

L'administration d'un serveur Apache sous Linux

Module 3 - Gérer les fichiers



Objectifs

- Savoir afficher le contenu des fichiers
- Savoir chercher dans des fichiers
- Savoir chercher des fichiers
- Savoir éditer des fichiers

Afficher du texte

- Pour **afficher le contenu** des fichiers, il existe plusieurs commandes
- La commande **more** permet d'afficher un fichier **avec pause** à chaque page écran
- La commande **less** est similaire à **more** avec quelques améliorations comme **la navigation avec les flèches** haut et bas
- La commande **cat** permet **d'afficher le contenu sans arrêt**

Afficher du texte

- Exemples :

```
$ more /etc/services
```

```
$ less /etc/services
```

- La touche [Espace] permet de passer les pages

- ```
$ cat /etc/services
```

# Afficher du texte

- Avec `less`, on peut utiliser certaines commandes :
  - `/chaîne` : permet de rechercher la chaîne de caractères
  - `G` : positionnement en fin de fichier
  - `gg` : positionnement en début de fichier

# Commande head

- Syntaxe : `head [options] fichier`
- Permet d'afficher les premières lignes d'un fichier
- Options :
  - `-n <nb>` : nombre de lignes à afficher (par défaut 10)
  - `-c` : nombre de caractères par ligne
- Exemple : `$ head -n 5 /etc/services`

# Commande tail

- Syntaxe : `tail [options] fichier`
- Permet d'afficher les dernières lignes d'un fichier
- Options :
  - `-n <nb>` : nombre de lignes à afficher
  - `-f` : lecture en temps réel de la fin du fichier
- Exemples :
  - `$ tail -n 5 /etc/services`
  - `$ tail -f /var/log/syslog`

Gérer les fichiers

# Afficher le contenu des fichiers

## Démonstration





# Recherche dans les fichiers

- Syntaxe : `grep [options] 'expression' <fichier>`
- Permet de rechercher `'expression'` dans un ou des fichiers
- Options :
  - `-n` : affiche le numéro des lignes contenant `'expression'`
  - `-l` : n'affiche que le nom du fichier contenant `'expression'`
  - `-v` : affiche les lignes ne contenant pas `'expression'`
  - `-i` : ignore la casse

# Recherche dans les fichiers

- Exemples :
- `$ grep -i smtp /etc/services`
- `$ grep -in '^http*' /etc/services`
- `$ grep -v tcp /etc/services`
- La seconde commande utilise une **expression régulière**

Gérer les fichiers

# La commande grep

## Démonstration



# Recherche des fichiers

- Syntaxe : `find <chemin> [critères] [action]`
- Commande permettant de **rechercher des fichiers** suivant **un ou des critères** et d'appliquer l'action
- Attention, c'est une commande **récursive**, qui peut donc être très consommatrice en ressources
- **<chemin>** : chemin à partir duquel faire la recherche  
Sans précision, c'est le dossier courant

# Recherche des fichiers

- **[critères]** : de nombreux critères existent et peuvent se **cumuler**
  - **-name "nom"** : recherche en fonction du nom
  - **-user** : recherche en fonction du propriétaire
  - **-type** : recherche en fonction du type
    - **f** pour fichier
    - **d** pour répertoire
  - **-size <nb>** : recherche en fonction de la taille
  - **-mtime <nb>** : recherche en fonction de la date de dernière modification

# Recherche des fichiers

- **[action]** : action à exécuter pour chaque fichier correspondant aux critères :
  - **-print** : par défaut, affiche les fichiers
  - **-exec cmd [options] {} \;** : la commande est exécutée avec ses options  
{} désigne le fichier correspondant aux critères  
La commande se termine par : **[espace]\;**
  - **-ok cmd [options] {} \;** : identique à **exec** avec confirmation pour chaque exécution

# Recherche des fichiers

- Exemple :

```
$ find /etc/ -name 'app*' -type f -exec ls -la {} \;
```

- **Attention** à la commande exécutée, les erreurs étant irrécupérables

Gérer les fichiers

# La commande find

## Démonstration





# Les éditeurs

- Tout est fichier, il faut donc pouvoir éditer les fichiers
- L'éditeur conventionnel sous Linux est **vi**
- Beaucoup d'utilitaires utilisent **vi**, il est donc important de savoir le manipuler
- L'extension **vim** permet de simplifier l'utilisation de vi

# Les éditeurs

- Il existe d'autres éditeurs suivant la distribution :
  - **emacs** : concurrent de vi
  - **nano** : éditeur simplifié
  - **joe**
  - **jed**
  - **jedit**
  - ...

# Gérer les fichiers

## vi

- vi dispose de deux **modes** de travail :
  - **mode édition** : permet d'éditer les fichiers
  - **mode commande** : permet d'exécuter des commandes
- Par défaut, lors du démarrage de l'éditeur, c'est le mode commande qui est activé
- La touche **[Échap]** permet de passer du mode édition au mode commande
- Les commandes doivent être précédées de « : »

Gérer les fichiers

# vi : mode édition

- Pour passer en **mode édition** :
  - **a** : insertion après le caractère courant
  - **i** : insertion avant le caractère courant
  - **o** : ajout d'une ligne vide après la ligne courante
  - **O** : ajout d'une ligne vide avant la ligne courante
- **[Échap]** pour repasser en **mode commande**



Gérer les fichiers

# vi : se déplacer

- Pour se déplacer :
  - **<nb>flèche haut** ou **<nb>k** : vers le haut
  - **<nb>flèche bas** ou **<nb>j** : vers le bas
  - **<nb>flèche gauche** ou **<nb>h** : vers la gauche
  - **<nb>flèche droite** ou **<nb>l** : vers la droite
  - **:nb** : aller à la ligne
  - **gg** : aller au début du fichier
  - **G** : aller à la fin du fichier

# vi : rechercher des chaînes

En mode commande sans le « : »

- **/<chaîne>** : recherche l'occurrence suivante de la chaîne
- **?<chaîne>** : recherche l'occurrence précédente de la chaîne
- **n** : répétition de la dernière recherche dans le même sens
- **N** : répétition de la dernière recherche dans le sens inverse

# vi : suppression

- En mode commande sans le « : »
- **<nb>x** : suppression des <nb> caractères suivants à partir de la position courante
- **dd** : suppression de la ligne courante
- **<nb>dd** : suppression de <nb> lignes à partir de la ligne courante

Gérer les fichiers

# vi : copier/coller

En mode commande sans le « : »

- Conseil : sous vim, utilisez le **mode visuel** pour sélectionner avec la touche **v** et les flèches les mots ou lignes à copier puis tapez la touche **y**.
- **<nb>p** : copie après la ligne courante <nb> fois le tampon
- **<nb>P** : copie avant la ligne courante <nb> fois le tampon



# Gérer les fichiers

## vi : sortie

- **:wq** : sauvegarde et sortie de vi
- **:q** : sortie sans sauvegarde
- **!** forcer la commande qui précède

Exemple : sortie sans sauvegarde

**:q !**

Gérer les fichiers

# todos / fromdos

- La déclaration de fin de ligne sous Unix est `\n`
- La déclaration de fin de ligne sous Windows est `\r\n`
- Il faut convertir les fichiers pour ne pas avoir de problèmes
- Unix vers Windows : `todos <fichier>`
- Windows vers Unix : `fromdos <fichier>`



Gérer les fichiers

# L'éditeur vi

## Démonstration



# Conclusion

- Vous savez comment afficher le contenu d'un fichier
- Vous savez comment éditer les fichiers