

TRAVAUX PRATIQUES

Introduction au shell

Dans le cours d'informatique de MP2I/MPI, nous utiliserons un système d'exploitation (*operating system*, ou *OS*) GNU/Linux : Ubuntu (version 22.04 LTS). Celui-ci a l'avantage d'être l'un des plus populaires auprès des débutant-es, et d'avoir beaucoup d'aide en ligne accessible.

Nous ferons une grosse session d'installation de Linux, avant les vacances¹. Aujourd'hui, nous utilisons obligatoirement les machines du lycée. Dans les prochains TP, vous pourrez soit utiliser les machines du lycée, soit utiliser les sites internet que je proposerai depuis votre ordinateur personnel.

Le but de ce TP est de vous apprendre à utiliser un *interpréteur de commande* (aussi appelé *terminal*, ou *shell* dans la suite). Il s'agit d'une sorte de "ligne directe" avec l'ordinateur, dans laquelle on pourra donner des instructions directes à l'OS.

Le terminal nous sera très utile : taper des commandes dedans est souvent plus rapide que d'enchaîner les clics dans une fenêtre, et permet plus d'adaptabilité. C'est un outil indispensable à tout-e futur-e informaticien-ne.

Appelez-moi si quelque chose n'est pas clair ou si vous bloquez!

A Jeu en shell

1. Ouvrez un terminal. Pour cela, cliquez sur la petite icône de fenêtre noire présente dans votre menu ; on peut aussi l'ouvrir en utilisant le raccourci 'Ctrl+Alt+T'.

Pour la suite de ce TP, nous utiliserons un jeu pédagogique pour vous apprendre des commandes de bases du shell.

2. Lancer `bash gameshell.sh` depuis `~`, et faites les missions. Le but est d'aller le plus loin possible!

Vous n'irez pas tous aussi loin en fonction de votre pratique antérieure du shell. Ce n'est pas grave. L'important est d'apprendre et de progresser.

B Compléments sur l'arrêt d'un programme

La plupart des programmes qui ouvrent une fenêtre graphique peuvent être quittés avec la combinaison de touches `Ctrl+Q` ou `Ctrl+W`.

La plupart des programmes interactifs qui sont uniquement en terminal (comme `gameshell`) peuvent être quittés avec la combinaison de touches `Ctrl+D`.

Enfin, si vous êtes dans un terminal qui est en train de faire tourner un programme que vous voulez terminer de force (qu'il soit interactif ou non, en fenêtre graphique ou non) vous pouvez utiliser `Ctrl+C` ou `Ctrl+Z`.

C Compléments sur l'arborescence des fichiers

Dans cette section, les chevrons < > jouent le rôle de guillemets : ils ne font pas réellement partie des chemins.

Dans votre système, les dossiers sont organisés de manière arborescente : un dossier est inclus dans un autre, etc, et ce jusqu'à un dossier appelé racine. Chaque fichier appartient à un dossier.

La racine est appelée `/`. Elle contient différents dossiers comme par exemple `var`. `var` contient lui-même d'autres fichiers et dossiers, comme par exemple `lib`, qui contient lui-même `systemd`². A chaque fichier ou dossier, on associe son chemin d'accès (*access path* ou simplement *path*) : c'est son "adresse". Ainsi, celle de `lib` est `/var/lib`.

1. Si vous souhaitez l'installer vous-même avant cela, venez me voir à la fin du cours.

2. Si vous trouvez cet exemple polémique, vous pouvez probablement passer à section suivante.

Définition 1 (Chemins absolus et relatifs).

Le chemin de la racine est `/`.

Le chemin d'un fichier ou d'un dossier qui n'est pas la racine se décompose ainsi :

```
<chemin_vers_le_dossier_parent_du_fichier>/<nom_du_fichier>
```

Pour différencier les chemins de dossier des chemins de fichiers, on fait souvent terminer les premiers par `/` : `/etc/iptables/` est un chemin de dossier, mais `/etc/group` peut-être un chemin de fichier ou de dossier.

On distingue deux types de chemins :

- les **chemins absolus**, c'est à dire adressés depuis la racine. Par exemple : `/home/<nom_de_l_utilisateur>/Documents/sous_dossier_favori`.
- les **chemins relatifs**, qui ne sont pas adressés depuis la racine. Par exemple : `Documents/sous_dossier_favori`. Lorsqu'une commande essaye de les évaluer, il les complète en chemin absolu en ajoutant le dossier courant comme préfixe. Ainsi, si on se trouve dans `/etc` lorsqu'on évalue le chemin relatif précédent, il sera évalué comme `/etc/Documents/sous_dossier_favori`.

Trois noms de dossiers sont réservés pour servir de raccourci à des dossiers particuliers :

- `~` est le chemin absolu `/home/<nom_de_l_utilisateur>/`, qui est le chemin du dossier personnel de l'utilisateur actuellement connecté.
- `.` est un lien vers le dossier courant.
- `..` est un lien vers le dossier parent.

`~` donc est votre dossier personnel d'utilisateur, où sont rangés vos dossiers et fichiers personnels. Les dossiers en dehors de `~` sont des dossiers du système d'exploitation et des programmes installés (où ils stockent tout ce qui est nécessaire à leur fonctionnement, y compris... eux-mêmes) et les dossiers des autres utilisateurs.

Remarque. La touche de tabulation propose des autocomplétions. Utilisez-la autant que possible! Le temps gagné n'est pas négligeable.

D Continuer chez soi

Si vous avez déjà un système d'exploitation basé sur Unix (Linux ou MacOS), vous pouvez continuer chez vous en téléchargement `gameshell.sh` à ce lien :

```
https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh
```