



Systeme de fichiers

Table des matières

1. ATELIER 1 : Ajout et préparation d'un disque	1
1.1. Exercice 1.1 : Ajouter un disque	1
1.2. Exercice 1.2 : Préparer le nouveau disque	1
1.3. Exercice 1.3 : Formater les nouvelles partitions	2
1.4. Exercice 1.4 : Monter les disques dans l'arborescence	2
2. ATELIER 2 : Les liens	4
2.1. Exercice 2.1 : Lien physique	4
2.2. Exercice 2.2 : Lien symbolique en absolu	6
2.3. Exercice 2.3 : Lien symbolique en relatif	6
3. ATELIER 4 : Droits particuliers	8
3.1. Exercice 4.1 : Création d'une boîte aux lettres	8
3.2. Exercice 4.2 : Droits d'endossement	8
4. ATELIER 5 : Sécuriser les fichiers	10
4.1. Exercice 5.1 : Evaluer ses droits sur un répertoire	10
4.2. Exercice 5.2 : Interdire la modification du contenu d'un répertoire	10
4.3. Exercice 5.3 : Interdire l'accès à un répertoire	11
4.4. Exercice 5.4 : Autoriser les droits en lecture seule sur un fichier	12
4.5. Exercice 5.5 : Droits d'exécution sur un fichier exécutable	13
4.6. Exercice 5.6 : Droits d'exécution sur un script	14

Chapitre 1. ATELIER 1 : Ajout et préparation d'un disque

Objectifs

- utiliser la console VirtualBox pour ajouter un disque dur à un serveur ;
- partitionner avec l'utilitaire cfdisk le nouveau disque dur ;
- formater les partitions avec l'utilitaire mkfs ;
- monter les partitions avec l'utilitaire mount.

1.1. Exercice 1.1 : Ajouter un disque

- Ajouter un disque de 10 Go à votre serveur virtuel.

1.2. Exercice 1.2 : Préparer le nouveau disque

Démarrer à nouveau votre machine virtuelle et connectez-vous avec l'utilisateur root.

- Visualiser les disques accessibles :

```
[root]# fdisk -l
```

- Créer 6 partitions: 4 de 2 Go et 2 de 1 Go.

```
[root]# cfdisk /dev/sdc
```

Créer une nouvelle partition :

Choisir l'option [Primaire] :

Donner une taille à la partition :

Répéter l'opération 3 fois :

Faire un premier lecteur logique :

Répéter l'opération 3 fois :

Penser à écrire la nouvelle table de partition :

1.3. Exercice 1.3 : Formater les nouvelles partitions

- Formater les nouvelles partitions en ext4.

```
[root]# mkfs -t ext4 /dev/sdc1
[root]# mkfs -t ext4 /dev/sdc2
[root]# mkfs -t ext4 /dev/sdc3
[root]# mkfs -t ext4 /dev/sdc5
[root]# mkfs -t ext4 /dev/sdc6
[root]# mkfs -t ext4 /dev/sdc7

[root]# ls -l /dev/sdc*
```

1.4. Exercice 1.4 : Monter les disques dans l'arborescence

- Monter les volumes sur /Disque[1-6].

Création des points de montage:

```
[root]# mkdir /Disque1 /Disque2 /Disque3 /Disque4 /Disque5 /Disque6
```

Montage des partitions:

```
[root]# mount /dev/sdc1 /Disque1
[root]# mount /dev/sdc2 /Disque2
[root]# mount /dev/sdc3 /Disque3
[root]# mount /dev/sdc5 /Disque4
[root]# mount /dev/sdc6 /Disque5
[root]# mount /dev/sdc7 /Disque6
```

Visualisation des systèmes de fichiers:

```
[root]# mount
...
/dev/sdc1 on /Disque1 type ext4 (rw)
...
```

- Faire en sorte qu'à chaque redémarrage, le disque ajouté (ses partitions) soit pris en compte.

```
[root]# vim /etc/fstab
/dev/sdc1    /Disque1    ext4    defaults    0 0
...
/dev/sdc7    /Disque6    ext4    defaults    0 0
```

- Rediriger le contenu du fichier de configuration des montages et la table associée dans le fichier « Resultat » que vous placerez sur la première partition du disque ajouté dans un répertoire « SuiviInstallation ».

```
[root]# cd /Disque1
[root]# mkdir SuiviInstallation
[root]# cat /etc/fstab /etc/mtab > SuiviInstallation/Resultat
```

Chapitre 2. ATELIER 2 : Les liens

Objectifs

- gérer les liens physiques et symboliques.

Pré-requis

- Créer le répertoire /Disque1/programmeTux.

```
[root]# mkdir /Disque1/programmeTux
```

- Changer le propriétaire de ce dossier pour stagiaire.

```
[root]# chown stagiaire /Disque1/programmeTux
```

2.1. Exercice 2.1 : Lien physique

- Se connecter sur une console avec l'utilisateur stagiaire.
- Créer l'arborescence src/version1 dans votre répertoire de connexion.

```
[stagiaire]$ mkdir -p src/version1
```

- Créer un fichier tux.exe dans src/version1. Il contiendra le texte « version 1 »

```
[stagiaire]$ echo "version 1" > src/version1/tux.exe
```

- Créer un lien physique tux-v1.exe dans le répertoire src vers le fichier src/version1/tux.exe.

```
[stagiaire]$ ln src/version1/tux.exe src/tux-v1.exe
```

- Que pouvons-nous remarquer à propos des propriétés du lien et de sa source ?

Les deux fichiers ont le même numéro d'inode.

```
[stagiaire]$ ls -li src/version1/tux.exe src/tux-v1.exe
263883 -rw-r--r-- 2 stagiaire users 0 fev 11 14:37 src/tux-v1.exe
263883 -rw-r--r-- 2 stagiaire users 0 fev 11 14:37 src/version1/tux.exe
```

- Détruire le fichier `src/version1/tux.exe`. Que se passe-t-il pour le lien `src/tux-v1.exe` ?

Le compteur revient à 1.

```
[stagiaire]$ rm -f src/version1/tux.exe
[stagiaire]$ ls -li src/tux-v1.exe
263883 -rw-r--r-- 1 stagiaire users 0 fev 11 14:37 src/tux-v1.exe
```

- Recréer `tux.exe` dans `src/version1/`.

```
[stagiaire]$ touch src/version1/tux.exe
```

- Le lien physique est-il toujours un lien vers `src/version1/tux.exe` ? Vérifier le contenu du fichier `src/version1/tux.exe`.

```
[stagiaire]$ ls -li src/version1/tux.exe src/tux-v1.exe
263883 -rw-r--r-- 1 stagiaire users 10 fev 11 14:37 src/tux-v1.exe
263884 -rw-r--r-- 1 stagiaire users 0 fev 11 14:38 src/version1/tux.exe
```

Non car les numéros d'inode sont différents et le compteur vaut 1.

```
[stagiaire]$ cat src/version1/tux.exe
[stagiaire]$ cat src/tux-v1.exe
version1
```

Le fichier source est maintenant vide.

- Créer un lien physique latest vers `src/tux-v1.exe` dans le répertoire « `/Disque1/programmeTux` ».

```
[stagiaire]$ ln src/tux-v1.exe /Disque1/programmeTux/latest
```

- Que se passe-t-il ? Pourquoi ?

Impossibilité de créer le lien physique car la source et le lien sont sur deux systèmes de fichiers différents.

2.2. Exercice 2.2 : Lien symbolique en absolu

- Copier le dossier src/version1 vers src/version2. Le fichier src/version2/tux.exe contiendra le texte « version 2 »

```
[stagiaire]$ cp -R src/version1 src/version2  
[stagiaire]$ echo "version 2" > src/version2/tux.exe
```

- Créer un lien symbolique latest dans le répertoire « /Disque1/programmeTux/ » vers src/version2/ en utilisant un chemin absolu.

```
[stagiaire]$ ln -s /home/stagiaire/src/version2/ /Disque1/programmeTux/latest
```

- Afficher le contenu du fichier /Disque1/programmeTux/latest/tux.exe

```
[stagiaire]$ less /Disque1/programmeTux/latest/tux.exe  
version 2
```

- Renommer le dossier src en sources.

```
[stagiaire]$ mv src sources
```

- Le lien fonctionne-t-il toujours ?

```
[stagiaire]$ less /Disque1/programmeTux/latest/tux.exe  
/Disque1/programmeTux/latest/tux.exe: Aucun fichier ou dossier de ce type  
[stagiaire]$ ll /Disque1/programmeTux/latest  
lrwxrwxrwx 1 stagiaire users 29 fev 11 16:20 /Disque1/programmeTux/latest ->  
/home/stagiaire/src/version2/
```

Non : le lien n'est plus actif.

- Pourquoi ?

Le chemin est absolu et la source a changé de place.

2.3. Exercice 2.3 : Lien symbolique en relatif

- Dans le répertoire sources, créer un lien symbolique latest.exe vers sources/version2/tux.exe en utilisant un chemin relatif.


```
[stagiaire]$ cd sources  
[stagiaire]$ ln -s ../version2/tux.exe latest.exe
```

- Déplacer (ne pas copier) le répertoire sources et son contenu dans « /Disque1/programmeTux ».

```
[stagiaire]$ cd ..  
[stagiaire]$ mv sources /Disque1/programmeTux/
```

- Le lien /Disque1/programmeTux/latest.exe fonctionne-t-il toujours ?

```
[stagiaire]$ ls -l /Disque1/programmeTux/sources/latest.exe  
lrwxrwxrwx 1 stagiaire users 24 fev 11 16:20 /Disque1/programmeTux/sources/latest.exe  
-> ../version2/tux.exe
```

Oui : le lien est actif.

Chapitre 3. ATELIER 4 : Droits particuliers

Objectifs

- positionner des droits particuliers ;
- positionner le suid.

3.1. Exercice 4.1 : Création d'une boîte aux lettres

- Créer l'arborescence `/home/BAL/resultats/`.

```
[root]# mkdir -p /home/BAL/resultats
```

- Faire en sorte que les utilisateurs puissent uniquement déposer des fichiers dans « resultats » en ne pouvant que passer dans BAL .

```
[root]# chmod 711 /home/BAL
[root]# cd /home/BAL
[root]# chmod 733 resultats
```

- Lister les droits de « BAL » ainsi que ceux de ses sous répertoires.

```
[root]# cd ..
[root]# ls -ld BAL
drwx--x--x 5 root root 1024 fev 11 08:21 BAL

[root]# cd BAL
[root]# ls -l
drwx-wx-wx 5 root root 1024 fev 11 08:21 resultats
```

3.2. Exercice 4.2 : Droits d'endossement

La commande « `chfn` » permet de formater le champ commentaire des comptes utilisateurs. Cette commande écrit dans le fichier « `/etc/passwd` ».

- Visualiser les droits de la commande et ceux du fichier.

```
[root]# ls -l /usr/bin/chfn
-rwx--x--x 1 root root 16464 oct 16 2007 /usr/bin/chfn

[root]# ls -l /etc/passwd
-rw-r--r-- 1 root root 1638 fev 5 13:23 /etc/passwd
```

- Faire en sorte que les utilisateurs puissent utiliser cette commande et ainsi changer le commentaire les concernant.

```
[root]# chmod u+s /usr/bin/chfn
```

- Listez à nouveau les droits de la commande et ceux du fichier.

```
[root]# ls -l /usr/bin/chfn
-rws--x--x 1 root root 16464 oct 16 2007 /usr/bin/chfn

[root]# ls -l /etc/passwd
-rw-r--r-- 1 root root 1638 fev 5 13:23 /etc/passwd
```

Chapitre 4. ATELIER 5 : Sécuriser les fichiers

Objectifs

- évaluer les droits sur un fichier ou un répertoire ;
- sécuriser l'accès à un fichier ou à un répertoire ;
- rendre un fichier exécutable.

4.1. Exercice 5.1 : Evaluer ses droits sur un répertoire

- Créer le répertoire « cours » dans votre répertoire de connexion.

```
[stagiaire]$ cd  
[stagiaire]$ mkdir cours
```

- Observer les droits d'accès appliqués à ce répertoire :

```
[stagiaire]$ ls -lisad cours  
drwxr-xr-x 5 stagiaire users 4096 fev 11 08:21 cours
```

- Quel utilisateur êtes-vous vis-à-vis de ce répertoire ?

Je suis le propriétaire

- De quels droits disposez-vous sur ce répertoire ?

Je dispose des droits de lecture, écriture et de passage dans le répertoire.

4.2. Exercice 5.2 : Interdire la modification du contenu d'un répertoire

- Modifier les droits du répertoire cours pour qu'ils deviennent dr-xr-xr-x.

```
[stagiaire]$ chmod u-w cours
```

- Créer un fichier linux.txt dans ce répertoire.

```
[stagiaire]$ touch cours/linux.txt
```

- Que se passe-t-il ?

cp: ne peut créer le fichier régulier 'cours/linux.txt' permission non accordée.

- Créer un sous répertoire « windows » dans le répertoire cours.

```
[stagiaire]$ mkdir cours/windows
```

- Que se passe-t-il ?

mkdir : ne peut créer le répertoire 'windows' : permission non accordée.

- Rétablir les droits drwxr-xr-x du répertoire cours. Créer un fichier linux.txt dans cours puis créer le répertoire cours/windows.

```
[stagiaire]$ chmod u+w cours  
[stagiaire]$ touch cours/linux.txt  
[stagiaire]$ mkdir cours/windows
```

- Remettre les droits dr-xr-xr-x sur le répertoire cours. Tenter de détruire le fichier linux.txt et le sous-répertoire windows.

```
[stagiaire]$ chmod u-w cours  
[stagiaire]$ rm -f cours/linux.txt  
rm: ne peut enlever 'cours/linux.txt' permission non accordée.
```

```
[stagiaire]$ rm -Rf cours/windows  
rm: ne peut détruire le répertoire 'cours/windows' : permission non accordée.
```

- Copier le fichier cours/linux.txt dans le répertoire cours/windows en le renommant srv2k8.txt.

```
[stagiaire]$ cp cours/linux.txt cours/windows/srv2k8.txt
```

- Commenter le résultat.

Les droits du répertoire cours permettent le passage, et ceux sur windows la copie du fichier.

4.3. Exercice 5.3 : Interdire l'accès à un répertoire

- Modifier les droits du répertoire cours pour qu'ils deviennent drw-r-xr-x.

```
[stagiaire]$ chmod 655 cours
```

- Se positionner dans cours.

```
[stagiaire]$ cd cours
```

- Que se passe-t-il ?

```
bash : cd : cours : permission non accordée
```

Il manque le droit x qui permet le positionnement.

- Afficher le contenu du fichier cours/linux.txt.

```
[stagiaire]$ less cours/linux.txt  
less: cours/linux.txt : permission non accordée
```

- Afficher ou tenter d'accéder au sous répertoire windows.

```
[stagiaire]$ cd cours/windows
```

- Que se passe-t-il ?

```
bash : cd : cours/windows : permission non accordée
```

4.4. Exercice 5.4 : Autoriser les droits en lecture seule sur un fichier

- Mettre les droits afin de pouvoir se positionner dans le répertoire cours et donner les droits de lecture seule au fichier « linux.txt ».

```
[stagiaire]$ chmod 755 cours  
[stagiaire]$ cd cours  
[stagiaire]$ chmod u-w linux.txt
```

- Afficher le contenu de ce fichier.

```
[stagiaire]$ less linux.txt
```

- Tenter de modifier son contenu (à l'aide d'un éditeur de texte ou par la commande « cat >>linux.txt »).

```
[stagiaire]$ vim linux.txt  
[stagiaire]$ cat >>linux.txt
```

- Tenter de détruire le fichier « linux.txt ».

```
[stagiaire]$ rm -f linux.txt
```

- Que se passe-t-il ? Pourquoi ?

La destruction du fichier est possible car le répertoire où il est situé possède le droit « w » pour « stagiaire ».

4.5. Exercice 5.5 : Droits d'exécution sur un fichier exécutable

- Copier le fichier « /bin/ls » dans votre répertoire de travail. Renommer le fichier en « ls1 ».

```
[stagiaire]$ cd  
[stagiaire]$ cp /bin/ls ls1
```

- Quel est son type ?

```
[stagiaire]$ file ls1
```

- Exécuter ce fichier par la commande « ./ls1 ».

```
[stagiaire]$ ./ls1
```

- Enlever les droits d'exécution à ce fichier.

```
[stagiaire]$ chmod a-x ls1
```

- Tenter de le lancer à nouveau « ./ls1 ». Que se passe-t-il ?

```
[stagiaire]$ ./ls1  
bash: ./ls1: permission non accordée
```

4.6. Exercice 5.6 : Droits d'exécution sur un script

- Créer un script nommé « Prog » dans ce script vous écrirez ceci :

```
#!/bin/bash  
clear  
echo "essai de script Shell"
```

- Vérifier les droits du fichier « Prog ».

```
[stagiaire]$ ls -l Prog
```

- Afficher le type de ce fichier.

```
[stagiaire]$ file Prog
```

- Modifier ses droits pour qu'il soit exécutable par le propriétaire.

```
[stagiaire]$ chmod u+x Prog
```

- Exécuter ce fichier « ./Prog » et vérifier le résultat.

```
[stagiaire]$ ./Prog
```

Affichage à l'écran du message « essai de script Shell ».