

Préparation LPI

Exam 101

103.2 Filtrés et flux de texte

- Poids : 3
- Envoyer des flux de texte au travers de filtre pour en modifier la sortie
- Les commandes sont regroupées dans le paquetage textutils

- Traitement des fichiers texte à l'aide de commandes simples
- Ces commandes agissent:
 - Sur le contenu d'un fichier
 - Sur un flux
- Grosse utilisation des méta-caractères de redirection dans l'utilisation des commandes de manipulation des fichiers textes
 - Cf module 103.4 pour plus de détails
- Flux standards :
 - stdin : flux standard d'entrée (clavier)
 - stdout : flux standard de sortie (écran)
 - stderr : flux standard des erreurs (écran)

- L'option « - », disponible sur la plupart de ces commandes, indique que l'entrée de la commande vient de stdin

- `cat` : concaténation de fichiers (concatenate)
`cat [OPTION] ... [FILE] ...`
- Affiche le résultat sur `stdin`
- Avec un paramètre : affiche le contenu du fichier

```
$ cat liste1
Pommes
Paires
Anannas
```

- Avec plusieurs paramètres : concatène l'ensemble des fichiers

```
$ cat liste1      $ cat liste2      $ cat liste3
Pommes           Javel             Céréales
Paires           Tire-bouchon     Huile
Anannas          Crème
                 Beurre
```

```
$ cat liste2 liste3 liste1
Javel
Tire-bouchon
Céréales
Huile
Crème
Beurre
Pommes
Paires
Anannas
```

- Avec aucun paramètre : attend son entrée depuis stdin
- Utilisé habituellement avec une redirection pour créer un fichier texte
- Utiliser Ctrl-d pour signaler à la commande cat la fin du fichier

\$ cat > liste4

Chaussure

Imper

Bérêt

- `tac` : commande inverse de `cat`

\$ cat liste1

Pommes

Poires

Anannas

\$ tac liste1

Anannas

Poires

Pommes

- Options de la commande cat :
 - -v : affiche les caractères non imprimables
 - -E : affiche \$ à chaque fin de ligne
 - -T : affiche ^I à la place des tabulations
 - -n : numérote les lignes
 - -A : équivalent à vET

```
$ cat liste1
Pommes
  Poires
Anannas
```

```
$ cat -A liste1
Pommes$
^IPoires$
Anannas$
```

```
$ cat -n liste1
1 Pommes
2  Poires
3Anannas
```

- `split` : commande duale de `cat`. Eclate le fichier en plusieurs parties

```
split [OPTION]... [INPUT [PREFIX]]
```

- Par défaut : fichiers résultats nommés « x » + « aa », « ab »,... « az », « ba », « bb »
- Taille par défaut des fichiers résultats : 1000 octets
- Options de la commande `split` :
 - `-b taille`: répartition en fichiers de taille « taille » (b, kB, MB, GB)
 - `-l nbe-lignes` : répartition en fichiers de nbe-lignes


```
$ split -l 3 ficsplit1  
[franck@port-135 lpi-tests]$ ls x*  
xaa xab xac xad
```

```
$ split -b 10 ficsplit1 bouts  
[franck@port-135 lpi-tests]$ ls -l bouts*  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsaa  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsab  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsac  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsad  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsae  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsaf  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsag  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsah  
-rw-rw-r-- 1 franck franck 10 févr. 1 22:38 boutsai
```

- `od` : affiche le contenu du fichier en octal (octal dump)

```
od [OPTION]... [FILE]...
```

```
od [-abcdfilosx]... [FILE] [[+]OFFSET[.] [b]]
```

```
od --traditional [OPTION]... [FILE]
```

```
[[+]OFFSET[.] [b] [+] [LABEL][.] [b]]
```

- Options de la commande `od` :
 - `-t TYPE`: sélectionne le format d'affichage des octets
 - `-c` : affichage en caractères ASCII (équivalent `-t c`)
 - `-t a` : affiche le nom des caractères
 - `-A RADIX` : sélectionne le mode d'affichage du positionnement des octets (décimal, octal, aucun) visibles en début de chaque ligne. A 0 par défaut

\$ od liste1

0000000 020040 067520 066555 071545 004412 067520 071151 071545

0000020 040412 060556 067156 071541 000012

0000031

\$ ll liste1

-rw-rw-r-- 1 franck franck 25 févr. 1 22:15 liste1

[franck@port-135 lpi-tests]\$ od -A d liste1

0000000 020040 067520 066555 071545 004412 067520 071151 071545

0000016 040412 060556 067156 071541 000012

0000025

\$ od -t c liste1

0000000 P o m m e s \n \t P o i r e s

0000020 \n A n a n n a s \n

0000031

\$ od -t a liste1

0000000 sp sp P o m m e s nl ht P o i r e s

0000020 nl A n a n n a s nl

0000031

- `expand` : transforme les tabulations en espaces
`expand [OPTION]... [FILE]...`
- Options de la commande `expand` :
 - `-i`: pas de conversion si la tabulation suit un caractère non vide
 - `-t num` : remplacement d'une tabulation par « num » espaces au lieu de 8 par défaut

\$ cat liste1

 Pommes

 Paires

Anannas

\$ expand liste1

 Pommes

 Paires

Anannas

\$ expand -t1 liste1

 Pommes

 Paires

Anannas

\$ expand -t2 liste1

 Pommes

 Paires

Anannas

- unexpand : transforme les espaces en tabulation
unexpand [OPTION]... [FILE]...
- Options de la commande unexpand :
 - -t num : remplacement de « num » espaces pour une tabulation
- Attention : deux espaces minimum pour conversion en tabulation

```
$ cat liste2
```

```
Javel Déboucheur
```

```
Tire-bouchon Serpillière
```

```
$ od -a liste2
```

```
0000000 sp J a v e l sp sp sp D C ) b o u c
```

```
0000020 h e u r nl sp sp T i r e - b o u c
```

```
0000040 h o n sp sp sp sp sp sp sp S e r p i
```

```
0000060 l l i C ( r e nl
```

```
0000070
```

```
$ unexpand -t1 liste2 | od -a
```

```
0000000 sp J a v e l ht ht ht D C ) b o u c
```

```
0000020 h e u r nl ht ht T i r e - b o u c
```

```
0000040 h o n ht ht ht ht ht ht ht S e r p i
```

```
0000060 l l i C ( r e nl
```

```
0000070
```

```
$ unexpand -t2 liste2 | od -a
```

```
0000000 sp J a v e l ht sp D C ) b o u c h
```

```
0000020 e u r nl ht T i r e - b o u c h o
```

```
0000040 n ht ht ht ht S e r p i l l i C ( r
```

```
0000060 e nl
```

```
0000062
```

- `tr` : transformation de caractères (translate)
`tr [OPTION]... SET1 [SET2]`
- Options de la commande `unexpand` :
 - `-d` : supprime le caractère dans SET
 - `-s` : supprime la répétition de caractère
- Fonctionne sur un flux et pas sur des fichiers
- `man tr` pour avoir la liste exhaustive des chaînes de caractères utilisables
- Très utile pour changer les formats de fichiers (Mac, Linux, Windows)


```
$ cat liste2
```

```
Javel Déboucheur
```

```
Tire-bouchon      Serpillière
```

```
$ tr -s ' ' < liste2
```

```
Javel Déboucheur
```

```
Tire-bouchon Serpillière
```

```
$ cat fic-unix
```

```
Ligne1
```

```
Ligne2
```

```
$ od -a fic-unix
```

```
0000000 L i g n e 1 nl L i g n e 2 nl
```

```
0000016
```

```
$ cat fic-unix | tr '\n' '\r' | od -a
```

```
0000000 L i g n e 1 cr L i g n e 2 cr
```

```
0000016
```

- `wc` : nbre de lignes, mots, octets (word count)

```
wc [OPTION] ... [FILE] ...
```

- Options de la commande `wc` :
 - `-c` : compte les octets
 - `-m` : compte les caractères
 - `-l` : compte les lignes (car nl)
 - `-w` : compte les mots

```
$ wc -l liste1
```

```
3 liste1
```

```
$ wc -c liste1
```

```
25 liste1
```

```
$ wc -m liste1
```

```
25 liste1
```

- `head` : affiche les premières lignes (10 premières par défaut)

```
head [OPTION] ... [FILE] ...
```

- Options de la commande `head` :
 - `-n` : affiche les `n` premières lignes

```
$ head -n 5 ficsplit1
```

```
Ligne 1
```

```
Ligne 2
```

```
Ligne 3
```

```
Ligne 4
```

```
Ligne 5
```

```
$ head -5 ficsplit1
```

```
Ligne 1
```

```
Ligne 2
```

```
Ligne 3
```

```
Ligne 4
```

```
Ligne 5
```

- `tail` : affiche les dernières lignes (10 dernières par défaut)
`tail [OPTION]... [FILE]...`
- Options de la commande `tail` :
 - `-f` : affiche en continu (sortie avec Ctrl-c)
 - `-f` et `-pid` : termine quand le processus se termine
 - `-n` : affiche les n dernières lignes
 - `--line=+n` : affiche les dernières lignes à partir de la nième

```
$ tail -n 2 ficsplit1
```

```
Ligne 10
```

```
Ligne 11
```

```
$ tail --line=+6 ficsplit1
```

```
Ligne 6
```

```
Ligne 7
```

```
Ligne 8
```

```
Ligne 9
```

```
Ligne 10
```

```
Ligne 11
```

- `pr` : préparation d'un texte pour impression. Pagination et en-têtes

```
pr [OPTION] ... [FILE] ...
```

- Options de la commande `pr` :
 - `-h 'texte'`: affiche en en-tête un texte en argument au lieu de du nom de fichier
 - `-m` : fusionne plusieurs fichiers et représentation en colonnes

```
$ pr ficsplit1
```

```
2010-02-01 22:29
```

```
ficsplit1
```

```
Page 1
```

```
Ligne 1
```

```
Ligne 2
```

```
Ligne 3
```

\$ pr -h 'Mes courses' -m liste1 liste2

2010-02-02 22:48

Mes courses

Page 1

Pommes
Poires
Anannas

Javel
Déboucheur
Tire-bouchon
Serpillière

- `nl` : Numérote les lignes

```
nl [OPTION] ... [FILE] ...
```

- Options de la commande `nl` :

→

```
$ nl ficsplit1
 1Ligne 1
 2Ligne 2
 3Ligne 3
 4Ligne 4
 5Ligne 5
 6Ligne 6
 7Ligne 7
 8Ligne 8
 9Ligne 9
10  Ligne 10
11  Ligne 11
```


- `fmt` : formatage de texte

```
fmt [-WIDTH] [OPTION] ... [FILE] ...
```

- Options de la commande `fmt` :

→ `-w n`: formate en n colonnes

```
$ pr -h 'Mes courses' -m liste1 liste3 | fmt -w 20
```

```
2010-02-02 22:57
```

```
Mes courses
```

```
Page 1
```

```
Pommes
```

```
Céréales Poires
```

```
Huile Anannas
```

```
Crème
```

```
Beurre
```

- `sort` : tri de lignes de texte

```
sort [OPTION]... [FILE]...
```

- Options de la commande `sort` :

- `-f`: Pas de casse

- `-n`: tri numérique

- `-r` : tri inverse

- `-k POS` : Utilisation du champ situé à la colonne POS (numérotation à partir de 1)

- `-t SEP` : utilise le caractère SEP comme séparateur de champs

- `LC_COLLATE` : var qui contient le jeu de car utilisé pour le tri \$ locale

...

```
LC_COLLATE="fr_FR.UTF-8"
```

```
$ cat liste1
```

```
Pommes
```

```
Poires
```

```
Anannas
```

```
$ sort liste1
```

```
Anannas
```

```
Poires
```

```
Pommes
```

```
$ nl liste1 | sort
```

```
1Pommes
```

```
2Poires
```

```
3Anannas
```

```
$ nl liste1 | sort -k2
```

```
3Anannas
```

```
2Poires
```

```
1Pommes
```

- `uniq` : Supprime les lignes identiques
`uniq [OPTION]... [INPUT [OUTPUT]]`
- Options de la commande `uniq` :
 - `-c` : préfixe les lignes dupliquée avec le nombre d'occurences
 - `-i`: pas de casse
- A utiliser conjointement avec la commande `sort`

\$ cat dup

Ligne1

Ligne4

Ligne2

Ligne3

Ligne4

Ligne4

Ligne2

\$ sort dup

Ligne1

Ligne2

Ligne2

Ligne3

Ligne4

Ligne4

Ligne4

\$ sort dup |uniq

Ligne1

Ligne2

Ligne3

Ligne4

- `cut` : sélectionne des champs particuliers du fichier
`cut OPTION... [FILE]...`
- Options de la commande `cut` :
 - `-d DELIM` : Utiliser `DELIM` à la place de la tabulation
 - `-f num`: sélectionne le champ à la position `num`

```
$ nl liste1  
 1Pommes  
 2Paires  
 3Anannas
```

```
$ nl liste1 | cut -f2  
Pommes  
Paires  
Anannas
```

- `paste` : fusionne lignes à lignes les fichiers

```
paste [OPTION]... [FILE]...
```

- Options de la commande `paste` :

→ `-d DELIM` : Utiliser `DELIM` à la place de la tabulation

```
$ cat liste*
```

```
Pommes
```

```
Poires
```

```
Anannas
```

```
Javel
```

```
Déboucheur
```

```
Tire-bouchon
```

```
Serpillière
```

```
Céréales
```

```
Huile
```

```
Crème
```

```
Beurre
```

```
Chaussure
```

```
Imper
```

```
Bérêt
```

```
$ paste liste1 liste2 liste3
```

```
Pommes Javel Céréales
```

```
Poires Déboucheur Huile
```

```
Anannas Tire-bouchon Crème
```

```
Serpillière Beurre
```

- `join` : permet de faire une jointure entre deux fichiers sur la base d'un champ commun.
Affiche la ligne sur `stdout` correspondant à chaque jointure. Par défaut les champs sont séparés par des espaces.

```
join [OPTION]... FILE1 FILE2
```

- Important : les deux fichiers doivent être triés sur le champ de jointure (avec la commande `sort`)

- Options de la commande join :
 - -i : ignore la casse
 - -t char : définit le caractère séparateur des champs (espace par défaut)
 - -1 champ : jointure sur le champ « champ » du fichier 1
 - -2 champ : jointure sur le champ « champ » du fichier 2
 - -a filename : Affiche les lignes du fichier filename (1 ou 2) qui ne satisfont pas à la condition de jointure
 - -v filename : idem option -a mais supprime les jointures correctes

```
$ cat table1  
user1 5437 Jean Bon  
user2 8756 Henri Cochet  
user3 9845 Sarah Croche  
user4 3423 Nico Rette
```

```
$ cat table2  
5437 Tennis  
9845 Kayak
```

```
$ sort -n -k2 table1 > table1.sort
```

```
$ cat table1.sort  
user4 3423 Nico Rette  
user1 5437 Jean Bon  
user2 8756 Henri Cochet  
user3 9845 Sarah Croche
```

```
$ sort -n -k1 table2 | join -1 2 -2 1 table1.sort -  
5437 user1 Jean Bon Tennis  
9845 user3 Sarah Croche Kayak
```

```
$ sort -n -k1 table2 | join -1 2 -2 1 -a 1 table1.sort -  
3423 user4 Nico Rette  
5437 user1 Jean Bon Tennis  
8756 user2 Henri Cochet  
9845 user3 Sarah Croche Kayak
```

```
$ sort -n -k1 table2 | join -1 2 -2 1 -v 1 table1.sort -  
3423 user4 Nico Rette  
8756 user2 Henri Cochet
```

- `sort` : permet trier les lignes d'un fichier
Beaucoup d'options. Possibilité de trier sur plusieurs champs

```
sort [OPTION]... [FILE]...
```

- Options :

- `-d` : ordre alphabétique
- `-f` : ignore la casse
- `-n` : trie sur un ordre numérique
- `-r` : inverse le tri
- `-kPOS1[,POS2]` : la clé de tri commence à POS1 et se termine à POS2 (n° de champ commence à 1)
- `-o fichier` : fichier résultat du tri
- `-t char` : caractère séparateur des champs

```
$ cat table1
```

```
user1 5437 Jean Bon  
user42 6765 Arthur Eiffel  
user2 8756 Henri Cochet  
user54 3487 Sacha Touille  
user3 9845 Sarah Croche  
user4 3423 Nico Rette
```

```
$ sort -k4 table1
```

```
user1 5437 Jean Bon  
user2 8756 Henri Cochet  
user3 9845 Sarah Croche  
user42 6765 Arthur Eiffel  
user4 3423 Nico Rette  
user54 3487 Sacha Touille
```

```
$ sort -nk2 table1
```

```
user4 3423 Nico Rette  
user54 3487 Sacha Touille  
user1 5437 Jean Bon  
user42 6765 Arthur Eiffel  
user2 8756 Henri Cochet  
user3 9845 Sarah Croche
```

- `sed` : Editeur de flux (stream editor)
Permet la transformation de fichiers textes sur le flux d'entrée (dans un tube) et affiche le résultat sur stdout (pas de modification du fichier d'origine)
`sed [OPTION]... {script-seulement-si-pas-d-autre-script} [fichier-d-entrée]..`
- Sed utilise les expressions régulières pour rechercher les lignes à modifier, rechercher les motifs à modifier et remplacer le texte correspondant
- Options
 - `-e script` : ajouter le script aux commandes
 - `-f fichier` : le script est un fichier externe

- Syntaxe d'une commande sed
- `[adresse1 [, adresse2]] fonction [argument]`
`[/regexp1/ [, /regexp2/]] fonction [argument]`
- Adresse (permet de sélectionner un groupe de ligne)
 - N° de ligne
 - \$: dernière ligne
 - 1 : première ligne
- `/regexp1/` et `/regexp2/`: expressions régulières qui permettent de sélectionner un ensemble de lignes sur la base d'un motif
- Utilisation de la virgule pour définir un ensemble
- Utilisation du point virgule pour séparer les commandes

- Syntaxe d'une commande sed ...
- `[adresse1 [, adresse2]] fonction [argument]`
`[/regexp1/ [, /regexp2/]] fonction [argument]`
- Fonction
 - d : suppression de la ligne
 - s : substitution de texte à l'intérieur de la ligne
s/regexp/remplacement/modificateur
- Modificateur
 - g : remplacement sur l'ensemble de la ligne
- Options
 - -i : applique les modifications dans le fichier d'entrée
Attention : ne pas rediriger la sortie de sed dans le fichier d'entrée. Cela vide le fichier !

```
$ cat table4
```

```
M Franck Duprat
```

```
Mme Catherine Moutarde
```

```
M Jean-louis Crottin
```

```
Mme Solène Chamon
```

```
$ sed -e 's/^M /Monsieur /' table4
```

```
Monsieur Franck Duprat
```

```
Mme Catherine Moutarde
```

```
Monsieur Jean-louis Crottin
```

```
Mme Solène Chamon
```

```
$ sed -e '1,2s/^M /Monsieur /' table4
```

```
Monsieur Franck Duprat
```

```
Mme Catherine Moutarde
```

```
M Jean-louis Crottin
```

```
Mme Solène Chamon
```


- `col` : élimine les sauts de ligne arrière
`cut OPTION... [FILE]...`
- Intérêt pour mettre en forme les pages de manuel

