

TP n°2 :L'éditeur Vi, Redirection et Pipe, Manipulation des Filtres

1 L'éditeur *vi*

Description

L'éditeur *vi* est l'éditeur standard du système d'exploitation UNIX. Pour utiliser l'éditeur *vi*, vous n'avez qu'à entrer à votre terminal *vi*, suivi d'une *espace* puis du *nom du fichier* que vous voulez éditer. Si le fichier n'existe pas, *vi* le créera automatiquement.

Fonctionnement

L'éditeur *vi* est un éditeur modal, c'est-à-dire qu'il se retrouve toujours dans l'un ou l'autre des deux modes suivants *Mode d'insertion* ou *Mode de commandes*. Selon le mode dans lequel il se trouve, l'éditeur interprète différemment les caractères qu'il reçoit du clavier :

Lorsque vous le lancez, *vi* se place initialement dans le *mode de commandes*.

2 Redirection et Pipe

Les commandes que nous avons vues jusqu'ici prennent leurs arguments au clavier et renvoient leurs résultats dans la fenêtre de terminal. En fait, chaque commande utilise trois *périphériques* principaux : l'*entrée*, la *sortie* et la *sortie d'erreur*. Il est possible de dire à l'interpréteur de commandes d'utiliser une autre source de données que l'*entrée standard* ou d'afficher les résultats dans un fichier (par exemple) plutôt que sur la *sortie standard*.

- Pour modifier l'entrée standard, on utilise l'opérateur `<`. Par exemple, pour utiliser le fichier *entrees.txt* à la place du clavier dans la commande *maCommande* , on utilisera la syntaxe : *maCommande < entrees.txt*.
- Pour modifier la sortie standard, on utilise l'opérateur `>`. Par exemple, pour stocker les résultats de la commande *maCommande* dans le fichier *sorties.txt*, on utilisera la syntaxe : *maCommande > sorties.txt*.
- L'opérateur de redirection de la sortie d'erreur standard est l'opérateur `2>`. Par exemple *maCommande 2> sorties.txt*

Exercice 1

1. Redirigez la sortie de `ls -l /bin` dans le fichier `list.txt` ; affichez le contenu de ce fichier.
2. Essayez de lire le contenu du fichier `/var/log/messages`. Maintenant exécutez la même commande en redirigeant la sortie d'erreurs dans le fichier `Error`. Affichez le contenu de ce fichier

Un autre opérateur permet de faire communiquer deux commandes en redirigeant la sortie de l'une sur l'entrée de l'autre. Il s'agit de l'opérateur `|` (*tube*). Par exemple, **maCommande1|maCommande2**

Exercice 2

Utilisez les mécanismes de redirections (`'<', '>', '|', '> >'`) et les commandes Unix simples (`cat`, `wc`, `tail`, `head`) pour répondre aux questions suivantes :

1. Copiez le fichier `/etc/mailcap` dans votre répertoire `$HOME` ;
2. Combien le fichier `mailcap` comporte-t-il de lignes ? De mots ? De caractères ?
3. Affichez la ligne 18, les 20 dernières, les lignes 40 à 50 du fichier `mailcap`.
4. Combien de mots y a-t-il entre les lignes 30 et 40 ?
5. Créez un fichier `début` formé des 10 premières lignes de `mailcap` et un fichier `fin` formé des 10 dernières.
6. Créez un fichier `resultat` constitué de la concaténation des fichiers `début` et `fin`. Renommez le fichier `resultat` en `resultat.old`.
7. Essayez de reconstruire le fichier `resultat`, mais en une seule ligne de commandes, sans utiliser les fichiers intermédiaires `debut` et `fin`¹

3 Manipulation des Filtres

1. La commande `tr`

Exercice

Que font ces commandes ?

1. `tr 'a' 'A' < fich1 >fich2`
2. `tr -cs 'a-zA-Z0-9' '\n' < fich1`

2. La commande `wc`

Exercice

Donnez la suite des commandes permettant de compter le nombre des fichiers dans le répertoire courant :

- a En utilisant les commandes `ls` et `wc -w`
- b En utilisant les commandes `ls` , `tr` et `wc -l`

¹On rappelle que le caractère `'>'` permet d'enchaîner la commande qui lui succède à la commande qui le précède.

3. La commande cut

Exercice

Soit la fichier nombres suivant :

```
$cat nombres
```

```
1,un,one,ein
```

```
2,deux,tow,zwei
```

```
3,trois,three,drei
```

```
4,quatre,four,vier
```

```
5,cinq,five,fünf
```

1. Supprimez du fichier nombres les trois dernières colonnes de chaque ligne
2. Donnez la ligne de commande permettant d'obtenir le résultat suivant :

```
1 one
```

```
2 tow
```

```
3 three
```

```
4 four
```

```
5 five
```

3. Interprétez le résultat de la ligne de commande suivante :

```
$cut -f3- -d',' nombres
```

4. La commande grep

Exercice

Soit la fichier fich suivant :

```
$cat fich
```

```
le travail
```

```
Afin de gagner sa vie un travailleur devra travailler
```

```
travail
```

```
travailleur
```

```
ne jamais sous estimer son ennemi
```

```
1 pour tous et tous pour un
```

-Trouvez les options de la commande grep permettant de :

1. Chercher toutes les lignes commençant par **a** ou **A**
2. Chercher toutes les lignes finissant par **er**
3. Chercher toutes les lignes contenant au moins un chiffre
4. Cherchez tous les mots dont la seconde lettre est un **r**
5. Apparaître les lignes où ne figurent pas un mot dont la seconde lettre est un r.
6. Ne rechercher que les lignes où figure le mot tel quel, et non pas ses variantes?
exemple : on cherche le mot **travail**, mais pas de **travailleur** ou **travailler**
7. Compter le nombre des lignes contenant la chaîne **ail**
8. Afficher les noms des fichiers contenant la chaîne **ail**
9. Afficher et numéroter les lignes contenant la chaîne **ail**

-Que font ces commandes ?

1. `ls -l /usr/bin | grep '^ [^0-9]*$'`
2. Afficher les noms de fichiers dont le nom ne contient qu'un seul chiffre

3. `ls -l /usr/bin | grep '[0-9].*[0-9]'`

5. La commande find

Exercice

En utilisant la commande find :

1. Rechercher toutes les apparitions de fichiers "s*.h" à partir du répertoire /usr/include.
2. Affiner la recherche précédente pour n'obtenir que
 - les apparitions de fichiers "s*.h" qui sont des fichiers classiques.
 - les apparitions de fichiers "s*.h" avec toutes les informations relatives à ces fichiers
3. Concatener tous ces fichiers dans un fichier unique concat.txt
4. Où se trouve le fichier passwd
5. Trouver les fichiers qui ont été modifier pendant les 10 dernières minutes
6. Trouver tous les fichiers du groupe 500
7. Chercher tous les fichiers de l'utilisateur **isi** se trouvant sous */home*
8. Effacer tous les fichiers dont l'extension est *.tmp* qui se trouvent sous */home*.

6. Compte rendu

Exercice 1

Ecrire la ou les commande(s) permettant d'effectuer l'action demandée.

On suppose que votre répertoire courant est « isi ».

1. Afficher tous les fichiers ordinaires du répertoire courant , ayant un nom dont le troisième caractère est « d ».
2. Créer un fichier « res » sous le répertoire courant, ayant comme contenu toutes les informations sur les fichiers ordinaires contenus sous le répertoire /home/isi.(utiliser la commande find).
3. Afficher et numéroter toutes les lignes du fichier fich qui contiennent exactement la chaîne « il ». (On suppose que fich est un fichier du répertoire courant)
4. Supprimer tous les fichiers ordinaires du répertoire /home/isi ,dont l'extension est « .txt »

Exercice 2

Décrire la sortie de chacune des commandes suivantes :

1. `[isi]$ cat f1 | tr ' '\n' | grep '.*t$'`
2. `[isi]$ ls | grep '^[mM][a-zA-Z0-9]*[5-8]'`
3. `[isi]$ cut -f2- -d ':' /etc/passwd >> f3`
4. `[isi]$ head -15 /etc/passwd | tail -9 > f4`