

Corrections des entraînements

1. Chapitre Gestion des droits utilisateurs

Réponses

- Vérifiez l'endroit où vous êtes, votre identité de compte avec notamment l'UID et le GID par la commande `id`.

```
pwd
```

```
id
```

- Quelles sont les permissions en octal de votre répertoire personnel ? et celles du répertoire parent ?

```
755 (drwxr-xr-x) visible par ls -al /root
```

même chose

- À quel groupe appartenez-vous ?

```
root (le GID)
```

- Créez un fichier par la commande `touch liste.sh`.

Pas de correction nécessaire...

- À qui appartient-il ? À quel groupe ?

```
au root, groupe root (ls -l)
```

- Quels sont les droits de ce fichier en octal ?

```
644 (rw-r--r--)
```

- Ajoutez le droit en écriture pour le groupe par la méthode octale.

```
chmod 664 liste.sh
```

- Enlevez le droit en écriture pour le groupe par la méthode symbolique.

```
chmod g-w liste.sh
```

- Le fichier `liste.sh` contiendra des commandes shell. Aussi, donnez-lui les permissions que l'on donne généralement à un programme, c'est-à-dire 755 et vérifiez-le.

```
chmod 755 liste.sh ou chmod ugo+x liste.sh
```

2. Chapitre Session de travail en mode console

- Tapez la commande permettant de voir dans quel répertoire vous êtes.

```
pwd (dans /root)
```

- Listez normalement le contenu de votre répertoire.

```
ls
```

- Listez le contenu du répertoire avec les fichiers cachés et le format long.

```
ls -al
```

- Utilisez le manuel en ligne pour connaître et comprendre la commande `sleep`.

```
man sleep
```

- Tapez maintenant `sleep --help`, quelle en est l'utilité ?

donne une aide plus concise

- Si vous tapez `sleep 360`, quelle combinaison de touches permet de l'interrompre ? Testez-la.

[Ctrl] c

- Quelle est la taille du répertoire `/usr` ?

```
du -hs /usr
```

- Allez dans le répertoire `/etc` par le chemin absolu.

```
cd /etc
```

- Faites un tube avec d'un côté la commande `ls *.conf`, de l'autre `wc -l`, quel en est le résultat ?

```
ls *.conf | wc -l (donne 22 fichiers sur mon serveur)
```

- Tapez tout simplement `cd` et validez. Que s'est-il passé ? Et que signifie le tilde (`~`) ?

L'utilisateur se retrouve dans son répertoire, le tilde indique son espace personnel.

- Allez dans votre répertoire et créez deux sous-répertoires : `hobbit` et `elfe`, vérifiez la création par la commande appropriée.

```
mkdir hobbit elfe
```

- Allez dans `hobbit` et créez trois fichiers `merry`, `pipin` et `gollum` par la commande `touch`.

```
cd hobbit
```

```
touch merry pipin gollum
```

- Sans changer de répertoire créez le sous-répertoire `elfedesbois` dans `elfe` et par le chemin relatif.

```
mkdir ../elfe/elfedesbois
```

- Sans changer de répertoire, créez le fichier `legolas` dans votre répertoire personnel et par le chemin absolu.

```
touch /root/legolas
```

- Allez dans `elfedesbois` obligatoirement en trois commandes (niveau par niveau).

```
cd ..
```

```
cd elfe/
```

```
cd elfedesbois/
```

- Sans changer de répertoire, déplacez `legolas` pour le mettre là où vous êtes et par le chemin relatif.

```
mv ../../legolas .
```

- Sans changer de répertoire renommez `gollum` en `sam` par le chemin absolu.

```
mv /root/hobbit/gollum /root/hobbit/sam
```

- Trouvez, par la commande `find`, le fichier `interfaces` à partir de `/etc`.

```
find /etc -name interfaces
```

- Retournez dans votre répertoire et créez le fichier `riri.txt`.

```
cd
```

```
touch riri.txt
```

- Dupliquez-le par la commande `cp` en changeant son nom en `fifi.txt`.

```
cp riri.txt fifi.txt
```

- Créez un lien par `ln` en changeant son nom en `loulou.txt`.

```
ln riri.txt loulou.txt
```

- Éditez les inodes par `ls` et l'option adéquate et concluez.

```
ls -li et les inodes de riri.txt et loulou.txt sont identiques.
```

- Par `VIM`, mettez une phrase dans `riri.txt`.

Pas de correction nécessaire...

- Éditez `fifi.txt` et `loulou.txt` ; concluez.

Le contenu est le même car ce sont en fait les mêmes fichiers.

- Effacez `riri.txt`, que s'est-il passé pour `fifi.txt` ?

Rien pour `fifi.txt` mais le nombre de lien de `loulou.txt` est passé de 2 à 1.

- Quel est le type du fichier `/etc/passwd` ?

```
file /etc/passwd ce qui donne le type ASCII text.
```

3. Chapitre Maintenance avancée du système

Exercices - Partie 1

- Affichez les lignes contenant la chaîne `ar`.

```
grep -n 'ar' test1.txt (1, 3, 4, 5, 6, 7, 9, 10)
```

- Affichez les lignes dont au moins une chaîne contient au minimum 3 j consécutifs.

```
grep -n '[j]\{3,\}' test1.txt (15)
```

- Affichez les lignes ayant au moins une chaîne se terminant par age.

```
grep -n 'age\>' test1.txt (11, 12, 13, 14, 15)
```

- Affichez les lignes ayant au moins une chaîne se terminant par guage ou gage.

```
grep -n 'g[u]*age\>' test1.txt (11, 12, 13, 14, 15)
```

- Affichez les lignes (pas les mots) se terminant par age ou ava.

```
grep -n 'age$\|ava$' test1.txt (11, 12, 13, 14)
```

- Affichez les lignes vides.

```
grep -n '^$' test1.txt (2, attention aux caractères invisibles)
```

- Affichez les lignes contenant les sous-chaînes 35 ET ans (vous ferez un "tube").

```
grep -n 'age$\|ava$' test1.txt (11, 12, 13, 14)
```

- Affichez les lignes ne commençant pas par un chiffre.

```
grep -n "^[^0-9]" test1.txt (toutes sauf la 2, 16, 17, 18, 22)
```

- Affichez les lignes ne contenant que des chiffres.

```
grep -n '^[0-9]$' test1.txt (aucune, l'espace est  
considéré comme un caractère)
```

- Affichez les lignes commençant par une voyelle ou un chiffre.

```
grep -n '^[aeiouy]\|^[0-9]' test1.txt (7, 16, 17, 18, 22)
```

- Affichez les lignes ne se terminant pas par un chiffre.

```
grep -n '[^0-9]$' test1.txt (toutes sauf la 2, 6, 16, 18, 22)
```

- Affichez les lignes faisant moins de 10 caractères.

```
grep -n '^.\{1,10\}$' test1.txt (1, 4, 5, 7, 8, 9, 10,  
11, 12, 13, 14, 18, 22)
```

Exercices - Partie 2

- Affichez à l'écran par cat le contenu du fichier.

```
cat test2.txt
```

- Quel est le nombre de mots, de lignes et de caractères de test2.txt ?

```
wc -wlc test2.txt
```

- Combien de lignes contient ce fichier ?

```
wc -l test2.txt
```

- Affichez à l'écran le deuxième champ de toutes les lignes.

```
cut -f2 -d: test2.txt
```

- Quel est le nombre de canards ?

```
grep 'canard' test2.txt | wc -l
```

- Envoyez le fichier `test2.txt` avec uniquement le premier champs de toutes les lignes dans un fichier `noms.txt`.

```
cut -f1 -d: test2.txt > noms.txt
```

- Triez le fichier `noms.txt` par ordre alphabétique.

```
sort -o noms.txt noms.txt
```

- Coupez le fichier `test2.txt` en **deux fichiers** `test2aa` et `test2ab` (voir l'option nécessaire dans le manuel en ligne).

```
split -l 4 test2.txt test2
```

- Réunissez-les dans un fichier `disney.txt`.

```
paste test2aa test2ab > disney.txt
```