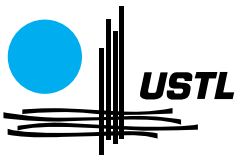

Licence d'informatique
Module de C/Unix

Travaux Pratiques

`cl` — Compilation Locale

Philippe Marquet

1 Un problème ?

Depuis l'installation de postes de travail puissants (les `pcu*`), il survient parfois quelques problèmes lors de la compilation de programmes C[1].

Un remède est de travailler localement sur la machine, c'est-à-dire sans utiliser les fichiers hébergés par le serveur. On travaille donc dans le répertoire `/tmp` plutôt que dans `/home/licence/duchmol`. Cependant, les avantages de travailler dans un répertoire situé sous `/home` sont :

1. Ces répertoires sont montés, donc visibles, depuis toutes les machines du réseau enseignement.
2. Un mécanisme de sauvegarde est assuré.

Une solution est d'organiser ainsi une session de travail :

1. Transférer temporairement un répertoire dans le `/tmp` de la machine locale.
2. Travailler dans ce répertoire.
3. Recopier le contenu de ce répertoire en fin de session.

2 Une commande !

Nous allons donc développer une commande Bourne shell `cl` qui permet de transférer des fichiers d'un répertoire situé sous `/home` vers le répertoire local de la machine `/tmp` et inversement.

La syntaxe de la commande `cl` est la suivante :

`cl` [`--install`|`--uninstall`|`--show`] [`--force`] *`répertoire`*

`répertoire` désigne le nom d'un répertoire sous `$HOME`. Par exemple, `cunix/tp3` désigne `$HOME/cunix/tp3`.

`--install` (option par défaut) installe sous `/tmp`[2] le répertoire désigné. Une trace de cette installation est gardée dans le fichier de log de `cl`. Si une installation du même répertoire a déjà eu lieu sur une autre machine, `cl` demande confirmation avant de poursuivre l'installation.

`--uninstall` transfert le répertoire désigné de `/tmp` vers `/home`. Si le répertoire n'a pas été installé sur la machine locale ou si il a été installé sur une autre machine, `cl` demande confirmation avant de continuer. Le ficher de log est mis à jour.

`--force` Lors d'une installation ou désinstallation, `cl` ne demande pas confirmation.

`--show` Liste les machines sur lesquelles le répertoire désigné est installé.

[1] Lors d'une compilation, la machine locale produit un fichier exécutable relativement volumineux que la machine serveur de fichiers ne sait pas écrire sur le disque au rythme de production. Il en découle des erreurs d'écriture dans le fichier exécutable, donc des erreurs lors de l'exécution du programme...

[2] Par exemple dans `/tmp/$USER/`*`répertoire`*

Id : cl.tex,v 1.5 2000/11/28 18 :34 :55 marquet Exp

Les lignes du fichier de log de `cl`[3] pourront être de la forme :

<*répertoire*>@<*machine*>@<*date*>

3 Annexe

Quelques commandes et options sont décrites ci-dessous. Elles peuvent être utiles pour la mise en place de votre commande `cl`. Utilisez les pages du manuel en ligne (`man`) pour plus de précisions.

- L'option `-b` de la commande `cp` permet de créer une sauvegarde d'un fichier destination déjà existant. L'option `-a` préserve les informations de droits... L'option `-u` ne recopie que les fichiers dont la destination est plus ancienne.
- L'option `-v` de la commande `grep` n'affiche que les lignes ne contenants pas un motif donné.
- L'option `-p` de la commande `mkdir` permet de créer un répertoire et tous ses parents si nécessaire.
- Les commandes `basename` et `dirname` permettent d'isoler les deux composantes d'un nom de fichier : (i) le nom dans le répertoire et (ii) le nom du répertoire.
- La commande `uname -n` affiche le nom de la machine.
- La commande `touch` change les dates d'accès et de dernière modification d'un fichier. Si le fichier n'existe pas, il est créé (vide).

[3] Par exemple `$HOME/.cllog`