

TD 1 d'Environnement Logiciel

Lundi 13 septembre 2004

1 Fichiers et commandes de base

Répertoires

- Tapez les commandes `ls`, `ls -l`, `ls -a` et `ls -i`. Combinez les options avec `ls -lai`. Expliquez les différentes sorties obtenues (en cas de doute, voir `man ls`).
- Créez un sous-répertoire `Unix` dans votre répertoire `travail` avec les commandes `mkdir` et `cd`.
- Dans la sortie de `ls`, que représentent les répertoires `.` et `..` ?
- Explorez le système. Par exemple regardez ce qui se trouve à la racine `/`. Revenez à votre répertoire en tapant `cd`.

Les fichiers

- Allez dans votre répertoire `travail/Unix` et créez un fichier `test.txt` avec votre éditeur de texte favori (si vous n'avez pas d'éditeur favori, lancez `emacs...`). Tapez quelques lignes dans ce fichier.
- Visualisez le contenu du fichier avec les commandes `cat`, `more` et `less`.
- Renommez votre fichier avec la commande `mv test.txt test_2.txt`. Observez le résultat en comparant les sorties avant et après des commandes `ls` et `ls -i`.
- Créez un répertoire `Test` ainsi qu'un sous-répertoire `Test/Tst` et déplacez-y le fichier avec la commande `mv test_2.txt Test/Tst/`. Tapez `ls -l`; `ls -l Test/Tst`.
- Tapez `ls -Rlai`. Observez les numéros des *inodes* dans la colonne de gauche.
- Déplacez/Renommez le répertoire `Tst` avec la commande `mv Test/Tst Nouveau_Tst`. Utilisez `ls -Rlai` pour observer que les *inodes* n'ont pas changé.
- Dupliquez le fichier avec la commande `cp Nouveau_Tst/test_2.txt test_3.txt`. Utilisez `ls -Rlai` pour vérifier que le fichier a bien un nouveau numéro.

- Créez un lien physique sur le fichier `test_3.txt` avec la commande `ln test_3.txt phy_ln.txt`. Comparez les *inodes* de ces deux fichiers. Observez également la seconde colonne du résultat de `ls -l`.
- Créez un lien symbolique sur le fichier `test_3.txt` avec la commande `ln -s test_3.txt sym_ln.txt`. Comparez les *inodes* de ces deux fichiers. Observez également comment ce nouveau fichier apparaît au travers des commandes `ls` et `ls -l`.

Utilisateurs, groupes

- Quel est votre nom d'utilisateur (*login*) ? Cf commande `whoami`.
- En regardant le fichier `/etc/passwd` et la page de manuel associée (`man 5 passwd`) indiquez l'*uid* de `root`, son shell, son répertoire de travail (*home*).
- En regardant le fichier `/etc/group` et la page de manuel associée (`man 5 group`) indiquez le ou les groupes dont `root` fait parti.
- En observant la sortie de `ypcat passwd` dont le format est le même que celui de `/etc/passwd`, indiquez votre *uid*, votre nom réel dans le système, la localisation de votre répertoire de travail dans le système de fichier, votre shell.
- En observant la sortie de `ypcat group` dont le format est le même que celui de `/etc/group`, indiquez le ou les groupes dont vous faites parti.
- Y a-t-il des utilisateurs dont le shell est `/bin/false` ? Lancez la commande `false` et regardez sa page de manuel. Quel intérêt peut il y avoir à donner `/bin/false` comme shell de login? (voir aussi `man login`).

Droits d'accès aux fichiers

- Tapez `ls -l` dans le répertoire `travail/Unix/Test`. Observez les droits des fichiers et des répertoires. Ils sont classés dans l'ordre {utilisateur, groupe, autres}, `r` étant le droit de lecture, `w` le droit d'écriture et `x` le droit d'exécution.
- Tapez `chmod -r test_3.txt; ls -l test_3.txt; more test_3.txt`. Restaurez l'ancien état avec `chmod +r test_3.txt`.
- Dupliquez le fichier `test_3.txt`, supprimez les droits d'écriture et supprimez le fichier avec la commande `rm`. Les droits d'écritures sur un fichier ont-ils un effet sur la possibilité de supprimer ce fichier?
- Tapez `chmod -w .; ls -l`. Essayez de créer un nouveau fichier dans le répertoire `Test`. Essayez de créer un nouveau fichier dans le répertoire `Nouveau_Tst`. Restaurez l'ancien état par `chmod +w ..`
- Tapez `chmod -x Nouveau_Tst; cd Nouveau_Tst`.

- Tapez `less Nouveau_Tst/test_2.txt`. Restaurez l'ancien état par `chmod +x Nouveau_Tst`.
- Comment sont gérés les droits sur les liens physiques?
- Créez à votre racine un répertoire `incoming` où n'importe quel membre de votre groupe puisse déposer des fichiers mais sans pouvoir lire le contenu ni du répertoire, ni des fichiers déposés. Vous devez disposer de tous les droits sur ce répertoire et les utilisateurs qui ne font pas partie de votre groupe ne doivent disposer d'aucun droit.

Commande `umask`

- La commande `umask` contrôle les droits par défaut des fichiers créés. Sa syntaxe est `umask 0ooo` avec `0ooo` un nombre octal commençant par 0 et comprenant les droits `ugo` avec `read=4`, `write=2`, `execute=1`.
- Tapez `umask` pour voir votre masque courant.
- Tapez `umask 124; mkdir T; ls -l`.
- Créez et sauvegardez un nouveau fichier texte avec un éditeur de texte lancé depuis le *shell*. Observez ses droits.
- Créez et sauvegardez un nouveau fichier mais avec un éditeur texte lancé depuis un autre *shell*.
- Restaurez votre ancien `umask` (probablement `umask 022`).

Les processus

- Tapez `ps`, puis `ps ax`.
- Écrivez un fichier C qui fait une boucle infinie `for (;;) ;`.
- Interrompez le programme avec `CTRL-c`.
- Relancez le programme et interrompez le avec `CTRL-z`. Tapez `ps`. Relancez le programme par `fg` (*foreground*) puis interrompez le à nouveau par `CTRL-z`. Tapez `bg` (*background*). Maintenant le programme s'exécute en tâche de fond.
- Tuez le programme par la commande `kill -KILL numéro du processus`.
- Lancez le programme en tâche de fond en ajoutant `&` à la ligne de commande.
- Lancez la commande `top`. Appuyez sur la touche `h` pour accéder à l'aide. Tuez le processus de fond à l'aide de `top`. Quittez `top` (touche `q`).

Édition de lien, gcc, ld, make.

Exemple d'un projet complet: *ecm*.

Ecm pour *Elliptic Curve Method* est un logiciel permettant de factoriser des nombres entiers via la méthode des courbes elliptiques. C'est surtout un projet contenant plusieurs modules en C que l'on va utiliser pour illustrer la compilation séparée.

- Récupérez les sources du projet avec la commande:
`cvs -d:pserver:cvs@cvs-sop.inria.fr:/CVS/spaces co ecm`. Un répertoire `ecm` contenant le projet va être créé dans le répertoire courant.
- Combien de fichiers C composent le projet?
- Où se trouve le point d'entrée du programme?
- Tapez `make` et observez la sortie à l'écran. Avec `ls -la` avant et après l'exécution de `make`, regardez les fichiers créés par la commande. `make` est une commande très utile pour le développeur, voir <http://www.gnu.org/software/make/> pour plus d'informations.
- Tapez `make clean` et recommencez la compilation à la main (sans utiliser `make`) avec l'utilisation de `gcc` et `ld`.
- Plus fort: observez les diverses étapes de la compilation d'un fichier C de votre choix en utilisant `gcc -E`, `gcc -S`, `gas`...
- Après avoir tapé `make`, retapez `make`. Normalement, il ne se passe rien.
- Exécutez `touch median.o` puis `make`. Que se passe-t-il? Pourquoi?
- Éditez la fonction `main` du projet pour afficher un message de bienvenu de votre choix à chaque démarrage et tapez `make`. Quels étapes de la création du fichier exécutable final sont effectuées?