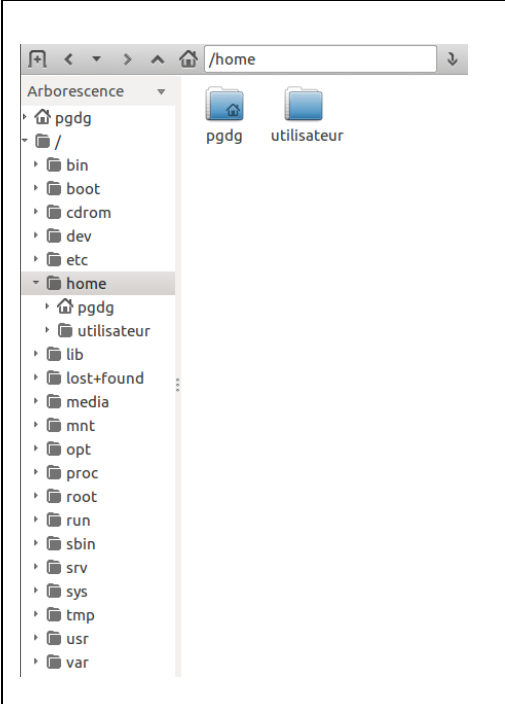


A- Arborescence du système de fichiers de Linux depuis la racine (root).

	<p>/bin : contient les programmes (exécutables) qui sont susceptibles d'être utilisés par tous les utilisateurs de la machine.</p> <p>/boot : fichiers de démarrage de Linux.</p> <p>/dev : fichiers contenant les périphériques.</p> <p>/etc : fichiers de configuration.</p> <p>/home : répertoires personnels des utilisateurs.</p> <p>/lib : dossier contenant les bibliothèques partagées (généralement des fichiers .so) utilisées par les programmes.</p> <p>/Media, mnt : Accès aux périphériques amovible (SDcard ou une clé USB).</p> <p>/root : dossier personnel du super-utilisateur <i>root</i>.</p> <p>/sbin : contient des programmes système importants.</p> <p>/tmp : Stockage temporaire des fichiers.</p> <p>/usr : Dossier d'installation de la plupart des programmes demandés par l'utilisateur.</p> <p>/var : Contient des données « variables » (fichiers logs, fichiers de serveur web, ...)</p>
---	--

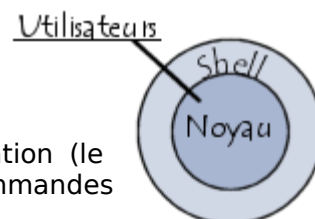
<https://doc.ubuntu-fr.org/arborescence>

B- Les principales commandes linux

1. Le shell

L'interpréteur de commandes est l'interface entre l'utilisateur et le système d'exploitation, d'où son nom anglais «**shell**», qui signifie «coquille».

Le shell est ainsi chargé de faire l'intermédiaire entre le système d'exploitation (le « noyau » ou « **kernel** » en anglais) et l'utilisateur grâce aux lignes de commandes saisies par ce dernier.



2. Qu'est-ce qu'une commande ?

Une commande est un fichier exécutable agissant sur le système ou sur des fichiers. Elles se trouvent généralement dans le dossier `/bin`.

Seul l'utilisateur `root` (**super-utilisateur ou administrateur**) a accès à toutes les commandes. (**Attention : utilisez la commande `sudo su` pour devenir `root` puis tapez le mot de passe.**)

3. Format d'une commande

Une commande est constituée d'au moins un mot : le nom de la commande. D'autres éléments sont susceptibles de la préciser. Les mots constituant la commande sont séparés par un espace. Certains caractères ont une signification spéciale. On les appelle méta-caractères.

```
nom_cmd [ option ... ] [ argument ... ]
```

Par exemple :

```
ls -l /home/pgdg/M*
```

- `ls` : nom de la commande qui Affiche les informations des fichiers
- `-l` : option
- `/home/pgdg/M*` : argument
- `*` : méta-caractère. Remplace une ou plusieurs lettres

Pour obtenir la syntaxe et le format précis d'une commande, utilisez comme option `--help`.

4. Facilités de saisie des commandes

Comme les commandes Unix sont souvent longues à saisir, diverses facilités sont offertes :

- **Les flèches Haut et bas**

Elles permettent de rappeler les commandes de l'historique. Elles s'affichent dynamiquement, il suffit de valider celle qui nous intéresse (Entrer).

- **Le clic-droit**

Un clic-droit dans un terminal console permet d'accéder au copier/coller.

- **L'opérateur tilde**

Le caractère **tilde** `~` (alt 126) seul renvoie au dossier personnel de l'utilisateur actuel.

Si l'utilisateur actif est toto, chaque occurrence du caractère `~` est remplacé par le chemin `/home/toto`

Le tilde `~` suivi d'un nom d'utilisateur, renvoie à son répertoire personnel (ainsi la commande `cd ~stage` conduit à `/home/stage`)

- **Compléter une commande - touche TAB**

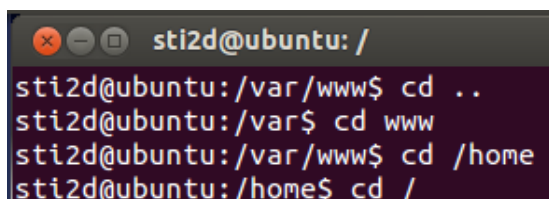
Lorsqu'on tape une ligne de commande incomplète, puis sur la touche **TAB**, l'interpréteur cherche à compléter le nom du fichier ou le nom de la commande, suivant le contexte.

S'il y a plusieurs propositions, le système attend un complément d'info de la part de l'utilisateur.

Un autre **TAB** et l'interpréteur affiche toutes les possibilités ou en indique le nombre, s'il y en a beaucoup !

5. Chemin absolu - chemin relatif

La racine du système de fichier de linux est représentée par le symbole `/`. Ce même symbole permet de séparer les dossiers.



```
sti2d@ubuntu: /  
sti2d@ubuntu:/var/www$ cd ..  
sti2d@ubuntu:/var$ cd www  
sti2d@ubuntu:/var/www$ cd /home  
sti2d@ubuntu:/home$ cd /
```

Exemple de la commande `cd` (change directory)

- chemin relatif, remonte d'un répertoire
- chemin relatif, défini depuis le répertoire courant
- chemin absolu, défini depuis la racine
- chemin absolu : va à la racine root

Chemin relatif : nécessite de savoir où on se trouve dans l'arborescence

Chemin absolu : peut être long à écrire si on se trouve loin dans l'arborescence

Attention : pour définir l'arborescence, utilisez `\` pour windows et `/` pour linux.



C- Gestion des fichiers sous linux - Les droits d'accès

Sous linux, tout fichier et tout dossier appartiennent à un propriétaire unique et à un groupe d'utilisateurs. On définit ainsi trois groupes d'utilisateurs :

- Le propriétaire,
- Le groupe d'utilisateurs propriétaire,
- Les autres utilisateurs.

Pour chacun d'eux on définit trois droits d'accès (autorisation de lecture, d'écriture et d'exécution). Seul le propriétaire du fichier et *root* peuvent changer ses droits d'accès :

Le propriétaire par défaut est celui qui a créé le fichier. Toutefois, l'utilisateur *root* ou son propriétaire peuvent modifier le propriétaire d'un fichier. On utilise alors la commande *chown*.

```
# chown propriétaire fichier
```

Le groupe par défaut est le groupe principal auquel appartient le propriétaire. Le propriétaire peut modifier le groupe à condition qu'il fasse partie de ce groupe. *root* peut également le faire.

```
# chgrp groupe fichier
```

Les droits d'accès sont représentés par les lettres :

		Fichier F	Répertoire R
r	read	On peut lire F	On peut voir le contenu de R
w	write	On peut écrire dans F ou le détruire	On peut créer des fichiers ou des répertoires dans R
x	execute	On peut exécuter F s'il est exécutable	On peut rentrer dans R avec cd

La commande *chmod* permet de modifier les droits d'accès d'un fichier. On utilise une représentation numérique octale pour représenter les droits.

```
# chmod droits fichier
```

Binaire	---	Droit	---	Octal
000	-----	(---)	-----	0
001	-----	(--x)	-----	1
010	-----	(-w-)	-----	2
011	-----	(-wx)	-----	3
100	-----	(r--)	-----	4
101	-----	(r-x)	-----	5
110	-----	(rw-)	-----	6
111	-----	(rwx)	-----	7

Par convention la présence d'un droit est noté 1, l'absence 0.

```
eleve10@vps751350: ~/Cours_1NSI/TP_unix
eleve10@vps751350:~/Cours_1NSI/TP_unix$ ls -l
total 20
drwxrwxr-x 3 eleve10 eleve10 4096 déc. 18 16:36 Python
-rwxrw-r-- 1 eleve10 eleve10 100 déc. 18 18:28 script1.sh
-rw-rw-r-- 1 eleve10 eleve10 108 déc. 19 10:22 script_python_os.py
-rw-rw-r-- 1 eleve10 eleve10 90 déc. 18 17:45 Test_TP_Linux.txt
drwxrwxr-x 4 eleve10 eleve10 4096 déc. 18 15:28 Unix
eleve10@vps751350:~/Cours_1NSI/TP_unix$
```

# chmod 777 fichier1	(Attribution de tous les droits à tout le monde)
# chmod 775 fichier2	(Attribution de tous les droits au propriétaire et aux membres du groupe, lecture-exécution aux autres utilisateurs)
# chmod 700 fichier3	(Attribution de tous les droits au propriétaire et aucun aux autres utilisateurs)
# chmod +wx fichier4	(Ajout des droits d'écriture et d'exécution à tout le monde)