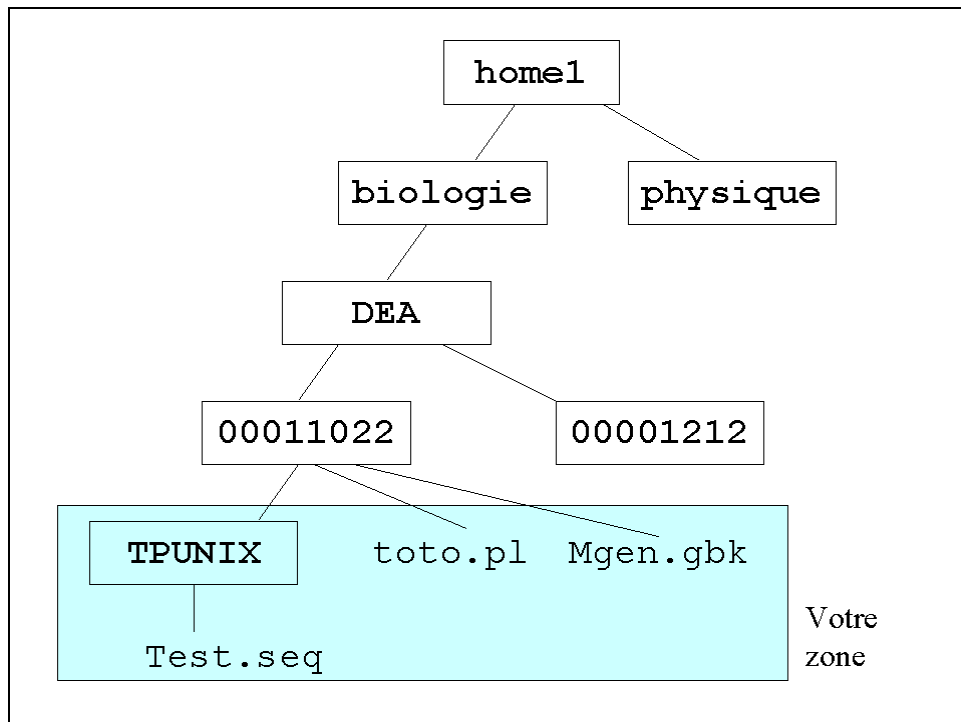


# **Notions d'Unix**

**Daniel Gautheret**

**2003-2004.2**

# Les Chemins sous UNIX



Un exemple d'arborescence Unix

1. Chemin absolu:

**/home1/biologie/dea/00011022/toto.pl**

2. Chemin relatif (pour un utilisateur dans 00011022):

**toto.pl**

3. Chemin relatif (pour un utilisateur dans TPUNIX):

**../toto.pl**

4. Chemin du répertoire courant:

**.**

# Les Principales commandes UNIX

D'après [l'abrégé Unix de Annie Danzart](#)

Nom	Description	Options	Arguments
<b>Gestion fichiers et répertoires</b>			
<a href="#">cd</a>	se positionne sur le répertoire désigné		chemin d'accès à un répertoire
<a href="#">ls</a>	liste le contenu d'un répertoire	-a : prise en compte des fichiers cachés -F : renseigne sur le type de fichier (*, /, @) -i : précision du numéro d'inode des fichiers -R : liste récursivement le contenu du répertoire -l : informations détaillées -g : ajout du nom du groupe -d : renseigne sur le répertoire lui-même -t : liste par date de modification -u : liste par date d'accès -r : ordre inverse	nom de répertoire
<a href="#">pwd</a>	retourne la référence absolue du répertoire courant		
<a href="#">cp</a>	copie du fichier source vers la destination	-i : demande confirmation -p : conservation de date et droits -r : recopie récursive d'un répertoire	source - destination
<a href="#">mv</a>	déplace fichiers et répertoires	-i : demande confirmation	source - destination
<a href="#">rm</a>	supprime la référence du fichier dans le répertoire	-f : force la commande sans s'occuper des droits -i : demande confirmation -r : destruction récursive	nom de fichier ou de répertoire
<a href="#">mkdir</a>	création d'un répertoire		nom de répertoire
<a href="#">rmdir</a>	suppression d'un répertoire vide		nom de répertoire

<a href="#"><u>cat</u></a>	imprime le contenu du fichier mentionné		nom de fichier
<a href="#"><u>more</u></a>	liste le contenu d'un fichier page par page		nom de fichier
<b>Divers</b>			
<a href="#"><u>man</u></a>	retourne le mode d'emploi de la commande s'il existe		nom de commande
<a href="#"><u>gzip</u></a>	compresse un fichier (voir compress) par défaut, la destination est la sortie standard qu'on redirige vers un fichier si on veut obtenir un fichier compressé	- <b>c</b> : résultat sans modification du fichier source - <b>f</b> : écrasement d'un fichier compressé préexistant - <b>v</b> : taux de compression - <b>d</b> : force une décompression - <b>9</b> : niveau maximum de compression	nom de fichier
<a href="#"><u>gunzip</u></a>	décompresse un fichier	- <b>c</b> : le résultat est produit sans modification du fichier d'origine - <b>v</b> : taux de compression	nom de fichier
<a href="#"><u>ps</u></a>	liste des processus BSD	- <b>a</b> : processus détenus par les autres utilisateurs - <b>u</b> : nom du propriétaire du processus - <b>x</b> : processus sans terminal associé - <b>l</b> : description complète - <b>t</b> : terminal particulier	
<a href="#"><u>kill</u></a>	envoi d'un signal à un processus	- <b>l</b> : liste des signaux disponibles - <b>n</b> : numéro du signal à envoyer <b>2</b> : INT (ctrl-c : ) <b>3</b> : QUIT (ctrl-\) <b>9</b> : KILL	numéro du process
<a href="#"><u>sort</u></a>	tri des lignes du fichier suivant l'ordre ASCII	- <b>r</b> : tri inverse - <b>u</b> : élimine les lignes identiques	fichier
<a href="#"><u>tar</u></a>	archivage/désarchivage de fichiers	<b>c</b> : création de l'archive <b>x</b> : restauration de l'archive <b>t</b> : listage du contenu de l'archive <b>v</b> : mode verbeux <b>p</b> : préserve dates et droits à la restauration	archive - répertoire ou fichiers

		<b>f</b> : le nom du fichier spécifié sera celui de l'archive	
<a href="#"><u>telnet</u></a>	établir une connexion avec une machine distante(sortie : exit)		nom de machine
<a href="#"><u>which</u></a>	retourne le chemin complet d'accès à une commande		nom de la commande
<a href="#"><u>who</u></a> <a href="#"><u>(am i)</u></a>	retourne le nom des utilisateurs qui ont ouvert une session		
<a href="#"><u>chmod</u></a>	change les permissions en lecture, écriture, exécution	<b>-R</b> change les droits récursivement à partir du noeud (mode = ugo)	mode - nom de fichier ou de répertoire

## Les Propriétés des fichiers

Résultat de la commande ls -l:

```

drwxr-xr-x  2 root    root      4096 Oct  9 16:44 docs
drwxr-xr-x  2 root    root      4096 Oct  9 16:44 vms
drwxr-xr-x  6 root    root      4096 Oct  9 16:44 widgets-mesa
drwxr-xr-x  2 root    root      4096 Oct  9 16:44 widgets-sgi
-rwxr-xr-x  1 2001    2001    139227 Jul 20 17:25 configure
-rw-r--r--  1 2001    2001    12685 Jul 20 17:25 Makefile.in
-rw-r--r--  1 2001    2001    18838 Jul 20 17:25 aclocal.m4
-rw-r--r--  1 2001    2001    13920 Jul 20 17:24 configure.in
-rw-r--r--  1 2001    2001    25326 Jul 18 15:36 Makefile.X11
-rwxr-xr-x  1 2001    2001     5598 Jul 18 02:19 install-sh
-rwxr-xr-x  1 2001    2001     6283 Jul 18 02:19 missing
-rwxr-xr-x  1 2001    2001       722 Jul 18 02:19 mkinstalldirs
-rw-r--r--  1 2001    2001         0 Jul 18 02:19 stamp-h.in
-rw-r--r--  1 2001    2001    1060 Jul 18 02:19 conf.h.in

```

# Trucs utiles sous Unix

## ***La redirection des sorties ">"***

Envoie le résultat d'une commande dans un fichier (au lieu de la sortie standard : l'écran)

```
ls > fichier
```

```
blast seq1.fasta seq2.fasta > blastout
```

```
cat titi > toto
```

```
cat titi >> toto (>> n'écrase pas le fichier de sortie)
```

## ***La redirection des entrées "<"***

```
Fasta < inputfile
```

Lit les entrées dans un fichier (au lieu de l'entrée standard : le clavier)

(inputfile contient toutes les réponses aux questions posées par le programme.)

## ***Le pipeline "|"***

Utilise le résultat d'une commande comme argument d'une autre commande

```
grep "ACCGAAG" ecoli.fasta | wc
```

```
blast seq1.fasta seq2.fasta | more
```

## ***Le contrôle des tâches***

"&" permet de récupérer le shell après le lancement d'un programme

```
netscape &
```

```
nedit &
```

Sans "&", le shell reste bloqué tant que le programme tourne. On peut toujours suspendre l'exécution d'un programme avec CTRL-Z, puis le redémarrer en arrière-plan avec "bg", ou en premier plan avec "fg".

La commande "jobs" permet de voir la liste des tâches en cours d'exécution (tâches nous appartenant)

La commande "kill < no de tâche > " tue une tâche en cours.

La commande "top" permet de visualiser toutes les tâches en cours d'exécution, y compris celles des autres utilisateurs.

# La commande GREP

"grep" est un *filtre*, c'est à dire un programme qui selectionne automatiquement les lignes d'un fichier possédant telle ou telle propriété. Le filtre "grep" permet de sélectionner les lignes contenant une certaine expression régulière, c'est à dire un motif flexible décrivant un ensemble de chaînes de caractères.

Par exemple, l'expression "...di" décrit les chaînes "lundi" et "mardi" (et toute autre chaîne de 3 caractères se terminant par "di"). L'expression "[Pp]hosphorylase" décrit les chaînes "Phosphorylase" et "phosphorylase"

La commande "grep" a la forme:

```
grep <expression régulière> <fichier>
```

(si l'on veut afficher toutes les lignes répondant à l'expression), ou bien

```
grep -c <expression régulière> <fichier>
```

(si l'on veut juste compter toutes les occurrences).

L'option -i permet d'ignorer les majuscules/minuscules. Ex :

```
grep -i "(adn)|(arn)" fichier
```

Voici les caractères que l'on peut utiliser dans les expressions régulières Unix.

.	Tout caractère (sauf newline)
^	Le début d'une ligne
\$	La fin d'une ligne
	Choix. A B: A ou B
()	Groupement de caractères
[]	Classe de caractères. [AGUC]: A,G,U ou C
\	Avant un caractère spécial qu'on ne veut pas prendre en compte comme tel car il fait partie de la chaîne recherchée
Les commandes suivantes sont à placer après le caractère concerné	
*	0 fois ou plus
+	une fois ou plus
?	une fois ou zero
{n}	exactement n fois
{n,}	au moins n fois
{n,m}	de n a m fois

# Les Shells

Les shell est l'environnement permettant d'entrer des commandes UNIX. Il existe plusieurs types de shell, notamment : csh, tcsh et bash (bash est souvent le shell par défaut sous Linux).

Selon le shell utilisé, on dispose de plus ou moins de facilité pour entrer des commandes. Par exemple, tcsh et bash permettent de revenir aux commandes précédentes avec les flèche-haut ou les touches CTRL-R + une chaîne de caractères à rechercher.

## ***Les variables d'environnement***

Le fonctionnement du shell est contrôlé par des variables d'environnement, qui stockent des informations comme :

- l'endroit où se trouvent certains programmes (variable PATH)
- l'endroit où se trouvent certaines bases de données,
- etc.

On peut définir le contenu de ces variables dans un fichier qui est lu automatiquement lors de l'ouverture d'un nouveau shell. La façon d'affecter ces variables diffère selon les shell :

### **Pour un shell Tcsh : Fichier `.tcshrc`**

```
setenv PATH ./~/Bin/:$PATH
setenv PLPLOT_LIB /usr/local/share/EMBOSS
```

### **Pour un shell Bash : Fichier `.bashrc`**

```
export PATH=./usr/X11R6/bin:~/Bin/:$PATH
export PLPLOT_LIB=/usr/local/share/EMBOSS
```

Après modification d'un fichier `.tcshrc` ou `.bashrc`, les changements peuvent être pris en compte au moyen de :

```
source .bashrc
ou
source .tcshrc
```

ou encore en ouvrant un nouveau shell.