

Cours des Systèmes d'Exploitation LINUX

ENSAM – Casablanca
2022-2023

Chapitre 2

Systeme de fichier Linux

Systeme de fichier Linux

Sur un système Linux, **tout est fichier**.

Cela peut sembler évident à première vue. Un document texte est évidemment un fichier, tout comme un document OpenOffice, une image, une vidéo ou un MP3.

Mais qu'en est-il des répertoires ? Il s'agit aussi de fichiers, d'un type particulier, qui contiennent des informations sur d'autres fichiers.

Les lecteurs de disques sont des gros fichiers.

Les connexions réseau sont des fichiers.

Même les processus en cours d'exécution sont des fichiers.

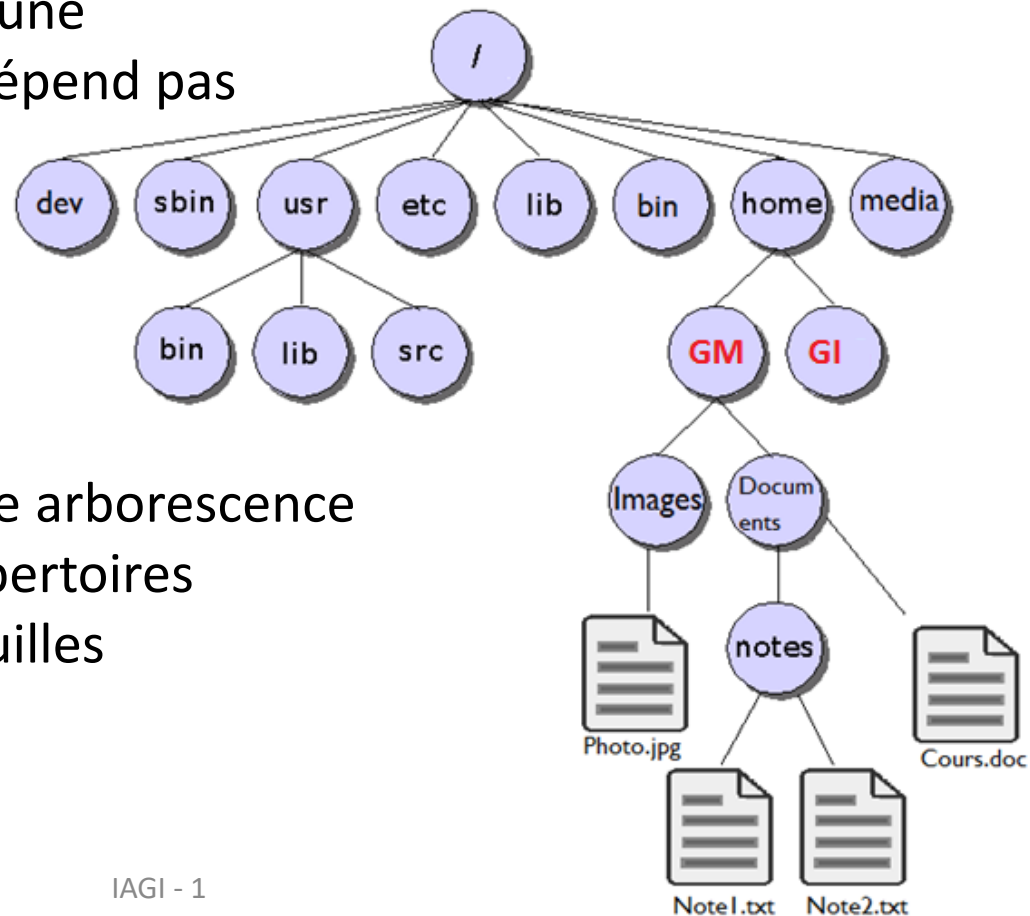
Bref, sous Linux... **tout est fichier**.

Système de fichier Linux

Le principe de l'arborescence sous Linux est totalement différent de celui de Windows.

Le principe est d'avoir une arborescence qui ne dépend pas des disques durs et de leurs partitions.

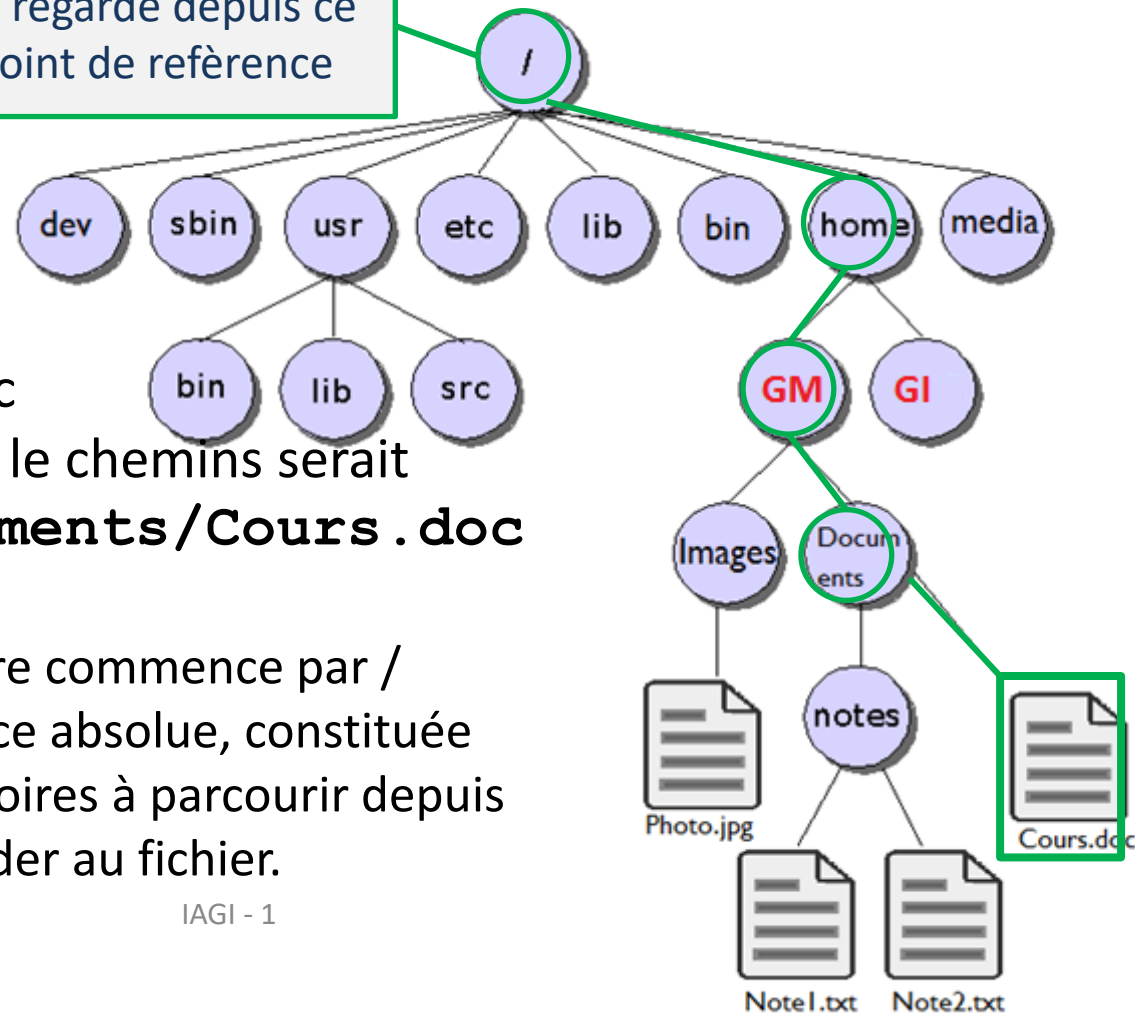
/ : est la racine de cette arborescence
Les nœuds sont les répertoires
Les fichiers sont les feuilles



Chemins absolus et relatifs

Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

on regarde depuis ce point de référence



Le chemin absolu :

Exemple : Cours.doc

Partant de la racine le chemins serait

/home/GM/Documents/Cours.doc

Si le nom de répertoire commence par /
il s'agit d'une référence absolue, constituée
d'une liste des répertoires à parcourir depuis
la racine / pour accéder au fichier.

Chemins absolus et relatifs

Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

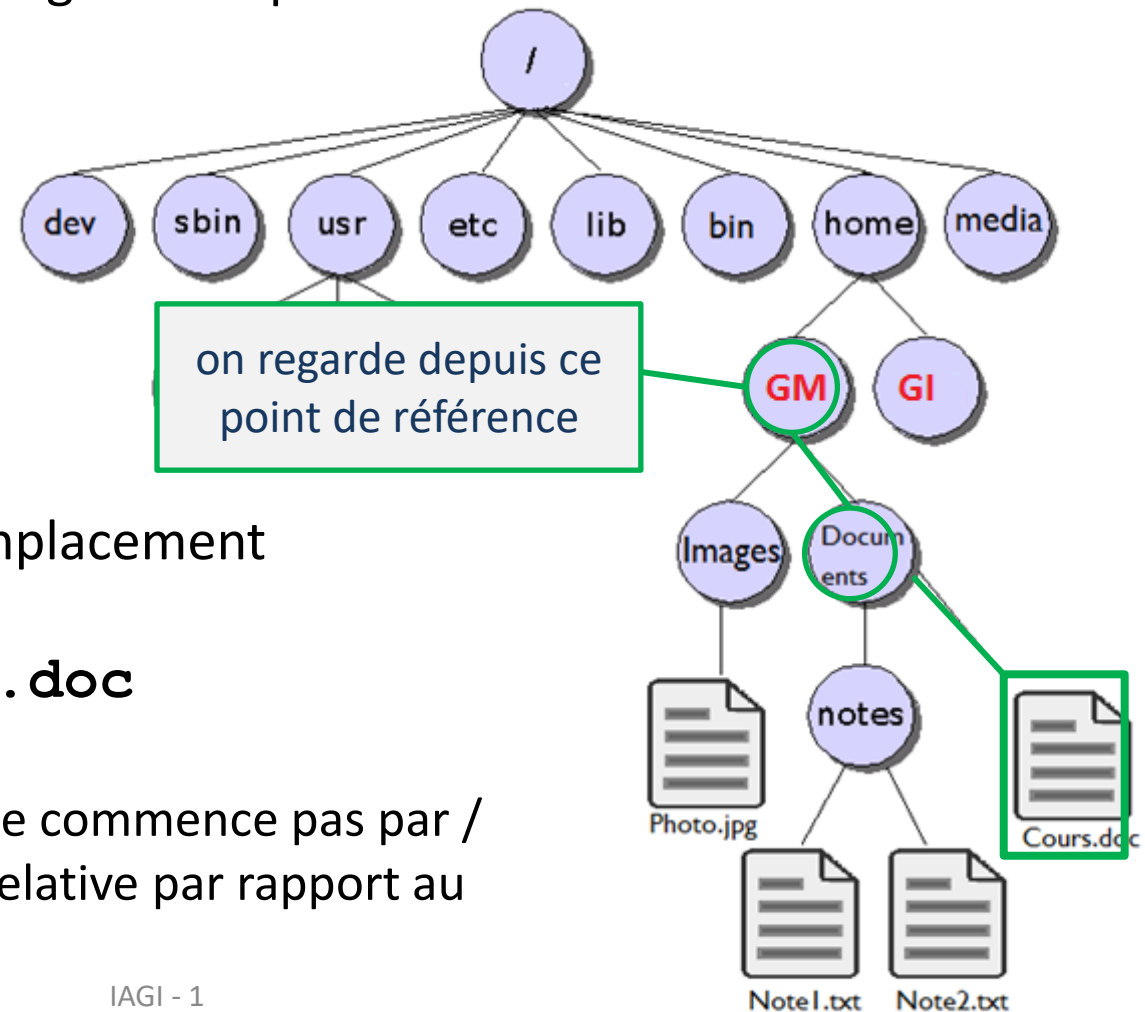
Le chemin relatif :

Exemple : Cours.doc

Le chemin relatif à l'emplacement
ou on se trouve :

Documents/Cours.doc

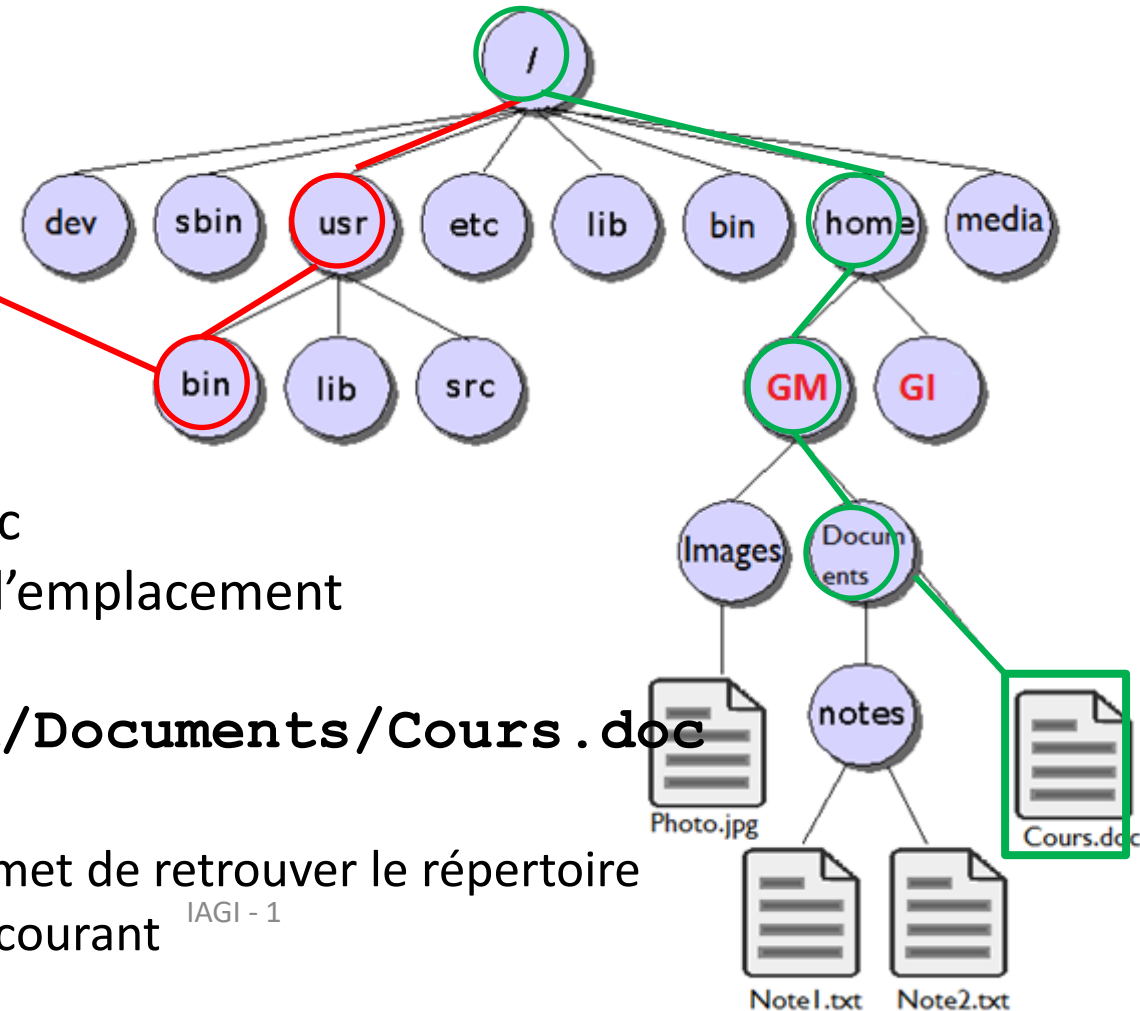
Si le nom de répertoire ne commence pas par /
il s'agit d'une référence relative par rapport au
répertoire courant.



Chemins absolus et relatifs

Selon ce schéma, 2 désignations possibles du chemin vers un fichier :

on regarde depuis ce point de référence



Le chemin relatif :

Exemple : Cours.doc

Le chemin relatif à l'emplacement
ou on se trouve :

`../../home/GM/Documents/Cours.doc`

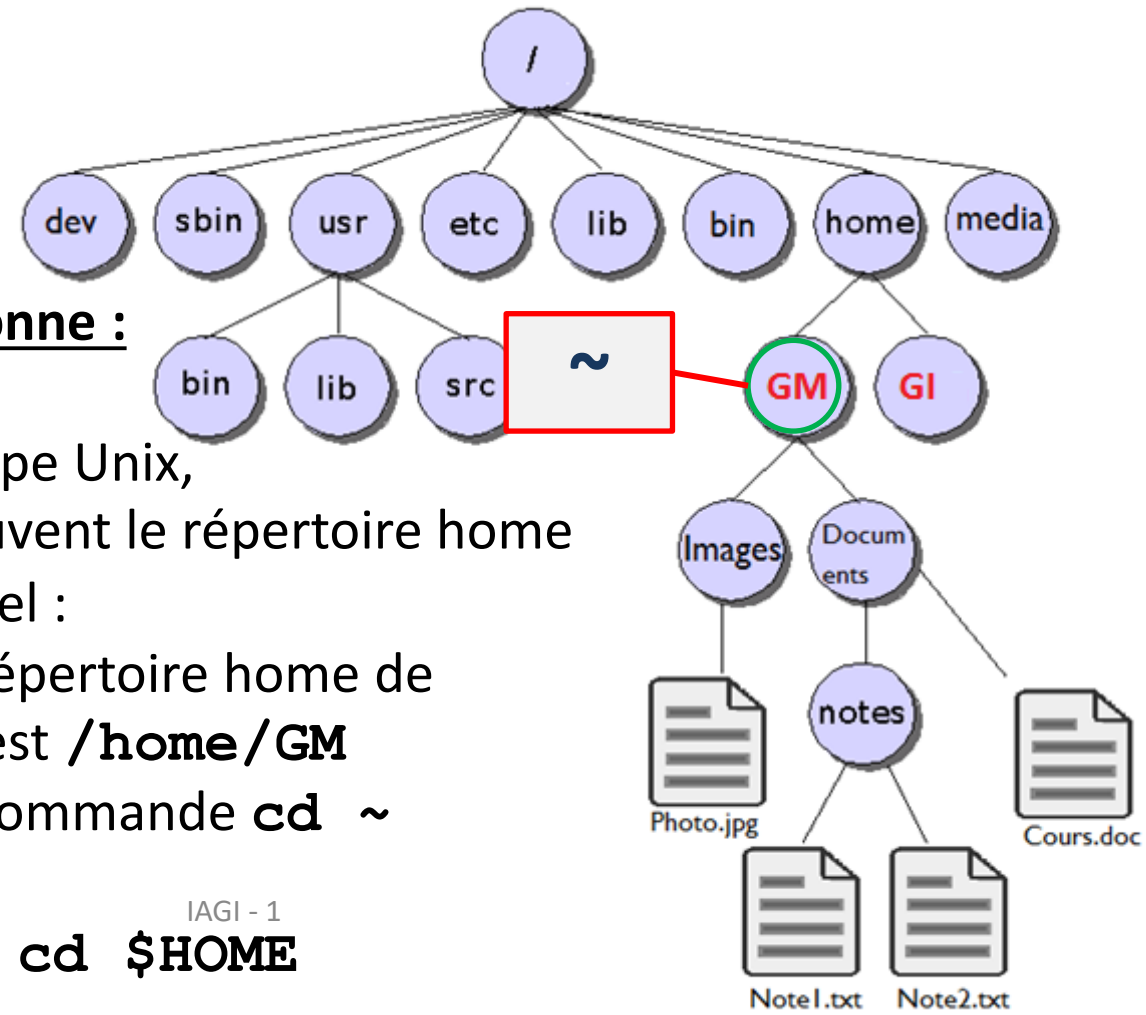
L'utilisation de .. Permet de retrouver le répertoire
parent au répertoire courant

Chemin dossier personnel

L'opérateur `~` (tilde) : Il permet de se positionner facilement et directement dans le dossier personnel sans taper son chemin.

Le répertoire personne :

Sur les systèmes de d'exploitation de type Unix, tilde `~` indique souvent le répertoire home de l'utilisateur actuel :
par exemple, si le répertoire home de l'utilisateur actuel est `/home/GM` on peut utiliser la commande `cd ~` à la place de `cd /home/GM` ou `cd $HOME`



Commandes de manipulation des fichiers

Afficher le dossier courant

Syntaxe : **pwd** [options]

(Eng – *print working directory*)

La commande **pwd** indique votre position actuelle au sein du système de fichiers. Elle permet de savoir où vous êtes dans le système de fichiers. La commande **pwd** commande n'a pas d'arguments, il suffit de taper **pwd** sur le Shell :



```
info@ubuntu: ~/Documents
info@ubuntu:~$ pwd
/home/info
info@ubuntu:~$ cd Documents
info@ubuntu:~/Documents$ pwd
/home/info/Documents
info@ubuntu:~/Documents$
```

Commandes de manipulation des fichiers

Lister les fichiers

Syntaxe : **ls** [options] [paramètres]

(Eng - list)

Permet de lister le contenu d'un dossier :

- ☐ par défaut, le dossier courant si utilisée sans paramètres.
- ☐ un dossier ou plusieurs différent si vous faites passer leurs chemins en paramètres.

Principales options (cumulables) :

- l** : affichage au format long des informations relatives aux objets
- g** : affichage des groupes propriétaires des objets
- R** : liste récursive
- i** : affichage du **inode** des fichiers listés
- d** : affichage des noms des objets et non de leurs contenus
- F** : affichage des objets avec un suffixe désignant le type de l'objet
- a** : affichage des objets dont les noms commencent par « . » (fichiers cachés)



Commandes de manipulation des fichiers

Changer de répertoire

Syntaxe : **cd** [options] [paramètres]

(Eng – *change directory*)

Permet de se déplacer au sein du système de fichiers.

- ❑ Si elle est utilisée sans argument, il vous renvoie dans votre répertoire personnel.
- ❑ Le chemin de ce répertoire est nécessaire comme argument, pour déplacer vers un autre répertoire :



```
info@ubuntu: /etc
info@ubuntu:~$ cd /etc
info@ubuntu:/etc$ pwd
/etc
info@ubuntu:/etc$
```


Commandes de manipulation des fichiers



Création de répertoire

Syntaxe : **mkdir** [options] paramètres

(Eng – *make directory*)

Permet de créer un répertoire (ou plusieurs) dans l'emplacement donné en paramètres.

❑ Utilisée avec plusieurs arguments, elle permet de créer plusieurs répertoires dans un même niveau de l'arborescence. Exemples : **mkdir rep1 rep2 rep3**

❑ Si on veut créer une suite de répertoires l'un inclus dans l'autre :

mkdir -p rep1/rep2/rep3 (similaire à : **mkdir rep1 rep1/rep2 rep1/rep2/rep3**)

```
info@ubuntu: ~  
info@ubuntu:~$ mkdir mon_repertoire  
info@ubuntu:~$ ls  
Desktop      history2.txt  Music        Templates  
Documents    IAGI         Pictures     Videos  
Downloads    mon_repertoire  Public  
info@ubuntu:~$
```


Commandes de manipulation des fichiers

Supprimer un répertoire

Syntaxe : **rmdir** [options] paramètres

(Eng – *remove directory*)

Permet de supprimer un répertoire (ou plusieurs) dont l'emplacement est donné en paramètres.

- ❑ Utilisée avec plusieurs arguments, elle permet de supprimer plusieurs répertoires dans un même niveau de l'arborescence. Exemples : **rmdir rep1 rep2 rep3**
- ❑ Si on veut supprimer une suite de répertoires l'un inclus dans l'autre :

rmdir -p rep1/rep2/rep3



```
info@ubuntu: ~  
info@ubuntu:~$ rmdir mon_repertoire  
info@ubuntu:~$ ls  
Desktop      Downloads    IAGI         Pictures     Templates  
Documents    history2.txt Music        Public       Videos  
info@ubuntu:~$
```

Commandes de manipulation des fichiers

Création de fichiers vides

Syntaxe : **touch** [options] paramètres

(Eng – *change file timestamps*)

Permet de mettre à jour le temps d'accès et de modification d'un fichier (ne marche pas sur les répertoires) dont l'emplacement est donné en paramètres.

❑ Utilisée pour créer un fichier vide s'il n'existe pas auparavant

touch fich



```
info@ubuntu: ~  
info@ubuntu:~$ touch fich  
info@ubuntu:~$ ls  
Desktop      fich          Music         Templates  
Documents    history2.txt  Pictures      Videos  
Downloads    IAGI         Public  
info@ubuntu:~$
```

Commandes de manipulation des fichiers

Création et édition de fichiers

Syntaxe : **gedit** [options] [paramètres]

(Eng – text editor for the GNOME Desktop)

Permet de créer un fichier et d'éditer son contenu.

gedit est l'éditeur par défaut et officiel de l'environnement de travail GNOME. Il est facile et simple à utiliser. Son mode d'exécution est **graphique**. Il existe d'autre éditeur en **ligne de commandes** (comme Vi, Vim, nano, Emacs...) que nous rencontrerons dans la suite de ce cours.

- ❑ Si utiliser sans paramètres, cela permet d'éditer un fichier sans qu'il soit créé physiquement, en attendant de l'enregistrer à partir de l'éditeur.



Commandes de manipulation des fichiers

Afficher le contenu d'un fichier

Syntaxe : **cat** [options] [paramètres]

(Eng – concatenate files and print on the standard output)

Permet d'afficher le contenu d'un fichier (ou plusieurs) en ligne.

❑ Si utiliser avec l'option -n, cela permet d'énumérer les lignes du fichier.



```
info@ubuntu: ~  
info@ubuntu:~$ cat fich1  
J'écrit dans l'editeur de l'interface graphique GNOME  
Gedit  
info@ubuntu:~$ cat -n fich1  
  1  J'écrit dans l'editeur de l'interface graphique GNOME  
  2  Gedit  
info@ubuntu:~$
```


Commandes de manipulation des fichiers

Supprimer un fichier

Syntaxe : **rm** [options] paramètres

(Eng – *remove files or directories*)

Permet de supprimer un ou plusieurs fichiers.

Par défaut, elle ne permet pas de supprimer les répertoires. Mais :

- ☐ Avec l'option **-r** cela est possible.
- ☐ Elle peut supprimer toute une arborescence on utilisant l'option **-rf**. **Dangereuse?!?! Sans confirmation, suppression définitive.**
- ☐ L'option **-i** exécute la tâche d'une manière interactif ; demande la confirmation de chaque opération.



```
info@ubuntu: ~  
info@ubuntu:~$ ls  
Desktop    fich          IAGI          Pictures    Videos  
Documents  fich1         mon_repertoire Public  
Downloads  history2.txt Music          Templates  
info@ubuntu:~$ rm fich  
info@ubuntu:~$ rm -i fich1  
rm : supprimer 'fich1' du type fichier ? y  
info@ubuntu:~$ rm mon_repertoire  
rm: impossible de supprimer 'mon_repertoire': est un dossier  
info@ubuntu:~$ rm -f mon_repertoire  
rm: impossible de supprimer 'mon_repertoire': est un dossier  
info@ubuntu:~$ rm -r mon_repertoire  
info@ubuntu:~$ ls  
Desktop    Downloads     IAGI    Pictures  Templates  
Documents  history2.txt Music     Public    Videos  
info@ubuntu:~$
```

Commandes de manipulation des fichiers

Copier un fichier

Syntaxe : `cp [options] paramètres`

(Eng – *copy files and directories*)

Permet de copier une source vers une destination. Elle est valable pour copier des fichiers et des répertoires.

Exemples d'utilisation.

- ❑ Copier un fichier vers un autre fichier:

`cp fich1 fich2`

- ❑ Copier des fichiers dans un répertoire :

`cp fich1 fich2 rep_fich`



```
info@ubuntu: ~  
info@ubuntu:~$ ls  
Desktop      fich1          Music          repl  
Documents    history2.txt  Pictures       Templates  
Downloads    IAGI          Public         Videos  
info@ubuntu:~$ cp fich1 fich2  
info@ubuntu:~$ ls  
Desktop      fich1          IAGI           Public         Videos  
Documents    fich2          Music          repl  
Downloads    history2.txt  Pictures       Templates  
info@ubuntu:~$ cp fich1 fich2 repl  
info@ubuntu:~$ ls repl  
fich1  fich2  
info@ubuntu:~$
```

Commandes de manipulation des fichiers

Déplacer (renommer) un fichier

Syntaxe : `mv [options] paramètres`

(Eng – *move (rename) files*)

Permet de déplacer une source vers un répertoire de destination. Elle est valable pour déplacer des fichiers et des répertoires. Elle peut aussi être utilisée pour renommer un fichier

Exemples d'utilisation :



```
info@ubuntu: ~  
info@ubuntu:~$ mkdir rep2  
info@ubuntu:~$ ls ~ rep2  
/home/info:  
Desktop    fich1          IAGI          Public    Templates  
Documents  fich2          Music         rep1      Videos  
Downloads  history2.txt   Pictures      rep2  
  
rep2:  
info@ubuntu:~$ mv fich2 fich3  
info@ubuntu:~$ ls  
Desktop    fich1          IAGI          Public    Templates  
Documents  fich3          Music         rep1      Videos  
Downloads  history2.txt   Pictures      rep2  
info@ubuntu:~$ mv fich1 fich3 rep2  
info@ubuntu:~$ ls ~ rep2  
/home/info:  
Desktop    Downloads      IAGI          Pictures    rep1    Templates  
Documents  history2.txt   Music         Public      rep2    Videos  
  
rep2:  
fich1  fich3  
info@ubuntu:~$
```


Commandes de manipulation des fichiers

Création d'un lien **physique** vers un fichier

Créant un fichier dans le dossier personnel :

```
$ touch fich.txt
```

Affichant ces propriétés étendues (-li pour faire apparaître le inode de ce fichier) :

```
$ ls -li fich.txt
```

```
info@ubuntu:~$ touch fich.txt
info@ubuntu:~$ ls -li fich.txt
399989 -rw-rw-r-- 1 info info 0 oct. 11 22:29 fich.txt
info@ubuntu:~$ █
```



Nous constatons que ce fichier a pour inode 1078748, et ne possède qu'un seul lien.

On peut créer un nouveau lien sur ce fichier éventuellement dans un répertoire différent si ce répertoire **est sur la même partition que le fichier `fich.txt`**, grâce à la commande `ln` (*link*) dont un exemple d'utilisation :

```
$ ln fich.txt Documents/L_fich.txt
```


Commandes de manipulation des fichiers

Création d'un **lien physique** vers un fichier

Affichant ces propriétés étendues du lien créé et du fichier original:

```
info@ubuntu: ~  
info@ubuntu:~$ ls -li fich.txt  
399989 -rw-rw-r-- 1 info info 0 oct. 11 22:29 fich.txt  
info@ubuntu:~$ ln fich.txt Documents/L_fich.txt  
info@ubuntu:~$ ls -li fich.txt Documents/L_fich.txt  
399989 -rw-rw-r-- 2 info info 0 oct. 11 22:29 Documents/L_fich.txt  
399989 -rw-rw-r-- 2 info info 0 oct. 11 22:29 fich.txt  
info@ubuntu:~$
```

Nous constatons que ce fichier a pour inode 1078748, et ne
on peut alors constater que le fichier fich.txt possède maintenant
deux liens physiques. Dans cette situation, il y a donc une seule
représentation des données sur le disque (ce qui est confirmé par
l'inode qui est identique (**1078748**) pour les deux liens) et deux
noms associés à cette représentation.

Commandes de manipulation des fichiers

Création d'un **lien physique** vers un fichier

Ce qui implique que l'on peut modifier le fichier par le biais de l'un ou l'autre des liens. Pour ce qui est de la suppression d'un lien, la règle est la suivante :

- ☐ Tant que le nombre de liens physiques sur le fichier en question est strictement supérieur à **1**, la commande **rm** aura pour effet la suppression du lien et non des données.
- ☐ Lorsque le nombre de liens physiques est égal à 1, **rm** efface effectivement les données, puisqu'il n'y a plus qu'un lien unique sur celles-ci.

Il peut être intéressant de noter que les commandes internes `cd` et `pwd` comprennent deux options :

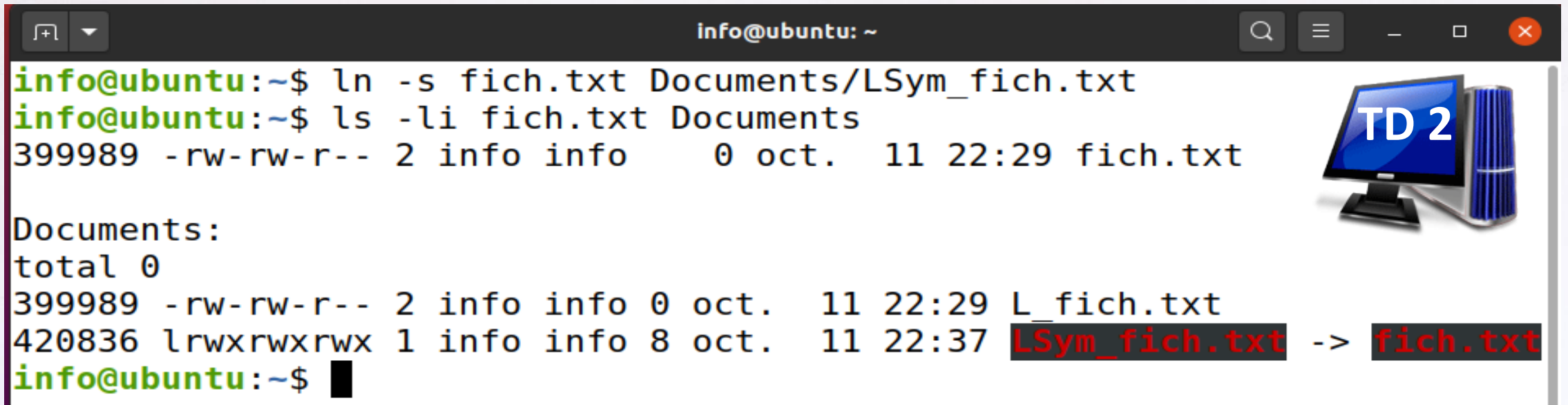
- `-L` pour demander explicitement à suivre les liens symboliques**
- `-P` pour demander à suivre la structure physique de l'arborescence.**

Commandes de manipulation des fichiers


Création d'un lien symbolique vers un fichier

Lorsque l'on veut lier un fichier par l'intermédiaire d'un lien ne se trouvant pas sur la même partition que le fichier lui-même, il est nécessaire de passer par l'utilisation d'un lien dit **symbolique**. Nous allons par exemple créer un lien symbolique sur notre fichier **fich.txt** depuis le répertoire **/Documents**.

```
ln -s fich.txt Document/Lfich.txt
```



```
info@ubuntu: ~  
info@ubuntu:~$ ln -s fich.txt Documents/LSym_fich.txt  
info@ubuntu:~$ ls -li fich.txt Documents  
399989 -rw-rw-r-- 2 info info 0 oct. 11 22:29 fich.txt  
  
Documents:  
total 0  
399989 -rw-rw-r-- 2 info info 0 oct. 11 22:29 L_fich.txt  
420836 lrwxrwxrwx 1 info info 8 oct. 11 22:37 LSym_fich.txt -> fich.txt  
info@ubuntu:~$
```



Commandes de manipulation des fichiers

Création d'un **lien symbolique** vers un fichier

La plupart des opérations sur le lien symbolique est effectuée sur le fichier sur lequel il pointe. Par contre la suppression de ce lien obéit aux règles suivantes :

1. La commande **rm** supprime le lien symbolique lui-même (qui est un fichier à part entière) et n'a pas d'influence sur le fichier auquel il se réfère .
2. Par conséquent, si on supprime le fichier, le lien symbolique existe toujours et pointe sur un fichier qui n'existe pas.
3. La taille du liens ne dépend pas de la taille fichier pointé, mais plutôt de la taille qui correspond aux nombres de caractères du chemin du fichier auquel il se réfère.

Enfin la dernière particularité du lien symbolique (qui le distingue du lien physique) provient du fait qu'il est possible de créer un tel lien sur un répertoire.

Manipulation



IAGI - 1