Informatique

Travaux Pratiques

Axel LE BOT

Table des matières

Table des figures

Shell

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 <url>` permet de télécharger un fichier présent à cette adresse. Ici nous récupérons une archive que nous pouvons manipuler avec `tar`. `tar` a trois options intéréssante :

- 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'utilise le fichier archive F ou le périphérique F (par défaut /dev/rmt0).

On découvre que l'archive télécharger contient 9 fichiers.

2. Manipulation de fichiers

- La commande `file <filename>` nous permet de savoir le format d'un fichier.
- La commande `mv <filename> <filename2>` me permet de déplacer mais aussi de renommer un "fichier1" en "fichier2". Ici j'ai utilisé la commande suivante : `mv image4.jpg image4.jpg2 .`
- La commande `ls -lh <filename>` permet d'afficher les informations plus détaillé lisible par l'humain. Ici la commande nous apprend que le fichier `script.txt` fait 170Ko.
- La commande `gzip <filename>` permet de compresser un fichier. Ici le fichier `script.txt` a été compressé. Il fait maintenant 65Ko. La compréssion est donc d'environ 38.235%.

— La commande `gunzip <filename>` permet de décompresser un fichier. Ici le fichier `script.txt` fait maintenant 170Ko, qui est bien la taille initial du fichier.

3. Création d'une nouvelle archive

— La commande `tar` permet de créer une archive sans la compresser. On peut tout de même utiliser `tar` pour archiver puis compresser des fichiers
— La commande `tar -z` permet de compréssser l'archive au format `gzip`.
— La commande `tar -cz *.jpg *.txt *.jp2` n'est pas exécutée car il est impossible d'écrire des données compressées dans le terminal. Pour sela il faut rajouter l'option `-f`.
— La commande `tar -cz *.jpg *.txt *.jp2 > nouvelleArchive3.tar.gz` redirige bien le résultat dans un fichier. En effet le symbole > permet de rediriger la sortie vers un fichier. symbole >.

La redirection du flux dans un fichier recréer une archive compréssé "archive3" similaire à "archive2" créé. En conclusion l'archive 2 et 3 donne le même résultat et sont plus petit que l'archive 1 puisqu'elles sont compréssés.

1.2 Exercice 2 : Utilisation des masques de création de fichiers

Cet exercice permet de comprendre les droits UNIX à la création de fichier à l'aide des masque utilisateur. La commande `umask` permet de définir les permissions par défaut de fichiers ou de répertoires créees. Pour sela il faut préciser les droits que l'on veut supprimer. Ci-dessous un tableau de droits UNIX pour rappel.

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.

Afin de créer des fichiers avec les droits adecquates, il faudra effectuer les commandes suivantes :

```
touch Raphael.txt
umask 0666
```

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

2. et 3.

Après plusieurs test, on découvre qu'il n'est pas possible de donner plus de droit que la limitation par défaut du systeme.

— `umask 666` sur les fichiers
— `umask 777` sur les répertoires

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

L'objectif de cet exercice est de manipuler les commandes de navigation dans l'arborescence et de création de fichier.

— La commande `mkdir <dirname>` permet de créer un répertoire.

1.

L'archive contient 5 images.

3.

Le chemin absolue d'un fichier ou d'un repertoire correspond au chemin vers le fichier ou repertoire depuis la racine. Par exemple : `/home/axel/Documents/Shell/TPs/TP1/Ex3/images/Chin`

4.

Le chemin relatif d'un fichier ou d'un repertoire correspond au chemin vers le fichier ou repertoire par rapport à un autre repertoire. Par exemple : `../P-Z/Vamporc.png`

6.

La commande `tar -xczf ITC313_TP_Shell_lebot.axel.tar.gz` permettra de décompresser et extraire l'archive.

7.

Après transfert de l'archive compressé toutes les permissions sont conservés.

1.4 Exercice 4 : Manipulation d'expression régulière

L'objectif de cet exercice est de manipuler les basiques des expressions régulière.

1.

La commande `wget https://cloud.infotro.fr/ITC313/unGrosBordel.txt` permettra de récupérer le fichier à analyser.

2.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "ette"` contiennent toutes la suite de lettres "ette".

3.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "T"` contiennent toutes la lettre "T".

4.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "^T"` contiennent toutes la lettre "T" en début de ligne.

5.

L'expression ^ signifie donc "début".

6.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "te$"` contiennent toutes la suite de lettres "te" en fin de ligne.

7.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "c.r"` contiennent toutes la suite de lettres "c", un caractère quelconque, "r".

8.

Les lignes affichées par `cat unGrosBordel.txt | grep -E "(oui|non)"` contiennent toutes soit "oui", soit "non".

9.

— $ représante la fin d'une ligne.
— | représante une condition "ou".
— . représante un caractère quelconque.

10.

L'option `-o` dans la commande `cat unGrosBordel.txt | grep -o -E "c.r"` permet de n'afficher que la partie de la ligne correspondant à l'expression "c.r" appelé "motif".

11.

Les motifs affichées par `cat unGrosBordel.txt | grep -o -E "[A-Z]"` contiennent une suite de 4 lettres majuscule.

12.

Les motifs affichées par `cat unGrosBordel.txt | grep -o -E "[A-Z][a-z]+"` contiennent une suite d'au moins une majuscule et une minuscule.

13.

Les motifs affichées par `cat unGrosBordel.txt | grep -o -E "[A-Z][a-z]*"` contiennent une suite de 0, 1 ou plus de une majuscule et une minuscule.

14.

— + représante 1 ou plusieurs.
— * représante 0, 1 ou plusieurs.

15.

La commande

```
cat unGrosBordel.txt | grep -o -E "[A-Za-z0-9\.\_]+@([A-Za-z0-9\-]+\.)*[a-zA-Z]{2,4
```

permet de récupérer les addresse e-mail.

16.

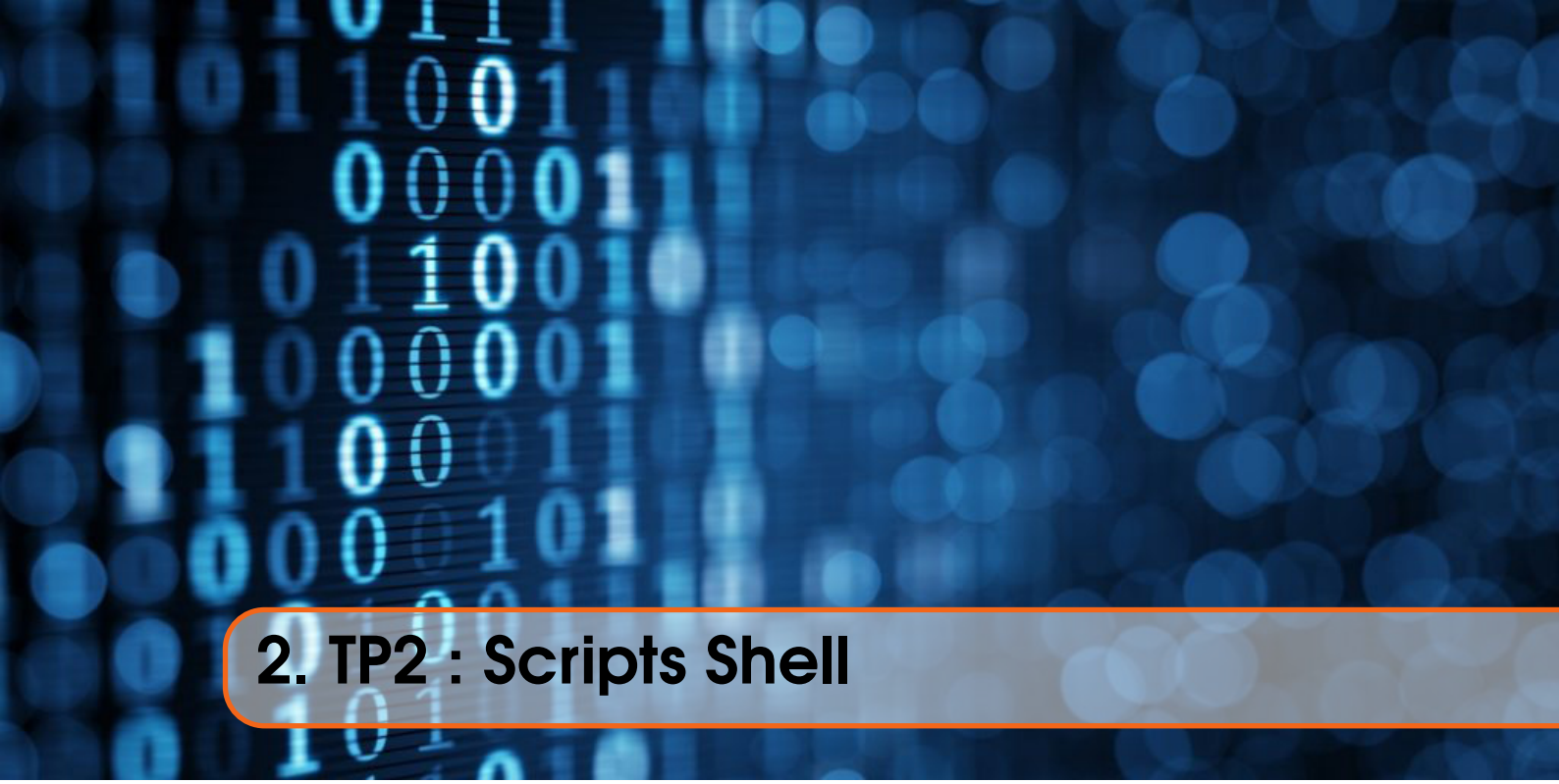
La commande

```
cat unGrosBordel.txt | grep -o -E "(\+33|0)(\.| )?[0-9]((\.| )?[0-9]{2}){4}"
```

permet de récupérer les numéro de téléphone.

17.

La commande `cat unGrosBordel.txt | grep -o -E "\(\([a-z]+\)\)"` permet de trouver la phrase secrète : "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 "\"\e[9mhello world\e[0m\" \"hello Kitty\""
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

II

C++

3. TP1+TP2 : Tableaux, matrices et Fonctio

3.1 Exercice sur des tableaux

3.1.1 Fonction sur les tableaux non triés

Exercice 1 : Algorithmes de parcours classiaues sur tableau non triés

Quelques copier coller ont suffit. Les tests ont bien été éffectué.

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 : Definition de fonction recursive

Exercice 7 : Algorithme recursif sur matrice

4. TP3+TP4 : Manipulation des arbres

5. TP5+TP6 : Hiérarchie de processus, signa

6. TP7+TP8 : Communication socket

7. TP9+TP10 : Héritage multiple et modélisa

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