

Travaux pratiques n°1  
Manipulation d'un système d'exploitation  
Environnement de développement intégré

F. CUVELLIER, J. TANOI

M.P.S.I. 1, 2016–2017

## 1 Manipulation d'un système d'exploitation

### 1.1 Créer un répertoire sous Linux

#### Exercice 1.

1. Ouvrir sa session sous Linux et accéder à un émulateur de terminal.
2. Exécuter la commande `tcsh`. Qu'effectue-t-elle ? (Si cette commande n'est pas disponible, c'est sans doute que `bash` est le seul shell de commandes installé sur la machine.)
3. Créer un répertoire à l'aide de la commande `mkdir -m 770 TP01`.
4. Accéder à ce répertoire (`cd TP01`).
5. Exécuter la commande `which vim`. Quel en est le résultat ?
6. Exécuter la commande `which emacs`. Quel en est le résultat ?

### 1.2 Création d'un fichier et exécution de son contenu par Python

#### Exercice 2.

1. Utiliser l'un des éditeurs de texte Vim et Emacs pour créer un fichier `essai.py` (par exemple, `vim essai.py`).
2. Entrer la ligne `print('Bonjour', 'monde!')`, l'enregistrer et fermer le fichier `essai.py`. (Avec Vim, taper successivement les caractères `i`, `p`, `r`, `i`, `n`, `t`, etc., et finir avec la touche d'échappement et `:x`.)
3. Exécuter le script `essai.py` à l'aide de l'application `python` (`python essai.py`).

### 1.3 Quelques opérations de gestion de fichiers avec le shell de commandes

#### Exercice 3.

1. Renommer le fichier `essai.py` : `mv -iv essai.py essai0.py`. Exécuter la commande `ls -al` pour vérifier le résultat.

2. Copier le fichier `essai0.py`: `cp -iv essai0.py essai1.py`. Exécuter la commande `ls -al` pour vérifier le résultat.
3. Copier plusieurs fois le fichier `essai0.py`: avec le shell de commandes `tcsh`, exécuter les commandes suivantes, ligne par ligne.

```
@ num=0
while ( $num < 100 )
@ num ++
cp -ipv essai0.py essai$num.py
end
```

Vérifier le résultat avec la commande `ls -al`.

(Avec le shell de commandes `bash`, exécuter les commandes suivantes, ligne par ligne,

```
num=0
while (( $num < 100 ))
do let num++
cp -ipv essai0.py essai$num.py
done
```

ou, en *une seule* ligne,

```
num=0 ; while (( $num < 100 )) ; do
  let num++ ; cp -ipv essai0.py essai$num.py ; done
```

(Dans la troisième ligne, ne pas mettre d'espace entre `num` et `++`.) Pour vérifier le résultat, la commande `ls -al` convient encore.)

4. Supprimer les fichiers `essai10.py`, `essai11.py`, etc.: `rm -v essai[1-9][0-9]*`.
5. Supprimer le répertoire `TP01` (commande `rmdir`).

## 1.4 Pour voir

**Exercice 4.** Reproduire les mêmes opérations avec le système d'exploitation Windows de la machine.

# 2 Environnement de développement intégré

## 2.1

Un *environnement de développement intégré* est un logiciel qui permet d'écrire des programmes à l'aide d'un éditeur de texte adapté, de les exécuter et d'en corriger les erreurs.

Certains éditeurs de texte performants, comme Emacs et Vim, peuvent faire office d'environnements de développement intégré. On profite alors de toute la puissance de l'éditeur (mais il faut d'abord apprendre à l'utiliser). On peut aussi se servir de l'environnement de développement intégré qui accompagne parfois le logiciel. Par exemple, le concepteur de Python a créé un environnement de développement intégré, appelé IDLE, d'apparence sommaire mais d'usage com-

## 2.2 IDLE

### Exercice 5.

1. Trouver IDLE.
2. Ouvrir une session pour Python.
3. Activer les différents menus déroulants pour voir ce qu'ils proposent.

## 2.3 Premiers pas en Python

Les opérations élémentaires (addition, multiplication, soustraction, division) s'effectuent en entrant le premier opérande, le signe opératoire (+, \*, -, /) et le deuxième opérande, puis en appuyant sur la touche d'entrée (retour-chariot).

Les commandes suivantes permettent de calculer le produit des nombres entiers de 1 à 10.

```
p = 1

for k in range(1, 11):
    p = p * k
```

p

Le signe = sert à affecter une variable (en première approximation). L'instruction `range(a, b)` retourne l'ensemble des nombres entiers de  $a$  à  $b$ ,  $a$  inclus,  $b$  exclu. Noter la présence du `:` à la fin de la ligne du `for`, ainsi que la ligne blanche à la fin du bloc qui commence par ce `for`.

Pour afficher la valeur d'une variable dans un bloc de commandes, on se sert de la fonction `print()`. En ligne, il suffit d'entrer le nom de la variable.

### Exercice 6.

1. On définit une suite de nombres réels  $(x_n)_{n \in \mathbf{N}}$  par les relations  $x_0 = 3$  et  $x_{n+1} = 2x_n + 5$ . Imiter le script précédent pour afficher  $x_n$  pour les indices  $n$  plus petits que 10.
2. Même question pour la suite de nombres réels  $(F_n)_{n \in \mathbf{N}}$  définie par les relations  $F_0 = 0$ ,  $F_1 = 1$  et, pour tout nombre entier naturel non nul  $n$ ,  $F_{n+1} = F_n + F_{n-1}$ .

**Exercice 7.** Voici la première strophe du *Bateau ivre*, d'Arthur Rimbaud (texte de la copie de Verlaine) :

Comme je descendais des Fleuves impassibles  
Je ne me sentis plus guidé par les haleurs :  
Des Peaux-Rouges criards les avaient pris pour cibles  
Les ayant cloués nus aux poteaux de couleurs.

La recopier (en respectant la ponctuation) et l'afficher avec Python. La chaîne de caractères qui représente un texte sur plusieurs lignes est délimité par des guillemets triples (`'''` ou `"""`), au début et à la fin.