

Vous avez déjà coupé du texte dans un éditeur de texte, non ?

La commande `cut` vous propose de faire cela au sein d'un fichier afin de conserver uniquement une partie de chaque ligne.

Couper selon le nombre de caractères

Par exemple, si vous souhaitez conserver uniquement les caractères 2 à 5 de chaque ligne du fichier, vous taperez :

```
$ cut -c 2-5 noms.txt
```

console

```
ran  
arce  
lber  
ean  
tép  
atri  
ince  
onat
```



`cut` a quelques soucis avec les mots contenant des accents. Comme vous pouvez le voir, certains mots ici coupés ont quatre lettres (comme prévu) alors que d'autres en ont trois.

Ceci est dû à l'encodage des caractères, aux accents. La commande `cut` se base sur le nombre d'octets, et comme nous l'avons vu plus tôt, celui-ci n'est pas forcément égal au nombre de caractères. À l'heure actuelle on ne peut rien y faire, c'est la commande `cut` qui devra être mise à jour par les programmeurs.

Pour conserver du 1er au 3ème caractère :

```
$ cut -c -3 noms.txt
```

console

```
Fra  
Mar  
Alb  
Jea  
St  
pat  
Vin  
jon
```

Comme vous pouvez le voir, si on ne met pas de chiffre au début, `cut` comprend que vous voulez parler du premier caractère.

De même, pour conserver du 3ème au dernier caractère :

```
$ cut -c 3- noms.txt
```

console

```
ançois  
rce  
bert  
an  
éphane  
trice  
ncent
```

nathan

Là encore, pas besoin de donner le numéro du dernier caractère, la commande `cut` comprend comme une grande qu'elle doit couper jusqu'à la fin.

Couper selon un délimiteur

Faisons maintenant quelque chose de bien plus intéressant. Plutôt que de s'amuser à compter le nombre de caractères, nous allons travailler avec ce que l'on appelle un **délimiteur**.

Prenons un cas pratique : les fichiers CSV (*Comma Separated Values*). Ce sont des fichiers dont les valeurs sont séparées par des virgules. Notez qu'Excel utilise plutôt le point-virgule comme séparateur, mais le principe reste le même.) Vous en avez peut-être déjà vu : ils sont générés par des tableurs — tels qu'Excel ou Calc — pour faciliter l'échange et le traitement de données.

Imaginons que vous ayez une (petite) classe et que vous rendez les notes du dernier contrôle. Vous avez fait un joli tableur et vous avez enregistré le document au format CSV. Le fichier sur lequel nous allons nous baser sera le suivant :

```
Fabrice,18 / 20,Excellent travail
Mathieu,3 / 20,Nul comme d'hab
Sophie,14 / 20,En nette progression
Mélanie,9 / 20,Allez presque la moyenne !
Corentin,11 / 20,Pas mal mais peut mieux faire
Albert,20 / 20,Toujours parfait
Benoît,5 / 20,En grave chute
```

console

Comme le nom CSV l'indique, les virgules servent à séparer les colonnes. Ces dernières contiennent, dans l'ordre :

- le prénom ;
- la note ;
- un commentaire.

C'est un exemple tout à fait fictif, bien entendu. ;-)

Créez, avec le texte que je viens de vous donner, un nouveau fichier que vous appellerez par exemple `notes.csv`.

Imaginons que nous souhaitions extraire de ce fichier la liste des prénoms. Comment nous y prendrions-nous ? On ne peut pas utiliser la technique qu'on vient d'apprendre car les prénoms ne font pas tous la même longueur. Nous allons donc nous servir du fait que nous savons que la virgule sépare les différents champs dans ce fichier.

Vous allez avoir besoin d'utiliser deux paramètres :

- `-d` : indique quel est le délimiteur dans le fichier ;
- `-f` : indique le numéro du ou des champs à couper.

Dans notre cas, le délimiteur qui sépare les champs est la virgule. Le numéro du champ à couper est 1 (c'est le premier).

Testez donc ceci :

```
$ cut -d , -f 1 notes.csv
```

console

```
Fabrice
Vincent
Sophie
Mélanie
Corentin
Albert
Benoît
```

C'est pas beau, ça ? :-)

Après le `-d`, nous avons indiqué quel était le délimiteur (à savoir la virgule « , »).

Après le `-f`, nous avons indiqué le numéro du champ à conserver (le premier).

Si nous voulons juste les commentaires :

```
$ cut -d , -f 3 notes.csv
```

console

```
Excellent travail
Nul comme d'hab
En nette progression
Allez presque la moyenne !
Pas mal mais peut mieux faire
Toujours parfait
En grave chute
```

Pour avoir les champs n°1 et n°3 (le prénom et le commentaire) :

```
$ cut -d , -f 1,3 notes.csv
```

console

```
Fabrice,Excellent travail
Vincent,Nul comme d'hab
Sophie,En nette progression
Mélanie,Allez presque la moyenne !
Corentin,Pas mal mais peut mieux faire
Albert,Toujours parfait
Benoît,En grave chute
```



De même, il est possible de conserver toute une série de champs avec le tiret comme tout à l'heure : `cut -d , -f 2-4 notes.csv` a pour effet de conserver les champs n°

2, 3 et 4.

D'autre part, `cut -d , -f 3- notes.csv` conserve les champs du n°3 jusqu'à la fin.

Vous êtes bien obligés d'admettre que, quand on sait bien s'en servir, la console de Linux peut vous permettre d'effectuer des opérations vraiment puissantes que vous ne pensiez même pas pouvoir faire aussi simplement jusqu'à présent. ;-)

En résumé

- `grep` est une commande couramment utilisée pour rechercher un mot dans un fichier.
- On peut utiliser des expressions régulières, un système complexe mais puissant, pour effectuer des recherches précises. On fait dans ce cas appel à la commande `egrep`.
- `sort` trie des lignes de texte par ordre alphabétique. Le paramètre `-n` permet de trier par ordre numérique.
- `wc` compte le nombre de lignes, de mots et d'octets dans un fichier.
- `uniq` supprime les doublons d'un fichier.
- `cut` coupe une partie d'un fichier.