

DEV1 – ENVL – Laboratoires d'environnement

TD 1 – Introduction

Table des matières

1	Une introduction à Linux	2
1.1	Présentation	2
1.2	Se connecter	3
1.3	Le dossier personnel et le dossier courant	4
1.4	Visualiser le contenu d'un dossier/fichier	4
1.5	La structure d'une commande	5
1.6	Changer le mot de passe sous Linux	6
1.7	Se déplacer	6
1.8	L'éditeur	7
1.9	Quelques commandes courantes	8
1.10	La racine du système de fichiers	9
1.11	Chemin absolu et relatif	9
1.12	Raccourcis pour des chemins	10
2	Conclusion	11

Conseils

Quelques conseils pour bien travailler et progresser.

- ▷ Faites bien tous les exercices proposés.
- ▷ Vous pouvez **coopérer** avec vos condisciples mais nous vous demandons de ne **pas copier** les réponses. Si vous voulez progresser, **chercher** la réponse est plus important que de la trouver.
- ▷ N'hésitez pas à **montrer votre travail** à votre professeur.
- ▷ N'hésitez pas à **poser des questions** si vous n'avez pas bien compris ce qu'on vous demande.
- ▷ **Prenez des notes!** Ce que vous allez apprendre aujourd'hui vous servira les semaines prochaines mais vous en aurez oublié une grande partie si vous ne notez rien. Le plus pratique est probablement d'annoter la version **papier**.

1 Une introduction à Linux

« *Linux ? Il y a moins bien mais c'est plus cher.* » – Auteur inconnu

1.1 Présentation

Vous ne travaillerez pas directement sur votre PC durant les laboratoires d'environnement. Celui-ci vous servira pour vous connecter au serveur Linux (son nom est `linux1`).

Linux

Linux est un système d'exploitation comme le sont également Windows ou MacOS. Il permet d'utiliser l'ordinateur et ses périphériques. La plupart des systèmes d'exploitation actuels proposent une interface graphique à l'utilisateur. Ainsi, les concepts de fenêtre, de menus, de souris... vous sont familiers. Linux n'échappe pas à la règle et propose un environnement proche de ce qu'on peut trouver sur d'autres systèmes.

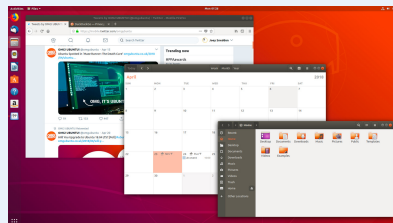


FIGURE 1 – Ubuntu 18.04

Bien que nous pourrions utiliser le mode graphique, nous allons, pour des raisons pédagogiques, vous apprendre à utiliser Linux en mode console.

Le mode console

En mode console^a, nous n'allons pas utiliser des fenêtres et une souris mais nous allons donner (**écrire**) des commandes au système.

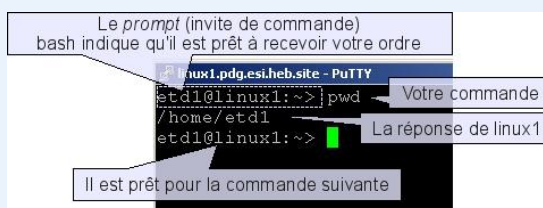
Shell

Un programme dont le but est d'accepter des commandes de la part de l'utilisateur et de les exécuter. C'est ce shell qui tourne dans la console.

Bash

Plusieurs shells existent en Linux. Bash est l'un d'entre eux.

Dans l'exemple ci-dessus, vous pouvez constater que l'utilisateur entre la commande `pwd` qui a pour effet d'afficher une réponse^b. Ensuite, bash attend la commande suivante.



^a. On parle aussi de mode « commande » ou encore « terminal ».

^b. Vous comprendrez bientôt ce que signifie `pwd`.

Pour travailler en mode console, il faut savoir dialoguer avec ce shell. Comment lui indiquer par exemple qu'on veut copier un fichier, exécuter un programme Java...? **C'est le principal objectif de ces TDs !**

1.2 Se connecter

`linux1` n'est pas la machine sur laquelle vous êtes. Vous allez devoir vous y connecter à distance.

putty

putty est une application qui permet (notamment) d'ouvrir un *terminal* sur une machine distante, de s'y connecter et de dialoguer avec elle en mode console.



Vous, sur le PC Windows



linux1

Lorsque vous allez vous connecter, `linux1` va vous demander de vous identifier.

Votre login (username)

Un 'g' **minuscule** suivi de votre numéro d'étudiant (ex : g52010). **Note** : pour Linux, les minuscules et les majuscules sont toujours des caractères différents.

Votre mot de passe

Le même que votre **mot de passe initial sous Windows** ^a.

^a. Le mot de passe sous Windows, sous Linux et pour votre mail HE2B sont trois mots de passe différents mais initialisés à la même valeur.

Tutoriel 1

Connectez-vous à linux1

Il y a 4 étapes :

- ☑ Lancez l'application PUTTY (vous la trouverez dans le menu ou comme raccourci sur le bureau).
 - ☑ Indiquez à PUTTY le nom de la machine (*Host Name*) à laquelle vous voulez vous connecter (ici `linux1`);
 - ☑ Cliquez sur "**Open**"; la connexion se fait ! S'il vous présente une boîte de message avec un "**Security Alert**", cliquez sur "**Yes**" en toute confiance.
 - ☑ Identifiez-vous !
 - ☑ Tapez votre nom d'utilisateur (cf. plus haut) puis sur la touche **ENTREE**.
 - ☑ Tapez votre mot de passe (cf. plus haut) puis sur la touche **ENTREE**.
- Note** : Rien ne s'affiche quand vous tapez votre mot de passe ; c'est normal.

1.3 Le dossier personnel et le dossier courant

Fichiers et dossiers

Comme sur la plupart des systèmes d'exploitation les données sur le disque dur sont organisées en fichiers et dossiers :

- ▷ Un **fichier** contient de l'information : un code Java, un exécutable, un texte. . .
- ▷ Un **dossier** contient des fichiers. Il sert à organiser/regrouper les nombreux fichiers du disque. On parle aussi de **répertoire**.
- ▷ Un dossier peut aussi contenir des dossiers qui, à leur tour, peuvent contenir des fichiers/dossiers qui. . . On a une organisation **hiérarchique** des fichiers.

Le répertoire personnel

Sur Linux, chaque utilisateur dispose d'un « **dossier personnel** » (on dit aussi sa « **home** »). C'est dans ce dossier qu'il peut créer des fichiers (et des dossiers). Par convention, ce dossier porte le même nom que le login de l'utilisateur ^a.

^a. À ne pas confondre avec le dossier qui s'appelle **home** et qui contient tous les répertoires personnels.

Le dossier courant

À tout moment, on se trouve dans un dossier précis, le « **dossier courant** ». Au départ, lorsqu'on se connecte, c'est le *répertoire personnel*.

1.4 Visualiser le contenu d'un dossier/fichier

Commençons notre apprentissage en douceur en voyant comment visualiser ce qui se trouve dans notre dossier.

Experience 1

Visualiser le contenu d'un dossier

- ✍ Entrez la commande `ls` (n'oubliez pas la touche ENTREE).

```
linux1:~> ls
dev1 welcome
```

Le bash vous montre le contenu de votre dossier courant (votre dossier personnel dans ce cas ci). Vous constatez qu'il contient déjà un dossier **dev1** et un fichier **welcome**. Pour vous aider à faire la différence entre un dossier et un fichier, la commande `ls` affiche les noms de dossiers dans une couleur différente.

- ✍ Entrez à présent la commande `ls dev1`.

```
linux1:~> ls dev1
readme
```

Cette fois, il vous montre que le dossier **dev1** contient un fichier appelé **readme**.

Experience 2

Visualiser le contenu d'un fichier

Vous avez vu que votre dossier contient un fichier **welcome**. Qu'y est-il écrit ?

✍ Entrez la commande `cat welcome`.

```
linux1:~> cat welcome  
Bienvenue sur la machine linux1 de l'ÉSI.
```

La commande n'est pas la même pour voir le contenu d'un fichier ou d'un dossier.

Visualiser le contenu

- ▷ `ls` (sans rien derrière) liste le contenu du dossier courant.
- ▷ `ls nomDuDossier` liste le contenu du dossier indiqué.
- ▷ `cat nomDuFichier` affiche à l'écran le contenu du fichier dont le nom est donné (on peut juste le voir, pas le modifier!).

1.5 La structure d'une commande

Faisons quelques expériences pour comprendre la structure générale d'une commande.

Experience 3

La casse

✍ Entrez la commande `LS` (en majuscule).

Vous voyez que le résultat est différent : il vous affiche un message d'erreur car il ne comprend pas ce que vous lui voulez.

En Linux, les majuscules et les minuscules n'ont pas le même sens, vous devez respecter la casse.

Experience 4

Les espaces

Tapez ces 3 commandes qui ne se différencient que par la présence ou non d'espaces.

✍ `ls dev1`

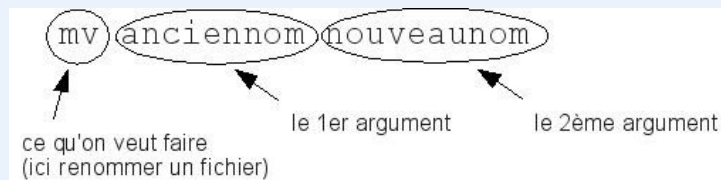
✍ `lsdev1`

✍ `ls dev 1`

À nouveau le résultat est différent dans les 3 cas. **Les espaces ont de l'importance.**

Une commande décomposée

Une commande est composée de plusieurs parties séparées par un ou plusieurs espaces.



La 1^{re} partie indique ce qu'on veut faire. Ensuite on donne des arguments pour préciser ce qu'on veut faire.

Exemples :

- ▷ Si on veut détruire un fichier (commande `rm`), il faut donner un argument : le nom du fichier à détruire.
- ▷ Si on veut renommer un fichier (commande `mv`), il faut donner 2 arguments : le nom du fichier à renommer et son nouveau nom.

1.6 Changer le mot de passe sous Linux

Faites une courte pause dans votre apprentissage des commandes et profitez-en pour changer le mot de passe.

Tutoriel 2

Changez votre mot de passe !

✍ Tapez la commande `passwd` pour changer votre mot de passe.

1. Le système vous demande de taper le mot de passe actuel (vous ne le voyez pas quand vous le tapez, c'est normal!)
2. Ensuite, vous entrez le nouveau mot de passe. Vous pouvez reprendre le même mot de passe que celui que vous avez choisi pour Windows.
3. Vous retapez une deuxième fois ce mot de passe pour le confirmer.

Pour vérifier que tout s'est bien passé, vous pouvez vous déconnecter et vous reconnecter.

✍ Entrez `exit` pour quitter proprement `linux1`.

✍ Reconnectez-vous.

FAQ

Quand je tape la commande rien ne se passe ! Une seule personne à la fois peut changer son mot de passe et vous êtes tous connectés à la même machine. Soyez patient !

Après avoir tout entré, il me met un message d'erreur ! Lisez le message ! Il est en général assez explicite : message trop court, trop simple. . .

J'ai quitté en fermant la fenêtre, ce n'est pas plus simple ? Oui ! Mais c'est impoli de quitter quelqu'un sans lui dire au revoir ! ;) Plus sérieusement, vous coupez brutalement la conversation avec `linux1` ce qui peut laisser trainer des processus actifs et vous empêcher de vous connecter la prochaine fois.

Si jamais j'oublie mon mot de passe sous Linux. Je dois aussi aller voir les techniciens ? Non ! Votre professeur de labo peut réinitialiser le mot de passe Linux à sa valeur initiale.

1.7 Se déplacer

Il est important de pouvoir se *déplacer* dans la structure des dossiers, de changer de dossier courant. Voyons pourquoi et comment.

Experience 5

Le bash ne cherche pas un fichier

On voudrait visualiser le contenu du fichier `readme` qui se trouve dans `dev1`.

✍ Entrez la commande `cat readme`.

Ça ne fonctionne pas !

Quand on indique un **nom** de fichier ou de dossier sans plus de précision, il doit se trouver dans le **dossier courant**. Le shell ne va **pas chercher** ailleurs ou dans un sous-dossier pour le trouver.

Par exemple, si on crée un dossier sans préciser où, il sera créé dans le dossier courant.

Experience 6

Se déplacer

- ✍ Tapez la commande `cd bin`.
 - ▷ Cette commande demande de se **déplacer** dans le dossier *bin*. Elle **change le dossier courant**.
- ✍ Retapez la commande `ls` du début.
 - ▷ Le résultat est différent. Est-ce que vous comprenez pourquoi ?

Se déplacer

- ▷ `cd` (*change directory*) : permet de changer de dossier courant.
 - ▷ `cd nomDuDossier` vous déplace dans le dossier indiqué (qui doit être dans le dossier courant).
 - ▷ `cd ..` vous amène dans le dossier juste au-dessus de celui où vous êtes, c-à-d celui qui contient le dossier courant actuel. On parle de **répertoire parent**.
 - ▷ `cd` (sans rien derrière) vous ramène toujours dans votre dossier personnel.
- ▷ `pwd` (*print working directory*) : affiche le (chemin du) dossier courant.
- ▷ `tree` : affiche une vue hiérarchique du dossier courant et de ses sous-dossiers.

Exercice 1

Assimiler les commandes de base

Afin d'assimiler les commandes qu'on vient de voir, on vous demande de :

1. Vous placer dans votre répertoire personnel ;
2. Vérifier que c'est bien là que vous êtes ;
3. En afficher le contenu ;
4. En afficher le contenu complet (sous-dossiers y compris) ;
5. Vous déplacer dans le fichier **dev1** ;
6. Vérifier que vous avez bien changé de dossier courant ;
7. Revenir dans le dossier courant.

Exercice 2

cd vs ls

Supposons que vous êtes dans votre répertoire personnel. Quelle est la différence entre :

1. La commande `ls dev1` ;
2. La commande `cd dev1` suivie de `ls` ?

Comment le mettre en évidence ?

1.8 L'éditeur

Un **éditeur de texte** (ou, plus court, un **éditeur**) est un programme qui vous permet d'entrer, modifier le contenu d'un fichier texte. Vous connaissez probablement NOTEPAD sous WINDOWS. Sur LINUX, il en existe beaucoup. Celui que nous vous proposons dans un premier temps au laboratoire s'appelle NANO. Il est assez facile pour débuter et faire des choses simples. Par la suite, vous pourrez choisir de rester sous NANO ou d'utiliser VIM.

Tutoriel 3

Premiers pas avec nano

- ✍ En préambule, tapez la commande `cd` pour revenir dans *votre home*.
- ✍ Tapez `nano test` pour commencer à éditer le fichier `test` (comme il n'existe pas encore, il est créé).
Une fenêtre s'ouvre. Vous voyez qu'elle est scindée en 2 parties : la partie supérieure où vous écrivez votre texte et la partie inférieure où sont indiquées les différentes commandes (le `^` représente la touche `Ctrl`)
- ✍ Entrez quelques mots.
- ✍ Appuyez sur la combinaison de touches `Ctrl X`, confirmez que vous voulez sauvegarder vos modifications et sortez.
Vous êtes maintenant revenu à l'invite de commande.
- ✍ Tapez à présent la commande `ls`. Vous pouvez constater que le fichier `test` est apparu dans la liste ;)

Pour aller plus loin dans l'apprentissage de cet éditeur, consultez ce document :

<http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/nano-l-editeur-de-texte-du-debutant>.

1.9 Quelques commandes courantes

Exercice 3

Se rappeler les commandes vues

Vous rappelez-vous des commandes déjà vues ? La commande pour :

- ▷ voir le contenu d'un dossier est
- ▷ éditer le contenu d'un fichier est
- ▷ changer de dossier courant est
- ▷ voir le chemin du dossier courant est
- ▷ voir une vue arborescente de tous le contenu d'un dossier est

Il est temps de voir quelques commandes supplémentaires.

Commandes courantes

- ▷ `mkdir nomDuDossier` crée un dossier (vide) nommé `nomDuDossier` ;
- ▷ `mv nomDuFichier nouveauNomDeFichier` renomme le fichier donné `nomDuFichier` sous le nom `nouveauNomDeFichier` ;
- ▷ `mv nomDuFichier nomDuDossier` déplace le fichier donné dans le dossier indiqué ;
- ▷ `cp nomDuFichier nouveauNomDeFichier` crée une copie du fichier sous le nom `nouveauNomDeFichier` ;
- ▷ `cp nomDuFichier nomDuDossier` copie le fichier donné dans le dossier indiqué (il garde le même nom) ;
- ▷ `rm nomDuFichier` détruit le fichier dont on donne le nom ;
- ▷ `rmdir nomDuDossier` détruit le dossier dont on donne le nom (Attention, le dossier doit être vide !) ;
- ▷ `touch nomDuFichier` crée un fichier vide de nom donné.

Exercice 4

Assimiler ces commandes courantes

Afin d'assimiler les commandes qu'on vient de voir :

1. Dans `dev1`, créez un dossier `td1`.
2. Déplacez-y le fichier `test` que vous avez déjà créé (vous pouvez le faire en 2 étapes).
3. Prenez une copie de votre fichier `test` (appelez-la `test2`).
4. Éditez ce fichier et ajoutez-y quelques mots.
5. Affichez le contenu des 2 fichiers pour vérifier qu'ils sont bien différents.
6. Créez, dans votre dossier `td1`, un dossier `monDossier`.
7. Déplacez-y votre fichier `test2`.
8. Détruisez le dossier `monDossier` (ainsi que son contenu).

1.10 La racine du système de fichiers

Rappelez-vous que les fichiers sont organisés en hiérarchie : un dossier contient d'autres dossiers qui lui-même...

On a vu que chaque utilisateur dispose d'un dossier personnel. Où se trouve-il ? Dans un dossier appelé `home`. Et où se trouve ce dossier `home` ? Dans un dossier appelé « / » (la barre oblique). Et où se trouve ce dossier « / » ? Nulle part ;) C'est le dossier principal (la **racine**).

La racine du système de fichiers

La **racine** est le dossier principal, tout en haut de la hiérarchie des dossiers.

1.11 Chemin absolu et relatif

« Tout est relatif, et cela seul est absolu. » – Auguste Comte

Régulièrement, il faut indiquer un endroit du système de fichiers (par exemple pour y aller). Désigner simplement le nom du dossier ne suffit pas.

Chemin absolu et relatif

Chemin

Un **chemin** est une suite de dossiers à traverser pour arriver au dossier ou au fichier qui nous intéresse. On les sépare par « / ».

Par exemple : `home/g12345/dev1/td1/test` indique que dans le dossier `home` il y a un dossier `g12345` qui contient un dossier `dev1` qui contient un dossier `td1` qui contient un fichier (ou un dossier) `test`.

Chemin absolu

Un chemin est **absolu** s'il commence à partir de la **racine**. Il commence donc par « / ». Par exemple : `/home/g12345/td1/test`

Chemin relatif

Dans le cas contraire il est **relatif**. Il s'agit d'un chemin à suivre à partir du **dossier courant**.

Un chemin **absolu** désigne toujours le même endroit. Un chemin **relatif** non ! ça dépend d'où on est, du dossier courant.

Exercice 5 pwd

Entrez la commande `pwd` (que fait-elle encore?). Comprenez-vous la notation qu'elle utilise pour la réponse? Est-ce un chemin absolu ou relatif?

Rappel : dans une commande, un nom de fichier (ou de dossier) désigne toujours un fichier (ou un dossier) se trouvant dans le dossier courant
Pour indiquer un **fichier** (ou un dossier) se trouvant **ailleurs**, il faut donner son **chemin** (absolu ou relatif).

Par exemple : `ls /home/g12345` permet de visualiser le contenu de la home de `g12345` (quel que soit l'endroit où on se trouve puisque c'est un chemin absolu).

Exercice 6 Utilisation d'un chemin

Supposons que vous êtes dans votre home et que vous ne pouvez pas vous déplacer (la commande `cd` est interdite!).

1. Affichez le contenu du fichier `test` (qui se trouve dans le dossier `td1`) en utilisant un chemin **relatif**.
2. Faites la même chose en utilisant un chemin **absolu**.
3. Créez une copie du fichier `welcome` dans le dossier `td1`.
4. Auriez-vous pu faire cette copie sans utiliser de chemin, en vous déplaçant (`cd`) ?

1.12 Raccourcis pour des chemins

Raccourcis

Dans un chemin :

- ▷ `~g12345` : désigne la home de l'utilisateur `g12345` ;
- ▷ `~` : désigne la home de l'utilisateur qui entre la commande ;
- ▷ `..` : désigne le dossier parent du dossier courant ;
- ▷ `.` : désigne le dossier courant.

Exercice 7 Se familiariser avec les raccourcis

Placez-vous dans votre dossier personnel. En utilisant les raccourcis afin de trouver la solution la plus courte possible :

1. Affichez le contenu de votre dossier personnel en utilisant un chemin absolu.
2. Affichez le contenu de la home de votre professeur en utilisant un chemin absolu.
3. Refaites la même chose mais en utilisant un chemin relatif.

Exercice 8 Comprendre un chemin

Que désigne le chemin suivant : `~mcd/../../home` ?

Exercice 9**Chemins absolus et relatifs**

Parmi tous les chemins suivants, quels sont ceux qui sont **relatifs** ?

- ☐ ~/../g12345/td2
- ☐ /home/g12345/../g54321/Hello.java
- ☐ ./tds/td2
- ☐ tds/td2/Hello.java
- ☐ ~g12345/tds/td2

2 Conclusion**Notions importantes de ce TD**

Voici les notions importantes que vous devez avoir assimilées à la fin de ce TD.

- ▷ Ce que font les commandes : `cd`, `ls`, `cp`, `mv`, `pwd`, `tree`, `exit`, `nano`, `rm`, `rmdir` et `touch`.
- ▷ Savoir éditer un petit texte avec **nano**.
- ▷ Comprendre la notion de hiérarchie de dossiers et savoir ce qu'est la racine.
- ▷ Comprendre la notion de dossier courant.
- ▷ Comprendre la notion de chemin absolu et relatif. Savoir les utiliser dans les commandes.
- ▷ Connaître la signification dans un chemin de : « ~ », « . » et « .. ».

Exercice 10**Exercice récapitulatif**

Cet exercice qui va vous permettre de vérifier que toutes les notions importantes vues aujourd'hui ont bien été assimilées.

1. Placez-vous dans votre dossier personnel.
2. Allez dans le dossier `td1`.
3. Créez un fichier vide nommé `cv`.
4. Éditez-le en y mettant votre nom.
5. Affichez son contenu.
6. Revenez dans votre dossier personnel simplement.
7. Afficher le contenu du fichier `cv` en utilisant un chemin relatif.
8. Faites de même avec un chemin absolu.
9. En restant dans votre dossier personnel :
 - (a) Créez un dossier **recap**.
 - (b) Déplacez le fichier `cv` dans ce dossier.
 - (c) Faites une copie du fichier `cv` appelée `cv2`
10. Supprimez tout ce que vous avez créé pour cet exercice.

Félicitations !

Vous êtes arrivé au bout de ce premier TD. Avant de quitter le laboratoire, n'oubliez pas de quitter proprement la connexion avec `linux1` (`exit`) et d'éteindre l'ordinateur (ou au moins de vous déloguer).

À la semaine prochaine et soyez à l'heure !

FAQ

Euh ! Moi, j'ai pas fini !

Vous n'êtes pas le seul ;) Pour beaucoup d'entre-vous, le travail demandé vous prendra plus que les 2 heures du laboratoire. À vous de terminer à la maison ou au laboratoire en libre accès (le 305). Faites-le ! C'est important. Vous aurez du mal à suivre la fois prochaine si vous n'avez pas terminé le TD d'aujourd'hui.

Est-ce que je peux me connecter à `linux1` de la maison ?

Non ! Cette machine n'est accessible que de l'école.

J'aimerais terminer à la maison mais je n'ai pas Linux ?

Sur poÉSI, vous trouverez un document qui vous explique comment avoir facilement un Linux fonctionnel sur votre PC.

J'ai beaucoup de mal à faire les exercices. J'ai l'impression de ne rien comprendre.

Ne vous découragez pas, vous n'êtes pas seul.

- ▷ Pendant les heures de laboratoire, votre professeur sera heureux de répondre à toutes vos questions, de vous réexpliquez les points confus.
- ▷ Des séances de rattrapages sont organisées certains midis. Un laboratoire est ouvert et un professeur est à votre disposition pour vous aider. Consultez les valves dédiées aux rattrapages pour les détails.
- ▷ Parfois, une notion n'est pas comprise parce que les bases ne sont pas là. Si vous ne comprenez pas quelque chose, n'hésitez pas à revenir en arrière pour consolider les notions de base. C'est reculer pour mieux avancer par la suite !

Exercice 11

Pour aller plus loin...

Vous avez fini tous les exercices proposés aujourd'hui et vous vous ennuyez ? Voici quelques pistes pour aller plus loin. Vous devrez probablement faire des recherches sur Internet pour trouver la réponse.

1. Comment créer un nom de fichier qui contient un espace ?
2. Comment créer en une seule commande un dossier `bro1` et, dedans, un dossier `bro12` ?
3. Comment, en une seule commande, supprimer tout un dossier qui n'est pas vide ?
4. Comment, en une seule commande, copier tout un dossier qui n'est pas vide ?