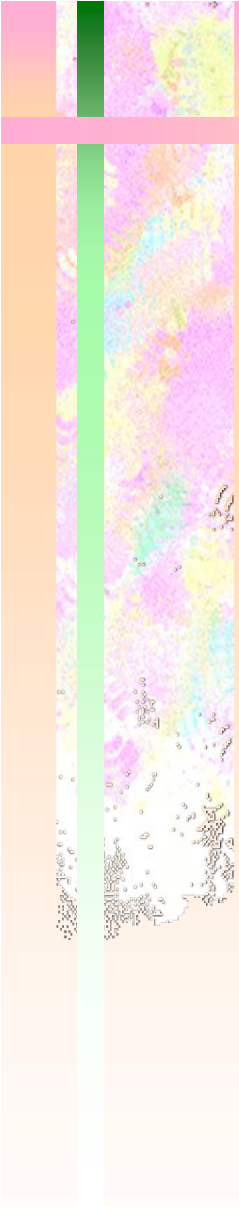




Utiliser Unix

Qu'est-ce qu'Unix ?

- Un système plus ancien que MsDos (1970)
- Au départ, un système pour informaticiens
 - Plus clair, **mieux outillé**
 - Disponible sur tous les types de machines
 - Version gratuite à code public (Linux).

- 
- Inventeur de l'interface graphique (fenêtre)
 - On a toujours les notions de base : système, logiciels, données

Linux à l'IUT


- Choix du système : choisir Linux au chargement
- Les logiciels sont dans le menu "Applications"
- On travaille avec 4 écrans virtuels (dans la barre des tâches)
 - ⇒ répartir les fenêtres

Les logiciels disponibles

- Moins de bureautique que Windows
- Serveur Web (Apache)
- Bases de données
- Compilateurs (ex : gcc) et outils gnu
- Éditeurs de texte
- Nombreux utilitaires

Organisation de l'espace disque

- Où suis-je :
 - Le disque est organisé en répertoires (ou catalogues, ou dossiers) qui contiennent des fichiers et d'autres répertoires.
 - Un répertoire se dit en anglais directory
 - Notion de répertoire courant

- 
- La commande `pwd` (`print working directory`) écrit le nom du *répertoire courant*
 - Ce nom s'abrège par `.`
 - Changer de répertoire courant
`cd` `_répertoire_destination_`
(notation `_nom_`)

Exemple (en couleur claire ce que l'utilisateur tape)

```
localhost<fl 57> pwd
```

```
/Volumes/HD-Mac OS 9.2/users/fl9/IUT
```

```
localhost<fl 58> cd intro1A
```

```
localhost<fl 59> pwd
```

```
/Volumes/HD-Mac OS 9.2/users/fl9/IUT/intro1A
```

`/Volumes/HD-Mac OS 9.2/users/fl9/IUT/intro1A`
est un **chemin**

Racine

- En Unix, l'utilisateur n'a pas à savoir le nom des disques, ni combien il y en a.
- Tous les fichiers et les répertoires sont accessibles à partir d'un répertoire particulier appelé le *répertoire racine* (ou : la *racine*)
- Le répertoire racine est écrit /

Chemins relatifs et absolus

- Si le chemin commence par /, il part de la racine (chemin absolu)
- Si il commence par ~/ , il part du home directory (**C-shell, Bash**)
- Si il commence par ./ ou un nom, il part du répertoire courant.

Exemples :

Le répertoire courant est le répertoire de chemin absolu :

`/home/users/TPINFO/1ereAnnee/`

- `./introInfo/TD2/.emacs` est le fichier `introInfo/TD2/.emacs` dans le répertoire courant.

autrement dit le fichier `.emacs` dans le répertoire

`/home/users/TPINFO/1ereAnnee/introInfo/TD2/`

./ est facultatif ici

- `~/introInfo/TD2/.emacs` est le fichier `introInfo/TD2/.emacs` dans le home `/home/users/INFO1A/B3/10203040/` (par ex) autrement dit le fichier `.emacs` dans le répertoire `/home/users/INFO1A/B3/10203040/introInfo/TD2/`
- `/introInfo/TD2/.emacs` part de la racine (et ne correspond probablement à aucun fichier sur le disque)

Qu'y a-t-il dans le répertoire courant ?

```
localhost<fl 59>pwd
```

```
/Volumes/HD-Mac OS 9.2/users/fl9/IUT/intro1A
```

```
localhost<fl 60> ls
```

list



```
IntroCours1    codeInt.c      listeNbrs      table.html
IntroCours2    brouillons    parOctet.c     tailleInt
IntroCours3    hello.html    plan           tailleInt.c
alphabet.txt   intro1A       recupCours1    touches
codeInt        intro1A.doc   source         touches.c
```

Une version plus informative

```
localhost<fl 60> ls -F
```

IntroCours1	codeInt.c	listeNbrs*	table.html
IntroCours2	brouillons/	parOctet.c	tailleInt*
IntroCours3	hello.html	plan	tailleInt.c
alphabet.txt	intro1A	recupCours1	touches*
codeInt*	intro1A.doc	source/	touches.c

répertoires

Exécutables

L'aide en ligne

La commande **man** décrit une commande, ou une fonction C

Taper man

Le prompt
(ne pas
taper)

> man _nom_recherché

Taper le nom de la commande
ou de la fonction recherchée

Ex :

> man pwd

PWD(1)

Manuel des commandes générales

PWD(1)

NOM : pwd - renvoie le nom du catalogue courant

SYNOPSIS : pwd [-LP]

← Crochet = facultatif

← options

DESCRIPTION

pwd écrit le chemin absolu du catalogue courant sur la sortie standard.

VOIR AUSSI

3 = Langage C

cd(1), csh(1), getcwd(3)

← 1 = Commandes

Options et arguments

- **Argument** : à quoi la commande est appliquée
- **Option** : modifie le comportement de la commande
- ex : ls **-aF** /etc/X11

Ranger les fichiers

- Copier (copy):
cp _fichier_ _nouveau_fichier_
- Déplacer (move):
mv _fichier_ _nouveau_fichier_
mv _fichier_ _catalogue_
- Détruire (remove):
rm _fichier_

- Exemple

localhost<fl 61>cp essai.html table.html

*(crée un second fichier table.html identique à essai.html.
Il y a donc deux fichiers distincts)*

localhost<fl 62>rm essai.html

(efface le fichier essai.html. Il ne reste que 1 des fichiers)

localhost<fl 63> mv table.html tab9.html

*(rebaptise le fichier table.html. Il n'y a toujours qu'un
fichier)*

localhost<fl 64>mv tab9.html ../TD

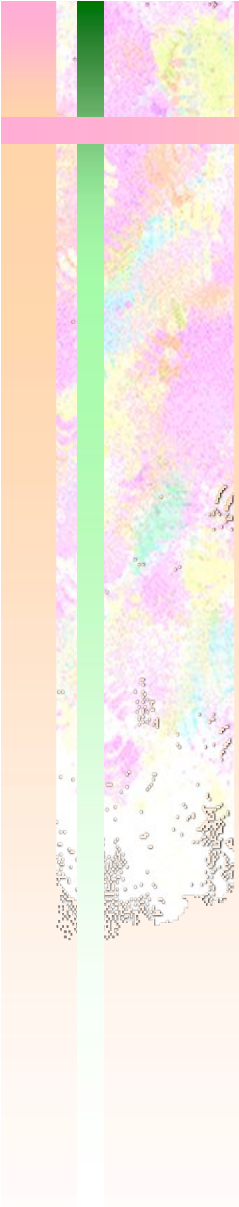
(déplace le fichier tab9.html dans un autre catalogue)


Les protections

- Trois droits possibles sur chaque fichier: lire (r), écrire (w), exécuter (x)
- Ces droits sont définis séparément pour l'utilisateur (u), son groupe (g), le reste du monde (o)
- On voit les droits en tapant ls -l

Changer les protections

- L'ensemble des droits s'appelle le *mode*, la commande qui les change est **chmod**
- Donner le droit de lecture sur `essai.c` :
> `chmod o+r essai.c`
- Retirer le droit d'écriture :
> `chmod o-w essai.c`
- Pour les répertoires, `x` est le droit d'accès

- 
- `chmod o+rx ~/introInfo/td2`
 - Encore faut-il pouvoir rentrer dans `introInfo`
 - Changement récursif sur tout le contenu : option `-R`
 - Donner à tous l'accès à tout `introInfo` :
 - > `chmod o+rx ~`
 - > `chmod -R o+rX ~/introInfo`(+X : on ne met le droit x pour o que s'il est déjà mis pour u ou g)



Un peu plus pour être efficace

- Caractères jocker

? autorise un seul caractère à cette place,

* autorise n'importe quelle suite de n'importe quoi



Exemples :

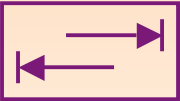
```
> mv essai.?tml ../TD
```

(déplace essai.html et essai.xml)

```
> rm *.html
```

(supprime tous les fichiers html qui restent).

Facilités d'écriture

- Compléter les noms de commande et les noms de fichier : utiliser la touche <tab>  (^d pour voir les choix)
- ex :
<localhost fl 67> ema<tab>
<localhost fl 68> ls Doc<tab>
- Rappeler les commandes déjà tapées :
<flèche haut>, <flèche bas>



Voir ce qu'il y a dans un fichier texte

- La commande *more* permet de lire les fichiers page à page
- *head* et *tail* affichent les premières et les dernières lignes.
- *cat* permet de mettre bout à bout plusieurs fichiers (concaténer)

Voir le manuel pour les options

Exemples

Brassens fl 34> `head -5 *.c`

(affiche les en-têtes de tous les fichiers C)

Brassens fl 35> `more longFichier.c`

(Affiche longFichier.c écran par écran, en laissant le temps de lire)

Brassens fl 36> `cat -n bizarre.txt`

(affiche bizarre.txt en numérotant les lignes)