

IF 110 - Outils indispensables



Joachim Bruneau-Queyreix

ENSEIRB-MATMECA

Bordeaux-INP

jbruneauqueyreix@enseirb-matmeca.fr

D'après le cours d'introduction aux systèmes d'exploitation de Télécom SudParis



□ Outils incontournables

- Nature d'une entrée
- Pour les fichiers texte : affichage, tri, recherche de motif
- Occupation disque
- Archivage de fichiers
- Recherche de fichiers



Nature d'une entrée du système de fichiers

- Traitement applicable à un fichier dépend de sa nature
 - Est-ce un fichier texte ? Une image ? Une archive ? Un pdf ?
- Commande `file` : affiche la nature d'une entrée
 - Si texte, précise le type d'encode
 - » ASCII s'il n'y a que des caractères, UTF-8 si caractères accentués, etc.

```
$ file *
TP3              directory
tp3.html        exported SGML document, UTF-8 Unicode text
cm3.pdf         PDF document, version 1.5
cm3.pptx        Microsoft Powerpoint 2010
Notes.txt       ASCII text
Pedagogie.txt   UTF-8 Unicode text
```

3



Nature d'une entrée du système de fichiers

- La commande `test` teste aussi la nature d'un fichier
(rappel `test cond` peut s'écrire [`cond`] avec bash)
 - [`-e fichier`] : vrai si fichier existe
 - [`-f fichier`] : vrai si fichier existe et est normal
 - [`-d fichier`] : vrai si fichier existe et est répertoire
 - [`-L fichier`] : vrai si fichier existe et est un lien symbolique

(remarque : les autres tests suivent les liens symboliques)

4



Affichage d'un fichier en mode texte

Consultation du contenu d'un fichier ordinaire

more <fichier>
less <fichier>

} affichage simple page par page

head [-n] <fichier> : affichage des n premières lignes

tail [-n] <fichier> : affichage des n dernières lignes

cat <fic1> [fic...] : affiche la concaténation des fichiers indiqués

wc [-l] [-w] [-c] <fic> : compte les lignes, mots et caractères du fichier

Option -l, compte uniquement les lignes ; -w, les mots ; -c, les caractères

5



Extraire des parties de lignes

- cut -c plage fic : extrait des caractères de chaque ligne de fic
(stdin si fic absent)
 - plage : num ou num1, num2, ... ou num1-num2

Exemple : cut -c3-7 fic.txt
⇒ extrait les caractères 3 à 7 de fic.txt

- cut -d car -f plage fic : extraits des champs
 - -d car : car = séparateur de champs (tabulation par défaut)
 - plage comme dans le cas précédent

Exemple : cut -d' ' -f2,4 fic.txt
⇒ extrait les 2^{ème} et 4^{ème} mots de chaque ligne de fic.txt

6



Supprimer ou transformer des caractères

- `tr s1 s2` : transforme chaque caractère de `s1` en ceux de `s2`
(à partir de `stdin`)
 - Exemple : `cat fic | tr '\n' 'ab'`
⇒ transforme les retours à la ligne en a et les espaces en b
 - Exemple : `cat fic | tr '\n' 'a'`
⇒ transforme les retours à la ligne et espaces en a

- `tr -d s` : élimine chaque caractère de la chaîne `s`
 - Exemple : `cat fic | tr -d 'aeiouy'`
⇒ élimine les voyelles de `fic`

7



Trier les lignes de fichiers texte

- `sort fic...`
 - Par défaut, tri lexicographique
 - » Option `-n` pour un tri numérique
 - Par défaut, tri appliqué en tenant compte de toute la ligne
 - » Option `-k x[,y]` pour un tri selon les champs `x` à `y`
 - `sort -k 2 fichier` : tri selon le 2^{ème} champ de chaque ligne
 - `sort -k 2,4 fichier` : tri selon les 2 à 4^{ème} champs de chaque ligne
 - Par défaut, le séparateur de champs est l'espace
 - » Option `-t <caractère>` pour changer le séparateur
 - Option `-r` pour inverser l'ordre du tri appliqué
 - Peut s'appliquer sur un ensemble de fichiers
 - D'autres options à consulter dans la page du manuel

8



Recherche d'un motif dans un fichier texte

- ❑ grep motif fichier fic ...
 - Affiche les lignes des fichiers contenant le motif
 - Le motif est une expression régulière (ou rationnelle)
 - » grep = *global regular expression print*
 - Pour IF110, seul un sous-ensemble d'expressions régulières
 - » Chaînes de caractères
 - » Attention! Les métacaractères vus précédemment n'ont pas la même signification dans les expressions régulières!
 - . : n'importe quel caractère
 - ? / * / + : répétition du caractère précédent 0 ou une fois / 0 ou plusieurs fois / au moins une fois
 - [...] (/ [^...]) : met en correspondance un caractère de (/hors) l'ensemble
 - Option -v affiche les lignes ne contenant pas le motif
 - Option -r permet de chercher récursivement le motif dans la sous-arborescence du répertoire donné en argument
 - D'autres options à consulter dans la page de manuel

9



Taille de l'occupation disque

- ❑ df : connaît l'état d'occupation des partitions
- ❑ ls -lh chemin ... : taille des chemins cibles
 - Si répertoire, donne la taille nécessaire au stockage de sa table d'entrées mais n'inclut pas celle de ses sous-entrées
 - Si lien symbolique, donne sa taille, i.e. l'espace nécessaire au stockage du chemin vers sa cible, ce qui correspond au nombre de caractères de ce chemin
- ❑ du : totalise l'occupation disque d'une entrée
 - Si répertoire, parcours récursif de son arborescence
 - Par défaut, donne le nombre de blocs occupés
 - » Option -h, pour afficher l'équivalent de ce nombre de blocs de manière « lisible pour un humain » en o/K/M/G
 - » Option -d0 pour éviter l'affichage des tailles des sous-répertoires

10



Archivage

- Rassemblement d'une arborescence de fichiers en un seul fichier
- Commande `tar` (pour **tape archive**) ⇒ manipuler des archives
 - Option `-c` `chem`, pour créer l'archive à partir du chemin `chem`
 - Option `-v`, pour un affichage en mode verbeux
 - Option `-z`, pour une compression des données au format gzip
 - Option `-f nom.tgz`, pour préciser le nom de l'archive voulue
 - Par convention, extension `.tgz` ou `.tar.gz`
 - Option `-x`, pour extraire (`-z`, pour la décompression via gzip)
⇒ décompression dans le répertoire courant

11



Recherche dans une arborescence

- `find` : recherche des entrées satisfaisants un ensemble de critères de sélection dans une arborescence
 - Parcourt récursivement et teste les critères à chaque étape
 - `find rep_de_recherche liste des critères`
 - » `-name <chaine>` : précise le nom des entrées à rechercher
 - Peut inclure des méta-caractères
 - » `-print` : permet l'affichage des résultats
 - » Exemple : `find . -name core -print`
 - affiche les chemins des entrées nommées `core` de mon répertoire courant
 - `find /usr -name "*.c" -print`
 - affiche les chemins des entrées dont le nom se terminent par `.c` sous `/usr`

12



Conclusion

□ Commandes clés :

- more, less, head, tail, cat, wc
- cut, tr, sort, grep
- df, du, ls -lh
- tar
- find, locate