
TP 1 : prise en main

Tout exercice non marqué d'un ♣ est à terminer pour la semaine prochaine.

ENVIRONNEMENT DE BUREAU

Lorsque vous démarrez l'ordinateur, choisissez **Ubuntu** comme système d'exploitation.

Une fois votre nom d'utilisateur et mot de passe vérifiés, apparaît l'environnement de bureau. Il s'agit d'un ensemble de programmes permettant de manipuler l'ordinateur à travers une interface graphique qui fait analogie à un bureau, avec des icônes, des fenêtres et des menus.

Dans nos salles de TP, l'environnement de bureau est Xfce, tournant au dessus du système d'exploitation GNU/Linux, le tout étant distribué par Ubuntu.

Le menu principal (bouton en haut à gauche) permet de lancer des applications, d'accéder aux outils de paramétrage du système ou encore de fermer la session ou d'éteindre l'ordinateur (GNU/Linux, comme tout système moderne, doit être arrêté proprement et non pas en éteignant physiquement la machine).

La plupart des programmes ainsi que le menu principal disposent d'une entrée « aide » (*help*) permettant d'accéder à l'aide en ligne. Si vous êtes coincé(e), n'hésitez pas à la consulter.

MANIPULATION DE FICHIERS ET RÉPERTOIRES

Les informations stockées sur un ordinateur sont généralement organisées en un système de *fichiers* qui permet aux utilisateurs d'enregistrer leurs données et leurs programmes. Ces fichiers peuvent correspondre à divers types de données : textes, musique, programmes, etc. Certains fichiers particuliers, appelés *dossiers* (ou *répertoires*), servent de « boîtes » pouvant contenir d'autres fichiers. Du point de vue de l'utilisateur, un répertoire est un ensemble de fichiers et de sous-répertoires, désignés par des noms.

Exercice 1 (Gestionnaire de fichiers).

- (1) Ouvrez le gestionnaire de fichiers en cliquant sur l'icône « répertoire personnel » du bureau.
- (2) Quels dossiers et fichiers se trouvent dans votre répertoire personnel ?
- (3) Allez dans le dossier « Documents ».
- (4) Quels dossiers et fichiers se trouvent dans votre dossier Documents ?
- (5) Remontez dans votre répertoire personnel.
- (6) Créez un dossier « Essai » dans votre répertoire personnel.
- (7) Renommez « Essai » en « Info111 ».
- (8) Fermez le gestionnaire de fichiers.

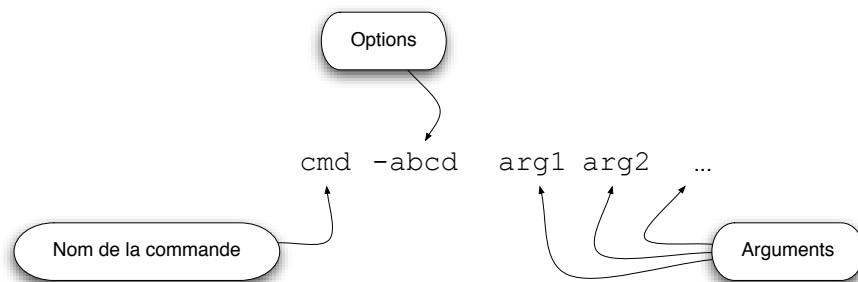
Exercice 2 (Éditeur de texte).

- (1) Lancez un éditeur de texte, par exemple **gedit**. Pour cela utilisez le menu principal (bouton en haut à gauche), où **gedit** se trouve dans la section **Accessoires**.
- (2) Ecrivez un court texte et enregistrez-le sous le nom « **essai.txt** » en faisant bien attention à l'endroit où vous l'enregistrez : on veut qu'il se trouve dans le dossier « **Info111** » que vous avez créé dans votre répertoire personnel.
- (3) Fermez **gedit**.
- (4) Ouvrez le gestionnaire de fichiers en cliquant sur l'icone « répertoire personnel » du bureau.
- (5) Allez dans le dossier « **Info111** » et vérifiez que le nouveau fichier « **essai.txt** » s'y trouve bien.
- (6) Ouvrir « **essai.txt** » en cliquant sur son icone et vérifiez le texte qu'il contient.
- (7) Fermez toutes les fenêtres ouvertes (éditeur de texte et gestionnaire de fichiers).

LE TERMINAL

Une autre façon d'interagir avec le système est d'utiliser un *terminal* dans lequel vous pouvez taper une à une des commandes (qui ne sont rien d'autre que des noms de programmes). Le programme avec lequel vous interagissez pour exécuter les commandes s'appelle le *shell*. À chaque fois, il vous indique qu'il est prêt en affichant en début de ligne une invite (ou *prompt*), ici le caractère dollar (\$). Vous lui indiquez que vous avez fini de taper la commande suivante en appuyant sur la touche **Entrée** (**Enter**).

Pour affiner le comportement d'une commande, on peut lui adjoindre ce que l'on appelle des *options*, représentées par un tiret suivie d'une ou plusieurs lettre(s). On peut également utiliser certaines commandes avec des *arguments*. La forme générale d'une ligne de commande est souvent (mais pas toujours) la suivante :



On peut distinguer diverses sortes de commandes :

- des commandes « simples » qui affichent leur résultat directement dans le terminal (**ls**, **cd**, **cp**, ...). Ces commandes, puisqu'elles utilisent le terminal pour leurs interactions, ne peuvent être exécutées en dehors de celui-ci.
- des commandes qui lancent des applications (**firefox**, **gedit**, etc.). Ces programmes peuvent également être exécutés à l'aide d'une icône de l'interface graphique.

L'utilisation du terminal peut paraître rebutante au premier abord ; pourtant elle n'a rien d'obsolète, bien au contraire. Cela demande un apprentissage supplémentaire par rapport à une interface graphique, mais à la longue, c'est l'une des façons les plus productives de faire exécuter des tâches à un ordinateur, car elle permet d'automatiser simplement de nombreuses actions. Pour l'utilisateur régulier du système que vous êtes amenés à devenir, c'est un outil indispensable.

Exercice 3 (Manipulation de fichiers et répertoires).

Vous allez maintenant apprendre quelques commandes de base, qui vous permettront d'effectuer toutes les opérations utiles sur le système de fichiers (parcours, copie, déplacement, etc.) par l'intermédiaire du terminal. Et vous verrez que cela se révèle souvent bien plus rapide qu'utiliser l'interface graphique.

- (1) Lancez un terminal. Pour cela vous pouvez utiliser le menu principal (bouton en haut à gauche) et cliquer sur « Émulateur de terminal ».
- (2) Exécutez la commande `pwd` et observez le résultat.

`pwd` : La commande `pwd` (*print working directory*) indique dans quel répertoire vous vous trouvez actuellement, autrement dit le répertoire courant.

À l'ouverture d'un nouveau terminal, le répertoire courant est toujours votre répertoire personnel.

- (3) Affichez le contenu de votre répertoire personnel (en utilisant `ls`).

`ls` : La commande `ls` (*list directory*) affiche le contenu du répertoire courant : ses fichiers et ses sous-répertoires.

- (4) Allez dans le dossier « Documents » (en utilisant `cd` et le bon argument).

`cd` : La commande `cd` `<dossierX>` change (*change directory*) le répertoire courant en le répertoire `dossierX`. Elle permet de se déplacer parmi vos dossiers. Utilisée seule (ou avec l'argument `~`), elle vous ramène dans votre répertoire personnel. Utilisée avec l'argument `..` elle vous amène dans le répertoire parent, c'est-à-dire le dossier qui contient le répertoire courant.

Astuce : Complétion automatique : pour aller plus vite, au lieu de taper le nom d'un fichier ou dossier en entier, vous pouvez indiquer le début du nom et essayer de le compléter avec la touche `Tab` (tabulation, à gauche de la touche `A`). Si plusieurs fins sont possibles elles vous seront proposées. Ici, pour « Documents », tapez juste les touches « `D` » puis « `Tab` ».

Exemple :

```
~$ ls
Bureau Image Modèle Musique Téléchargements Vidéo
~$ cd Documents
~/Documents$ ls
~/Documents$ cd ..
~$
```

- (5) Exécutez la commande `pwd` et observez le résultat.
- (6) Affichez dans le terminal le contenu du dossier « Documents ».
- (7) Retournez dans votre répertoire personnel (toujours avec le terminal !).
- (8) Allez dans le dossier « Info111 ».
- (9) Exécutez la commande `pwd` et observez le résultat.
- (10) Affichez le contenu du dossier « Info111 ».
- (11) Dans le dossier « Info111 », utilisez la commande `mkdir` pour créer un sous-dossier « Semaine1 », puis des sous-dossiers « Semaine2 » et « Semaine3 ».

`mkdir` : La commande `mkdir` `<dossierX>` crée (*make directory*) un dossier vide nommé `dossierX`. Si un dossier de même nom existe déjà à cet emplacement, vous obtiendrez un message d'erreur.

Astuce : en utilisant les flèches de déplacement du clavier, vous pouvez récupérer (flèche du haut, flèche du bas) et modifier (flèche de gauche, flèche de droite) les commandes précédentes.

- (12) Dans le dossier « Semaine1 », créez un sous-dossier « Laby ».
- (13) À l'aide de `ls` (et `cd` si besoin), vérifiez que les dossiers ont été créés au bon endroit.
- (14) Affichez dans le terminal le contenu du fichier « `essai.txt` ».

`less` : La commande `less <fichierX>` affiche le contenu du fichier `<fichierX>`.
Pour quitter, taper la touche `Q`.

Exercice 4 (Lancement d'applications et interruption de programme).

Pour lancer une application, il suffit la plupart du temps de taper son nom dans le terminal.

- (1) Allez dans le répertoire `Info111/Semaine1`.
- (2) Lancez l'éditeur de texte `gedit` en tapant dans le terminal la ligne de commande :
`gedit`
- (3) Avec `gedit`, créez un nouveau fichier texte « `essai2.txt` » et enregistrez-le dans le dossier « `Semaine1` » (qui se trouve dans le dossier « `Info111` »).
- (4) Fermez `gedit`.
- (5) En utilisant `ls` (et `cd` si besoin), vérifiez que le fichier a bien été créé.
- (6) Ouvrez à nouveau votre fichier texte `essai2.txt` en tapant dans le terminal la ligne de commande : `gedit essai2.txt`
Pour gagner du temps, pensez à utiliser la complétion automatique (tapez juste le début du nom du fichier et utilisez la touche `Tab` pour compléter, comme expliqué plus haut).
- (7) Fermez `gedit`.

`gedit` : La commande `gedit <fichierX>` ouvre à l'aide de `gedit` le fichier `fichierX` si celui-ci existe déjà, ou crée un nouveau fichier vide nommé `fichierX` et l'ouvre avec `gedit` si ce fichier n'existe pas encore.

- (8) En tapant une seule ligne de commande, créez un fichier vide nommé « `essai3.txt` » et ouvrez-le avec `gedit`.
- (9) Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `ls`). Que se passe-t-il ?
- (10) Toujours sur le terminal, gardez la touche `Ctrl` enfoncée puis pressez et relâchez la touche `C`. Que se passe-t-il ?
- (11) Répétez les opérations en ajoutant cette fois le caractère `&` en fin de commande, c'est-à-dire ouvrez votre fichier texte `essai3.txt` en tapant dans le terminal la ligne de commande `gedit essai3.txt &`
Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `ls`). Voyez-vous une différence ?

Exercice 5 (Aller un peu plus loin ♣).

Pour gagner du temps, pensez à utiliser la complétion automatique autant que possible !

- (1) Renommez le fichier « `essai.txt` » en « `essai1.txt` » (en utilisant `mv`).

`mv` : La commande `mv` `<cheminFichierX>` `<cheminFichierY>` déplace (*move*) le fichier de chemin `<cheminFichierX>` pour lui attribuer le chemin `<cheminFichierY>`. Par exemple si `<cheminFichierX>` est le nom d'un fichier du répertoire courant et `<cheminFichierY>` est un nouveau nom de fichier, la commande change le nom du fichier. Si `<cheminFichierX>` est le nom d'un fichier du répertoire courant et `<cheminFichierY>` est le nom d'un sous-dossier du répertoire courant, la commande déplace le fichier et le met dans le sous-dossier.

- (2) Déplacez le fichier « `essai1.txt` » pour le mettre dans le dossier « `Semaine1` ».

- (3) Copiez « `essai1.txt` » en un nouveau fichier « `essai0.txt` ».

`cp` : La commande `cp` `<fic1>` `<fic2>` duplique (*copy*) le fichier de nom `<fic1>` en un nouveau fichier de nom `<fic2>`. Après exécution de la commande, `<fic1>` et `<fic2>` sont deux fichiers de même contenu mais complètement autonomes.

- (4) Modifiez « `essai0.txt` » (en l'ouvrant avec `gedit` depuis le terminal).

- (5) Vérifiez avec `less` que les deux fichiers sont maintenant différents.

- (6) Effacez le fichier « `essai3.txt` » (en utilisant `rm`).

`rm` : La commande `rm` `<fic>` efface (*remove*) le fichier `<fic>`.

- (7) Par défaut, `rm` ne demande pas de confirmation lorsque vous tentez de supprimer un fichier. Ceci peut se révéler assez dangereux. Trouvez l'option qui permet de demander confirmation.

`man` : Le manuel (*manual*) en ligne pour toutes les commandes accessibles depuis le terminal. Il suffit de taper `man` `<cmd>` pour accéder à la description complète de la commande `cmd`.

Exercice 6 (Premiers programmes en jouant).

Le jeu laby¹ propose plusieurs niveaux de labyrinthes ; pour chacun d'entre eux, le but est de guider la fourmi pas à pas vers la sortie à l'aide d'un programme.

- (1) Lancez laby depuis le terminal.
- (2) Sélectionnez le langage de programmation « c » (en haut à droite).
- (3) Suivez les instructions (en haut à gauche) pour exécuter pas à pas la démonstration.
- (4) Résolvez le maximum de niveaux en une heure ! Vous pouvez vous aider de l'API ci-dessous. Faites attention aux astuces données en bas à gauche pour résoudre les niveaux de façon intelligente. Enregistrez dans le dossier Semaine1 votre solution à chaque niveau en copiant-collant votre programme dans un fichier texte.

. API du jeu laby

```

////////////////////////////////////
// Instructions
////////////////////////////////////

avance();           // Fait un pas vers l'avant
droite();           // Fait pivoter la fourmi vers la droite
gauche();           // ou la gauche
prend();            // Prend le caillou situé sur la case devant la fourmi
pose();             // Pose le caillou devant la fourmi
dit("bonjour");     // Dit bonjour
regarde();          // Renvoie Vide, Caillou, Mur, Toile, Sortie, ou Inconnu
                    // selon ce qui se trouve sur la case devant la fourmi

ouvre();            // Ouvre la porte située devant la fourmi (fin du niveau)

////////////////////////////////////
// Constructions
////////////////////////////////////

// Tant que la condition est respectée, répète les instructions
while(condition) {
    instructions;
}

// Si condition est vrai, exécute les instructions 1 sinon les instructions 2
if (condition) {
    instructions1;
} else {
    instructions2;
}

// Défini une fonction f
void f() {
    instructions;
}

```

Exercice ♣ 7.

Reprendre laby avec un autre langage de programmation (par exemple Python). Saurez vous faire tous les niveaux ?

Exercice 8 (À faire pour la semaine prochaine).

Une heure supplémentaire de Laby en salle libre-service.

1. <https://sgimenez.github.io/laby/>