

TRAVAUX PRATIQUES

ESIEE ENGINEERING

ISBS1

Travaux pratiques N 2: Les commandes de bases sous Unix

Décembre
2010

Ramzi Mahmoudi
mahmoudr@esiee.fr

Les commandes de base sous Unix

ISBS 1 – TP2 DEC 2010

Les réponses aux questions posées sont à rendre pour la fin de la séance de TP (4H)

1) INODES ET LIENS:

- Quel est le numéro d'inode de votre répertoire de connexion ?
- Créer un nouveau répertoire nommé « **reper** ».
- Combien il y a-t-il de liens sur ce répertoire ?
- A quoi correspondent-ils ?
- Dans le répertoire **reper**, créer un sous-répertoire « **sousrep** ».
- Combien il y a-t-il maintenant de liens sur le répertoire « **reper** » ? Pourquoi ?
- Créer un fichier « **titi** » quelconque dans le sous répertoire « **sousrep** ».
- Quel est son numéro d'inode ?
- Changez le nom de ce dernier fichier, pour l'appeler « **toto** ».
- Le numéro d'inode change-t il ?
- Créer un lien (dur) nommé **liendur** sur le fichier « **toto** ».
- Vérifier que le nombre de liens est bien 2 et que les inodes sont identiques.
- Créer un lien symbolique nommé **liensymb** sur le fichier « **toto** ». Que constatez-vous ?
- Déplacer le fichier « **toto** » dans « **reper** ». Le **liendur** est il encore valide ?
- Et le **liensymb** ?
- Même question si l'on supprime le fichier « **toto** ».
- Dans le répertoire « **sousrep** », créer un lien sur le répertoire « **reper** ».
- Que se passe-t-il si l'on utilise la commande **ls -Ral reper** ?

2) CHAÎNAGE DE COMMANDES

- Pour exécuter plusieurs commandes à la file en les mettant sur la même ligne.
- Il faut les séparer par des « ; ».
- On peut ajouter des parenthèses aux commandes ou aux séquences de commandes.
- Tapez : **mkdir Test ; cd Test ; touch toto ; ls -l ; rm toto ; cd .. ; rmdir Test**.
- Que s'est-il passé ?
- Tapez **ls ; wc test > resultat** puis **(ls ; wc test) > resultatb**.
- Comparez les fichiers **resultat** et **resultatb**.
- Que remarquez-vous ?
- Comment chaîner **ls** et **wc** pour compter le nombre de fichiers contenus dans le répertoire courant ?

3) REDIRECTION DE STDIN ET DE STDOUT :

- Créer un nouveau répertoire nommé « ex3 ».
- Dans le répertoire « ex3 », créer trois sous-répertoire « sousrep1 », « sousrep2 », et « sousrep3 » puis exécuter les commandes suivantes :
 - `echo 'Resultat de la commande ls -t : '>resultat.txt`
 - `ls -t >>resultat.txt`
 - `echo '-----'>resultat.txt`
 - `echo 'Resultat de la commande ls -t : '>resultat.txt`
 - `ls -r >>resultat.txt`
 - `cat resultat.txt`
- Expliquer ce résultat ?
- Combien de ligne y a-t-il dans resultat.txt ?
- Essayez d'exécuter la commande `wc tutu`.
- « Le fichier tutu n'existe pas » Ce message est affiché non sur le canal de sortie mais sur le canal d'erreur.
- **Essayez** de rediriger ce message dans un fichier nommé erreur par la commande :
 - `wc trtrtr > erreur.txt`
- Consultez le contenu du fichier erreur.
- Exécutez la commande `wc trtrtr >& erreur.txt` et observez le résultat.

4) COMMANDE CUT

- Consulter le manuel de la commande `cut` et énoncer les différentes utilisations de la commande.
- Créer un fichier ex4.txt:

```
un, deux, trois, quatre  
cinq, six, sept, huit  
neuf, dix
```

- Donner la ligne de commande permettant de donner la première colonne.
- Donner la ligne de commande permettant de donner la première et la troisième colonne.
- Donner la ligne de commande permettant de donner du 5^{ème} au 9^{ème} caractères.

5) COMMANDE GREP

- Consulter le manuel de la commande `grep` et énoncer les différentes utilisations de la commande.
- Créer un fichier util.txt:

```
util1:x:501:501::/home/util1:/bin/bash  
util2:x:501:501::/home/util1:/bin/bash  
util3:x:501:501::/home/util1:/bin/bash  
util4:x:501:501::/home/util1:/bin/bash  
util5:x:501:501::/home/util1:/bin/bash  
util6:x:501:501::/home/util1:/bin/bash
```

- Afficher les lignes et les numéros de lignes contenant la chaîne « **ut** ».
- Afficher les lignes et les numéros de lignes contenant exactement le mot « **ut** ».
- Afficher toutes les lignes commençant par « **u** » ou « **U** ».
- Afficher toutes les lignes contenant au moins un chiffre.
- Afficher toutes les lignes commençant par une majuscule.
- Afficher toutes les lignes contenant « **a** », « **b** » ou « **c** ».
- Afficher les lignes contenant « **a** » comme deuxième caractère.
- Afficher les lignes qui se terminent par « **h** ».

6) PIPES

- Quelles sont, à votre avis, les commandes suivantes qui fonctionnent :
- `ls -R | more`
- `echo util.txt | rm`
- `echo util.txt | mkdir`
- `ls -R | tail -3`
- `cat util.txt | head -2`
- `ls -Rl | grep rwxr-- | more`

7) SÉLECTION DE TEXTE

- Créer un fichier `mami.txt`:

```
aaabc2 ;
absdsc.
aafdsfsdf;

cbccvf45.
ab333c5;
aa34c;
```

- Affichez les lignes contenant la chaîne « **ds** »
 - Affichez les lignes contenant un chiffre.
 - Affichez les lignes contenant une lettre **a** et contenant plus loin une lettre **c**.
 - Affichez les lignes contenant une lettre **c** suivie d'un chiffre.
 - Affichez les lignes commençant par autre chose qu'un **a**.
 - Affichez les lignes se terminant par un point virgule.
 - Comment générer ces lignes à partir du fichier «**mami**» et de l'application de **grep** ?
- ```
1:aaabc2;
2:absdsc.
3:aafdsfsdf;
4:
5:cbccvf45;
6:ab333c5;
7:aa34c;
```

## 8) REDIRECTION, TRI, COMPTAGE

- Lancez la commande « **cat /etc/passwd >passwords** ».
- Que fait-elle ?
- On utilisera le fichier « **passwords** » tout au long du reste de ce TP.
- Le format du fichier généré est :  
`login : mot de passe : UID numérique : GID numérique : gecos : homedir :shell`
- Le séparateur des champs est le caractère «:».
- Le fichier « **passwords** » que l'on a généré n'est pas trié.
- Triez le fichier par ordre croissant de login.
- Triez le fichier « **passwords** » par ordre croissant de **UID**.
- Combien de lignes le fichier **passwords** contient-il ?
- Affichez les 5 premières lignes de « **passwords** », les lignes étant ordonnées par UID.
- Affichez la quatorzième ligne de « **passwords** », (les lignes étant ordonnées alphabétiquement).
- Au moyen de la commande « **cut** », affichez les noms de login en triant le résultat par ordre alphabétique.

## 9) AWK

- Il existe une commande **UNIX** pratique pour travailler sur un fichier avec des colonnes. Il s'agit de la commande **awk**.
- Créez un fichier « papi » contenant les lignes :

|        |        |    |     |      |
|--------|--------|----|-----|------|
| or     | 100000 | aa | a11 | 5565 |
| argent | 40000  | cc | b12 | 9876 |
| bois   | 5      | kk | c13 | 4566 |

- Essayez les commandes suivantes :
  - `awk '{ print $2 }' papi`
  - `awk '{ print $2, $1 }' papi`
  - `awk '{ print $1, "/", $2 }' papi`
- Modifiez le fichier « papi » pour que les espaces soient remplacés par «:»
- Essayez les commandes suivantes :
  - `awk '{ print $2 }' data`
  - `awk '{ print $1 }' data`
  - `awk -F : '{ print $2 }' data`

## 10) SUPPRESSION DE DOUBLONS

- La commande « **uniq** » supprime dans un fichier plusieurs lignes consécutives strictement identiques.
- Créez un fichier « **greve** » et copiez-y les lignes :

```
La poste est en greve.
La poste est en greve.
La poste est en greve.
Les banquiers sont en greve.
Les banquiers sont en greve.
La poste est en greve.
La poste est en greve.
```

- Appliquez la commande **uniq** à ce fichier et constatez le résultat.
- Triez le fichier puis appliquez la commande **uniq** et constatez le résultat.