

DEV1 – ENVL – Laboratoire d'environnement système**TD 1 – Introduction à GNU/Linux****Table des matières****1 Présentation****2****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*.

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

GNU/Linux

GNU/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. GNU/Linux n'échappe pas à la règle et propose un environnement proche de ce qu'on peut trouver sur d'autres systèmes.

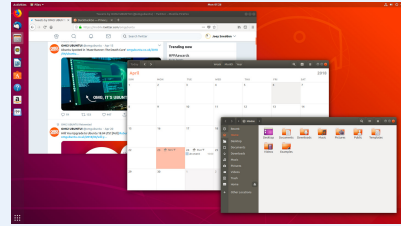
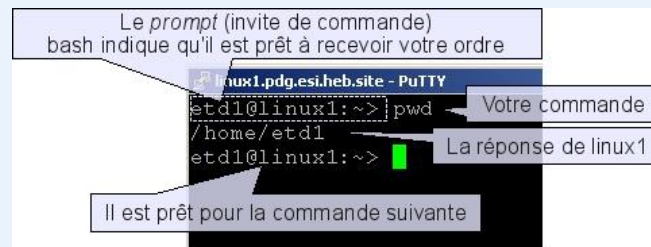


FIGURE 1 – UBUNTU 18.04

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

Le mode console

En mode console ¹, 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 GNU/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 ². Ensuite, Bash attend la commande suivante.

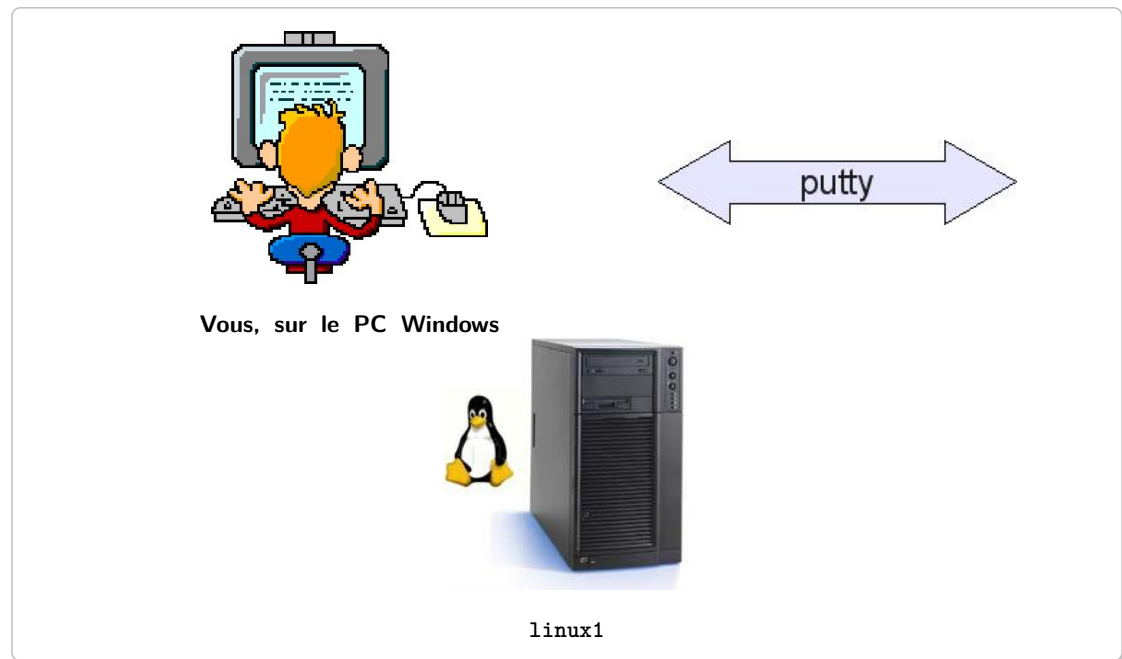
Pour travailler en mode console, il faut savoir dialoguer avec ce shell. Comment lui indiquer par exemple qu'on veut copier un fichier, exécuter un programme Java...? *C'est le principal objectif de ces TDs !*

section[Connexion]Connexion

`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.



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

Votre login (username)

Un 'g' *minuscule* suivi de votre numéro d'étudiant (p. ex. : g52010).

Note Pour GNU/Linux, les minuscules et les majuscules sont différentes.

Votre mot de passe

Le même que votre *mot de passe initial sous Windows*³.

Pas à pas I

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.

section[Dossier personnel et dossier courant]Dossier personnel et 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, etc.
- ▷ Un *dossier* contient des fichiers. Il sert à organiser/re-grouper 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 etc. On a une organisation *hiérarchique* des fichiers.

Le répertoire personnel

Sur GNU/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'utilisateur⁴.

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*.

section[Contenu d'un dossier ou d'un fichier]Contenu d'un dossier ou d'un fichier

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

Visualiser le contenu d'un dossier

Expérience I

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

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

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`.

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 !).

section[Structure d'une commande]Structure d'une commande

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

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 GNU/Linux, *les majuscules et les minuscules n'ont pas le même sens.*

Les espaces

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

 `ls dev1`

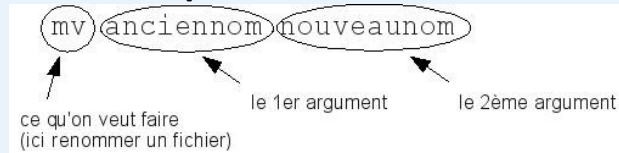
 `lsdev1`

 `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.

section[Modification du mot de passe sous GNU/Linux]Modification du mot de passe sous GNU/Linux

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

Pas à pas 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 : mot de passe 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 GNU/Linux. Je dois aussi aller voir les techniciens ? Non ! Votre professeur de labo peut réinitialiser le mot de passe GNU/Linux à sa valeur initiale.

section[Introduction aux chemins]Introduction aux chemins

Comment désigner un fichier qui se trouve dans un sous-dossier ?
Peut-on se contenter d'indiquer son nom ? Testons !

Expérience 5

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.

Expérience 6

Accès à un sous-dossier

Modifions légèrement l'expérience précédente.

✎ Entrez la commande `cat dev1/readme`.

Ça fonctionne cette fois ! Nous avons donné le *chemin* pour y arriver.

Une première approche des chemins

En première approche, un *chemin* est la suite des dossiers à traverser (séparés par /⁵) pour arriver au fichier (ou au dossier) désiré. Cette suite commence au dossier courant⁶.

section[Déplacement dans une arborescence de fichiers]Déplacement dans une arborescence de fichiers

Nous venons de voir qu'il est possible de désigner un élément dans un sous-dossier. Il est aussi utile de pouvoir se *déplacer* dans la structure des dossiers, de changer de dossier courant. Voyons comment.

Expérience 7

Se déplacer

✎ Entrez la commande `ls`.

▷ Elle affiche le contenu de votre dossier personnel.

✎ 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 dossier `dev1` ;
6. Vérifier que vous avez bien changé de dossier courant ;
7. Afficher le contenu du fichier `readme` ;
8. Revenir dans le dossier personnel ;
9. Afficher à nouveau le contenu du fichier `readme` .

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 ?

section[Commandes courantes]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
- ▷ 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

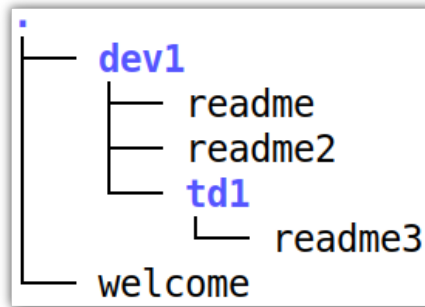
- ▷ `mkdir nomDuDossier` crée un dossier (vide) nommé `nomDuDossier` ;
- ▷ `mv nomDuFichier nouveauNomDeFichier` renomme le fichier donné `nomDuFichier` sous le nom `nouveauNomDeFichier` ;
- ▷ `mv nomDuFichier nomDuDossier` déplace le fichier donné dans le dossier indiqué ;
- ▷ `cp nomDuFichier nouveauNomDeFichier` 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.

Pour chaque commande, un nom de fichier (dossier) peut-être un chemin vers ce fichier (dossier). Par exemple, `rm dev1/readme` supprime le fichier `readme` du dossier `dev1`.

Exercice 4

Assimiler ces commandes courantes

Afin d'assimiler les commandes qu'on vient de voir on vous demande d'arriver à la situation suivante :



1. `readme2` est une copie de `readme`.
2. `td1` est un dossier créé dans `dev1`.
3. `readme3` est aussi une copie de `readme`.

Une fois que c'est fait, supprimez les 2 fichiers et le dossier créés.

Exercice 5

Sans `cd`

Si vous avez utilisé la commande `cd` à un moment ou un autre lors de l'exercice précédent, recommencez sans utiliser cette commande. Le répertoire courant sera en permanence votre dossier personnel.

section[Conclusion]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`, `rm`, `rmdir` et `touch`.
- ▷ **Comprendre la notion de hiérarchie de dossiers.**
- ▷ **Comprendre la notion de dossier courant et de dossier personnel.**
- ▷ **Savoir désigner un sous-dossier en utilisant un chemin simple.**

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 GNU/Linux ?

Sur poÉSI, vous trouverez un document qui vous explique comment avoir facilement un GNU/Linux fonctionnel sur votre PC. Attention toutefois que certaines choses pourront être différentes : autre nom d'utilisateur qui implique un autre nom de répertoire personnel par exemple.

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 !**

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 ?