

# TP shell n° 1

Rémi Nollet, d'après Juliusz Chroboczek

septembre 2022

## Interface graphique

### Exercice 1.

1. Ouvrez le gestionnaire de fichiers et explorez l'arborescence du système en repérant le répertoire racine puis votre répertoire *home*. (Attention — le répertoire qui s'appelle « *home* » n'est pas votre répertoire *home*.)
2. Créez un répertoire appelé *essai* dans votre répertoire *home*.
3. Renommez le répertoire *essai* en *test*.

### Exercice 2.

1. Lancez l'éditeur de texte Emacs.
2. Créez un fichier nommé *poeme.txt* dans votre répertoire *home*, et écrivez quelques vers de votre poème favori. Sauvegardez. Ne quittez pas Emacs.
3. À l'aide du gestionnaire de fichiers, localisez le fichier *poeme.txt* et visualisez-le.

## Shell

À partir de maintenant, vous n'êtes pas autorisés à utiliser le gestionnaire de fichiers. Quittez-le.

### Exercice 3.

1. Lancez un émulateur de terminal (*console*).
2. Dans le terminal, tapez ce qui suit :  

```
echo toto
```

puis validez.
3. Tapez (et validez)  

```
echo -n toto
```

**Notation** On utilise :

- la *police de caractères de machine* à écrire pour décrire les commandes à taper ;
- la *police italique* (ou parfois la *police penchée*) pour représenter une partie de commande à remplacer.

**Rappel** La commande `man` ouvre un programme qui permet de lire une page de documentation. Vous pouvez faire défiler le texte en utilisant la touche *espace*. Pour sortir, pressez la touche *q*.

#### **Exercice 4.**

1. Testez les commandes `whoami`, `who` et `w` et observez les informations affichées.
2. Lisez la page de manuel de la commande `w`. Quelles sont les options qu'accepte `w` ?
3. Combien d'options accepte la commande `ls` ?

**L'historique** Les flèches  $\uparrow$  et  $\downarrow$ , permettent de faire défiler les commandes précédentes. Avec les flèches  $\rightarrow$  et  $\leftarrow$ , vous pouvez corriger un détail de chaque ligne de commande.

**Exercice 5.** Consultez la pages de manuel de la commande `who` sans retaper la commande `man`.

#### **Exercice 6.**

1. À l'aide de la commande `pwd`, déterminez le répertoire dans lequel vous vous trouvez. Si vous n'êtes pas dans votre répertoire *home*, déplacez-vous à l'aide de la commande `cd ~`.
2. À l'aide de la commande `ls`, vérifiez que le fichier `poeme.txt` créé auparavant s'y trouve bien. Affichez son contenu, d'abord à l'aide de la commande `cat`, ensuite à l'aide de la commande `more`, enfin à l'aide de la commande `less`.
3. Affichez la liste des fichiers du répertoire `/etc`. Affichez le contenu du fichier nommé `/etc/profile`. (Vous n'avez pas besoin d'en comprendre le contenu.)

#### **Exercice 7.**

1. Créez un répertoire nommé `cours` dans votre répertoire *home*. Dans ce répertoire créez un répertoire `prog`.
2. Déplacez le fichier `poeme.txt` dans le répertoire `prog`. Vérifiez que vous ne vous êtes pas trompé (et notamment que vous n'avez pas créé un fichier nommé `prog` dans le répertoire courant).

3. A partir de votre répertoire personnel essayez de copier le répertoire `cours` sous le nom `cours-sauvegarde`. Lisez le message d'erreur. Trouvez l'option qui vous permet de copier ce répertoire.
4. Faites une copie de `poeme.txt` qui s'appelle `.poeme` (avec un point au début). Affichez le contenu du répertoire; que remarquez-vous? Réessayez en ajoutant l'option `-a` à `ls`.

**La complétion automatique** La touche *tabulation* demande au *shell* de compléter une commande ou un nom de fichier incomplets. Intuitivement, c'est la touche « fais ce que je pense ».

**Exercice 8.** Affichez votre fichier `poeme.txt` en tapant « `cat p` » puis en appuyant sur la touche tabulation.

**Exercice 9.**

1. À l'aide de la commande `rmdir`, essayez de supprimer le répertoire `cours-sauvegarde`. Que se passe-t-il?
2. Consultez la documentation de la commande `rm` pour trouver l'option qui permet d'effacer un répertoire et son contenu (on parle de *suppression récursive*). Appliquez cette commande au répertoire `cours-sauvegarde`.

**Exercice 10.** Par défaut, le résultat de la commande `ls` est minimal.

1. Trouvez l'option permettant d'afficher tous les fichiers, y compris les fichiers cachés. (Indication : vous vous en êtes déjà servis.)
2. Trouvez l'option permettant d'afficher le maximum d'informations sur les fichiers listés.
3. Que fait l'option `-F`?

**Emacs et le clavier** Dans Emacs, la notation `C-g` indique qu'il faut maintenir la touche *Control* enfoncée pendant qu'on tape un `x`. La notation `M-x` (« meta-x ») indique qu'il faut maintenir la touche *Alt* enfoncée pendant qu'on tape un `x`. Quelques commandes utiles :

- `C-g` : interrompre l'action en cours ;
- `C-x 2` : scinder la fenêtre courante ;
- `C-x 1` : supprimer toutes les fenêtres sauf la fenêtre courante ;
- `C-x o` (la lettre « o ») : changer de fenêtre ;
- `C-x C-s` : sauvegarder le fichier.

**Exercice 11.** Ouvrez simultanément le fichier `poeme.txt` et un fichier `poeme2.txt` dans Emacs. Faites un copier-coller depuis `poeme.txt` dans `poeme2.txt`. Sauvegardez. Vérifiez à l'aide de la commande `diff -u` que les deux fichiers sont bien identiques.

**Exercice 12** (Un premier programme C).

1. À l'aide d'Emacs (que vous n'avez toujours pas quitté), créez un fichier `.emacs` dans votre répertoire `home` contenant la ligne suivante :

```
(setq-default c-basic-offset 4)
```

Quittez Emacs et relancez-le. Assurez-vous qu'Emacs n'affiche pas d'erreur.

2. Toujours à l'aide d'Emacs, créez un fichier `bonjour.c` dans lequel vous taperez le code suivant en appuyant sur la touche `tab` au début de chaque ligne :

```
#include <stdio.h>

int
main(void)
{
    printf("Bonjour.\n");
    return 0;
}
```

3. Compilez votre programme à l'aide de la commande

```
$ gcc -Wall bonjour.c
```

S'il y a des erreurs, corrigez-les et recommencez. Quel fichier a été créé ? Exécutez-le à l'aide de la commande `./a.out`. Supprimez le fichier `a.out`.

4. Compilez votre programme de nouveau, mais cette fois-ci à l'aide de la commande

```
$ gcc -Wall -o bonjour bonjour.c
```

Quel fichier a été créé ? Exécutez-le.

5. Dans Emacs, tapez `M-x compile`. Remplacez la ligne de commande proposée par la ligne

```
gcc -Wall bonjour.c
```

puis validez.

**Exercice 13** (Erreurs à la compilation).

1. À l'aide du *shell*, copiez le fichier `bonjour.c` en un fichier `bonjour2.c`. Ouvrez-ce dernier à l'aide d'Emacs, et supprimez le point-virgule « ; » de la ligne 8. (Remarquez que le numéro de ligne courant est affiché tout en bas de votre tampon.)
2. Compilez le programme `bonjour2.c` à partir du shell. Lisez le message d'erreur. Lisez-le de nouveau, et assurez-vous de l'avoir compris.
3. Compilez le programme `bonjour2.c` à partir d'Emacs (`M-x compile`). Dans le tampon « *\*Compilation\** », cliquez sur le message d'erreur. Que se passe-t-il ?