

# Commandes du Shell

## Définition

Le shell (mot anglais signifiant « coquille ») est un logiciel fournissant une interface pour un utilisateur. Le shell est la partie la plus externe du système d'exploitation, c'est l'interface utilisateur du système d'exploitation.

## Les Shells graphiques

Explorateur des répertoires par les fenêtres: explorer ou nautilus

## Les Shells textuels

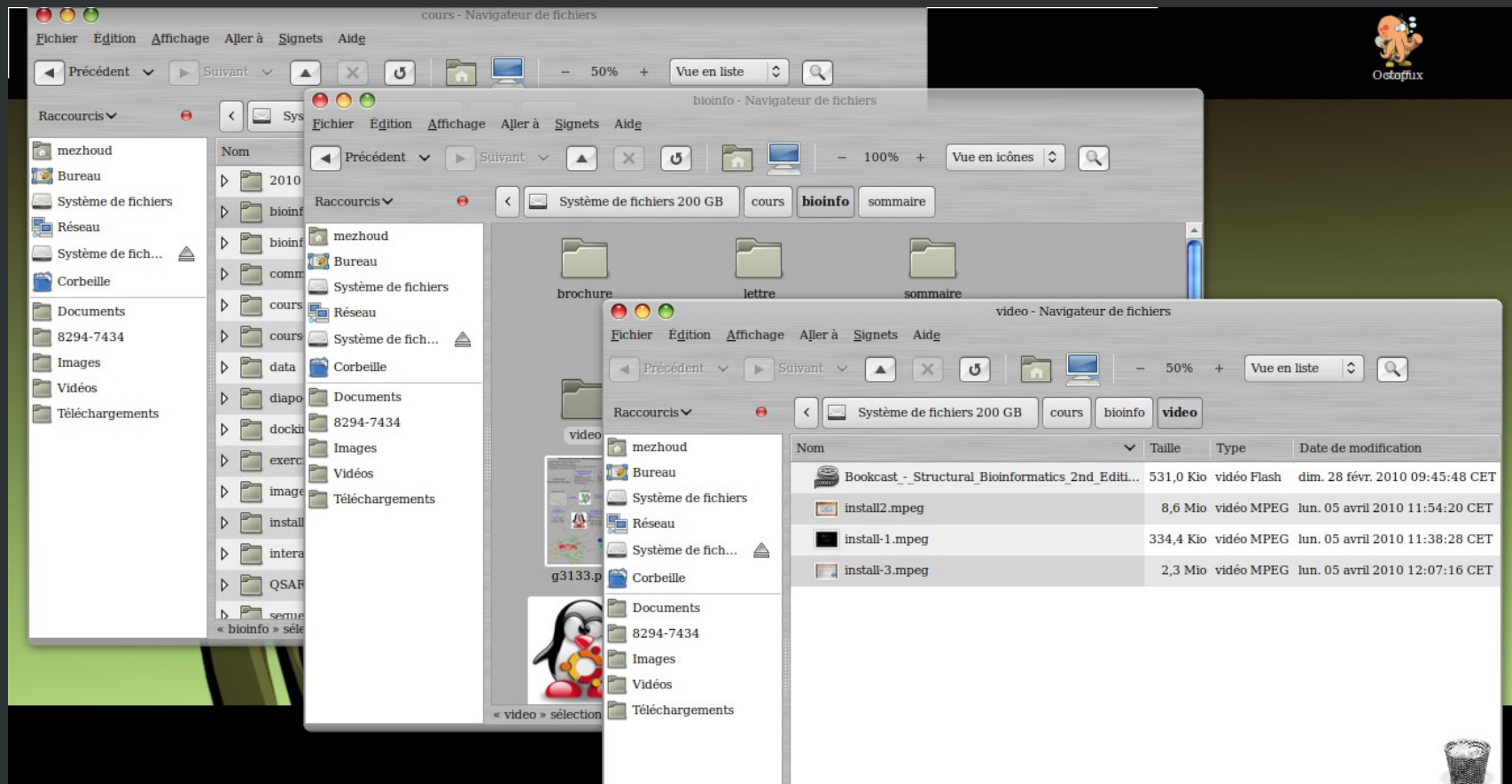
Explorateur des répertoires avec un terminal: Bash, csh



# Nautilus = shell graphique

## Les Shells graphiques

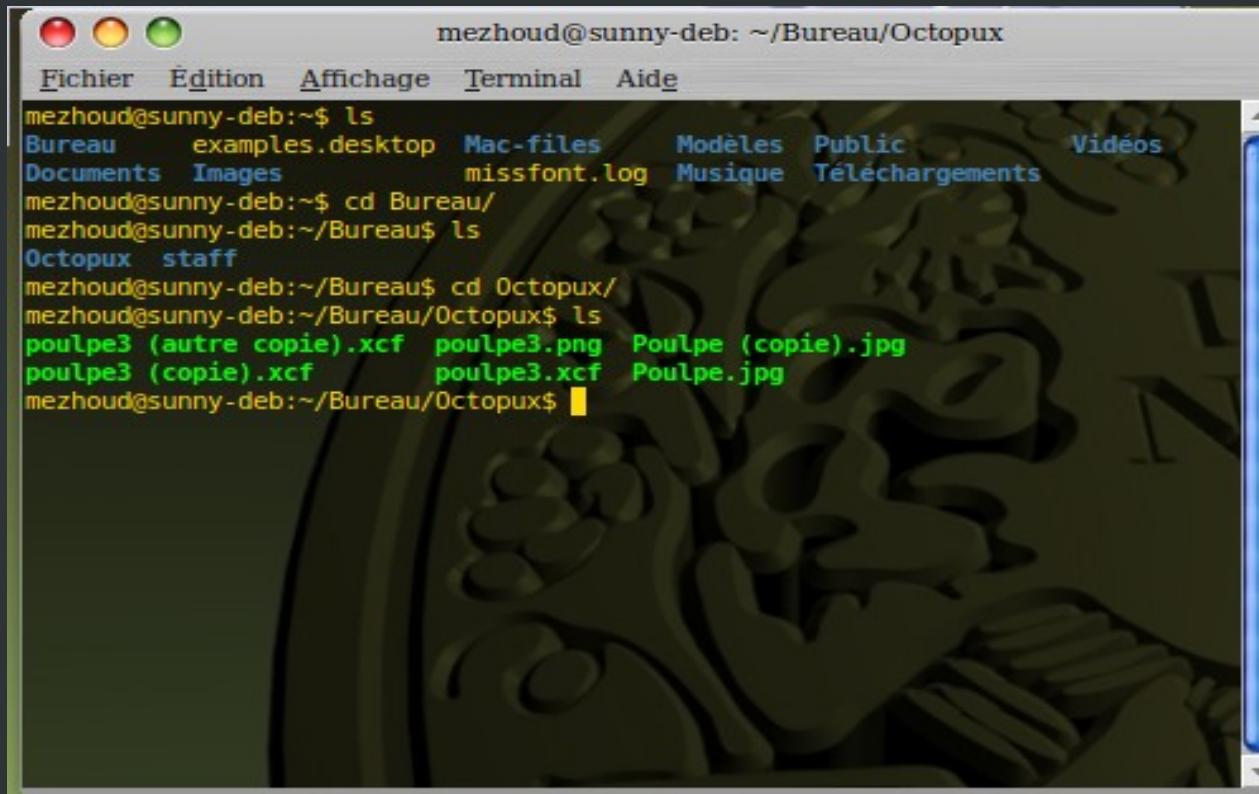
Explorateur des repertoires par les fenêtres: explorer ou nautilus



# Le Terminal

## Les Shells textuels

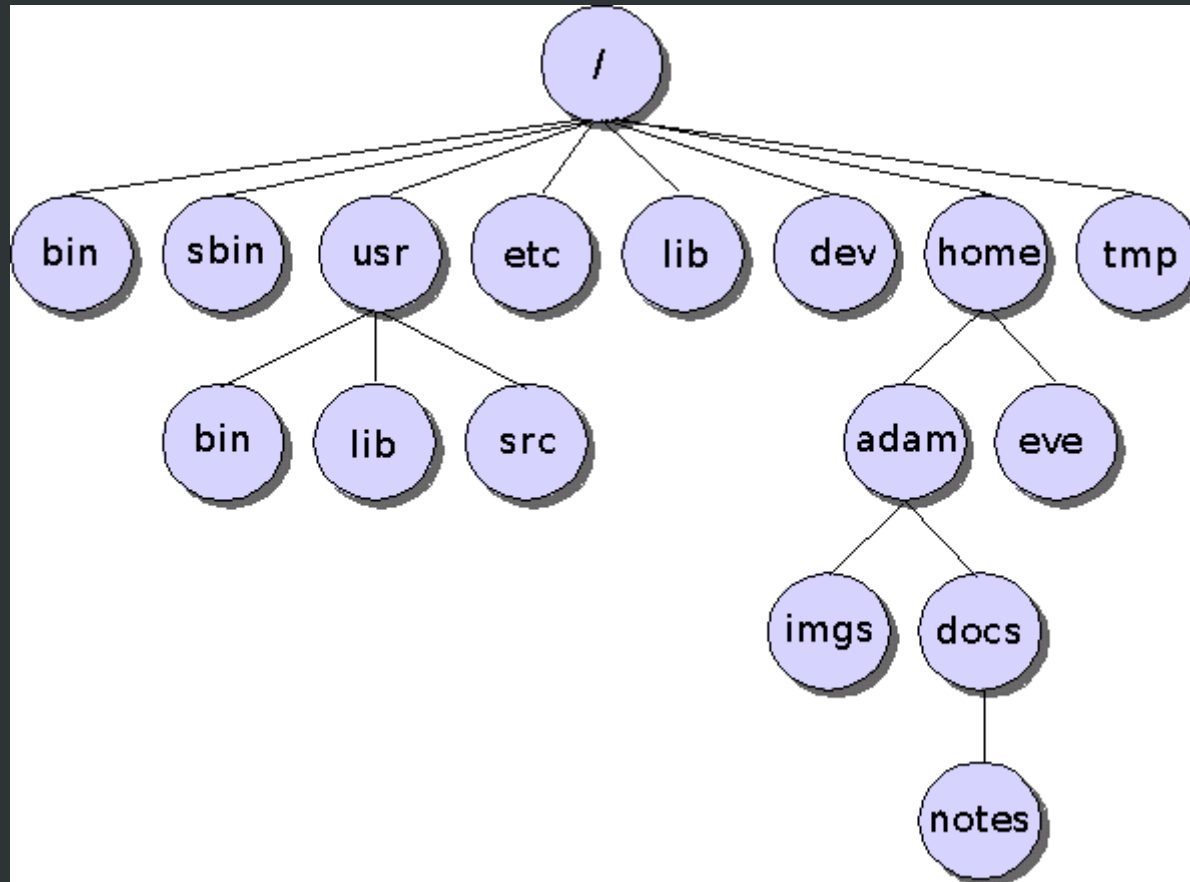
Explorateur des répertoires avec un terminal: Bash, csh



```
mezhoud@sunny-deb: ~/Bureau/Octopux
Fichier  Édition  Affichage  Terminal  Aide
mezhoud@sunny-deb:~$ ls
Bureau  examples.desktop  Mac-files  Modèles  Public  Vidéos
Documents  Images  missfont.log  Musique  Téléchargements
mezhoud@sunny-deb:~$ cd Bureau/
mezhoud@sunny-deb:~/Bureau$ ls
Octopux  staff
mezhoud@sunny-deb:~/Bureau$ cd Octopux/
mezhoud@sunny-deb:~/Bureau/Octopux$ ls
poulpe3 (autre copie).xcf  poulpe3.png  Poulpe (copie).jpg
poulpe3 (copie).xcf  poulpe3.xcf  Poulpe.jpg
mezhoud@sunny-deb:~/Bureau/Octopux$
```



# Arborescence du système Linux



# C'est quoi le shell?

- Le shell c'est un programme qui se trouve dans le répertoire /bin.
- Le shell permet d'exécuter des commandes, d'explorer l'arborescence du système, de créer, d'éditer et de supprimer des fichiers, etc.
- On distingue plusieurs shells
  - - le /bin/sh shell Bourne
  - - le /bin/bash shell Bourne Again SHell
  - - le /bin/csh C shell
  - - le /bin/ksh Korn shell
  - - le /bin/tcsh C shell amélioré
  - - le /bin/zsh/ Z shell



# Comment accéder à la ligne de commande?

- Pour accéder à la ligne de commande, il est possible d'utiliser un terminal (xterm, kterm, gterm) ou encore konsole.
  - Menu → accessoires → terminal
- En informatique, un terminal désigne un ensemble de périphériques de sortie (écran...) ou d'entrée (clavier, souris...), en quelque sorte l'extrémité d'un réseau.



# Pourquoi utiliser la ligne de commande ?

- Beaucoup de solutions sont données en ligne de commande. Pour certaines tâches, l'utilisation de la ligne de commande s'avère bien plus pratique et plus puissante que la fameuse souris



# Notions de commande

- Une commande est un fichier exécutable. L'exécution d'une commande peut être différente suivant le cas.

```
command -option argument
```



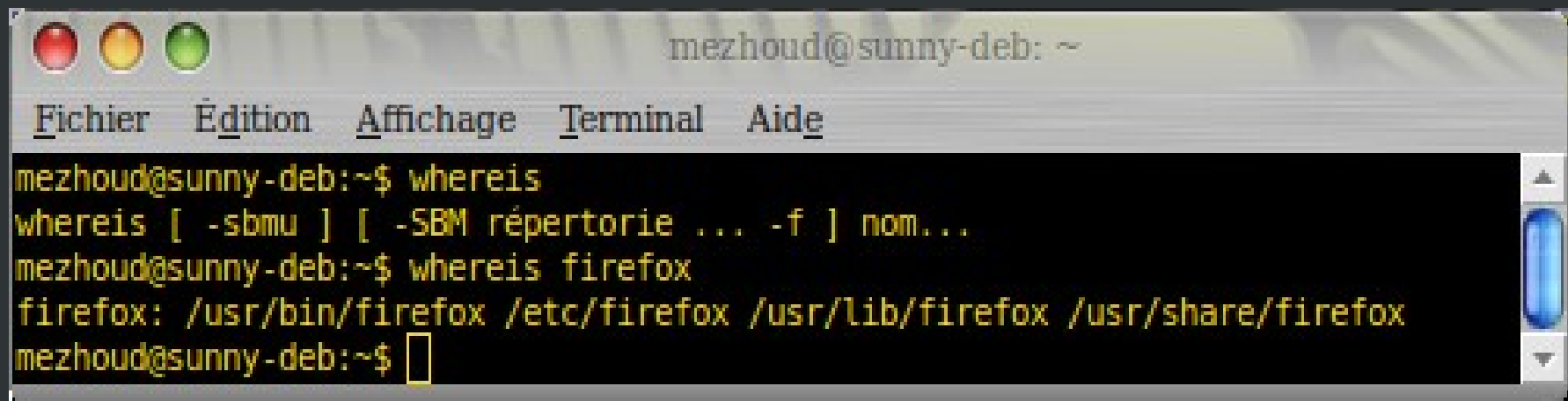


# Où se trouve les commandes?

Les commandes que vous pouvez exécuter depuis votre terminal se trouvent dans certains répertoires de votre système.

- Pour trouver l'emplacement d'une commande, on utilise "whereis" (en français "où est") :

```
whereis firefox
```



```
mezhoud@sunny-deb: ~  
Fichier  Édition  Affichage  Terminal  Aide  
mezhoud@sunny-deb:~$ whereis  
whereis [ -sbmu ] [ -SBM répertorie ... -f ] nom...  
mezhoud@sunny-deb:~$ whereis firefox  
firefox: /usr/bin/firefox /etc/firefox /usr/lib/firefox /usr/share/firefox  
mezhoud@sunny-deb:~$
```



# Commandes pour naviger dans les répertoires

<code>pwd</code>	Se localisé
<code>ls</code>	Lister ce qu'il y a
<code>cd</code>	Se déplacer dans /home
<code>cp</code>	copier
<code>rm</code>	Remove/ effacer
<code>mkdir</code>	Make directory
<code>touch</code>	Créer un fichier
<code>mv</code>	déplacer
<code>sudo</code>	Avoir les faveurs "administrateur"
<code>rmdir</code>	Effacer un dossier
<code>man</code>	manuel



# Recherche avec le terminal

## Dans le système de fichier

- **find**

```
find /home -name test
```

```
#Commande /répertoire -option fichier
```

```
Exemple:
```

```
sudo find / -name nom-utilisateur
```

- **locate**

```
locate nom-fichier
```

```
#Exemple:
```

```
locate bureau
```

```
locate Bureau
```

```
locate -i bureau (-i, insensible à la casse  
majuscule)
```



# Recherche avec le terminal

## Dans un fichier texte

- **grep ou egrep**

```
grep filtre fichier
```

Exemple:

```
grep P450 alldrugbase.fasta
```

```
grep -c P450 alldrugbase.fasta (print  
number line)
```

- **cat**

```
cat nom-fichier
```

#Exemple:

```
cat alldrugbase.fasta
```

```
cat -n alldrugbase.fasta
```



# Manipuler des fichiers avec le terminal

## Flux et redirection

- Le chevron >

Permet de rediriger le flux de sortie d'une commande vers une destination

```
echo truc > /home/session/Bureau/null
```

Explication: le chevron enverra le flux de la commande "echo truc" vers le fichier "null" contenu dans le Bureau.

```
echo truc >> ~/Bureau/null
```

Quand le chevron est mis deux fois à la suite, >>, la sortie est écrite dans le fichier sans l'écraser, mais en le continuant.

```
echo raja > fifi.txt  
gedit fifi.txt &  
echo est belle >> fifi.txt
```



# Manipuler des fichiers avec le terminal

## Flux et re direction

- Le tube ou Pipe |

La syntaxe:

cmd 1 | cmd 2, permet à la cmd 2 de travailler sur ce que la cmd1 renvoie.

```
ls /home
```

```
ls /home | grep me
```

```
ls /home | grep me | wc -l
```



# Les expressions rationnelles

## Regular Expression (REGEX)

### Définition:

Une expression rationnelle est composée d'une suite de caractères appelée 'motif' ou 'pattern' en anglais, qui est utilisée pour décrire une chaîne de caractère avec pour but de la retrouver dans un texte



# Les expressions rationnelles Regular Expression (REGEX)

**Le Point "."** .....N'importe quel caractère (sauf fin de ligne)

**a.c.**

**La paire de crochets "[ ]"** .....Séquence de caractères

**U****[uoe]nt[uo]**      **[0-9a-zA-Z]**      **[^0-9]** Exclure les chiffres

**L'astérisque "\*"**  .....Est un quantificateur qui indique une répétition du caractère ou élément qui le précède.....0 ou plusieurs fois

**Π = 3,14[0-9]\*    protein[a-z]\***

**Le signe plus "+"**  .....Est un quantificateur qui indique une répétition du caractère ou élément qui le précède.....1 ou plusieurs fois





# Les expressions rationnelles Regular Expression (REGEX)

**La paire de parenthèses "()"** .....est utilisée à la fois pour grouper certains éléments d'une expression régulière

**(a).c(...)** = a.c...      **(ab)\*** = abababab      **a(b)\*** = abbbbbbb

**Le point d'interrogation "?"** .....joue le rôle de quantificateur. Il signifie 0 ou 1 l'occurrence d'un symbole ou expression

```
egrep -c 'AA(RR)?' alldrug.fasta
```

**L'accent circonflexe "^"** .....Ancre de positionnement au début de ligne  
.....Négation d'une classe entre crochets

```
egrep -c '^>' alldrugbase.fasta
```

**Le dollar "\$"** .....Ancre de positionnement  
.....Positionnement en fin de ligne ou chaîne

```
egrep ' D$' alldrugbase.fasta  
^$      ^.*$      ^.+ $
```



# Les expressions rationnelles Regular Expression (REGEX)

**Le congrès-oblique "\** .....Symbole d'achappement des métacaractères

**. ? \$ ^ [ ] ( ) \* + | { }** .....ce sont des Métacaractères

**Le congrès-oblique entier "\n** Faire appelle à la nième parenthèse

**(Quatre) cent \1 = Quatre cent Quatre**

**La barre verticle "|** .....Alternative entre plusieurs termes

```
cat alldrub.fasta | egrep 'AAA|AAR'
```

**Accolade "{ }** Quantificateur avec précision du nombre de répétition

- 1) egrep 'A{5}' fichier = filtrer AAAAA
- 2) egrep 'A{,5}' fichier = filter au plus AAAAA
- 3) egrep '{5,}' fichier = filter au moins AAAAA



# Les expressions rationnelles Regular Expression (REGEX)

**Le congrès-oblique "\** .....Symbole d'achappement des métacaractères

**. ? \$ ^ [ ] ( ) \* + |** .....Se sont des Métacaractères

**La barre verticale "|** .....Alternative entre plusieurs termes

**[a-z]+\.(gif|jpeg|png)|**

**Les caractères de contrôle**

**\n** .....Fin de ligne

**\r** .....Retour chariot

**\t** .....Tabulation horizontale

**\w** .....Classe des caractères alphabétiques

**\d** .....Classe des nombres entiers

**\s** .....Classe des caractères d'espacement



# Les expressions rationnelles

## Regular Expression (REGEX)

### exemples:

```
cat alldrugbase.fasta | egrep 'DB[0-9]{5}'
```

```
grep ">\w*\|[0-9]*\s[^(]*[A-Z0-9;]" /mezhoud/Bureau/alldrugbase.fasta
```

```
find /home/mezhoud/Bureau -regex ".*\.(doc|png)"
```

```
find /home/mezhoud/Bureau -iname "*.doc" -or -iname "*.png"
```



# Les expressions rationnelles

## Regular Expression (REGEX)

**Table:** Regular expression special characters

Test	Special Characters
start	^
end	\$
any single character	.
set of characters	[ccc]
set of characters not equal	[^ccc]
range of characters	[c-c]
range of characters not equal	[^c-c]
zero or one of previous character	?
zero or multiple of previous characters	*
one or multiple of previous characters	+
OR operator	

**Table:** Examples of regular expressions

Test	Operation
begins with D	~ '^D'
contains D	~ 'D'
D in second position	~ '^.,D'
begins with D and contains e	~ '^D.*e'
begins with D, contains e, and then f	~ '^D.*e.*f'
contains A, B, C, OR D	~ '[A-D]' OR ~ '[ABCD]'
contains A OR a	~* 'a' OR ~ '[Aa]'
does not contain D	!~ 'D'
does not begin with D	!~ '^D' OR ~ '^[^D]'
begins with D, with one optional leading space	~ '^ ?D'
begins with D, with optional leading spaces	~ '^ *D'
begins with D, with at least one leading space	~ '^ +D'
ends with G, with optional trailing spaces	~ 'G *\$'

