

ATTENTION : IMPOSSIBLE DE MODIFIER LES DROITS DE test6 SOUS UBUNTU !

FONCTIONNE CEPENDANT SOUS FREEBSD (voir tuto\_install\_freeBSD\_VM)

```
cd shell00
```

```
mkdir ex02
```

```
cd ex02
```

// RESULTAT ATTENDU :

```
%> ls -l
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x---r-- 2 XX XX XX Jun 1 22:45 test2
-r-----r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r---x 1 XX XX 2 Jun 1 23:43 test4
-r-----r-- 2 XX XX 1 Jun 1 23:44 test5
lrwxr-xr-x 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

// test0 :

```
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
```

```
mkdir test0
```

// d indique que test0 est un répertoire (directory)

```
chmod 715 test0
```

// rwx--xr-x = 715

// autre syntaxe possible pour chmod :

// on indique par une des lettres suivantes le (ou les)

// ensembles de droits que l'on veut modifier :

// u : user (premier ensemble de droits)

// g : group (deuxième ensemble de droits)

// o : others (troisième ensemble de droits)

// a : all (tous les ensembles de droits)

// puis si on veut ajouter ou retirer un (ou des) droit(s)

// + : ajouter

// - : retirer

// enfin, le(s) droit(s) à ajouter ou retirer

// r : read

// w : write

```

touch -mt 2306012047 test0 // x : execute
// les droits par défaut de test0 (juste après sa création) sont
// rwxrwxr-x et on veut obtenir
// rwx--xr-x on peut donc utiliser la commande
// chmod g-rw test0
// pour retirer rw à group (2ème ensemble de droits)
// pour Jun 1 20:47
// (-m = modification time) : permet de mettre à jour
// la date de modification du fichier (ou du répertoire)
// (-t = STAMP) : permet d'utiliser la date indiquée
// à la place de la date actuelle
// YYMMDDhhmm

```

```
// test1 :
```

```
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
```

```
touch test1 // - indique que test1 est un fichier
```

```
// le fichier pèse 4 octets
```

```
vim test1
```

```
<i>
```

```
<space> x3
```

```
<escape>
```

```
:wq
```

```
<enter>
```

```
// ou
```

```
// echo " " > test1
```

```
chmod 714 test1 // rwx--xr-- = 714
// les droits par défaut de test1 (juste après sa création) sont
// rw-rw-r-- et on veut obtenir
// rwx--xr-x on peut donc utiliser les commandes
// chmod u+x test1
// chmod g-rw test1
// chmod o-x test1
```

```
touch -mt 2306012146 test1 // pour Jun 1 21:46
```

```
// test2 :
```

```
dr-x---r-- 2 XX XX XX Jun 1 22:45 test2
```

```
mkdir test2 // d indique que test2 est un répertoire (directory)
chmod 504 test2 // r-x---r-- = 504
// les droits par défaut de test2 (juste après sa création) sont
// rwxrwxr-x et on veut obtenir
// r-x---r--
// on peut donc utiliser les commandes
// chmod u-w test2
// chmod g-rwx test2
//chmod o-x test2
```

```
touch -mt 2306012245 test2 // pour Jun 1 21:46
```

// test3 et test5 :

```
-r-----r-- 2 XX XX 1 Jun 1 23:44 test3
```

```
-r-----r-- 2 XX XX 1 Jun 1 23:44 test5
```

// on remarque que test3 et test5 sont identiques (sauf leur nom)  
// et que le nombre de liens vers chacun de ces fichiers est 2  
// alors que ce ne sont pas des répertoires  
// (ce sont tous les deux des fichiers (- indique que ce sont des fichiers))  
// il faut donc créer un lien (hard link) entre eux  
// hard link (lien matériel) : crée une nouvelle entrée d'index dans le système  
// de fichier pour pointer vers le même contenu que le fichier cible  
// un hard link ne peut pas pointer sur des systèmes de fichiers différents  
// si le fichier cible est supprimé, le lien matériel reste intact tant que d'autres  
// liens pointent toujours vers le contenu

touch test3

// crée le fichier test3

ln test3 test5

// crée un lien (hard link) vers le fichier test3, appelé test5

// UNE FOIS LE LIEN CREE

// LES MODIFICATIONS SUR test3 SE REPERCUTERONT SUR test5 !

// les fichiers pèsent 1 octet

vim test3

<i>

<space>

<backspace>

<escape>

:wq

<enter>

// ou

// echo "" > test3

```
chmod 404 test3 // r----r-- = 404
// les droits par défaut de test3 (juste après sa création) sont
// rw-rw-r-- et on veut obtenir
// r----r-- on peut donc utiliser les commandes
// chmod ug-w test3
```

```
touch -mt 2306012344 test3 // pour Jun 1 23:44
```

```
// test4 :
```

```
-rw-r-----x 1 XX XX 2 Jun 1 23:43 test4
```

```
touch test4 // - indique que test4 est un fichier
// le fichier pèse 2 octets
```

```
vim test4
<i>
<space>
<escape>
:wq
<enter>
```

```
// ou
// echo " " > test4
```

```
chmod 641 test4 // rw-r---x = 641
// les droits par défaut de test4 (juste après sa création) sont
// rw-rw-r-- et on veut obtenir
// rw-r---x on peut donc utiliser les commandes
// chmod g+w test4
// chmod o-r test4
// chmod o+x test4
```

```
touch -mt 2306012343 test4 // pour Jun 1 23:43
```

// test6 :

```
lrwxr-xr-x 1 XX XX 5 Jun 1 22:20 test6 -> test0
```

// l indique que test6 est un lien symbolique

// test6 -> test0 indique que c'est un lien vers test0

ln -s test0 test6 // crée un lien symbolique (symlink) vers le fichier test0, appelé test6  
// fonctionne comme un raccourci vers le fichier (ou répertoire) cible  
// un lien symbolique peut pointer sur des systèmes de fichiers différents  
// si le fichier (ou répertoire) cible est supprimé,  
// le lien symbolique est rompu !

chmod 755 test6 // rwxr-xr-x = 755  
// ATTENTION : NE FONCTIONNE PAS SOUS UBUNTU !

touch -hmt 2306012220 test6 // pour Jun 1 22:20  
// ATTENTION A AJOUTER L'OPTION -h !  
// l'option h permet d'appliquer la modification  
// sur le lien symbolique au lieu de l'appliquer sur le fichier  
// qu'il référence  
// SANS CETTE OPTION, LA MODIFICATION SERA APPLIQUEE  
// A test0 !

tar -cf ex02.tar \* // compresse tous le contenu du répertoire courant dans l'archive ex02.tar  
// (\* = wildcard) : toute suite de caractère

// LE REPERTOIRE DE RENDU DE DOIT CONTENIR QUE LE FICHIER ex02.tar !  
rm -r test\* // (rm = remove) : efface un (ou des) fichier(s) ou répertoire(s)  
// l'option -r permet aussi de supprimer un (ou des) répertoire(s) et leur contenu  
// avec rm -r test\*  
// on supprime tous les fichiers et les répertoires (avec leur contenu)  
// dont le nom commence par test