

TP3 - Textutils

Laurent Tichit

24 août 2018

Plan

- 1 Fichiers texte
- 2 Périphériques standards
- 3 Redirections
- 4 /dev/null
- 5 Pipe

Fichiers texte

Affichage, Concaténation

- **cat fich1** : Affiche le fichier sur la sortie standard.
- **cat fich1 fich2 fich3...** : Concatène plusieurs fichiers.

Début et fin de fichier

- **head [-n] [fichier]** : affiche les n premières (10 par défaut) lignes du fichier.
- **tail [[+ nombre] [lbc]] [fichier]** : recopie sur stdout le morceau de fichier spécifié.

Affichage page par page

- **more [-n] fichier...** : permet d'afficher page par page un long fichier.
- **less fichier...** : permet d'afficher page par page un long fichier.
Permet les mêmes commande interactives que **man**.

Tri de lignes

- **sort [options] [[+n1] [-n2]] [fich1...]** : permet de trier lexicographiquement les lignes des fichiers d'entrée.

```
tichit@iml230:~$ sort -t: +2n /etc/passwd
```

- ▶ Trie le fichier /etc/passwd
 - ▶ sur le troisième champ (+2n),
 - ▶ séparateur ':' (-t:)
- **uniq [options] fichier1 fichier2** : permet de supprimer les lignes *adjacentes* identiques d'un fichier.

Découpage par colonne

- La commande **cut** permet d'extraire certains champs d'un fichier.
 - ▶ **-c** suivant le nombre de caractères
 - ▶ **-f** suivant le nombre de champs
 - ▶ **-d** spécifie le séparateur de champ
- **cut -c-10 /etc/passwd**
Extrait les 10 premiers caractères de chaque ligne.
- **cut -c2-10 /etc/passwd**
Extrait les caractères 2 à 10 de chaque ligne.
 - ▶ **n** : n^{ieme} caractère compté à partir de 1.
 - ▶ **m-** : du m^{ieme} caractère jusqu'à la fin de la ligne.
 - ▶ **m-n** : du m^{ieme} au n^{ieme} (inclus) caractère.
 - ▶ **-n** : du premier au n^{ieme} (inclus) caractère.
- **cut -d: -f 5 /etc/passwd**
Extrait le champ 5 de chaque ligne. Le séparateur de champs est ' : '.

Concaténation verticale

- **paste fich1 fich2...** : Concatène ligne à ligne plusieurs fichiers. En fait chaque fichier devient la colonne d'un tableau.

```
tichit@iml230:~$ cat premier
```

```
1
```

```
1
```

```
1
```

```
tichit@iml230:~$ cat deuxieme
```

```
2
```

```
2
```

```
2
```

```
tichit@iml230:~$ paste premier deuxieme
```

```
1 2
```

```
1 2
```

```
1 2
```

Comptage des caractères, mots, lignes

- **wc [options] [fichier...]** : compte le nombre de lignes, de mots ou de caractères d'un fichier.

Comparaison

- **diff [options] fichier1 fichier2** : permet de comparer deux fichiers ligne à ligne.

Commande **tr**

Recopier l'entrée sur la sortie en remplaçant certains caractères par d'autres.

- Remplace les 'a' par des 'A', les 'b' par des 'B' et les 'c' par des 'C' :
tr abc ABC < fichier1
- Remplace les majuscules par des minuscules :
tr "[A-Z]" "[a-z]" < fichier1
- Placer chaque mot d'un fichier donné sur une ligne différente.
tr -cs "[A-Za-z-']" "\n" < fichier1
- Créer un dictionnaire de tous les mots d'un texte en ordre alphabétique.
tr -cs "[A-Za-z-']" "[\n]" | sort | uniq

Shell : les redirections

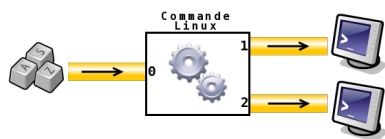
Périphériques standards

Unix possède trois périphériques standards qui sont :

- L'entrée standard (pseudo-fichier `/dev/stdin`) : le clavier (STDIN, descripteur 0)
- La sortie standard (pseudo-fichier `/dev/stdout`) : le terminal actif (STDOUT, descripteur 1)
- La sortie erreur standard (pseudo-fichier `/dev/stderr`) : le terminal actif (STDERR, descripteur 2)

Les diverses commandes utilisent par défaut ces trois moyens de dialogue. Cela signifie que :

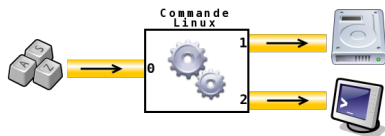
- la commande et ses paramètres sont lus au clavier,
- si la commande est interactive, elle lira au clavier,
- les messages de sortie, résultat ou erreur seront visualisés dans le terminal actif.



Redirections

Rediriger la sortie standard

- La commande **echo bonjour** imprimera le texte *bonjour* sur *stdout*.
- On peut rediriger cette sortie sur n'importe quelle sortie grâce au caractère de redirection **>** (on peut aussi utiliser **1>**).
- Si on désire écrire **à la suite** de *fich1* (sans perdre son contenu), le caractère de redirection sera **>>**.
echo Monsieur >> fich1 ajoutera la ligne Monsieur à la fin du fichier *fich1*.



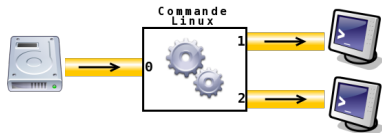
Rediriger la sortie standard

La commande **echo Bonjour > fich1** créera un fichier *fich1* contenant la ligne `Bonjour`. Au niveau du système ce *processus* est effectué de la façon suivante :

- Le *shell* crée le fichier *fich1*. Si celui-ci existe déjà, il est vidé.
- Le *shell* redirige la sortie standard de *echo* vers *fich1*
- Le *shell* fait exécuter *echo* par le système et s'endort
- *echo* écrit la ligne `Bonjour` dans *fich1* et se termine
- Le *shell* est réveillé, affiche le prompt, et attend la prochaine commande.

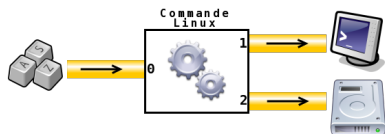
Rediriger l'entrée standard

- La commande **tr abc DEF** a pour but de recopier un texte lu sur l'entrée standard (i.e. au clavier) en remplaçant les *a* par des *D*, les *b* par des *E*, les *c* par des *F*.
- Elle est le plus souvent appelée en redirigeant son entrée et sa sortie vers des fichiers.
- L'entrée standard est redirigée au moyen du caractère de redirection `<`
- Ex. : **tr abc DEF < fich1.txt > fich2.txt** recopie le contenu de *fich1.txt* dans *fich2.txt* en remplaçant les *a,b,c* par des *D,E,F*.



Rediriger les erreurs

- Pour rediriger la sortie standard d'erreur, l'opérateur est : **2>**
- C'est utile pour conserver les messages d'erreurs produits par une commande vers un fichier de log.



`/dev/null`

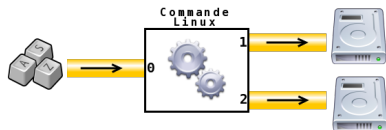
Le device `/dev/null`

- C'est un périphérique spécial qui est un fichier **toujours** vide.
- On redirigera toutes les sorties dont on ne désire pas prendre connaissance et que l'on souhaite voir disparaître.
- Ce périphérique sera souvent utilisé pour éliminer les messages d'erreur de l'affichage.

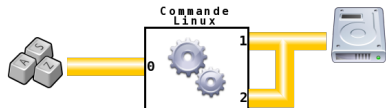
```
tichit@iml230:~$ cat texte 2>/dev/null
```

Redirection des 2 sorties

- Si on redirige les deux sorties vers des fichiers différents :
tichit@iml230:~\$ **cat texte > result.txt 2>/dev/null**



- Si on redirige les 2 sorties vers le meme fichier :
tichit@iml230:~\$ **find / -name "*.txt" > result.txt 2>&1**



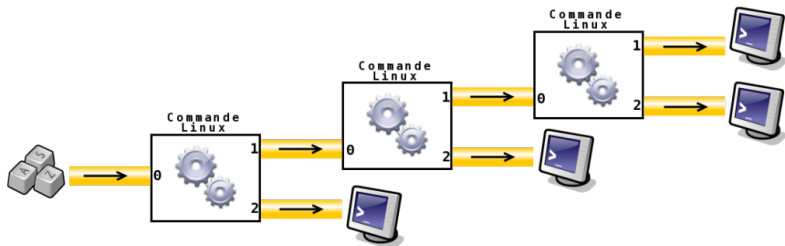
Pipe

Enchaîner des commandes

On peut aussi rediriger la sortie standard d'une commande vers l'entrée d'une autre.

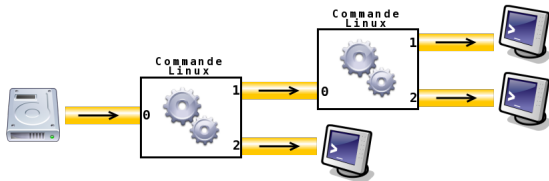
Pour lier deux commandes, on se sert du symbole `|`. Cela s'appelle un **pipe** (ou tube).

Exemple :
`tichit@iml230:~$ ls -l | wc | tr -s " "`



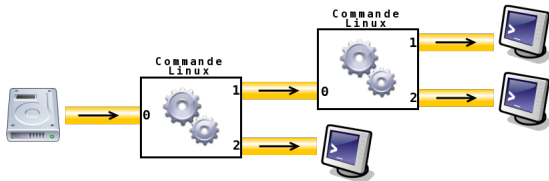
Exercice

On peut aussi mixer ces différents opérateurs. Ecrivez les commandes suivantes :

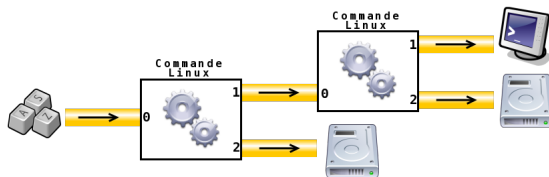


Exercice

On peut aussi mixer ces différents opérateurs. Ecrivez les commandes suivantes :

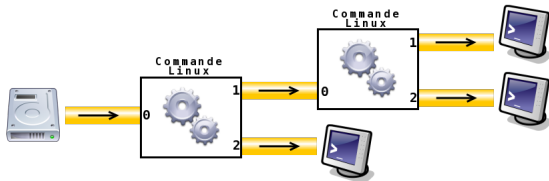


`cmd1 < f1.txt | cmd2`

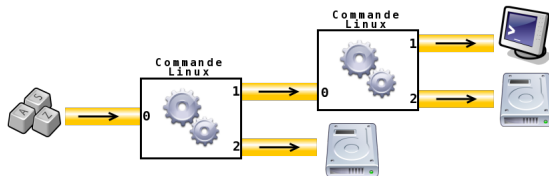


Exercice

On peut aussi mixer ces différents opérateurs. Ecrivez les commandes suivantes :



`cmd1 < f1.txt | cmd2`



`cmd1 2> err1.txt | cmd2 2> err2.txt`