

Manuel utilisateur

Ce guide à pour objectif de guider l'utilisateur dans l'utilisation du SGF écrit par Yanis KOUIDRI. Vous trouverez dans ce guide toutes les commandes qu'il est possible d'exécuter dans l'interface en ligne de commande du programme. Pour chaque commande il y aura une partie théorique (sémantique) de la commande suivie d'un ou plusieurs exemple d'utilisation.

Se déplacer

Pour se déplacer au sein de l'arborescence du SGF il faut utiliser la commande `cd`.

```
cd [prochain_saut]
```

`[prochain_saut]` peut être :

- Un dossier présent dans le répertoire courant
- Le dossier parent identifier par `..`



Si `prochain_saut` n'existe pas, alors un message d'erreur est affiché et le répertoire courant ne change pas.



Il n'est pas possible de se déplacer de plus d'un répertoire à la fois.

Exemple :

```
> cd Dossier_fils
```

On se déplace dans le dossier nommé Dossier_fils qui est présent dans le répertoire actuel.

```
> cd ..
```

On remonte d'un dossier.

Se repérer

Pour savoir où l'on se trouve dans l'arborescence il faut utiliser la commande `pwd` sans argument.

Le chemin absolu du répertoire courant sera alors affiché.

Exemple :

```
> pwd
/bin/boot/
```

On se trouve donc dans le dossier `/boot` qui se trouve dans le dossier `/bin` qui se trouve à la racine (`/`).

Lister

Pour connaître le contenu du répertoire courant il existe deux commandes.

Rapidement

Pour obtenir le nom de tous les éléments (fichier et dossier) du répertoire courant il faut utiliser la commande `ls` sans argument.

Exemple :

```
> ls
config.xml
index.html
note.txt
Archive
Autres
```

On voit ici que notre répertoire courant contient 5 élément. Cependant on ne peut pas savoir si Archive et Autres sont des dossiers ou bien des fichiers.

En détail

Pour obtenir le nom de tous les éléments (fichier et dossier) du répertoire courant plus leur taille et leurs droits il faut utiliser la commande `ll` sans argument.

Exemple :

```
> ll
drwx----- 10240 Archives
-rw-r--r-- 541 notes.txt
```

On voit ici qu'il y a un dossier et un fichier.

Créer

Un fichier

Pour créer un fichier il faut utiliser la commande `touch`

```
touch [nom_fichier]
```

Où `[nom_fichier]` est le nom souhaité du fichier à créer.



Il n'est possible de créer qu'un seul fichier à la fois.



Il n'est pas possible de créer fichier ayant le même nom qu'un autre fichier/dossier du répertoire courant.



Toutes les créations seront faites dans le répertoire courant.

Exemple :

```
> touch info
> touch voiture
```

J'ai créé deux fichiers nommés `info` et `voiture` dans le répertoire courant.

Un dossier

Pour créer un dossier il faut utiliser la commande `mkdir`

```
mkdir [nom_dossier]
```

Où `[nom_dossier]` est le nom souhaité du dossier à créer.



Il n'est possible de créer qu'un seul dossier à la fois.



Il n'est pas possible de créer dossier ayant le même nom qu'un autre fichier/dossier du répertoire courant.



Toutes les créations seront faites dans le répertoire courant.

Exemple :

```
> mkdir Src  
> mkdir Bin
```

J'ai créé deux dossiers nommés `Src` et `Bin` dans le répertoire courant.

Modifier

Il est possible d'éditer un fichier grâce à la commande `vi`.



Seul les fichiers peuvent être édités.

La commande `vi` permet d'ajouter du contenu à la fin d'un fichier. Il n'est pas possible de supprimer du contenu déjà présent dans le fichier.

Utilisation :

```
vi [nom_fichier]
```

Où :

- `[nom_fichier]` est le nom du fichier à modifier (il doit au préalable avoir été créé)

Une fois la commande tapée, l'utilisateur doit rentrer la chaîne de caractère qu'il veut saisir dans le fichier.

Exemple :

```
> vi compte  
J'ai dépensé 5 euros.  
>
```

Je viens d'ajouter la chaîne de caractères `J'ai dépensé 5 euros.` dans mon fichier `compte`.

Afficher

Pour afficher le contenu d'un fichier il faut utiliser la commande `cat`.

```
cat [nom_fichier]
```

Où :

- `[nom_fichier]` est le nom du fichier à afficher (il doit exister).

Exemple :

```
> cat rappel  
Faire les courses  
>
```

Copier

Pour copier un fichier ou un dossier vide il faut utiliser la commande `cp`.

```
cp [a_copier] [nom_copie]
```

Où :

- `[a_copier]` est le nom du fichier à copier.
- `[nom_copie]` est le nom que va recevoir la copie.

Exemple :

```
> cp liste copie_liste
```

Je viens de faire une copie du fichier `liste` qui s'appelle `copie_liste`.



Les copies ne peuvent s'effectuer que dans le répertoire courant.



Il n'est pas possible de copier un dossier qui contient un ou plusieurs éléments.

Renommer

Pour renommer un fichier ou un dossier il faut utiliser la commande `mv`.

```
mv [element] [nouveau_nom]
```

Où :

- `[element]` est le nom du fichier à renommer.
- `[nouveau_nom]` est le nouveau nom du fichier.

Exemple :

```
> mv archive archive
```

Je viens de renommer `archive` en `archive`.

Supprimer

Pour supprimer un fichier ou un dossier vide il faut utiliser la commande `rm`.

```
rm [element]
```

Où :

- `[element]` est le nom du fichier/dossier vide à supprimer.

Exemple :

```
> rm config.old
```

Je viens de supprimer le fichier `config.old` qui était présent dans mon dossier courant.



Il n'est pas possible de supprimer un dossier qui contient un ou plusieurs éléments.

Archiver

Pour créer une archive à partir d'un dossier il faut utiliser la commande `tar`.

```
tar [dossier]
```

Où :

- `[dossier]` est le nom du dossier à archiver

Exemple :

```
> tar impots
```

Je viens de créer une archive qui s'appelle `impots.tar` qui a une taille qui correspond à la somme des tailles des fichiers que contient le dossier `impots` et ses éventuels sous dossiers.



Il n'est pas possible de faire une archive à partir d'un fichier.



Il n'est pas possible de faire une archive d'un dossier si une archive de celui-ci existe déjà dans le répertoire courant.