertoires `.` et `..` représentent respectivement le répertoire lui-même et son répertoire père.

Les fichiers

- Le déplacement par `mv` sur un même système de fichier ne modifie pas l'*inode* du fichier. Par contre une copie par `cp` duplique effectivement les données sur le disque, et donc utilise une nouvelle *inode*.

Utilisateurs, groupes

- La ligne correspondant à `root` sur mon système dans `/etc/passwd` est la suivante:
`root:x:0:0:root:/root:/bin/bash`
Cela signifie que son *uid* est 0, son shell est `/bin/bash` et son répertoire *home* est `/root`.
- Sur mon système les utilisateurs `bind`, `sshd` et `Debian-exim` ont `/bin/false` pour shell. Cela permet d'être sûr que personne ne pourra se logger sur ce compte.

Droits d'accès aux fichiers

- Les droits en écriture sur un fichier n'indique pas si on a le droit d'effacer ce fichier; il faut avoir un droit en écriture sur le répertoire contenant ce fichier pour cela (effacer un fichier, ce n'est qu'enlever un lien nom → *inode* dans le répertoire parent).
- Les droits d'un fichier sont stockés dans son *inode*, donc changer les droits d'un fichier sera visible via tous les liens physiques qui pointent vers cette *inode*.
- Un répertoire `incoming` répondant à la question doit avoir les droits `rwX-wX--T` où le T indique le caractère *sticky*: seul `root` et le propriétaire d'un fichier peuvent l'effacer.

Édition de lien, `gcc`, `ld`, `make`.

- Le point d'entrée du programme en C est traditionnellement appelé `main` (mais cela peut être changé avec l'option `-e` de `ld`). Un `grep main *.c` donnait la réponse.
- `make` est utilisé dans la compilation de projet en lançant automatiquement les commandes de compilation et de liaison utiles pour créer l'exécutable défini dans le fichier `Makefile`. En particulier à la modification d'un fichier source C les autres fichiers du projet ne sont pas recompilés, seules les étapes nécessaires sont refaites.