

TP 2 : Manipulation des fichiers et de leur contenu

Février 2014

Objectifs :

- utiliser les méta-caractères et les expressions régulières ;
- enchaîner les commandes grâce aux redirections.

1 Redirection

Écrire les lignes de commandes effectuant les actions suivantes :

1. copie du fichier `/etc/services` dans le fichier `rezo` sans utiliser la commande `cp` ;
2. comptage des utilisateurs existants sur le système ;
3. comptage du nombre de personnes connectées ;
4. ajout du numéro de ligne à un fichier ;
5. liste les répertoires de BE du département info (`/mnt/be/in`) et redirige les erreurs dans un fichier.

2 Manipulation des fichiers et de leur contenu

2.1 Recherche et + avec la commande `find`

`find` est une commande permettant de rechercher des fichiers dans un répertoire et d'effectuer diverses opérations sur ces fichiers. Par exemple, `find / -name toto -print` va chercher à partir de / tous les fichiers s'appelant `toto` et les afficher dans le shell.

La syntaxe de la commande est : `find 'repertoire' [EXPRESSIONS]`

La liste des EXPRESSIONS est composée d'options, de tests, d'actions et d'opérateurs. Chacune retourne vrai ou faux et par défaut le traitement continue tant que les expressions retournent vrai.

Les tests permettent de sélectionner certains fichiers. Par exemple, `-name` indique le nom des fichiers à rechercher. Les tests portent sur : nom, taille, date, type,...

Les actions permettent d'exécuter une commande sur les fichiers trouvés. Par exemple, `-exec grep toto '{}'` \; cherche la chaîne de caractères `toto` dans le fichier.

Pour tout savoir, tapez `man find`.

▷ Exercice 1 :

Écrire les lignes de commandes effectuant les actions suivantes :

1. trouve le nombre de programme écrit en C ; quelle est la différence avec `ls *.c` ?
2. liste les fichiers Postscript de votre HOME ;
3. supprime les fichiers `.o` ;
4. supprime les fichiers `.ps` qui correspondent à un `.c`.

2.2 grep et expressions régulières

La commande `grep` permet de rechercher une expression régulière (une chaîne de caractères par exemple) dans un ou plusieurs fichiers.

La syntaxe est la suivante :

```
grep regex file
```

Voir « man `-s7 regex` » pour la syntaxe des expressions régulières.

Nous allons utiliser le fichier `utilisateurs` obtenu à la section 1.

▷ Exercice 2 :

Écrire les lignes de commandes effectuant les actions suivantes :

1. affiche les lignes commençant par `tr` ;
2. affiche les lignes ne commençant pas par `tr` ;
3. trouve les login de la forme `tr(g ou h)+2 chiffres` ;
4. indique le nombre d'utilisateurs ayant le même prénom que vous ;
5. trouve les lignes finissant par `/bash`.

2.3 Manipulation du contenu d'un fichier

Nous allons utiliser le fichier `rezo` obtenu à la section 1. Il contient une liste de protocoles et leurs ports associés.

▷ Exercice 3 :

1. Trouver le port d'un des protocoles utilisés pour la messagerie qui s'appelle `imap`.
2. Supprimer les lignes de commentaires.
3. Remplacer le `/` par un `' '`.
4. Extraire le premier champs et enlever les doublons.
5. Trier sur le nom, puis sur le port, puis sur le port de façon numérique.
6. Combien y-a-t-il de port `tcp` ? `udp` ? Créer les fichiers `rezotcp` et `rezoudp` contenant le résultat des commandes précédentes.
7. Extraire le premier champs de ses deux fichiers puis les comparer pour savoir quel(s) protocole(s) n'existe qu'en `udp` ou `tcp`.

2.4 Utilisation conjointe de find et grep

Pour retrouver un type de fichier contenant une chaîne spécifique, nous pouvons utiliser `find` et `grep` ensemble. Vous pouvez ainsi retrouver tous les fichiers Java contenant une commande, un commentaire précis, etc ...

▷ Exercice 4 :

1. Recherchez votre prénom dans tous les fichiers `.c` stockés dans votre répertoire racine.
2. Combien de fonctions `main` y-a-t-il dans vos fichiers `C` ?

3 Historique des commandes

Le shell gère une liste des dernières commandes exécutées. Ce mécanisme permet d'obtenir un historique des commandes exécutées et de redemander l'exécution de ces commandes. La commande **history** permet d'obtenir l'historique des dernières commandes avec un numéro d'ordre associé à chaque commande. Les commandes ainsi numérotées peuvent être ré-exécutées simplement :

- !! permet de ré-exécuter la dernière commande ;
- !n permet de ré-exécuter la commande numéro n ;
- !ba permet de ré-exécuter la commande la plus récente commençant par ba.

Testez la commande **history** et la ré-exécution de commandes.