



# Informatique

Travaux Pratiques

**Axel LE BOT**



Copyright © 2017-2018 Axel LE BOT

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



# Table des matières

## I

## Shell

<b>1</b>	<b>TP1 : Manipulations de l'environnement et des fichiers sous UNIX .....</b>	<b>11</b>
1.1	Exercice 1 : Découverte de quelques commandes d'archivage	11
1.2	Exercice 2 : Utilisation des masques de création de fichiers	12
1.3	Exercice 3 : Manipulation du Systeme de fichier et des droits de navigation	12
1.4	Exercice 4 : Manipulation d'expression régulière	13
<b>2</b>	<b>TP2 : Scripts Shell .....</b>	<b>15</b>
2.1	Exercice 5 : Un premier script	15
2.2	Exercice 6 : Comptage des paramètres	15
2.3	Exercice 7 : Portée des variables	16

## II

## C++

<b>3</b>	<b>TP1+TP2 : Tableaux, matrices et Fonctions recursives ..</b>	<b>19</b>
<b>3.1</b>	<b>Exercice sur des tableaux</b>	<b>19</b>
3.1.1	Fonction sur les tableaux non triés .....	19
3.1.2	Algorithmes de tri de tableaux .....	19

3.1.3	Fonctions sur les tableaux triés .....	19
3.1.4	Exercice sur les Fonctions récursives .....	19
<b>4</b>	<b>TP3+TP4 : Manipulation des arbres .....</b>	<b>21</b>
<b>4.1</b>	<b>Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées</b>	<b>21</b>
4.1.1	Exercice 1 : Mise en place d'ABR et premiers algorithmes .....	21
4.1.2	Exercice 2 : Algorithmes récursifs sur arborescences .....	21
<b>4.2</b>	<b>Modification et parcours d'ABR non équilibrées</b>	<b>21</b>
4.2.1	Exercice 3 : Insertion et suppression de valeurs dans une arborescence	21
<b>4.3</b>	<b>Parcours d'arborescences binaires</b>	<b>21</b>
4.3.1	Exercice 4 : Parcours sur arbres .....	21
<b>5</b>	<b>TP5+TP6 : Hiérarchie de processus, signaux .....</b>	<b>23</b>
<b>5.1</b>	<b>Gestion des signaux : envoi et reception</b>	<b>23</b>
5.1.1	Exercice 1 : Droits et signaux .....	23
5.1.2	Exercice 2 : capture de signal et traduction en langage C .....	23
5.1.3	Exercice 3 : Capture de signaux et redirections (exercice difficile) ....	23
5.1.4	Exercice 4 : envoi multiples et capture de signal en C .....	23
<b>5.2</b>	<b>Gestion des processus</b>	<b>23</b>
5.2.1	Exercice 5 : Processus en premier-plan / Arriere-plan .....	23
5.2.2	Exercice 6 : Duplication et recouvrement de processus .....	23
<b>5.3</b>	<b>Gestion des processus - Suite</b>	<b>23</b>
5.3.1	Exercice 7 : Duplication de processus .....	23
5.3.2	Exercice 8 : Creation et destruction de processus .....	23
5.3.3	Exercice 9 : Evaluation du nombre de processus .....	23
5.3.4	Exercice 10 : Conjonctions, Disjonctions, et Duplication .....	23
5.3.5	Exercice 11 : Terminaison normale de processus .....	23
<b>6</b>	<b>TP7+TP8 : Communication socket .....</b>	<b>25</b>
<b>6.1</b>	<b>Communication distante en utilisant l'outil netcat</b>	<b>25</b>
6.1.1	Exercice 1 : Découverte de la commande nc : netcat .....	25
6.1.2	Exercice 2 : Utilisation de la commande nc : netcat pour le transfert de fichier et l'évaluation de la bande passante .....	25
6.1.3	Exercice 3 : Une histoire de serveurs concurrents ...	25
6.1.4	Exercice 4 : Comprendre une requête HTTP .....	25
<b>6.2</b>	<b>Développement d'un client et d'un serveur en C</b>	<b>25</b>
6.2.1	Exercice 5 : Mise en place d'une communication en mode non connecte	25
6.2.2	Exercice 6 : Création d'une architecture (client UDP) - (relai UDP-TCP)-(serveur TCP) .....	25

<b>6.3</b>	<b>Exercices bonus</b>	<b>25</b>
6.3.1	Exercice 7 : Résolution de noms . . . . .	25
6.3.2	Exercice 8 : Serveur multi-client en mode connecte . . . . .	25
<b>7</b>	<b>TP9+TP10 : Héritage multiple et modélisation . . . . .</b>	<b>27</b>
7.1	Exercice 1 : Organisation d'un jeu de combat au tour par tour	27





## Table des figures

1.1	Permissions Unix .....	12
-----	------------------------	----





## **1 TP1 : Manipulations de l'environnement et des fichiers sous UNIX .. 11**

- 1.1 Exercice 1 : Découverte de quelques commandes d'archivage
- 1.2 Exercice 2 : Utilisation des masques de création de fichiers
- 1.3 Exercice 3 : Manipulation du Systeme de fichier et des droits de navigation
- 1.4 Exercice 4 : Manipulation d'expression régulière

## **2 TP2 : Scripts Shell ..... 15**

- 2.1 Exercice 5 : Un premier script
- 2.2 Exercice 6 : Comptage des paramètres
- 2.3 Exercice 7 : Portée des variables





# 1. TP1 : Manipulations de l'environnement

## 1.1 Exercice 1 : Découverte de quelques commandes d'archivage

*L'objectif de cet exercice est de découvrir et manipuler les commandes de téléchargement, d'archivage, de compression et de décompression de fichier*

### 1. Récupération et décompression d'une archive

La commande `wget https://cloud.infotro.fr/ITC313/archive.tar` permet de télécharger l'archive présent à cette adresse.

- L'option `-x` permet de restaurer les fichiers contenus dans une archive.
- L'option `-c` permet de créer une nouvelle archive.
- L'option `-f` permet d'utiliser le fichier archive F ou le périphérique F (par défaut `/dev/rmt0`).

9 fichiers étaient présents dans cette archive

### 2. Manipulation de fichiers

`file *` Afin de renommer le fichier j'utilise cette commande `mv image4.jpg image4.jpg2`. Le fichier `script.txt` fait 170Ko. La commande `gzip` sur `script.txt` a compressé le fichier. Le fichier fait maintenant 65Ko. La compression est donc d'environ 38.235%. Après décompression avec la commande `gunzip` le fichier fait maintenant 170Ko, qui est la taille initiale du fichier.

### 3. Création d'une nouvelle archive

La nouvelle archive fait 850Ko soit 1Ko de moins que l'ancienne archive. Sûrement à cause du nom de fichier `jp2` changé. La somme des tailles des fichiers dans l'archive est égale à 845479 soit 845Ko, on observe une différence de 5Ko. La compression de l'archive (créé précédemment) fait 617Ko soit une différence de 228Ko. L'option `-z` utilise `gzip` pour compresser l'archive. Elle revient totalement à créer une archive puis la compresser puisque

d'après les test la taille n'est pas différente. La Commande `tar -c -z *.jpg *.txt *.jp2` devrait normalement afficher dans le terminal le résultat. La Commande `tar -c -z *.jpg *.txt *.jp2 > nouvelleArchive3.tar.gz` redirige bien le résultat dans un fichier. La redirection du flux dans un fichier recrée une archive compressé similaire à la deuxième créé. En conclusion l'archive 2 et 3 donne le même résultat et sont plus petit que l'archive 1 puisqu'elles sont compressés.

## 1.2 Exercice 2 : Utilisation des masques de création de fichiers

Humain	Base 8	Base 2
---	0	000
--X	1	001
-W-	2	010
-WX	3	011
r--	4	100
r-X	5	101
rw-	6	110
rwX	7	111

FIGURE 1.1 – Permissions Unix

1.

```
touch Raphael.txt
umask 0666
touch Donatello.txt
umask 0331
touch Michelangelo.txt
umask 0661
touch Leonardo.txt
umask 0000
```

2. et 3.

Il n'est pas possible de créer de donner plus de droit que la limitation par défaut du systeme. ( application umask par défaut 666 sur fichier et 777 sur repertoire)

## 1.3 Exercice 3 : Manipulation du Systeme de fichier et des droits de navigation

1.

L'archive contient 5 images.

3.

```
/home/ESIREM-AD/al669724/Documents/Shell/TPs/TP1/Ex3/images/Chinpokomon/P-Z/Vamp
```

4.

```
../P-Z/Vamporc.png
```

6.

L'option `-z` permet de compresser l'archive. `tar -xzf ITC313_TP_Shelllebot.axel.tar.gz` permettra de décompresser et extraire l'archive.

7.

Toutes les permissions sont conservés.

## 1.4 Exercice 4 : Manipulation d'expression régulière

2.

Les lignes contenant la suite de lettres "ette".

3.

Les lignes contenant la lettre "T".

4.

Les lignes commençant par la lettre "T".

5.

`^` signifie "début".

6.

Les lignes finissant par "te".

7.

Les lignes contenant la suite de caractère "c", un caractère, "r".

8.

Les lignes contenant "oui" ou "non".

9.

- '\$' -> en fin
- '|' -> ou
- '.' -> un caractère

10.

Permet d'afficher uniquement la partie correspondant à la recherche.

11.

Une suite de 4 lettre Majuscule.

12.

Une suite d'au moins une Majuscule et un minuscule

14

**13.**

Les mots commençant par une majuscule ainsi que les lettres majuscules.

**15.**

Permet de récupérer les adresses e-mail.

**16.**

Permet de récupérer les numéros de téléphone.

**17.**

((bien))((joue))((tu))((as))((trouve))((la))((reponse))((a))((la))((derniere))((question))

## 2. TP2 : Scripts Shell

### 2.1 Exercice 5 : Un premier script

```
#!/bin/bash
# Axel LE BOT - 2017-10-01
clear
USER=$(whoami)
echo "Hello $USER"
echo -e "\n\e[9mhello world\e[0m\n" \ "hello Kitty\n"
echo "Do you like fishsticks(y/n)?"
read answer

if [ "$answer" = "y" ]
then
    echo "then you are a gayfish!"
else
    echo "OK"
fi
```

### 2.2 Exercice 6 : Comptage des paramètres

```
#!/bin/bash
# Axel LE BOT - 2017-10-02
echo "Ex6 - paramètres"
echo -n "-Liste des parametres entres : "
COUNT=0

for i in $*
```

```
do
    echo -n "$i "
    (( COUNT++ ))
done

echo -e "\n-Nombre de parametres : $COUNT"

#!/bin/bash
# Axel LE BOT - 2017-10-02
echo "Ex6 - paramètres"
echo -n "-Liste des parametres entres : "

A=$#
for ((i=0 ; i<A ; i++)) do
    echo -n "$1 "
    shift
done

echo -e "\n-Nombre de parametres : $A"
```

## 2.3 Exercice 7 : Portée des variables

### 1. Portée des variables locales

La variable créé dans le terminal n'est pas accessible depuis un script.

### 2. Portée limitée au shell

Les variables sont local au terminal

### 3. Étendre la portée de la valeur d'une variable locale



<b>3</b>	<b>TP1+TP2 : Tableaux, matrices et Fonctions recursives .....</b>	<b>19</b>
3.1	Exercice sur des tableaux	
<b>4</b>	<b>TP3+TP4 : Manipulation des arbres</b>	<b>21</b>
4.1	Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées	
4.2	Modification et parcours d'ABR non équilibrées	
4.3	Parcours d'arborescences binaires	
<b>5</b>	<b>TP5+TP6 : Hiérarchie de processus, signaux .....</b>	<b>23</b>
5.1	Gestion des signaux : envoi et reception	
5.2	Gestion des processus	
5.3	Gestion des processus - Suite	
<b>6</b>	<b>TP7+TP8 : Communication socket</b>	<b>25</b>
6.1	Communication distante en utilisant l'outil net-cat	
6.2	Développement d'un client et d'un serveur en C	
6.3	Exercices bonus	
<b>7</b>	<b>TP9+TP10 : Héritage multiple et modélisation .....</b>	<b>27</b>
7.1	Exercice 1 : Organisation d'un jeu de combat au tour par tour	





## 3. TP1+TP2 : Tableaux, matrices et Fonctions

### 3.1 Exercice sur des tableaux

#### 3.1.1 Fonction sur les tableaux non triés

Exercice 1 : Algorithmes de parcours classiques sur tableau non triés

Quelques copier coller ont suffi. Les tests ont bien été effectués.

Exercice 2 : Ajout et suppression d'éléments tableaux non triés

#### 3.1.2 Algorithmes de tri de tableaux

Exercice 3 : Trier des tableaux aléatoires

#### 3.1.3 Fonctions sur les tableaux triés

Exercice 4 : Algorithmes de parcours classiques sur tableau non triés

Exercice 5 : Ajout et suppression d'éléments sur tableaux triés

#### 3.1.4 Exercice sur les Fonctions récursives

Exercice 6 : Définition de fonction récursive

Exercice 7 : Algorithme récursif sur matrice





## **4. TP3+TP4 : Manipulation des arbres**

### **4.1 Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées**

**4.1.1** Exercice 1 : Mise en place d'ABR et premiers algorithmes

**4.1.2** Exercice 2 : Algorithmes récursifs sur arborescences

### **4.2 Modification et parcours d'ABR non équilibrées**

**4.2.1** Exercice 3 : Insertion et suppression de valeurs dans une arborescence

### **4.3 Parcours d'arborescences binaires**

**4.3.1** Exercice 4 : Parcours sur arbres





## 5. TP5+TP6 : Hiérarchie de processus, signaux

### 5.1 Gestion des signaux : envoi et reception

- 5.1.1 Exercice 1 : Droits et signaux
- 5.1.2 Exercice 2 : capture de signal et traduction en langage C
- 5.1.3 Exercice 3 : Capture de signaux et redirections (exercice difficile)
- 5.1.4 Exercice 4 : envoi multiples et capture de signal en C

### 5.2 Gestion des processus

- 5.2.1 Exercice 5 : Processus en premier-plan / Arriere-plan
- 5.2.2 Exercice 6 : Duplication et recouvrement de processus

### 5.3 Gestion des processus - Suite

- 5.3.1 Exercice 7 : Duplication de processus
- 5.3.2 Exercice 8 : Creation et destruction de processus
- 5.3.3 Exercice 9 : Evaluation du nombre de processus
- 5.3.4 Exercice 10 : Conjonctions, Disjonctions, et Duplication
- 5.3.5 Exercice 11 : Terminaison normale de processus







## 6. TP7+TP8 : Communication socket

### 6.1 Communication distante en utilisant l'outil netcat

- 6.1.1 Exercice 1 : Découverte de la commande nc : netcat
- 6.1.2 Exercice 2 : Utilisation de la commande nc : netcat pour le transfert de fichier et l'évaluation de la bande passante
- 6.1.3 Exercice 3 : Une histoire de serveurs concurrents ...
- 6.1.4 Exercice 4 : Comprendre une requête HTTP

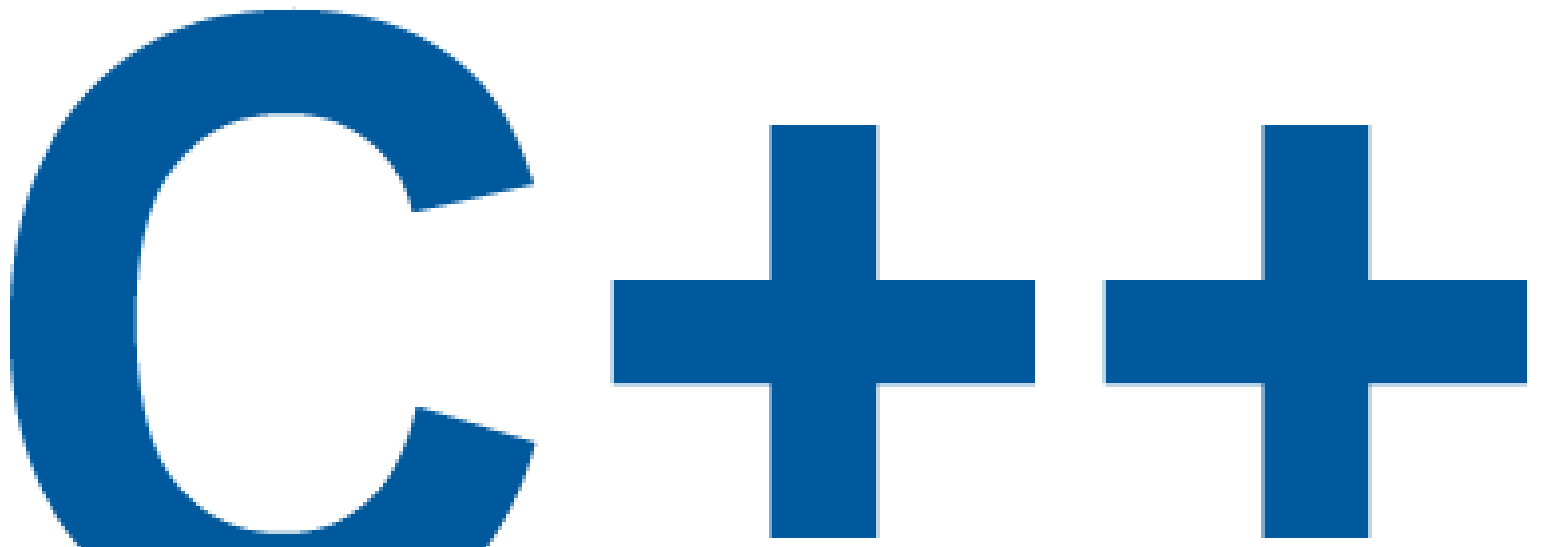
### 6.2 Développement d'un client et d'un serveur en C

- 6.2.1 Exercice 5 : Mise en place d'une communication en mode non connecte
- 6.2.2 Exercice 6 : Création d'une architecture (client UDP) - (relai UDP-TCP)- (serveur TCP)

### 6.3 Exercices bonus

- 6.3.1 Exercice 7 : Résolution de noms
- 6.3.2 Exercice 8 : Serveur multi-client en mode connecte





## 7. TP9+TP10 : Héritage multiple et modélisation

### 7.1 Exercice 1 : Organisation d'un jeu de combat au tour par tour