



UNIX

Cours 11 : Commandes de filtres

Filipe Vasconcelos¹

¹ESME, Lille, filipe.vasconcelo@esme.fr

- 1 Les filtres**
- 2 cut**
- 3 head**
- 4 tail**
- 5 tr**
- 6 uniq**
- 7 sed**
- 8 Quelques commandes en vrac**



AAV2

Reconnaitre, décrire et utiliser les commandes de bases de l'interpréteur de commandes bash ou trouver efficacement rapidement de la documentation sur ces commandes.

AAV3

Utiliser et enchaîner, au moyen de tubes, les principales commandes de filtre (grep, cut, head, tail ...) pour manipuler des flux de données qu'ils proviennent ou non dans des fichiers



1 Les filtres

2 cut

3 head

4 tail

5 tr

6 uniq

7 sed

8 Quelques commandes en vrac



Les filtres

- ❑ Un filtre est une commande qui lit les données sur l'entrée standard, effectue des traitements sur les lignes reçues et écrit le résultat sur la sortie.
- ❑ Le filtre permet de réaliser des traitements sur des textes.
- ❑ Le filtre peut être utilisé avec des fichiers ou appliqués sur les sorties standards via les tubes.



grep : « g/re/p »

- ❑ Usage : Cette commande cherche et affiche toutes les lignes qui satisfont un motif (chaine de caractère ou expression régulière).
- ❑ Syntaxe : grep [options] <motif> <fichier>
- ❑ Résultat : Lignes du fichier contenant le motif
- ❑ Remarque : grep -E est équivalent à egrep



① Les filtres

② cut

③ head

④ tail

⑤ tr

⑥ uniq

⑦ sed

⑧ Quelques commandes en vrac



NOM

`cut` - Supprimer une partie de chaque ligne d'un fichier.

DESCRIPTION

`cut` affiche une partie de chaque ligne de chacun des fichiers mentionnés, ou de l'entrée standard si aucun fichier n'est indiqué. Un nom de fichier '-' correspond également à l'entrée standard.

La partie affichée est sélectionnée par les options présentées ci-dessous.

- b, --bytes liste_d_octets
- c, --characters liste_de_caractères
- f, --fields liste_de_champs



```
cat compter.txt
```

```
cat compter.txt
```

```
un one um ichi  
deux two dois ni  
trois three très san  
quatre four quattro yon  
cinq five cinco go  
six six seis roku
```

```
cut -c<liste>
```

```
cut -c1-4 compter.txt
```

```
un o  
deux  
troi  
quat  
cinq  
six
```



```
cut -d<delimiteur> -f<champs>
```

```
cut -d ' ' -f 1,3 compter.txt
```

```
un um  
deux dois  
trois très  
quatre quattro  
cinq cinco  
six seis
```



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



NOM

head - Afficher le début des fichiers

SYNOPSIS

head [OPTION] ... [FICHIER] ...

DESCRIPTION

Afficher les 10 premières lignes de chaque FICHIER sur la sortie standard.

-c, --bytes=[-]K

afficher les K premiers octets de chaque fichier

-n, --lines=[-]K

afficher les K premières lignes au lieu des 10 premières



```
head compter2.txt
```

un one um ichi
deux two dois ni
trois three très san
quatre four quattro yon
cinq five cinco go
six six seis roku
sept seven sete nana
huit eight oito hachi
neuf nine nove kyū
dix ten dez jū

```
head -n 3 compter2.txt
```

un one um ichi
deux two dois ni
trois three très san



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



NOM

tail - Afficher la dernière partie de fichiers

SYNOPSIS

tail [OPTION] ... [FICHIER] ...

DESCRIPTION

Afficher les 10 dernières lignes de chaque FICHIER sur la sortie standard.

-c

afficher les K derniers octets ;

-f

afficher les données ajoutées lorsque le fichier grossit ;

-n, --lines=K

afficher les K dernières lignes, au lieu des 10 dernières

-s, --sleep-interval=N

avec -f, attendre environ N secondes (1,0 par défaut) entre les itérations



```
tail compter2.txt
```

trois three trois san
quatre four cuatro yon
cinq five cinco go
six six seis roku
sept seven sete nana
huit eight oito hachi
neuf nine nove kyū
dix ten dez jū
onze eleven onze jūichi
douze twelve doze jūni

```
tail -n2 compter2.txt
```

onze eleven onze jūichi
douze twelve doze jūni



Utiliser tail pour le « monitoring »

```
tail -f
```

```
simulation > log &
tail -f log
```

```
iter : 325      value = 18040  Elapsed Time : 1101  (s)
iter : 326      value = 3281   Elapsed Time : 1103  (s)
iter : 327      value = 7102   Elapsed Time : 1113  (s)
iter : 328      value = 5786   Elapsed Time : 1116  (s)
iter : 329      value = 13874  Elapsed Time : 1119  (s)
iter : 330      value = 19195  Elapsed Time : 1121  (s)
iter : 331      value = 134    Elapsed Time : 1123  (s)
iter : 332      value = 32008  Elapsed Time : 1143  (s)
```



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



NOM

tr - Convertir ou éliminer des caractères

SYNOPSIS

tr [OPTION]... ENSEMBLE1 [ENSEMBLE2]

DESCRIPTION

Convertir, compresser et/ou éliminer des caractères lus depuis l'entrée standard, les écrire sur la sortie standard.

-c, -C, --complement

utiliser le complément de l'ENSEMBLE1

-d, --delete

éliminer les caractères de l'ENSEMBLE1, ne pas convertir

-s, --squeeze-repeats

remplacer chaque séquence de caractères répétés en entrée et qui apparaît dans l'ENSEMBLE1 par une seule occurrence de ce caractère



```
cat exemple_cours_tr.txt
```

```
9143 777 258
```

Exemple tr (GitHub)

```
cat exemple_cours_tr.txt|tr 12345789 'achsm>db'
```

```
bash >>> cmd
```



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



NOM

uniq - Signaler ou éliminer les lignes répétées

SYNOPSIS

uniq [OPTION]... [ENTRÉE [SORTIE]]

DESCRIPTION

Filtrer les lignes successives identiques du fichier d'ENTRÉE (ou de l'entrée standard), écrire dans le fichier de SORTIE (ou vers la sortie standard).

Sans options, les lignes identiques sont fusionnées avec la première occurrence.

-c, --count

préfixer les lignes par le nombre d'occurrences

-d, --repeated

n'afficher que les lignes dupliquées, une par groupe

-i, --ignore-case

ignorer les différences de casse



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



La commande sed est un éditeur de flux de texte (« stream editor »). Elle agit sur un flux de texte avec des commandes (interne à sed)

Prenons le fichier suivant (github)

```
cat distrib.txt
```

```
Linux
Solaris
Ubuntu
Fedora
Debian
RedHat
Mandriva
<ligne blanche>
```

par défaut sed ne fait rien

```
cat distrib.txt | sed
```

commande d (delete) efface la première ligne

```
cat distrib.txt | sed '1d'
```

Solaris
Ubuntu
Fedora
Debian
RedHat
Mandriva
<ligne blanche>

commande d (delete) efface plusieurs lignes

```
cat distrib.txt | sed '1,4d'
```

Debian
RedHat
Mandriva
<ligne blanche>



Substitution

```
sed s/a/u/ distrib.txt (première occurence)
```

Linux
Soloris
Ubuntu
Fedoru
Debiun
RedHut
Mundriva

Substitution

```
sed s/a/u/g distrib.txt (global)
```

Linux
Soloris
Ubuntu
Fedoru
Debiun
RedHut



Effacer toutes les lignes commentaires // et les lignes vides

```
sed /^$/d lignes_vides.txt
```

```
sed -e '/^$/d' -e '/^*/d' code.c
```

```
#include <stdio.h>
int main(void)
{
    int n = 10;
    for (int i =0;i<n;i++)
    {
        printf("i = %d",i)
    }
}
```



- ① Les filtres
- ② cut
- ③ head
- ④ tail
- ⑤ tr
- ⑥ uniq
- ⑦ sed
- ⑧ Quelques commandes en vrac



- ❑ sort : trier les lignes d'un fichier
- ❑ uniq : affiche en supprimant les lignes répétées utilité : sort <fichier> | uniq
- ❑ comm : comparer lignes par lignes des fichiers triés
- ❑ diff : affiche les différences entre deux fichiers
- ❑ basename : supprimer des suffixes
- ❑ pr : formate la sortie pour l'impression
- ❑ awk : langage de traitement de fichier ligne par lignes (la syntaxe est inspirée du C)
- ❑ ...



Exercice 1

Le fichier `nat2019.csv` répertorie tous les prénoms donnés en France depuis 1900. Combien de prénoms distincts ont été donnés depuis 1900 ? Commandes qui peuvent être utilisés : `cat`, `sort`, `uniq`, `cut` et `wc`

Exercice 2

À partir du fichier des emails (`liste_inge1`) des étudiants de la promo 2020-2021 des Ingé1 de Lille, produire un fichier à deux colonnes avec "Prénom Nom" (avec les majuscules au début de chaque chaînes) Commandes qui peuvent être utilisés : `cat`, `sort`, `cut` et `sed`

