

Travaux pratiques: les filtres simples

1. Analyse de fichier de log d'un serveur Web

A l'aide des filtres `wc`, `cut` et `sort`, analysez le fichier [access_log.1](#):

- Comptez le nombre de requêtes faites au serveur Web.
- Triez le fichier par adresse IP de machines s'étant connectées au serveur Web.
- Affichez les différentes adresses IP des machines s'étant connectées au serveur Web.
- Combien de machines différentes se sont connectées au serveur?
- Comptez le nombre de connexions de chacune des machines.
- Quelle est la page la plus fréquemment lue?
- Affichez le nombre de navigateurs Web différents utilisés.
- Toutes les adresses IP de machines de l'INT commencent par 157.159. Comptez le nombre de requêtes provenant de machines de l'extérieur.

2. La commande `find`

Tester les différentes options de la commande `find`:

- lister les fichiers sous `/dev` qui ne sont pas de type bloc ou caractère
- lister les éléments modifiés durant les dernières 24 heures sous `/var`
- lister les fichiers sous `/etc` ayant une taille comprise entre 10000 et 20000 octets

3. Analyse du fichier magic

Le fichier `/usr/share/file/magic` ([magic](#)) contient la liste des "magic number" utilisés par la commande `file`. A l'aide des filtres `grep`, `sort` et `wc`:

- comptez le nombre de type `mime` différents;
- affichez les différents types `mime`;
- affichez les types `mime` correspondants au "magic number" `#!`

4. Analyse du fichier auto

À l'aide des filtres `grep` et `wc`, à partir du fichier [auto](#):

1. affichez la liste des véhicules gris du fichier `auto`
2. affichez la liste des modèles (2ème champ);
3. affichez par ordre alphanumérique la liste des modèles (2ème champ);
4. affichez la liste des véhicules de l'année 89;
5. affichez le nombre de véhicules qui ne sont pas rouges;
6. triez le fichier.

5. Analyse du fichier `/usr/dict/words`

Indication: soit exécutez les commandes directement en interactif soit créez un script shell (ce qui permettra de faire des modifications ou des corrections).

1. Le fichier `/usr/dict/words` contient une liste de mots :
 - afficher le nombre de mots ayant 2 caractères puis 2 caractères en majuscule,
 - afficher le nombre de mots commençant par une majuscule et se terminant par z,
 - le nombre de mots constitués uniquement de majuscules.
2. lister les utilisateurs sans mot de passe
3. lister les entrées sous `/` dont le nom commence par `.` sans afficher `.` et `..`
4. trier de différentes façons `/etc/group` et `/etc/passwd` (ordre ASCII, inverse, sur numéro d'UID et de GID ...)
5. trier les entrées du répertoire `/` selon l'ordre décroissant des tailles (et en cas d'égalité sur l'ordre ASCII du nom)
6. lister les utilisateurs connectés en commençant par les plus anciens
7. lister les sous-répertoires du répertoire `/` puis l'inverse c'est à dire les entrées de type fichier;
8. afficher le nom et le champ "information" pour chaque utilisateur déclaré ayant pour numéro de GID 100. (5 utilisateurs). On pourra en sortie remplacer le séparateur : par espace;
9. créer un script shell qui mémorise dans un fichier liste le contenu de votre répertoire mais sans que le fichier liste n'apparaisse dans la liste;
10. créer un script shell qui affiche à l'écran la date et les utilisateurs connectés et qui en même temps ajoute l'information dans un fichier;
11. lister tous les utilisateurs non affectés à un shell (lignes de `/etc/passwd` non terminées par : ou `/bin/sh` ou `/bin/csh`)

5.2 Utilisation des expressions régulières

Écrire un script shell qui affiche la lettre qui apparaît le plus souvent dans le fichier `/usr/dict/words` en tant que premier caractère d'un mot (en se limitant aux seuls mots communs c'est à dire sans majuscule).

Principe : éclater le fichier `words` en 26 fichiers, chacun contenant les mots commençant par chacune des 26 lettres. Ensuite, compter et trier le nombre de lignes de ces fichiers. On pourra utiliser les différents filtres (`grep`, `ed`...) pour récupérer dans `words` les 26 lettres.
