AGUILAR_Flavien

SH-File-System

i1-tp-file_system : Manipulation du système de fichiers sous Linux

1. Répertoire personnel de l'utilisateur :

1- Pour afficher la valeur du répertoire utilisateur, nous allons afficher la variable d'environement $HOME :

```
test@debian:~$ echo $HOME
/home/test
```

2- Pour afficher la valeur du répertoire courant, nous avons plusieurs possibilitées, soit utilisé la commande

```
test@debian:~/coucou$ pwd
/home/test/coucou
```

soit afficher la variable d'environnement $PWD :

```
test@debian:~/coucou$ echo $PWD
/home/test/coucou
```

3- Après l'ouverture d'une session, le répertoire courant sera le répertoire utilisateur de l'utilisateur ayant initialisé la session.

4- Pour nous déplacer vers la racine en une seule commande, nous allons utiliser la commade :

```
cd /
```

Nous pouvons vérifier que nous sommes bien à la racine avec la commande :

```
test@debian:/$ pwd
/
```

5- Pour afficher les fichiers sous forme de liste les fichiers du répetoire courant, nous allons utiliser l'option -l de ls :

```
test@debian:/$ ls -l
total 57
lrwxrwxrwx  1 root root     7 oct.   1 14:22 bin -> usr/bin
drwxr-xr-x  4 root root  1024 oct.   1 14:37 boot
drwxr-xr-x 18 root root  3200 oct.   1 16:34 dev
drwxr-xr-x 77 root root  4096 oct.   1 16:34 etc
drwxr-xr-x  3 root root  4096 oct.   1 14:37 home
lrwxrwxrwx  1 root root    31 oct.   1 14:24 initrd.img -> boot/initrd.
   img-4.19.0-11-amd64
lrwxrwxrwx  1 root root    31 oct.   1 14:24 initrd.img.old -> boot/
   initrd.img-4.19.0-11-amd64
lrwxrwxrwx  1 root root     7 oct.   1 14:22 lib -> usr/lib
lrwxrwxrwx  1 root root     9 oct.   1 14:22 lib32 -> usr/lib32
lrwxrwxrwx  1 root root     9 oct.   1 14:22 lib64 -> usr/lib64
lrwxrwxrwx  1 root root    10 oct.   1 14:22 libx32 -> usr/libx32
drwx------  2 root root 16384 oct.   1 14:22 lost+found
drwxr-xr-x  3 root root  4096 oct.   1 14:22 media
drwxr-xr-x  2 root root  4096 oct.   1 14:22 mnt
drwxr-xr-x  2 root root  4096 oct.   1 14:22 opt
dr-xr-xr-x 82 root root     0 oct.   1 16:34 proc
drwx------  3 root root  4096 oct.   1 16:34 root
drwxr-xr-x 17 root root   560 oct.   1 16:35 run
lrwxrwxrwx  1 root root     8 oct.   1 14:22 sbin -> usr/sbin
drwxr-xr-x  2 root root  4096 oct.   1 14:22 srv
dr-xr-xr-x 13 root root     0 oct.   1 16:49 sys
drwxrwxrwt  8 root root  4096 oct.   1 16:49 tmp
drwxr-xr-x 13 root root  4096 oct.   1 14:22 usr
drwxr-xr-x 11 root root  4096 oct.   1 14:22 var
lrwxrwxrwx  1 root root    28 oct.   1 14:24 vmlinuz -> boot/vmlinuz
   -4.19.0-11-amd64
lrwxrwxrwx  1 root root    28 oct.   1 14:24 vmlinuz.old -> boot/
   vmlinuz-4.19.0-11-amd64
```

2. Editeurs de textes :

1- Pour retourner au répertoire utilsateur en une seule commande, nous allons utiliser la commande :

```
test@debian:/$ cd ~
```

Nous pouvons vérifier que nous sommes au bon endroit avec la commande pwd :

```
test@debian:~$ pwd
/home/test
```

2 - Pour rentrer un mot dans un fichier, nous allons utiliser la commande :

```
test@debian:~$ vim monfichier
```

Cette commande va créer un fichier nommé monfichier et rentré en édition à l'intérieur du fichier.

Nous tapons notre texte et nous enregistrons en tapant :

```
:wq
```

3 / 4- Dans mon fichier, j'ai tapé 7 caractères, et lorsque je regarde la taille de mon fichier, je remarque qu'il fait 8o, j'en déduit que le fichier en lui même pèse 1 octer et que chaque caractères correspond à 1 octet.

5- Pour afficher le contenu de mon fichier dans la console, je vais utilser la commande :

```
test@debian:~$ cat monfichier
bonjour
```

3. Création d'une arborescence donnée :

1 - Nous nous déplaçons dans notre répertoire utilsateur avec la commande :

```
test@debian:/$ cd ~
```

2 - Nous affichons le contenu du répertoire utilisateur :

```
test@debian:/$ ls ~
coucou  monfichier
```

3 - Pour effacer en une seule commande tous les fichiers à l'intérieur du répertoire utilisateur, nous allons utiliser la commande :

```
test@debian:~$ rm -rf ~/*
```

4 - Nous pouvons vérifier en listant le contenu du répertoire utilisateur :

```
test@debian:~$ ls
```

5 - Pour créer l'arborescence demandée, nous allons utiliser les commandes suivantes depuis le répertoire personnel de l'utilisateur root afin d'avoir les droits en écriture sur l'ensemble des dossiers :

```
root@debian:/home/test# mkdir /home/rt1/
root@debian:/home/test# mkdir /home/rt1/xyz/
root@debian:/home/test# mkdir /home/rt1/xyz/public_html/
root@debian:/home/test# mkdir /home/rt1/xyz/public_html/docs/
root@debian:/home/test# mkdir /home/rt1/xyz/public_html/images/
root@debian:/home/test# touch /home/rt1/xyz/public_html/index.html
root@debian:/home/test# mkdir /home/rt1/xyz/programmes
root@debian:/home/test# mkdir /home/rt1/xyz/programmes/langage_c
root@debian:/home/test# touch /home/rt1/xyz/programmes/langage_c/a.c
root@debian:/home/test# mkdir /home/rt1/xyz/programmes/php
root@debian:/home/test# mkdir /home/rt1/xyz/programmes/java
root@debian:/home/test# touch /home/rt1/xyz/programmes/java/TP.java
```

6 - Nous allons rentrer du texte dans le fichier index.html avec la commande suivante :

```
root@debian:/home/test# vim /home/rt1/xyz/public_html/index.html
```

7 - Nous allons afficher le contenu du fichier avec la commande :

```
root@debian:/home/test# cat /home/rt1/xyz/public_html/index.html
<h1>Bonjour !!!</h1>
```

8 - Pour copier le contenu du répertoire programmes dans le dossier images, nous allons utiliser la commande suivante :

```
root@debian:/home# cp -r ./rt1/xyz/programmes/ ./rt1/xyz/public_html/
   images/
```

9 - Pour déplacer l'intégralité du répertoire programmes dans le dossier docs, nous allons utiliser la commande suivante :

```
root@debian:/home# mv ./rt1/xyz/programmes/ ./rt1/xyz/public_html/docs/
```

10 - Pour afficher le contenu détaillé du répertoire docs, nous allons utiliser la commande :

```
root@debian:/home# ls -R ./rt1/xyz/public_html/docs/
./rt1/xyz/public_html/docs/:
programmes

./rt1/xyz/public_html/docs/programmes:
java  langage_c  php

./rt1/xyz/public_html/docs/programmes/java:
TP.java

./rt1/xyz/public_html/docs/programmes/langage_c:
a.c

./rt1/xyz/public_html/docs/programmes/php:
```

---

i1-tp-script_1 : Script

1 Affichage:

1. 1 Utiliser la commande echo dans un script pour afficher «bonjour»

1 - Pour afficher bonjour « retour_a_la_ligne » tout le monde, nous allons utiliser la commande :

```
test@debian:~$ echo -e "Bonjour \ntout le monde"
Bonjour
tout le monde
```

2 - Pour afficher le même texte qu'à la question 1 le tout en 3 commandes et à l'aide d'un script, nous allons utiliser les commandes suivantes :

```
test@debian:~$ vim monscript.sh
test@debian:~$ chmod +x monscript.sh
```

```
test@debian:~$ ./monscript.sh
Bonjour
tout le monde
```

Le script « monscript.sh » contient les lignes suivantes :

```
#!/bin/bash
echo -e "Bonjour \n tout le monde"
```

3 - Pour afficher « Bonjour tout le monde » à l'aide d'un script et de deux commandes, nous allons utiliser les commandes :

```
test@debian:~$ vim monscript.sh
test@debian:~$ ./monscript.sh
Bonjour tout le monde
```

Le script « monscript.sh » contiendra alors les lignes :

```
#!/bin/bash
echo "Bonjour tout le monde"
```

2 Variables :

1 - Grace à la commande set, j'ai pu définir les deux variables correspondant au nom de l'utilisateur et au nom de la machine :

```
USER=test    -> Nom d utilisateur
HOSTNAME=debian
```

2 - Le script permettant d'afficher « je suis l'utilisateur x sur la machine y » est :

```
#!/bin/bash
echo "Je suis l'utilisateur $USER de la machine $HOSTNAME"
```

Nous pouvons vérifier son bon fonctionnement :

```
test@debian:~$ ./monscript.sh
Je suis l'utilisateur test de la machine debian
```

3 - Pour définir des variables et afficher « l'utilisateur x s'appelle y z », nous allons utiliser les commandes suivantes :

```
test@debian:~$ export nom=AGUILAR
test@debian:~$ export prenom=Flavien
test@debian:~$ echo "L'utilisateur $USER s'appelle $nom $prenom"
L'utilisateur test s'appelle AGUILAR Flavien
```

4 - Pour affecter la commande ls à une variable, nous allons utiliser les commades suivantes :

```
test@debian:~$ export list=$(ls)
test@debian:~$ echo $list
monscript.sh
```

5 - Pour demander à l'utilisateur de saisir son nom et son prénom afin de l'afficher, nous allons utiliser le script suivant :

```
#!/bin/bash

read -p "Veuillez saisir votre nom : " nom
read -p "Veuillez saisir votre prénom : "prenom

echo "Vous vous appellez $nom $prenom"
```

```
test@debian:~$ ./monscript.sh
Veuillez saisir votre nom : aguilar
Veuillez saisir votre nom : flavien
Vous vous appellez aguilar flavien
```

3 Analyse de la ligne de commande

1 - Pour afficher l'ensemble de la ligne de commande d'un script, nous allons utiliser la variable $* :

```
#!/bin/bash
echo $*
```

```
test@debian:~$ ./monscript.sh 1 2 3 1 2 1 3
1 2 3 1 2 1 3
```

2 - Pour afficher seulement la commande, nous allons utiliser la variable $0 :

```
#!/bin/bash
echo $0
```

```
test@debian:~$ ./monscript.sh 1 2 3 1 2 1 3
./monscript.sh
```

3 - Pour créer un groupe dont le nom sera donné en argument et dont nous allons affecter tous les droits à « others », nous allons utiliser le script suivant :

```
#!/bin/bash
mkdir $1
chmod 007 $1
```

4 - La variable permettant d'afficher le statut de l'exécution d'une commande est : $? , nous allons modifier notre script afin d'afficher ce retour entre chaque commande :

```
#!/bin/bash
mkdir $1
echo $?
chmod 007 $1
echo $?
```

```
test@debian:~$ ./monscript.sh test
0
0
```

5 - Pour créer un répertoire dans un répertoire ou nous avons les droits, nous allons utiliser le script suviant :

```
#!/bin/bash
mkdir ~/$1
```

6 - Pour créer un répertoire dans un répertoire ou nous n'avons pas les droits nous allons utiliser le script :

```
#!/bin/bash
mkdir /root/flavien
```

7 - Pour créer une arborescence à 2 niveaux, nous allons utiliser le script :

```
#!/bin/bash
mkdir niveau1
mkdir niveau1/niveau2/
```

8 - Lorsque nous voulons écrire dans un répertoire ou nous avons les droits, le retour de la variable $? sera 0, ce qui indique que la commande a bien pu s'exécuter sans erreur, en revanche,lorsque nous avons pas les droits sur le répertoire, la variable $? nous retourne un code d'erreur qui nous indique que la commande n'a pas pu s'exécuter normalement.

4 Utilisation du shebang

1 - Le shebang va permettre de communiquer au système que le fichier n'est pas un binaire mais un script. Il précise aussi avec quel interpréteur nous allons exécuter le script.

2 - Pour exéctuter un script à l'aide du shebang, nous allons devoir disposer des droits d'exécutions sur le fichier.

3 - La ligne à rajouter tout en hautd d'un script pour le rendre executable est :

```
#!/bin/bash
```

4 - Pour inclure le dossier bin situé dans notre répertoire personnel à la variable $PATH, nous allons utiliser la commande :

```
test@debian:/$ export PATH=$PATH:/home/test/bin/
```

5 - Cette méthode va nous permettre d'exécuter notre script depuis n'importe quel répertoire.