

Pratique des machines

TP1 : Terminal, shell, exemples de commandes

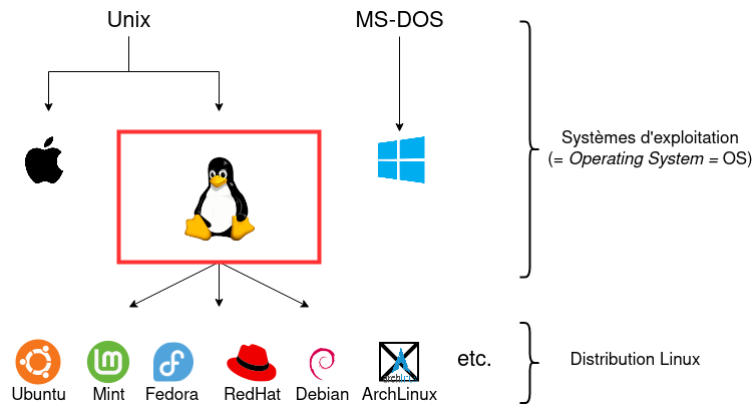
am@up8.edu

Septembre 2025

- **Ne pas copier / coller d'instructions depuis le PDF**
- Il n'y a rien à rendre. Par contre, **si vous ne savez pas répondre à une des questions du TP ou que vous ne comprenez pas le résultat d'une commande, vous devez poser la question**

Table des matières

1	Systèmes d'exploitation Unix	2
2	Découverte de quelques commandes usuelles	3
2.1	Ouverture du terminal	4
2.2	Se déplacer dans le terminal	5
2.3	Savoir "où" vous êtes	5
2.4	Voir ce qui se trouve dans un répertoire	6
2.5	Créer un répertoire	7
2.6	Effacer un fichier ou un dossier	7
3	Manier le terminal	8



1 Systèmes d'exploitation Unix

Au cours de votre formation et de votre pratique de l'informatique, vous allez travailler le plus fréquemment sur des machines dont le système d'exploitation est de type [Unix](#).

MacOS, comme les [distributions Linux](#) présentées ci-dessus sont des systèmes Unix.

Pour connaître le système d'exploitation de la machine sur laquelle vous travaillez, rendez-vous dans la section "À propos" (ou équivalent) de votre gestionnaire de paramètres.

1. → Quel est le système d'exploitation sur lequel vous vous trouvez ?

Le [système de fichiers](#) (*file system*) est l'organisation hiérarchique des fichiers au sein du système d'exploitation : il est composé de ce que nous appelons des [répertoires](#) (les dossiers) et des [fichiers](#) (par exemple, des images, des fichiers textes, des exécutables, etc.). Ceux-ci sont organisés de manière hiérarchique, comme illustré sur la Figure 1. Du point de vue du système, *Everything is a file*, c'est à dire que tout est un fichier¹. Dans la suite [FILE] peut désigner un fichier ou un répertoire.

La [racine du système](#) est le dossier qui contient tous les autres dossiers.

Chaque répertoire et chaque fichier est identifié dans le système de fichiers par un [chemin absolu](#) unique. Ce chemin contient les noms des répertoires séparés par des "/" dans lesquels il faut "descendre" depuis la [racine du système](#) pour atteindre sa destination. Le symbole utilisé pour la racine du système étant "/", tout chemin absolu commence par ce caractère.

Par exemple :

- le chemin absolu du répertoire `floppy` de la Figure 1 est `/media/floppy` ou `/media/floppy/`
- le chemin absolu du fichier `uname` de la Figure 1 est `/bin/uname`. Le chemin absolu d'un fichier ne peut pas terminer par un "/".

Pour désigner l'emplacement d'un fichier (ou dossier), on peut également utiliser une infinité de [chemins relatifs](#) : ce sont tous les chemins permettant de l'atteindre qui ne démarrent pas à la racine.

1. Voir la réponse à la question "What are directories, if everything on Linux is a file?" pour plus de détails.

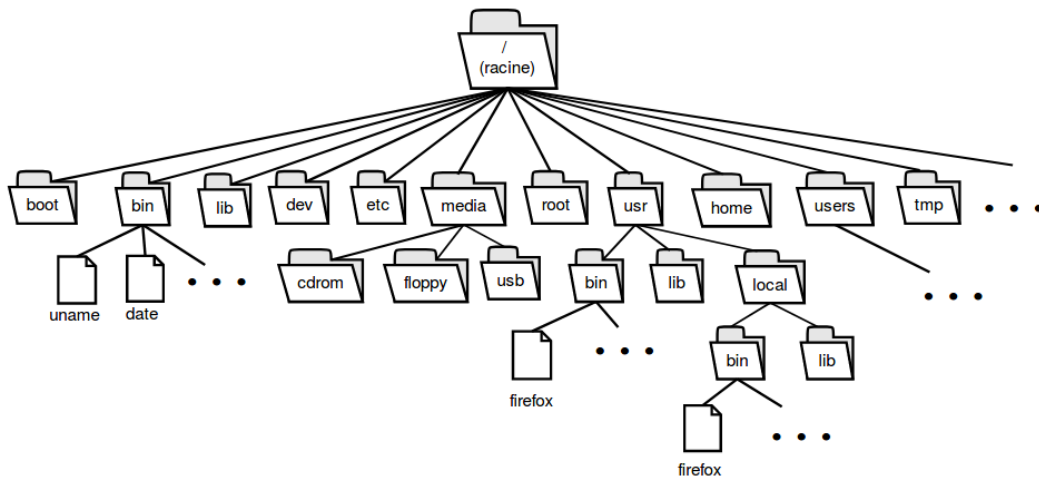


FIGURE 1 – arborescence des fichiers dans linux

Pour "remonter" dans la hiérarchie de fichiers, c'est à dire désigner le [répertoire parent](#), on utilise les symboles "`..`". Ainsi,

- Un chemin relatif permettant d'atteindre le fichier `uname` depuis le dossier `floppy` est `../../bin/uname`
- Comme je l'ai dit, il en existe une infinité :
`../../media/../../media/../../media/../../media/../../bin/uname`
est aussi un chemin relatif valide.

On a parfois besoin de désigner le répertoire courant, ou [répertoire de travail](#), dans ce cas on utilise le symbole "`.`" (point).

2 Découverte de quelques commandes usuelles

Un [terminal](#) (ou émulateur de terminal, ou console) est un logiciel qui sert de point d'accès de communication entre l'humain et la machine. Le plus souvent, on entend par terminal une fenêtre dans laquelle est lancé un shell.

On "lance" le terminal en cliquant sur une icône qui ressemble généralement à ceci :



Une autre manière d'ouvrir le terminal est de faire `ctrl+alt+t` au clavier (depuis une machine où ces touches clavier existent, pour les machines Apple je vous laisse chercher vous-même).

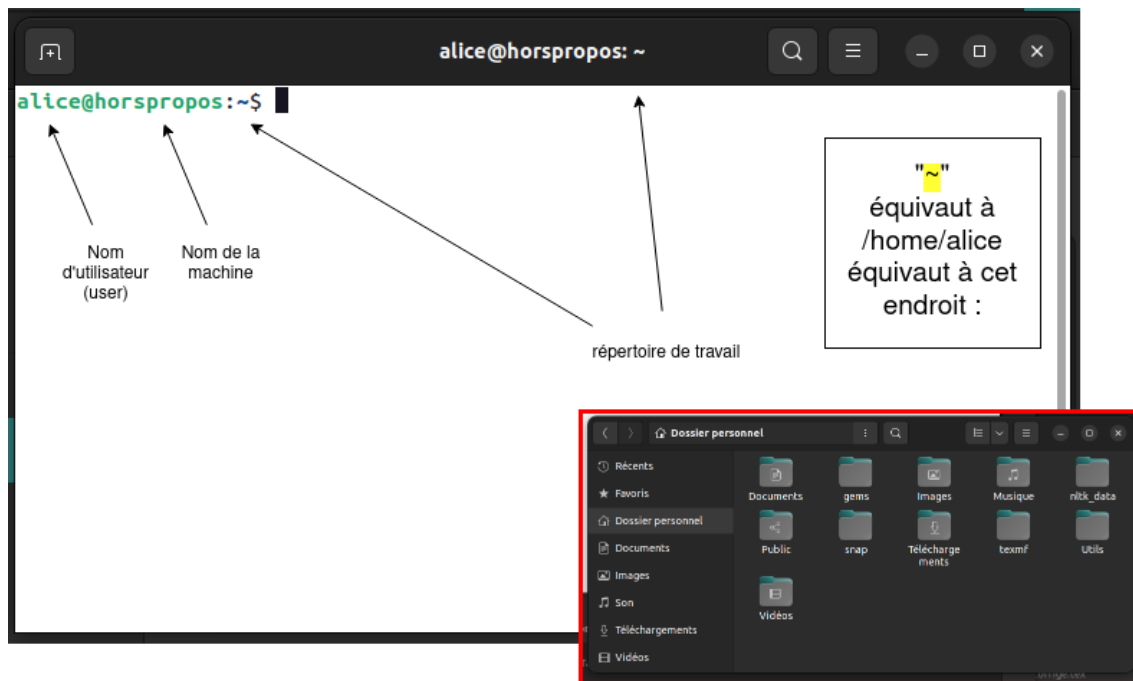


FIGURE 2 – Terminal

Un shell est un *interpréteur de commande*. Il en existe plusieurs, comme nous allons le voir dans ce TP. Les commandes saisies au clavier sont interprétées par le shell lorsqu'on tape "entrée".

Le terminal est l'interface qui permet d'accéder au shell pour lui fournir des instructions grâce aux lignes de commandes.

Une ligne de commande est composée d'une commande, d'éventuelles options (précédées du caractère "-" et d'éventuels arguments) :

ligne de commande

```
commande -option1 -option2 -option3 argument1 argument2
```

2.1 Ouverture du terminal

→ Ouvrir un terminal.

Par défaut, l'invite de commande (qui comme son nom l'indique vous invite à lui fournir une commande) est composée de votre **login**, puis d'un @, puis du **nom de la machine** sur laquelle vous êtes connecté.e, puis d'un :, puis du **répertoire de travail**, et enfin d'un \$ qui signifie que vous pouvez entrer une commande.

Par défaut encore, le répertoire de travail correspond à votre [répertoire personnel](#), ou « *home directory* », qui est la racine de la partie du système de fichier *qui appartient à votre compte*. Un alias du chemin absolu vers votre répertoire personnel est "~".

Rappel : la véritable [racine du système](#) correspond au répertoire qui se trouve à l'adresse « / ».

→ Quelle est l'invite de commande qui s'affiche dans votre terminal par défaut ?

Recopiez la commande suivante (sans le premier signe \$ qui n'est là que pour indiquer le début de la ligne de commande) et exécutez-la :

```
$ printf "Mon shell courant est : %s\n" "$SHELL"
```

→ Quel est votre shell ?

2.2 Se déplacer dans le terminal

la première commande que nous utilisons est la commande [cd](#).

```
$ cd <répertoire>
```

change directory : cette commande prend un argument, qui est le répertoire de destination.

L'argument est optionnel : si vous exécutez "cd" seul, vous vous déplacez dans votre répertoire personnel.

→ Exécutez dans le terminal et observez les changements de votre **prompt** :

```
$ cd /usr/bin
$ cd -
$ cd /usr/local/bin
$ cd ..
$ cd ~
```

2.3 Savoir "où" vous êtes

Quand on se pose la question "Où suis-je ?" dans le terminal, on cherche à connaître le **répertoire** dans lequel votre invite de commande est située. Votre localisation influe sur le chemin vous permettant d'accéder à un fichier contenu dans votre hiérarchie de fichiers.

Découvrons la commande [pwd](#). Normalement, suite aux dernières commandes que vous avez

exécutées vous devriez être revenu.e dans votre *home*. Si ce n'est pas le cas, exécutez une des commandes permettant de situer votre prompt dans votre *home*.

```
$ pwd [OPTION]...
```

print working directory : affiche le chemin du [répertoire de travail](#), c'est-à-dire le répertoire dans lequel vous vous trouvez quand vous exécutez l'instruction. Cette commande prend des options que nous n'utiliserons pas pour l'instant.

→ Vérifiez que votre répertoire de travail est bien votre *home*.

→ Donnez les trois commandes que vous connaissez désormais qui vous permettent de vous déplacer dans votre *home* quel que soit votre répertoire de travail.

2.4 Voir ce qui se trouve dans un répertoire

Nous allons découvrir la commande [ls](#) :

```
$ ls [OPTION]... [FILE]...
```

liste le contenu d'un répertoire.

Les crochets autour de [FILE] signifient que l'argument est optionnel : si vous exécutez "ls" seul, vous listez le contenu de votre répertoire de travail.

Options fréquemment utilisées :

- -a : montre les fichiers cachés
- -l : affiche le contenu du dossier sous forme de *listing* ordonné

→ Exécutez :

```
$ ls
$ ls -a
$ ls -l
$ ll
$ ls -la
```

→ Les répertoires et les fichiers ne devraient pas s'afficher de la même manière : comment les distinguez-vous sur votre machine ?

→ Quels sont les deux premiers répertoires qui s'affichent lorsque vous faites la 2^e commande ci-dessus (ls -a) ? Rappelez leur signification.

→ Que fait l'avant dernière commande ?

→ Que fait la dernière commande ?

2.5 Créer un répertoire

Voyons maintenant la commande [`mkdir`](#) qui nous permet d'interagir avec le système de fichiers :

```
$ mkdir [OPTION]... DIRECTORY...
```

MaKe DiRectory : crée un répertoire du nom de DIRECTORY. Ce paramètre est obligatoire. Vous pouvez créer un répertoire ailleurs que dans votre répertoire de travail en spécifiant le chemin relatif ou absolu vers l'emplacement que vous souhaitez.

→ Exécutez les commandes suivantes pour créer un répertoire de travail pour ce cours (vous pouvez remplacer `pdm` par `pratique_des_machines` ou une variante de votre choix).

```
$ cd ~
$ mkdir pdm
$ ls pdm
$ mkdir pdm/TP1
$ ls pdm
$ cd pdm/TP1
$ pwd
$ ls
```

D'après les commandes ci-dessus, vous vous trouvez normalement dans un répertoire vide équivalent à `~/pdm/TP1`.

2.6 Effacer un fichier ou un dossier

```
$ rm [OPTION]... [FILE]...
```

remove : supprime le ou les fichiers spécifiés.

Pour supprimer un répertoire, il faut spécifier l'option `-r` (récursive) : tout le contenu du répertoire sera également supprimé.

Depuis le répertoire équivalent à `~/pdm/TP1`, exécutez :

```
$ mkdir dir_parent/dir_enfant
$ mkdir dir_parent
$ mkdir dir_parent/dir_enfant
$ ls
$ ls dir_parent
$ ls dir_enfant
$ rm dir_parent
$ rm -r dir_parent
```

→ Expliquez brièvement le comportement de chaque commande et commentez les éventuels messages d'erreur.

Nommage des fichiers et dossiers

- **n'utilisez pas d'espaces dans vos nom de répertoires ou de fichiers** ;
- n'utilisez pas de caractères spéciaux ou de ponctuation. Vous n'avez droit qu'aux caractères de a à z, aux chiffres et à "-" et "_" ;
- donnez toujours une **extension** aux fichiers que vous créez. Les dossiers, eux, n'ont pas d'extension ;

→ Testons ce qui se passe si on décide de mettre des espaces dans le nom d'un dossier en exécutant les lignes suivantes :

```
$ cd ~  
$ mkdir test  
$ cd test  
$ mkdir pratique des machines  
$ ls
```

Que s'est-il passé ?

→ Effacez ce dossier de test grâce à la commande `rm` présentée ci-dessus.

3 Manier le terminal

Quelques astuces et raccourcis clavier pour utiliser le terminal :

- `Ctrl + Maj + t` : ouvrir un nouvel onglet dans le terminal
- `Ctrl + d` : arrêter le processus en cours ou fermer le terminal (ex : lancez la commande `cat` puis arrêtez le processus)
- la tabulation permet d'auto-compléter
- flèches du haut / bas : se déplacer dans l'[historique](#) des commandes lancées
- `Ctrl + a` : aller au début de la ligne
- `Ctrl + e` : aller à la fin de la ligne
- `Ctrl + k` : couper jusqu'à la fin de la ligne
- `Ctrl + Maj + C` : clic droit "copier" : copier la région en surbrillance
- `Ctrl + Maj + V` : clic droit "coller" ou clic molette ou `Ctrl + y` (seulement au sein du terminal) : coller
- `Ctrl + c` : arrêter la commande en cours
- `clear` ou `Ctrl + l` : effacer les commandes précédentes de l'interface graphique
- `reset` : réinitialiser l'environnement du shell