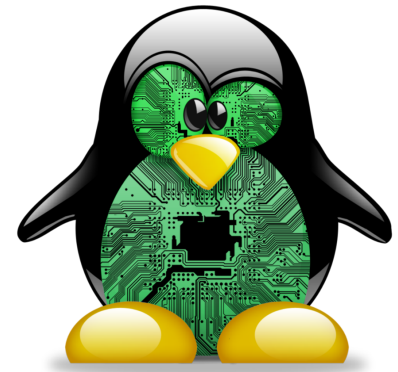


INTRODUCTION AUX COMMANDES UNIX

OBJECTIFS :

- ouvrir un terminal `tty`
- A partir d'un terminal (console) :
 - se repérer et se déplacer dans l'arborescence
 - créer et supprimer des dossiers
 - créer, copier et déplacer des fichiers
 - obtenir des informations
 - lancer des applications
- utiliser des raccourcis clavier



1 Premiers pas au labo d'informatique - salle 112 → 115

1.1 Découverte de la console

L'interface proposée au labo d'info est simple sans surcharge. Les différents logiciels que vous pourrez être amenés à utiliser se lancent à partir des onglets en bas, au milieu de l'écran.

Une autre façon d'utiliser l'ordinateur peut être réalisée grâce à la console (ou terminal). Pour la lancer, presser les touches `Ctrl+Alt+T`. Une console se lance. Taper simplement `firefox` puis presser la touche `Enter` pour se rassurer.

Aller sur la page `gondor-carnot.fr`. A l'aide du cahier de texte *PCSI-2*, rendez vous directement sur l'article `INTRO-Tp-1`.

REMARQUE : la console qui vous a permis de lancer Firefox est verrouillée jusqu'à la fermeture du navigateur. Faites simplement `Ctrl+Alt+T` pour lancer une nouvelle console.

Q - 1 : *Tester les commandes `w` et `whoami` puis observer les résultats.*

L'instruction `man` permet d'avoir de l'aide. **EXEMPLE :** `man whoami`. Tapez `q` pour quitter l'aide.

Q - 2 : *Tester la commande `pwd` (pour `print working directory`)*

La commande retourne le répertoire de travail actuel.

Q - 3 : *Taper successivement `ls` (listing) et `ls -l` (listing long et complet).*

La commande retourne la liste des fichiers et des dossiers qui composent le répertoire.

ATTENTION ! Oups, je crois que j'ai fait `Ctrl+Alt+F1`... mais ouf, pour revenir à l'environnement graphique, je n'ai qu'à faire `Ctrl+Alt+F7` !

```
Fichier Édition Onglets Aide
pi@raspberrypi:~$ w
18:48:59 up 13 min, 2 users, load average: 0,28, 0,57, 0,32
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
pi        tty7     :0            18:35   13:57 40.15s 1.07s /usr/bin/lxsess
pi        tty1     -             18:35   13:57 0.26s  0.22s -bash
pi@raspberrypi:~$ whoami
pi
pi@raspberrypi:~$ █
```

```
Fichier Édition Onglets Aide
pi@raspberrypi:~$ w
18:48:59 up 13 min, 2 users, load average: 0,28, 0,57, 0,32
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
pi        tty7     :0            18:35   13:57 40.15s 1.07s /usr/bin/lxsess
pi        tty1     -             18:35   13:57 0.26s  0.22s -bash
pi@raspberrypi:~$ whoami
pi
pi@raspberrypi:~$ w
19:00:02 up 24 min, 3 users, load average: 0,03, 0,10, 0,18
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
pi        tty7     :0            18:35   25:00 57.75s 1.23s /usr/bin/lxsess
pi        tty1     -             18:35   25:00 0.26s  0.22s -bash
pi        tty3     -             18:59   17:00s 0.51s  0.18s -bash
```

Pour revenir à l'environnement graphique, taper `Ctrl+Alt+F7` !

1.2 Arborescence de fichiers

1.2.1 A la racine

Les fichiers et répertoires sont organisés sous forme arborescente :

- La racine est le répertoire qui contient tout. Il se note `/` (slash) et contient un certain nombre de sous-répertoires, dont, en général :
 - **etc** : contient les fichiers de configuration du système, accessible seulement par l'administrateur système.
 - **bin** : contient les exécutables des programmes de base (binary).
 - **usr** : contient les autres exécutables (user).
 - **var** : contient fichiers variables tels que journaux systèmes, messages, fichiers temporaires.
 - **home** : contient les répertoires utilisateurs.
 - **dev** : périphériques (devices).
 - **tmp** : fichiers temporaires.
 - **media** : disques durs et clef USB (pratique quand on lance Linux depuis une clé USB ou un Dvd bootable) ...
- le répertoire courant s'obtient avec `."`, le répertoire parent avec `".."` . Le répertoire de l'utilisateur logué est raccourci en `"~"`.

1.2.2 Changement de répertoire

Dans un répertoire nommé `doss` qui contient un répertoire `doss-1` et un répertoire `doss-2`, il est possible d'accéder au répertoire `doss-2` en tapant `cd doss-2`. La commande `cd` signifie *change directory*.

Q - 4 : *En utilisant `cd`, choisir un dossier `y` entrer.*

Si cela a fonctionné, l'adresse entre votre nom et le `$` a dû changer.

Q - 5 : *Taper `cd . . .` Taper à nouveau `ls`. Que s'est-il passé?*

REMARQUE : penser à utiliser la touche `Tab` pour la complétion automatique. La console devient alors plus rapide que le clicodrome ! Une double tabulation retourne tous les résultats possibles. C'est également très pratique.

Q - 6 : *Remonter à la racine de l'ordinateur / en tapant `cd /`.*

1.2.3 Dossiers et fichiers

Pour créer un dossier, on utilise la commande `mkdir` (*make directory*).

- Pour créer un dossier INFO dans le répertoire courant, taper : `mkdir INFO`. Le résultat n'apparaît pas. Il faut faire `ls` pour voir si le nouveau dossier apparaît bien.
- Pour créer un dossier Dossier-3 à l'adresse `~/Dossier-1/Dossier-2`, écrire :
`mkdir ~/Dossier-1/Dossier-2/Dossier-3`

Q - 7 : A la racine de votre compte utilisateur (pour rappel « ~ ») créer un dossier INFO et dans ce dossier INFO, un dossier TEST.

REMARQUE : si vous avez raté en créant le dossier ailleurs ou en l'ayant appelé autrement (genre INFU), vous pouvez le supprimer en utilisant la commande `rmdir INFU`.

Q - 8 : Avec la commande `touch`, créer un fichier `test.txt` (`touch test.txt`).

Q - 9 : Ce fichier n'ayant rien à faire dans le répertoire INFO, le déplacer dans le dossier TEST avec `cp` (pour copy) et `mv` pour (move).

`mv ~/INFO/text.txt ~/INFO/TEST`

ATTENTION ! bien respecter la syntaxe pour `mv` sinon votre fichier risque de disparaître. !

REMARQUE : il est possible de profiter de cette étape pour renommer le fichier.

`mv ~/INFO/text.txt ~/INFO/TEST/texte.txt`

Q - 10 : Avec la souris et le navigateur de fichier en bas de l'écran, vérifier que le fichier a bien été créé.

```
pi@raspberrypi:~ $ mkdir Info
pi@raspberrypi:~ $ cd Info/
pi@raspberrypi:~/Info $ mkdir doss
pi@raspberrypi:~/Info $ mkdir doss/doss-1
pi@raspberrypi:~/Info $ mkdir doss/doss-2
pi@raspberrypi:~/Info $ tree
.
├── doss
│   ├── doss-1
│   └── doss-2
└── 3 directories, 0 files
pi@raspberrypi:~/Info $ cd doss
pi@raspberrypi:~/Info/doss $ ls
doss-1  doss-2
pi@raspberrypi:~/Info/doss $ cd doss-2
pi@raspberrypi:~/Info/doss/doss-2 $ cd ../doss-1
pi@raspberrypi:~/Info/doss/doss-1 $ cd ..
pi@raspberrypi:~/Info/doss $
```

1.3 Raccourcis clavier

Q - 11 : Ouvrir le fichier `texte.txt` avec `gedit` (`gedit texte.txt`).

Affranchissez-vous du clavier en utilisant les raccourcis ci-contre :

Q - 12 : A partir de ces raccourcis, écrire 1024 fois votre nom dans le fichier `texte` créé précédemment. Essayer d'être le plus rapide possible...

Raccourcis	Effet
Ctrl+a	sélectionne tout le texte
Ctrl+c	copie la sélection
Ctrl+x	coupe la sélection
Ctrl+v	colle la sélection
Alt+F4	ferme la fenêtre courante.

2 Environnement de développement intégré (IDE)

Coder dans un langage déterminé nécessite un environnement de travail adapté. On appelle cet environnement un IDE pour *Integrated Development Environment*. Nous en rencontrerons plusieurs :

- **Idle** : programme en **Python**. Léger et basique, il s'utilise simplement.
- **Pyzo** : programme en **Python**. Tout comme **Spyder**, son environnement regroupe un ensemble de fonctionnalités utiles pour du calcul numérique.
- **Scilab** : programme en...Scilab. Destiné au calcul numérique, dans ce domaine, il est moins lourd à traîner que **Python**; tout y est vecteur.

2.1 Démarrer avec Python

2.1.1 Python en console

Ouvrir un terminal en cliquant sur `Ctrl+Alt+T`. Taper l'instruction `python`. Vous devez voir apparaître dans la marge des chevrons `>>>`.

Q - 13 : Tapez `1 + 1` puis appuyer sur la touche *Entrée*. Tapez `1 == 1` puis appuyer sur *Entrée*.

Cependant, il vaut mieux consigner le programme dans un fichier pour l'éditer et l'exécuter à souhait. Pour sortie, taper `quit()`.

2.1.2 Interpréteur

Depuis la console, se placer dans le répertoire `INFO` créé précédemment et exécuter la commande `idle3` (le 3 signifie que l'on utilise la version 3 de Python).

Il apparaît une fenêtre appelée *Python Shell*, qui est l'interpréteur (littéralement l'emballage) de commandes Python.

Q - 14 : Exécuter (taper *Enter*) successivement dans l'interpréteur les expressions suivantes et comprendre ce qu'elles font.

C'est sympa mais **Python** est un peu plus qu'une calculatrice.

```
2 + 5
2 ** 5
2.1 / 3.5
19.0 / 5.0
19 // 5
19 / 5
19 % 5
```

2.1.3 Éditeur

Pour sauvegarder le travail, il convient d'utiliser l'éditeur. Ce dernier permettra d'écrire les instructions puis les programmes à venir dans un fichier (de terminaison `.py` ne contenant ni espace, ni caractères spéciaux) . Il sera possible de partager le fichier et de l'exécuter.

Q - 15 : Avec le raccourci `Ctrl+n` créer un premier fichier **Python** `exo-1.py` si vous êtes dans l'interpréteur ou taper `idle3 exo-1.py` dans une console.

Dorénavant, en Tp, nous travaillerons toujours dans des fichiers `.py`.

```
import os.path

LDos = ['INTRO', 'ALGO-I', 'TAB2D', 'ALGO-II', 'GRAPHES']

compt = 0
for Dos in LDos:
    compt += 1
    chemin = './' + str(compt) + '-' + Dos
    os.mkdir(chemin)
    for i in range(1, 6):
        os.mkdir(chemin + '/' + Dos + '-' + str(i))
```

FIGURE 1 – Script **Python** pour créer automatiquement l'arborescence.

Q - 16 : Télécharger le fichier `Doc-INFO-INTRO-Tp-1.py` depuis l'article lié au Tp (Fig 1). Placer le fichier dans le dossier `INFO` que vous avez créé.

2.2 Exercices

2.2.1 Commentaires, entrée/sortie et affectation : #, input, print, =

Q - 17 : Recopier le programme suivant dans la fenêtre principale:

```
print ( " Bonjour le monde ! " )
# On demande maintenant le nom de l'utilisateur
nom = input ( " Écrire votre nom : " )
print ( " Bonjour " , nom)
```

Dans l'exemple, `nom` est une variable. Elle stocke la chaîne de caractère entrée par l'utilisateur. L'opérateur `=` est l'opérateur d'affectation. Il permet d'écrire des données dans une variable.

Le texte derrière le dièse `#` est un commentaire. Il n'est pas pris en compte par **Python**, mais permet de rendre plus clair un programme par le lecteur. On peut également librement sauter des lignes dans un programme.

Q - 18 : Modifier le programme pour obtenir et afficher le nom et le prénom de l'utilisateur.

2.2.2 Jouons avec les nombres

Q - 19 : Reprendre dans l'éditeur les expressions tapées dans l'interpréteur à la partie 2.1.2. Obtenez vous le même résultats ?

Q - 20 : Reprendre la question précédente en utilisant **Python 2.7**. Y a-t-il un problème ?

ATTENTION ! dans le futur être très vigilant ! N'utiliser que **Python 3**.

Q - 21 : Recopier et exécuter le programme suivant et commenter le résultat. Adapter pour obtenir ce que vous pensiez voir.

```
n = input ( " Entrer un nombre : " )
n = n + 2
print (n)
```

2.2.3 Au boulot !

Q - 22 : En toute autonomie :

- Écrire un programme qui élève un nombre au cube.
- Écrire un programme qui, partant du prix hors taxe d'un article, ajoute la TVA (de 20%) et affiche "Le prix TTC est de. . ."
- Écrire un programme qui demande le prénom et l'année de naissance d'une personne, puis qui renvoie une phrase comprenant le prénom et l'âge de la personne.