

Les bases du shell

Sébastien GAGNÉ, Université d'Orléans

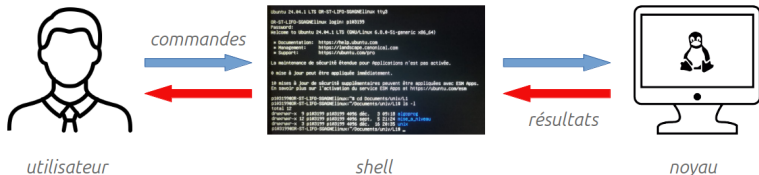
L1 Pratique du système UNIX — S2

shell et UNIX

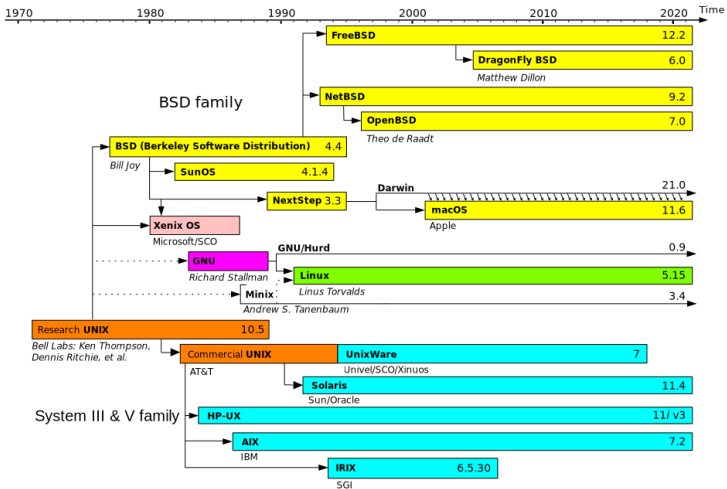
UNIX est un système d'exploitation multi-tâches et multi-utilisateurs, qui date des années 1970.

Le noyau du système (le **kernel**) gère les ressources et permet aux composants de communiquer.

Le **shell** est une interface entre l'utilisateur et le noyau. L'utilisateur lance des commandes via le shell.



La famille UNIX



source : https://fr.wikipedia.org/wiki/Type_Unix

Sitographie - Histoire d'UNIX

Genèse d'UNIX

- <https://tuteurs.ens.fr/unix/histoire.html>
- <https://fr.wikipedia.org/wiki/Unix>

Type UNIX

- https://fr.wikipedia.org/wiki/Type_Unix

Les co-créateurs d'UNIX

- https://fr.wikipedia.org/wiki/Ken_Thompson
- https://fr.wikipedia.org/wiki/Dennis_Ritchie

En mode console

Avant les interfaces graphiques, le shell était LE moyen de communication entre utilisateur et noyau.

```
Ubuntu 24.04.1 LTS OR-ST-LIFO-SGAGNElinux tty3
OR-ST-LIFO-SGAGNElinux login: p103199
Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

La maintenance de sécurité étendue pour Applications n'est pas activée.
0 mise à jour peut être appliquée immédiatement.

10 mises à jour de sécurité supplémentaires peuvent être appliquées avec ESM Apps.
En savoir plus sur l'activation du service ESM Apps at https://ubuntu.com/esm

p103199@OR-ST-LIFO-SGAGNElinux:~$ cd Documents/univ/L1
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1$ ls -l
total 12
drwxrwxr-x  9 p103199 p103199 4096 déc.  3 09:18 algoprogram
drwxrwxr-x 12 p103199 p103199 4096 sept.  5 21:24 mise_a_niveau
drwxrwxr-x  3 p103199 p103199 4096 déc. 16 20:35 unix
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1$ _
```

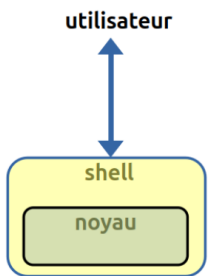
Les shell classiques

- Le shell historique : le **Bourne shell** **sh** (1977, Stephen Bourne) équipe la version 7 d'UNIX ;
- Le shell **cs****h** (1978, Bill Joy, Berkeley). Il permet d'accéder à l'historique des commandes passées ;
- Le shell **k****sh** (1983, David Korn). Ajoute des fonctions de script élaborées ;
- Le **B**ourne-**a**gain **s****h**ell ou **ba****sh** (*projet GNU*, 1988) : le shell par défaut dans nos salles TP.

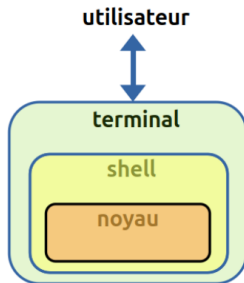
Utilisation d'un terminal

Le **terminal** est une **interface graphique** qui permet l'émulation de la console. C'est une console "graphique".

Il communique les commandes au shell, qui les exécute. Le shell reçoit les réponses et les retransfère au terminal.



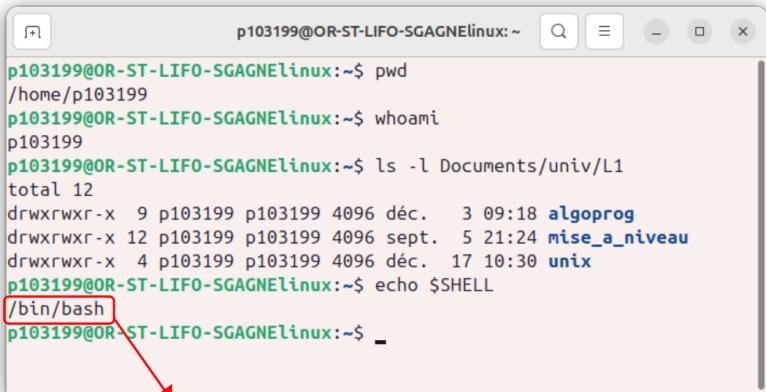
sans interface graphique



avec interface graphique

Utilisation d'un terminal

Le terminal installé sur nos machines est GNOME terminal



The screenshot shows a terminal window titled "p103199@OR-ST-LIFO-SGAGNElinux: ~". The user p103199 has executed several commands: `pwd` (returns /home/p103199), `whoami` (returns p103199), and `ls -l Documents/univ/L1` (lists files with permissions, owner, group, size, date, and filename). The user then enters `echo $SHELL`, which outputs `/bin/bash`. This output is highlighted with a red box, and a red arrow points from it to a text box below the terminal.

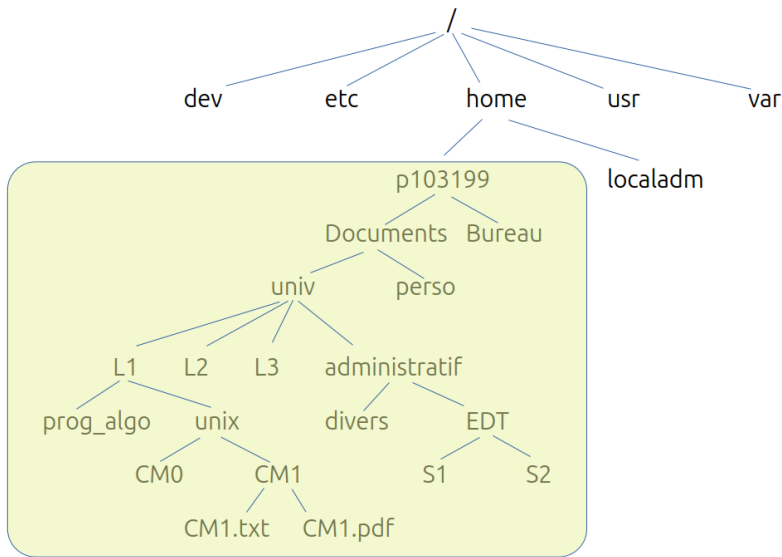
```
p103199@OR-ST-LIFO-SGAGNElinux:~$ pwd
/home/p103199
p103199@OR-ST-LIFO-SGAGNElinux:~$ whoami
p103199
p103199@OR-ST-LIFO-SGAGNElinux:~$ ls -l Documents/univ/L1
total 12
drwxrwxr-x  9 p103199 p103199 4096 déc.  3 09:18 algoprogram
drwxrwxr-x 12 p103199 p103199 4096 sept.  5 21:24 mise_a_niveau
drwxrwxr-x  4 p103199 p103199 4096 déc. 17 10:30 unix
p103199@OR-ST-LIFO-SGAGNElinux:~$ echo $SHELL
/bin/bash
p103199@OR-ST-LIFO-SGAGNElinux:~$ _
```

le shell par défaut pour
l'utilisateur **p103199**

Arborescence de fichiers

- Le système de fichiers est présenté sous la forme d'une **arborescence unique** ;
- La racine de cette arborescence est notée `/` ;
- Cette arborescence est composée de fichiers et de répertoires ;
- Tout fichier ou répertoire peut être désigné par un unique **chemin absolu** depuis la racine ;
- On peut accéder à tout emplacement à partir de n'importe où (**chemin relatif**).

Exemple d'arborescence



Chemin relatif / absolu

- Chemin absolu vers le fichier CM1.pdf :

```
/home/p103199/Documents/univ/L1/unix/CM1/CM1.pdf
```

- Chemin relatif de **prog_algo** vers **S2** :

```
../../administratif/EDT/S2
```

- Chemin relatif de **CM1.txt** vers **CM1.pdf** :

```
./CM1.pdf
```

•

désigne le répertoire courant

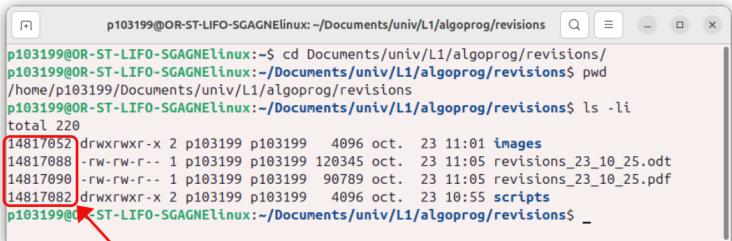
• •

désigne le répertoire parent

Les fichiers UNIX

Un **fichier UNIX** est caractérisé par :

- son nom ;
- le chemin absolu qui mène à lui ;
- son **inode** représenté par un numéro d'inode ;

A terminal window titled 'p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/algoprog/revisions'. The user has run 'cd Documents/univ/L1/algoprog/revisions/' and 'pwd', showing the current directory. Then they run 'ls -li', which lists files with their inodes. The first three lines of the output are highlighted with a red box: '14817052 drwxrwxr-x 2 p103199 p103199 4096 oct. 23 11:01 images', '14817088 -rw-rw-r-- 1 p103199 p103199 120345 oct. 23 11:05 revisions_23_10_25.odt', and '14817082 drwxrwxr-x 2 p103199 p103199 4096 oct. 23 10:55 scripts'. A red arrow points from the text below to the first line of this list.

```
p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/algoprog/revisions
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/algoprog/revisions$ cd Documents/univ/L1/algoprog/revisions/
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/algoprog/revisions$ pwd
/home/p103199/Documents/univ/L1/algoprog/revisions
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/algoprog/revisions$ ls -li
total 220
14817052 drwxrwxr-x 2 p103199 p103199 4096 oct. 23 11:01 images
14817088 -rw-rw-r-- 1 p103199 p103199 120345 oct. 23 11:05 revisions_23_10_25.odt
14817090 -rw-rw-r-- 1 p103199 p103199 90789 oct. 23 11:05 revisions_23_10_25.pdf
14817082 drwxrwxr-x 2 p103199 p103199 4096 oct. 23 10:55 scripts
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/algoprog/revisions$ _
```

chaque "fichier" a
son numéro d'inode

Les inodes

- Un **inode** est une **structure de données** correspondant à un fichier et contenant de nombreux champs qui le caractérisent ;
- Chaque fichier a son **numéro d'inode** qui l'identifie dans le **système de fichiers** ;
- Par contre, plusieurs fichiers peuvent avoir le même numéro d'inode. Dans ce cas, ces fichiers sont des **liens physiques** du même fichier ;
- L'inode contient notamment les blocs d'adresses et de données définissant le contenu du fichier ;

Les types de fichiers

Sous UNIX, tout est fichier. Les principaux types sont

- les **fichiers classiques** : concrètement, des séquence d'octets quelconques sur un espace de stockage : **CM1.pdf** ou **CM1.txt** ;
- les **répertoires** : **Documents**, **univ**, **L1**, ... ;
- les **liens symboliques** : des alias définis par un chemin (des fichiers contenant un chemin) ;

Les types de fichiers

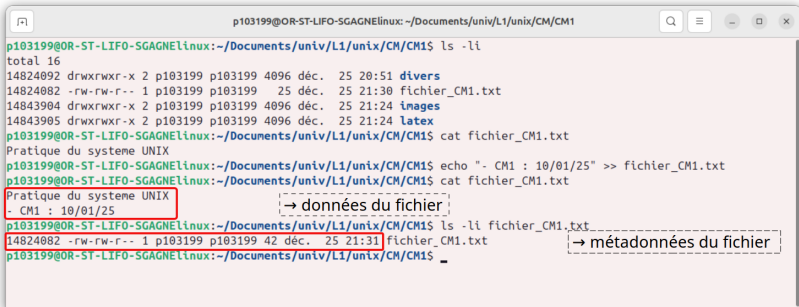
```
p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/unix/CM
p103199@OR-ST-LIFO-SGAGNElinux:~$ cd Documents/univ/L1/unix/CM
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM$ ls
CM0 CM1 cours1 pdf planning_CM.ods
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM$ ls -li
total 24
14843762 drwxrwxr-x 4 p103199 p103199 4096 déc. 15 20:22 CM0
14843903 drwxrwxr-x 4 p103199 p103199 4096 déc. 22 21:33 CM1
14818718 lrwxrwxrwx 1 p103199 p103199 12 déc. 22 21:40 cours1 -> pdf/CM_1.pdf
14843893 drwxrwxr-x 2 p103199 p103199 4096 déc. 22 21:38 pdf
14818715 -rw-rw-r-- 1 p103199 p103199 11470 déc. 22 21:29 planning_CM.ods
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM$ _
```

d → répertoire (directory)
l → lien symbolique
- → fichier classique

- D'autres types de fichiers existent. Voir plus tard.
- **ls -li** affiche les détails et le numéro d'inode.

Données et métadonnées

- Un fichier "classique" contient des **données**.
Par exemple un fichier texte qui contient les caractères "Pratique du système UNIX" ;
- Il possède aussi des **métadonnées** qui sont des renseignements divers sur le fichier.



A terminal window titled 'p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/unix/CM/CM1'. The terminal shows the following commands and output:

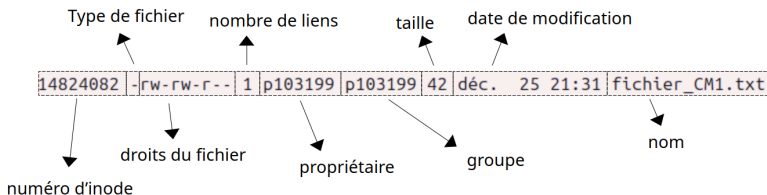
```
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ ls -li
total 16
14824092 drwxrwxr-x 2 p103199 p103199 4096 déc. 25 20:51 divers
14824082 -rw-rw-r-- 1 p103199 p103199 25 déc. 25 21:30 fichier_CM1.txt
14843904 drwxrwxr-x 2 p103199 p103199 4096 déc. 25 21:24 images
14843905 drwxrwxr-x 2 p103199 p103199 4096 déc. 25 21:24 latex
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ cat fichier_CM1.txt
Pratique du systeme UNIX
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ echo "- CM1 : 10/01/25" >> fichier_CM1.txt
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ cat fichier_CM1.txt
Pratique du systeme UNIX
- CM1 : 10/01/25
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ ls -li fichier_CM1.txt
14824082 -rw-rw-r-- 1 p103199 p103199 42 déc. 25 21:31 fichier_CM1.txt
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ _
```

Annotations on the terminal output:

- A red box highlights the file content: `Pratique du systeme UNIX` and `- CM1 : 10/01/25`. A dashed box to the right points to it with the text `→ données du fichier`.
- A red box highlights the file metadata line: `14824082 -rw-rw-r-- 1 p103199 p103199 42 déc. 25 21:31 fichier_CM1.txt`. A dashed box to the right points to it with the text `→ métadonnées du fichier`.

Détails des métadonnées

Les métadonnées importantes du fichier :



Permissions sur les fichiers

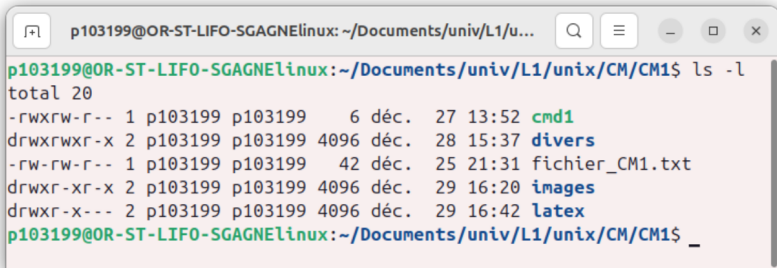
Un mécanisme de permissions gère "qui peut faire quoi" sur les fichiers.

Les permissions par paquets de 3 caractères : les 3 premiers pour l'utilisateur (u), les 3 suivants pour son groupe (g) et les 3 derniers pour les autres (o).

permission	fichier	répertoire
r	lire le contenu	listier les fichiers
w	modifier le contenu	supprimer / créer des fichiers
x	exécuter le fichier	traverser le répertoire
-	pas de permission	pas de permission

Permissions sur les fichiers

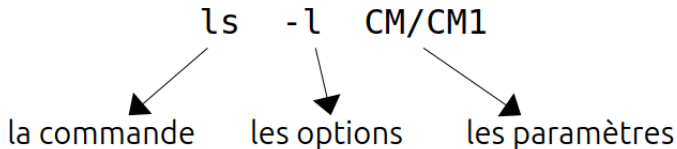
Analysez les permissions sur les fichiers et répertoires listés ci-dessous, pour l'utilisateur p103199, pour le groupe p103199 et pour tous les autres utilisateurs :

A terminal window titled 'p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/u...' displays the output of the 'ls -l' command. The output lists five files with their permissions, owner, group, size, date, and name. The permissions are: -rwxrw-r-- for cmd1, drwxrwxr-x for divers, -rw-rw-r-- for fichier_CM1.txt, drwxr-xr-x for images, and drwxr-x--- for latex. The user p103199 is the owner of all files, and the group p103199 is the group for cmd1, fichier_CM1.txt, and latex. The group 4096 is the group for divers and images. The terminal prompt is p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1\$ and the cursor is on the next line.

```
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ ls -l
total 20
-rwxrw-r-- 1 p103199 p103199  6 déc. 27 13:52 cmd1
drwxrwxr-x 2 p103199 p103199 4096 déc. 28 15:37 divers
-rw-rw-r-- 1 p103199 p103199 42 déc. 25 21:31 fichier_CM1.txt
drwxr-xr-x 2 p103199 p103199 4096 déc. 29 16:20 images
drwxr-x--- 2 p103199 p103199 4096 déc. 29 16:42 latex
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1$ _
```

Commande UNIX de base

Une commande UNIX de base est lancée avec des **options** (ou non) et des **paramètres** (ou non) :



La commande est interprétée par le shell qui interagit avec le noyau et produit (parfois affiche) les résultats.

Nous avons déjà exécuté dans ce cours quelques commandes : `ls`, `pwd`, `whoami`, `cd`, `cat`, `echo`.

Premières commandes UNIX

- **whoami** affiche l'utilisateur actuel ;
- **pwd** donne le chemin absolu du répertoire de travail actuel ;
- **date** donne la date actuelle du système ;
- **echo** affiche le(s) paramètre(s) sur la sortie standard (console) ;

Premières commandes UNIX

- **cat** *fichier1* : affiche le contenu de *fichier1* ;
- **touch** *fichier1* : crée le fichier (vide) décrit par le chemin *fichier1* ;
- **rm** *fichier1* : supprime le fichier décrit par le chemin *fichier1* ;
- **mv** *fichier1* *fichier2* : renomme (ou déplace, selon le contexte) le fichier1 en fichier2 ;
- **cp** *fichier1* *fichier2* : crée une copie du fichier1 en fichier2 (mêmes données, inodes différents) ;

Premières commandes UNIX

- **ls** *rep1* : liste le contenu du répertoire *rep1* décrit par un chemin. Voir les options -l, -i, -a ;
- **cd** *rep1* : change le répertoire de travail en celui décrit par le chemin donné (*rep1*) ;
- **mkdir** *rep1* : crée le répertoire décrit par le chemin *rep1* ;
- **rmdir** *rep1* : supprime le répertoire (qui doit être vide) dont le chemin est en paramètre ;

Premières commandes UNIX

La commande **chmod** est importante car elle modifie les permissions d'un fichier.

Exemples d'utilisations "symboliques" :

- On peut ajouter le droit **x** à l'utilisateur sur **unFichier** par **chmod u+x unFichier** ;
- On peut ajouter le droit **w** à l'utilisateur et au groupe par **chmod ug+w unFichier** ;
- On peut ajouter le droit **x** et retirer le droit **w** à l'utilisateur et aux autres utilisateurs par **chmod uo+x-w unFichier** ;

Premières commandes UNIX

Exemples d'utilisations "octales" :

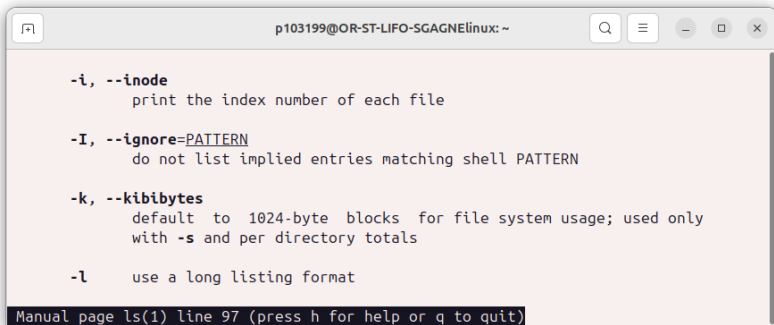
- On peut donner les droits **rw**x à l'utilisateur, **rw-** au groupe et **r-** - aux autres sur **unFichier** par **chmod 764 unFichier** ;

u	g	o	u	g	o	u	g	o
r	w	x	r	w	-	r	-	-
4	2	1	4	2	0	4	0	0
	7			6			4	

- De façon générale, on peut coder les droits en donnant la valeur 4 au droit **r**, la valeur 2 au droit **w** et la valeur 1 au droit **x**. Si la permission n'est pas donnée, la valeur est 0 ;
- Ainsi on peut donner une suite de 3 valeurs entre 0 et 7 pour les droits **u**, **g** et **o**.

Premières commandes UNIX

- la commande **man** donne la page du manuel relative à la commande passée en paramètre.
- Par exemple, la commande **man ls** présente toutes les options possibles de la commande **ls**.



The screenshot shows a terminal window titled "p103199@OR-ST-LIFO-SGAGNElinux: ~". The window displays the manual page for the 'ls' command. The content is as follows:

```
-i, --inode
    print the index number of each file

-I, --ignore=PATTERN
    do not list implied entries matching shell PATTERN

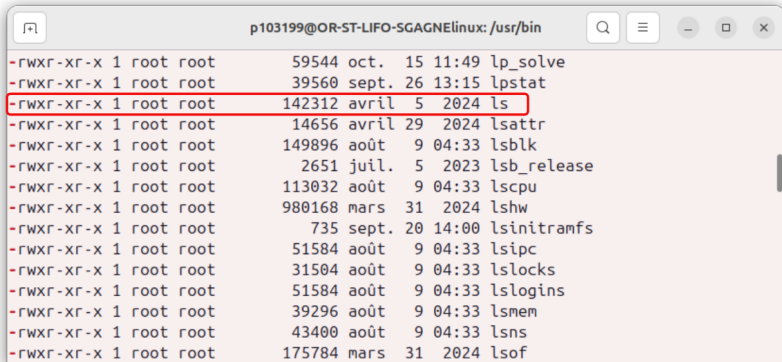
-k, --kibibytes
    default to 1024-byte blocks for file system usage; used only
    with -s and per directory totals

-l      use a long listing format
```

At the bottom of the terminal, a status bar reads: "Manual page ls(1) line 97 (press h for help or q to quit)".

Programmes

Un **programme** est un fichier **exécutable** composé d'instructions en *binaire*. Le répertoire **/usr/bin** en regroupe beaucoup, par exemple **ls** ;



```
p103199@OR-ST-LIFO-SGAGNElinux: /usr/bin
-rwxr-xr-x 1 root root 59544 oct. 15 11:49 lp_solve
-rwxr-xr-x 1 root root 39560 sept. 26 13:15 lpstat
-rwxr-xr-x 1 root root 142312 avril 5 2024 ls
-rwxr-xr-x 1 root root 14656 avril 29 2024 lsattr
-rwxr-xr-x 1 root root 149896 août 9 04:33 lsblk
-rwxr-xr-x 1 root root 2651 juil. 5 2023 lsb_release
-rwxr-xr-x 1 root root 113032 août 9 04:33 lscpu
-rwxr-xr-x 1 root root 980168 mars 31 2024 lshw
-rwxr-xr-x 1 root root 735 sept. 20 14:00 lsinitramfs
-rwxr-xr-x 1 root root 51584 août 9 04:33 lsipc
-rwxr-xr-x 1 root root 31504 août 9 04:33 lslocks
-rwxr-xr-x 1 root root 51584 août 9 04:33 lslogins
-rwxr-xr-x 1 root root 39296 août 9 04:33 lsmem
-rwxr-xr-x 1 root root 43400 août 9 04:33 lsns
-rwxr-xr-x 1 root root 175784 mars 31 2024 lsof
```

Programmes

Un fichier **texte** composé d'une suite d'instructions écrites dans un langage de programmation pouvant être compilé (comme C), ou interprété (comme python) est un **script** et pourra, par extension, aussi être considéré comme un **programme**.

programme.py ×

```
1 n = int(input("entrez un entier positif : "))
2 res = ""
3 while n > 0:
4     r = n % 3
5     n = n // 3
6     res = str(r) + res
7 print(res)
```

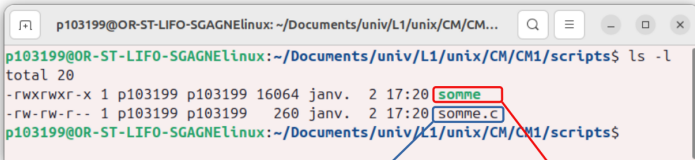
exemple de programme python (interprété)

Programmes

Le script *somme.c* est compilé en un exécutable *somme*.

C *somme.c*

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  int main() {
4      int a, b;
5      printf("entrez l'entier a : "); scanf("%d", &a);
6      printf("entrez l'entier b : "); scanf("%d", &b);
7      printf("La somme de %d et %d est %d.\n", a, b, a + b);
8      return EXIT_SUCCESS;
9  }
```



A terminal window showing the command `ls -l` in the directory `~/Documents/univ/L1/unix/CM/CM1/scripts`. The output lists two files: `somme` (executable) and `somme.c` (text file). A blue arrow points from `somme.c` to the text 'fichier texte destiné à être compilé' below. A red arrow points from `somme` to the text 'fichier compilé exécutable (binaire)' below.

```
p103199@OR-ST-LIFO-SGAGNElinux: ~/Documents/univ/L1/unix/CM/CM...
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1/scripts$ ls -l
total 20
-rwxrwxr-x 1 p103199 p103199 16064 janv.  2 17:20 somme
-rw-rw-r-- 1 p103199 p103199   260 janv.  2 17:20 somme.c
p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/CM/CM1/scripts$
```

fichier texte destiné à être compilé

fichier compilé exécutable (binaire)

Processus

- Un **processus** est une instance d'un exécutable en cours d'exécution dans le système ;
- Un programme qui s'exécute devient un processus, avec ses caractéristiques (mémoire allouée, ...). Le processus est l'aspect dynamique du programme (qui demeure statique) ;
- un processus a un identifiant (PID ou Process IDentifier) unique ;
- il a un cycle de vie (actif, endormi, arrêté, ...)

Exemples de processus

- L'exécutable **somme** peut être lancé dans un terminal par l'instruction **./somme** ;
- Cela génère un processus qui disparaît à la fin de l'exécution ;
- La commande **ps** liste les processus liés au terminal en cours ;
- pour "voir" le processus **somme**, on lance le programme en *arrière-plan* par **./somme&**
- La commande **fg** le ramène au premier plan.

Variables d'environnement

- Les **variables d'environnement** sont des paires clé/valeur d'informations liées à l'environnement du processus qui s'exécute ;
- Elles sont accessibles à tous les processus créés par le terminal en cours ;
- On peut en créer de nouvelles ou en modifier, voire en supprimer ;
- Sauf action spécifique, toute modification, création, suppression n'est valable que pour la session shell en cours.

Variables d'environnement

Variable	Description
PATH	Chemins de recherche des exécutable
HOME	Répertoire personnel de l'utilisateur connecté
USER	Nom de l'utilisateur connecté.
PWD	Répertoire de travail actuel
SHELL	Le shell utilisé par l'utilisateur connecté
LANG	Définition des paramètres régionaux (localisation, langue)
TERM	Type de terminal utilisé

Quelques variables d'environnement

Variables d'environnement

- `echo $PATH` : affiche la variable `PATH` ;
- `export VAR="uneValeur"` : crée la variable d'environnement `VAR` et lui donne une valeur (`VAR` accessible aux processus fils) ;
- `VAR2="uneAutreValeur"` : crée la variable `VAR2` (locale \neq d'environnement) et lui donne une valeur (non accessible aux processus fils) ;
- `unset VAR` : supprime la variable `VAR` ;

Caractères "magiques"

Les **caractères magiques** ou **métacaractères** sont des **symboles** qui permettent de définir des **modèles** (des "**patterns**") et ainsi rechercher de façon générique des fichiers.

Caractère	Description
*	Correspond à un certain nombre (≥ 0) de caractères (sauf /)
?	Correspond à un seul caractère (sauf /)
[]	Définit une plage ou un ensemble de caractères autorisés
[^]	Correspond à tout caractère sauf ceux mentionnés dans les crochets
{ }	Correspond à plusieurs options définies, séparées par des virgules
\	Échappe les métacaractères pour les traiter comme des caractères normaux

Quelques métacaractères et leur signification

Caractères "magiques"

```
p103199@OR-ST-LIFO-SGAGNELinux: ~/Documents/univ/L1/unix/CM/CM1/exemples
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls
corr23.txt  fichier21.csv  file1.log  suj14txt  suj23.odg  suj23.txt  sujet2.txt
fichier1.txt  fichier2.csv  suj0.txt  suj1.txt  suj23.odt  suj2.log
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj*
suj0.txt  suj14txt  suj1.txt  suj23.odg  suj23.odt  suj23.txt  suj2.log  sujet2.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj?.txt
suj0.txt  suj1.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj[01]*
suj0.txt  suj14txt  suj1.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj[02]*
suj0.txt  suj23.odg  suj23.odt  suj23.txt  suj2.log
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls *.log
file1.log  suj2.log
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls *.*[^l]*
corr23.txt  fichier21.csv  suj0.txt  suj23.odg  suj23.txt
fichier1.txt  fichier2.csv  suj1.txt  suj23.odt  sujet2.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls {suj,file}*
file1.log  suj0.txt  suj14txt  suj1.txt  suj23.odg  suj23.odt  suj23.txt  suj2.log  sujet2.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls {suj,file}*
file1.log  sujet2.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj??.txt
suj23.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$ ls suj??.{txt,od?}
suj23.odg  suj23.odt  suj23.txt
p103199@OR-ST-LIFO-SGAGNELinux:~/Documents/univ/L1/unix/CM/CM1/exemples$
```