

# INTRODUCTION SYSTÈME

## UNE INTRODUCTION AU SYSTÈME D'EXPLOITATION LINUX

Guillaume Santini

guillaume.santini@iutv.univ-paris13.fr  
IUT de Villeteuse

13 février 2012

Partie #3

# PLAN

- ❶ PROPRIÉTÉ ET DROITS SUR LES FICHIERS
  - Propriété des fichiers
  - Les droits sur les fichiers et répertoires
- ❷ ARCHIVES ET COMPRESSION DES FICHIERS
- ❸ ÉDITION ET MANIPULATION DE FICHIERS

# PROPRIÉTÉ DES FICHIERS

## IDENTIFICATIONS DES UTILISATEURS DANS UN ENVIRONNEMENT MULTI-UTILISATEURS

**UID (User IDentifier)** numéro unique associé à chaque utilisateur lors de la création de son compte.

**GID (Group IDentifier)** numéro unique d'un groupe d'utilisateurs. Chaque utilisateur peut être associé à un ou plusieurs groupes.

## UTILITÉ

- Chaque fichier (ou répertoire) et chaque processus du système est associé à un utilisateur : cet utilisateur est le propriétaire du fichier (ou répertoire) ou celui qui a lancé le processus.
- Être propriétaire d'un fichier ou d'un processus confère des droits sur ceux-ci.

## CONNAITRE L'IDENTITÉ DU PROPRIÉTAIRE D'UN PROCESSUS OU D'UN FICHIER

- Les commandes `top` et `ps` affichent le nom du propriétaire des processus.
- La commande `ls` avec l'option `-l` affiche le nom et le groupe du propriétaire d'un fichier ou d'un répertoire.

# ls(ter)

## SYNTAXE

```
ls -l <source>
```

## DESCRIPTION

- Affiche le contenu d'un répertoire en format long.
- Le format long donne le nom du propriétaire et son groupe, ainsi que les droits des différentes classes d'utilisateurs sur les fichiers et répertoires.

## EXEMPLE D'UTILISATION:

```
chez_moi/..... Répertoire Courant
├── public_html/
│   └── index.html
└── astronomie.txt
```

```
[ login@localhost ~ ] ls -l
total 32
drwxr-xr-x 2 santini ensinfo 4096 20 jui 15 :50 public_html
-rw-r--r-- 1 santini ensinfo  25 20 jui 15 :49 telluriques.txt
```

Ici, le nom de l'utilisateur est **santini**, nom du groupe est **ensinfo** et les droits sont colorés en **vert**.

# LES DROITS SUR LES FICHIERS ET RÉPERTOIRES

## 3 CATÉGORIES D'UTILISATEURS

-	r W X	r W X	r W X
Type de Fichier	Doits du propriétaire ( <b>U</b> ser)	Doits du groupe ( <b>G</b> roup)	Doits des autres ( <b>O</b> ther)

### TYPES DE FICHIERS

Types	
-	Fichier ordinaire
d	Répertoire
l	lien symbolique

### DROITS/PERMISSIONS

	Fichier	Répertoire
r ( <b>R</b> ead)	lire	lister le contenu
w ( <b>W</b> rite)	écrire et modifier	modifier le contenu
x ( <b>eX</b> ecute)	exécution	traverser

### TYPES D'UTILISATEURS

Cible	
u ( <b>U</b> ser)	Propriétaire du fichier/répertoire
g ( <b>G</b> roup)	Membre du même groupe que le propriétaire
o ( <b>O</b> ther)	Tous les autres
a ( <b>A</b> ll)	Tous les utilisateurs (réunion de 'u', 'g' et 'o').

# chmod

## SYNTAXE

```
chmod droit fichier
```

## DESCRIPTION

- Modifie les droits et permissions accordés par le propriétaire aux différents utilisateurs du système.

## EXEMPLE D'UTILISATION:

Retire au propriétaire le droit d'écriture sur le fichier `cv_2011.pdf`.

```
[ login@localhost ~ ] chmod u-w cv_2011.pdf
```

Retire aux utilisateurs qui ne sont ni le propriétaire ni membre de son groupe les droits de lecture, d'écriture et d'exécution.

```
[ login@localhost ~ ] chmod o-rwx listing.bash
```

Ajoute au propriétaire et aux membres de son groupe le droit d'exécution sur le fichier `listing.bash`.

```
[ login@localhost ~ ] chmod ug+x listing.bash
```

Ajoute à tous les utilisateurs, tous les droits.

```
[ login@localhost ~ ] chmod a+rx listing.bash
```

# chmod(bis)

## DESCRIPTION

Il existe plusieurs notations des droits.

- La notation alphanumérique : (u)go(a) (+/-) (rwx)
- La notation octale :

Droit	---	--x	-w-	-wx	r--	r-x	rw-	rwx
Binaire	000	001	010	011	100	101	110	111
Octale	0	1	2	3	4	5	6	7

Alphabétique	r	w	x	r	-	x	-	-	x
Binaire	1	1	1	1	0	1	0	0	1
Octale	7			5			1		

## EXEMPLE D'UTILISATION:

Alph.	Oct.	Alph.	Oct.
--- --- ---	000	rw- --- ---	700
rw- --- ---	600	rw- r-x r-x	755
rw- r-- r--	644	rw- rw- rw-	777
rw- rw- rw-	666		

```
[ login@localhost ~ ] chmod 700 dir_parano
```

```
[ login@localhost ~ ] chmod 644 fichier_pub
```

# PLAN

- 1 PROPRIÉTÉ ET DROITS SUR LES FICHIERS
- 2 ARCHIVES ET COMPRESSION DES FICHIERS
  - Compression
  - Archivage
- 3 ÉDITION ET MANIPULATION DE FICHIERS



# COMPRESSION

## RÉDUIRE LA TAILLE DES FICHIERS

- Pour économiser de la place sur les supports de stockage,
- Pour réduire la quantité de données à transférer sur un réseau.

## ALGORITHMES

On distingue les algorithmes de compression avec perte ou sans perte :

- Sans perte, signifie que l'algorithme de décompression **permettra** de retrouver les données telles qu'elles étaient avant la compression. Il s'agit juste d'une ré-écriture des données sous forme plus concise.
- Avec perte, signifie que l'algorithme de décompression **ne permettra pas** de retrouver les données telles qu'elles étaient avant la compression. Les données initiales sont modifiées pour prendre moins de place.

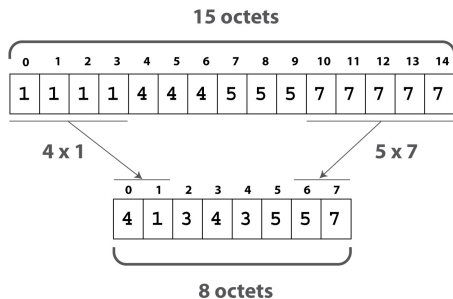
## FORMATS COMPRESSÉS

Les fichiers compressés sont des fichiers binaires

- Formats issus d'un programme de compression : `.gz`, `.bz2`, `.Z`, `.tgz`, `.zip`, ...
- Formats spécialisés : `.jpeg`, `.mpeg`, ...

# COMPRESSION

## PRINCIPE DE L'ALGORITHME RLE (RUN-LENGTH ENCODING)



## DESCRIPTION

- Une séquence de  $n$  répétitions du motif  $m$  est ré-écrite comme un couple  $(n,m)$ .
- L'ordre des couples permet de retrouver les données initiales,
- C'est un algorithme de compression sans perte.

# gzip

## SYNTAXE

```
gzip fichier <fichier.2 ...>
```

## DESCRIPTION

- Comprime un ou plusieurs fichiers dont le nom est passé en paramètre.
- Le fichier source (initial non compressé) est supprimé et seul subsiste le fichier compressé.
- Le fichier compressé qui apparaît porte le même nom que le fichier initial avec l'extension `.gz` ajoutée à la fin.

## EXEMPLE D'UTILISATION:

```
chez_moi/..... Répertoire Courant
├─ tellurique.tsv
└─ astronomie.txt..... Avant gzip
```

```
chez_moi/..... Répertoire Courant
├─ tellurique.tsv
└─ astronomie.txt.gz..... Après gzip
```

```
[ login@localhost ~ ] ls
astronomie.txt telluriques.tsv

[ login@localhost ~ ] gzip astronomie.txt

[ login@localhost ~ ] ls
astronomie.txt.gz telluriques.tsv
```

# gunzip

## SYNTAXE

```
gunzip fichier <fichier.2 ...>
```

## DESCRIPTION

- Décompresse un ou plusieurs fichiers dont le nom est passé en paramètre.
- Le fichier source (compressé) est supprimé et seul subsiste le fichier décompressé.
- Le fichier décompressé qui apparaît porte le même nom que le fichier initial sans l'extension .gz ajoutée à la fin.

## EXEMPLE D'UTILISATION:

```
chez_moi/..... Répertoire Courant
├─ tellurique.tsv
└─ astronomie.txt.gz..... Avant gunzip
```

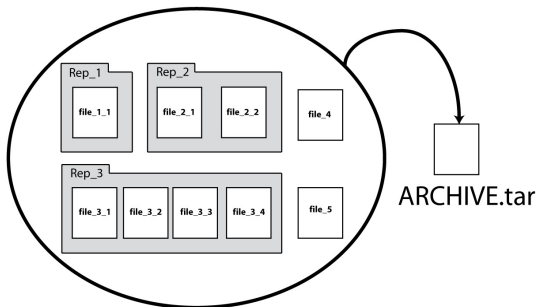
```
chez_moi/..... Répertoire Courant
├─ tellurique.tsv
└─ astronomie.txt..... Après gunzip
```

```
[ login@localhost ~ ] ls
astronomie.txt.gz telluriques.tsv

[ login@localhost ~ ] gunzip astronomie.txt.gz

[ login@localhost ~ ] ls
astronomie.txt telluriques.tsv
```

# ARCHIVAGE



## REGROUPER LES FICHIERS

- Pour simplifier le transfert des données, un groupe de fichiers et de répertoires sont concaténés dans un seul fichier qui peut être compressé.
- Pour archiver des données non utilisées (sauvegardes intermédiaires).

# tar

## SYNTAXE

```
tar cv nom_archive fichier_ou_repertoire <autres_sources>
```

## DESCRIPTION

- Crée un fichier archive dont le nom (chemin) est donné en premier argument et porte classiquement l'extension `.tar`.
- Les fichiers sources qui servent à créer l'archive sont préservés par la commande `tar`.
- L'option `c` (**C**reate), indique que la commande `tar` doit utiliser un algorithme d'archivage.
- L'option `v` (**V**erbose), permet d'afficher le déroulement de l'archivage.

## EXEMPLE D'UTILISATION:

Regroupe dans la même archive `espace.tar` le fichier `astronomie.txt` et le répertoire `Images/` et son contenu :

```
[ login@localhost ~ ] tar cv espace.tar astronomies.txt Images/
```

# tar(bis)

## SYNTAXE

```
tar xv nom_archive
```

## DESCRIPTION

- Extrait les fichiers et répertoires d'une archive.
- Les fichiers sont placés dans le répertoire courant.
- L'option x (eXtarct) indique que la commande tar doit utiliser un algorithme de désarchivage.

## EXEMPLE D'UTILISATION:

Extrait le contenu de l'archive espace.tar :

```
[ login@localhost ~ ] tar xv espace.tar
```

# tar(ter)

## SYNTAXE

```
tar cvz nom_archive fichier_ou_repertoire <autres_sources>
```

## SYNTAXE

```
tar xvz nom_archive
```

## DESCRIPTION

- L'option z permet de créer ou d'extraire une archive compressée.
- L'extension donnée aux fichiers contenant une archive compressée par ce moyen est classiquement : .tgz

## EXEMPLE D'UTILISATION:

Crée une archive compressée espace.tgz avec le fichier astronomie.txt et le répertoire Images/ et son contenu :

```
[ login@localhost ~ ] tar cvz espace.tar astronomie.txt Images/
```

Extrait le contenu d'une archive compressée espace.tgz :

```
[ login@localhost ~ ] tar xvz espace.tar
```



- 1 PROPRIÉTÉ ET DROITS SUR LES FICHIERS
- 2 ARCHIVES ET COMPRESSION DES FICHIERS
- 3 ÉDITION ET MANIPULATION DE FICHIERS
  - L'éditeur de texte
  - L'édition en ligne de commande
  - Astuces en ligne de commande

## L'ÉDITEUR DE TEXTE

## CARACTÉRISTIQUES

- Affiche le contenu de fichiers textes bruts,
- Le texte n'est pas mis en forme : pas de notion de titre ni de paragraphe, de taille ou de police de caractères ...
- Le texte ne contient que des caractères alpha-numériques.

```

Source de : http://www.lutv.univ-paris13.fr/lutv/index.php - Mozilla Firefox
Fichier Edition Affichage Aide
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
<HTML xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>IUT de VILLETANEUSE - Université Paris 13</title>
<meta name="description" content="Site officiel de l'IUT de Villetaneuse, Université Paris XIII" />
<meta name="keywords" content="iut, villetaneuse, Diplôme Universitaire de Technologie, Université Paris 13, Univ
<meta http-equiv="Content-Language" content="FR" />
<meta name="author" content="Reni DUPAC" />
<meta name="copyright" content="Copyright 2004-2007 IUTV" />
<meta name="robots" content="all" />
<meta name="verify-v1" content="F2v4B/EVeEMrZ0LB9bY0KZE3A1NjyIwpokM/p0+zRA=" />
<link rel="stylesheet" type="text/css" href="inc/style.css" media="all" />
<link href="inc/SpryCollapsiblePanel.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">AC_FL_RunContent = 0;</script>
<script src="inc/AC_RunActiveContent.js" type="text/javascript"></script>
<script src="inc/SpryCollapsiblePanel.js" type="text/javascript"></script>
<script type="text/javascript" src="inc/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="inc/iutv.js"></script>
<script type="text/javascript" src="actus/js/ajax.js"></script>
<script type="text/javascript">
var curId=0;
function AddChamp(){
    document.getElementById('nextdiv' + curId).innerHTML = '<div id="divActus"> </div>';
}
function SupprChamp(){
    document.getElementById('nextdiv' + curId).innerHTML = "";
}
</script>
<script type="text/javascript">
var ajax = new sack();
var articleList0b;
var activeArticle = false;
var clickedArticle = false;
var content0b;

function mouseoverArticle()

```




















Texte brut & coloration syntaxique

## LA COLORATION SYNTAXIQUE

- L'éditeur reconnaît le format du fichier et applique des règles de coloration sur certains mots clefs, délimiteurs, ...
- Contrairement aux documents mis en forme (pdf, doc), la coloration des caractères n'est pas sauvee dans le fichier, c'est le résultat d'un calcul de l'éditeur.

# L'ÉDITEUR DE TEXTE

## LES RACOURCIS CLAVIERS UTILES



Combinaisons de touches		Action
 + 		Tout sélectionner
 + 		Copier la sélection
 + 		Couper la sélection
 + 		Coller la sélection là où se trouve le curseur
 + 		Annuler la dernière opération
 + 		Enregistrer le document
 +  + 		Enregistrer le document sous ...
 + 		Ouvrir un document
 + 		Créer un nouveau document

## UTILISATION

Les fichiers textes sont utilisés pour :

- Écrire les sources des scripts et des programmes,
- Écrire les sources de certains documents (HTML,  $\text{\LaTeX}$ , ...),
- Enregistrer/Manipuler des données, ...

## PRENEZ L'HABITUDE DE SAUVER RÉGULIÈREMENT VOTRE TRAVAIL

- Une version précédente du document pourra toujours être obtenue en répétant la combinaison  +  ,
- Si il y a un problème avec votre ordinateur (coupure d'alimentation électrique, plantage du système, ...), vous aurez ainsi une version récente de votre travail.

# L'ÉDITION EN LIGNE DE COMMANDE

## DE NOMBREUX OUTILS

---

### Visualisation de Contenu

---

`more` Affiche le contenu page par page

`less` Ouvre un environnement permettant de naviguer dans le document

---

### Extraction de Parties

---

`head` Affiche les premières lignes

`tail` Affiche les dernières lignes

`cut` Affiche une colonne

`grep` Affiche les lignes comportant un motif particulier

---

### Manipulation de textes

---

`cat` Concatène plusieurs fichiers

`sed` Opère des substitutions

`sort` Trie les lignes selon un ordre

`uniq` Supprime les lignes consécutives identiques

---

### Analyse de contenu

---

`wc` Compte le nombre de lignes, de mots et d'octets



---

## more

### SYNTAXE

```
more fichier <fichier_2 ...>
```

### DESCRIPTION

- Affiche le contenu du (des) fichier(s) page par page,
- L'affichage s'adapte à la taille du shell,
- Pour passer à la ligne suivante, l'utilisateur presse la touche .
- Pour passer à la page suivante, l'utilisateur presse la touche .
- Une fois que tout le contenu du fichier a défilé, l'utilisateur retrouve un nouveau prompt.

### EXEMPLE D'UTILISATION:

- Cette commande est utilisée pour parcourir des documents dont l'affichage dépasse la taille de la fenêtre du terminal.
- Utilisée avec un tube (cf. Partie sur les Redirections) elle permet de visualiser tous les résultats d'une commande qui dépasserait la taille de la fenêtre du terminal. Par exemple, si un répertoire contient de très nombreux fichiers, la commande `ls` qui affiche le contenu du répertoire peut produire un affichage très long. Si l'on souhaite passer en revue tous les fichiers il faut alors utiliser la commande suivante :

```
[ login@localhost ~ ] ls Ma_Musique | more
```

# less

## SYNTAXE

```
less fichier
```






## DESCRIPTION







- Affiche le contenu d'un fichier,
- Permet de naviguer en avant et en arrière dans le fichier.
- Permet d'effectuer des recherches de mot(if)s.

La commande ouvre une interface dans la fenêtre du terminal. Contrairement à la commande `more`, on ne revient pas à la ligne de commande lorsqu'on atteint la fin du fichier, pour cela il faut quitter l'application.

## EXEMPLE D'UTILISATION:

Pour avoir une description complète des commandes de navigation dans l'interface de visualisation `less`, reportez-vous aux pages de `man`. Les commandes les plus utilisées sont :

Combinaison de touches	Action
	Affiche l'aide (abrégé des commandes)
	Avancer d'une page (forward)
	Reculer d'une page (backward)
	Avancer d'une ligne
	Reculer d'une ligne

Combinaison de touches	Action
	Quitter
	Aller à la première ligne
 + 	Aller à la dernière ligne
 + 	Aller à la ligne numéro num

# head

## SYNTAXE

```
head < -int > fichier
```

## DESCRIPTION

- Affiche par défaut les 10 premières lignes d'un fichier.
- Si un entier  $n$  précède le nom du fichier, la commande affiche les  $n$  premières lignes du fichier.

## EXEMPLE D'UTILISATION:

Soit le fichier `planetes.txt` contenant les lignes suivantes :

**planetes.txt**

```
# Premier groupe
1 Mercure Tellurique
2 Venus Tellurique
3 Terre Tellurique
4 Mars Tellurique
# Deuxième groupe
1 Jupiter Gazeuse
2 Saturne Gazeuse
3 Uranus Gazeuse
4 Neptune Gazeuse
```

La commande suivante affiche les 5 premières lignes du fichier :

```
[ login@localhost ~ ] head -5 planetes.txt
# Premier groupe
1 Mercure Tellurique
2 Venus Tellurique
3 Terre Tellurique
4 Mars Tellurique

[ login@localhost ~ ] █
```

# tail

## SYNTAXE

```
tail < -int > fichier
```

## DESCRIPTION

- Affiche par défaut les 10 dernières lignes d'un fichier.
- Si un entier *n* précède le nom du fichier, la commande affiche les *n* dernières lignes du fichier.

## EXEMPLE D'UTILISATION:

Soit le fichier `planetes.txt` contenant les lignes suivantes :

**planetes.txt**

```
# Premier groupe
1 Mercure Tellurique
2 Venus Tellurique
3 Terre Tellurique
4 Mars Tellurique
# Deuxième groupe
1 Jupiter Gazeuse
2 Saturne Gazeuse
3 Uranus Gazeuse
4 Neptune Gazeuse
```

La commande suivante affiche les 4 dernières lignes du fichier :

```
[ login@localhost ~ ] tail -4 planetes.txt
1 Jupiter Gazeuse
2 Saturne Gazeuse
3 Uranus Gazeuse
4 Neptune Gazeuse

[ login@localhost ~ ] █
```



# cut

## SYNTAXE

```
cut -d 'sep' -f n fichier
```

## DESCRIPTION

- Affiche une colonne du fichier.
- L'option `<-d 'sep'>` permet de changer le séparateur par défaut qui est la tabulation. Le séparateur est donné entre guillemets simples.
- L'option `<-f n>` indique que la commande doit afficher la n<sup>ème</sup> colonne.

## EXEMPLE D'UTILISATION:

Cas#1 : les mots (les champs) sont séparés par des tabulations :

tellur.tsv		
1	Mercure	Venus
2	Terre	Mars

Commande #1

```
[ login@localhost ~ ] cut -f 2 tellur.tsv
Mercure
Terre
[ login@localhost ~ ] █
```

Cas#2 : les mots (les champs) sont séparés par le caractère = :

jov.txt	
1=Jupiter=Saturne	
1=Uranus=Neptune	

Commande #2

```
[ login@localhost ~ ] cut -d '=' -f 3 jov.txt
Saturne
Neptune
[ login@localhost ~ ] █
```

# grep

## SYNTAXE

```
grep "motif" fichier
```

## DESCRIPTION

- Affiche les lignes du fichier qui comportent le "motif".
- Les lignes sont affichées dans leur ordre d'apparition dans le fichier.

## EXEMPLE D'UTILISATION:

Soit le fichier `planetes.txt` contenant les lignes suivantes :

### planetes.txt

```
# Premier groupe
1 Mercure Tellurique
2 Venus Tellurique
3 Terre Tellurique
4 Mars Tellurique
# Deuxième groupe
1 Jupiter Gazeuse
2 Saturne Gazeuse
3 Uranus Gazeuse
4 Neptune Gazeuse
```

Commandes :

```
[ login@localhost ~ ] grep 'Tellurique' planetes.txt
1 Mercure Tellurique
2 Venus Tellurique
3 Terre Tellurique
4 Mars Tellurique

[ login@localhost ~ ] grep '1' planetes.txt
1 Mercure Tellurique
1 Jupiter Gazeuse

[ login@localhost ~ ] █
```

# cat

## SYNTAXE

```
cat fichier <fichier_2 ...>
```

## DESCRIPTION

- Affiche le contenu des fichiers les uns à la suite des autres.
- Les fichiers sont concaténés dans l'ordre des paramètres.

## EXEMPLE D'UTILISATION:

Cette commande est en générale utilisée pour concaténer des fichiers textes. On l'utilise avec une commande de redirection (*cf.* Partie Redirections) pour enregistrer le résultat de la concaténation dans un nouveau fichier.

Soient les deux fichiers suivants :

tellur.txt

Mercure, Venus  
Terre, Mars

jov.txt

Jupiter, Saturne  
Uranus, Neptune

La commande :

```
[ login@localhost ~ ] cat tellur.txt jov.txt
Mercure, Venus
Terre, Mars
Jupiter, Saturne
Uranus, Neptune

[ login@localhost ~ ] █
```

# sort

## SYNTAXE

```
sort <-r> fichier
```

## DESCRIPTION

- Affiche les lignes du fichier triées par ordre croissant.
- L'option `-r` inverse l'ordre de tri.

## EXEMPLE D'UTILISATION:

Soit le fichier :

donnees.txt
a
A
1
7
8
71

```
[ login@localhost ~ ] sort donnees.txt
1
7
71
8
A
a
[ login@localhost ~ ] █
```

# uniq

## SYNTAXE

```
uniq fichier
```

## DESCRIPTION

- Affiche les lignes du fichier en supprimant les lignes consécutives identiques.

## EXEMPLE D'UTILISATION:

Soit le fichier :

```
donnees.txt
```

```
1
lune
Terre
Terre
lune
```

```
[ login@localhost ~ ] uniq donnees.txt
1
lune
Terre
lune

[ login@localhost ~ ] █
```

# sed

## SYNTAXE

```
sed 's/motif/new/g' fichier
```

## DESCRIPTION

La commande `sed` est une commande qui permet de faire de nombreuses opérations. Nous ne verrons ici que la syntaxe permettant de substituer un motif dans un texte.

- Affiche le contenu du fichier après avoir remplacé les occurrences du motif par `new`.

## EXEMPLE D'UTILISATION:

Soit le fichier :

```
dialogue.txt
```

```
- C'est par ici!!!  
-Où ça, "ici"?
```

```
[ login@localhost ~ ] sed 's/ici/là/' dialogue.txt  
- C'est par là!!!  
-Où ça, "là"?  
  
[ login@localhost ~ ] █
```

## WC

## SYNTAXE

```
wc fichier <fichier.2 ...>
```

## DESCRIPTION

- Affiche des statistiques sur le nombre de lignes, de mots et de caractères (comptés en nombre d'octets) contenus dans le fichier dont le chemin est donné en paramètre.

## EXEMPLE D'UTILISATION:

Soit le fichier suivant :

```
tellur.tsv
1 Mercure Venus
2 Terre Mars
```

Commande #1 :

```
[ login@localhost ~ ] wc tellur.tsv
2      6      29 tellur.tsv

[ login@localhost ~ ] █
```

L'affichage produit indique que le fichier `tellur.tsv` comporte :

- 2 lignes,
- 6 mots et
- 29 caractères. La taille du fichier texte est donc de 29 octets ...

# wc(bis)

## SYNTAXE

```
wc -l fichier <fichier_2 ...>
```

## DESCRIPTION

- L'option `-l` indique que l'on affiche que le nombre de lignes.

## EXEMPLE D'UTILISATION:

Soit le fichier suivant :

```
tellur.tsv
1 Mercure Venus
2 Terre Mars
```

Commande #1 :

```
[ login@localhost ~ ] wc -l tellur.tsv
2      tellur.tsv
[ login@localhost ~ ] █
```










L'affichage produit indique que le fichier `tellur.tsv` comporte :

- 2 lignes.



# SE FACILITER LA VIE EN LIGNE DE COMMANDE

## RACCOURCIS CLAVIER

Combinaisons de Touches	Fonction
	Complétion automatique
	Vide la fenêtre : place la ligne de commande en haut de la fenêtre
	Curseur au début de la ligne
	Tue le processus en cours d'exécution
	Curseur à la fin de la ligne
	Efface ce qui précède le curseur
	Interrompt l'exécution d'un processus
	Affiche la commande précédente dans l'historique
	Affiche la commande suivante dans l'historique

# wget

## SYNTAXE

```
wget chemin
```

## DESCRIPTION

- Client HTTP, HTTPS et FTP .
- Permet de récupérer du contenu d'un serveur Web ou FTP (télécharger).

## EXEMPLE D'UTILISATION:

```
[ login@localhost ~ ] wget http://www-lipn.univ-paris13.fr/~santini/intro_syste
me/2011_2012_S1D_cours_1.pdf .
Résolution de www-lipn.univ-paris13.fr... 10.10.0.68
Connexion vers www-lipn.univ-paris13.fr[10.10.0.68] :80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Longueur : 4568618 (4,4M) [application/pdf]
Sauvegarde en : «2011_2012_S1D_cours_1.pdf»

100%[=====>] 4 568 618 10,4M/s ds 0,4s

2012-01-02 16 :02 :59 (10,4 MB/s) - «2011_2012_S1D_cours_1.pdf» sauvegardé
[4568618/4568618]

[ login@localhost ~ ] ls -l ./2011_2012_S1D_cours_1.pdf
-rw-r--r-- 1 santini users 4,4M 2011-12-14 10 :33 ./2011_2012_S1D_cours_1.pdf
```

## diff

## SYNTAXE

```
diff fichier_1 fichier_2
```

## DESCRIPTION

- Compare deux fichiers, localise et affiche les différences (très utile pour suivre l'évolution d'un code).
- identifie les insertions, délétions et modifications.

## EXEMPLE D'UTILISATION:

planete.v1	planete.v2	
1 Mercure	1 Planetes	← Insertion
2 Venus	2 Mercure	
3 Terre	3 Venus	
4 Mars	4 Terre	
5 Jupiter	5 Mars	
6 Saturn	6 Jupiter	
7 Uranus	7 Saturne	← Modification
8 Neptune	8 Uranus	
9 Pluton	9 Neptune	← Délétion

```
[ login@localhost ~ ] diff planete.v1 planete.v2
0a1
> Planetes
6c7
< Saturn
---
> Saturne
9d9
< Pluton
[ login@localhost ~ ] █
```