

1^{ère} NSI — Architecture Système

TP Shell UNIX

tiré d'un TP de Charles Poulmaire

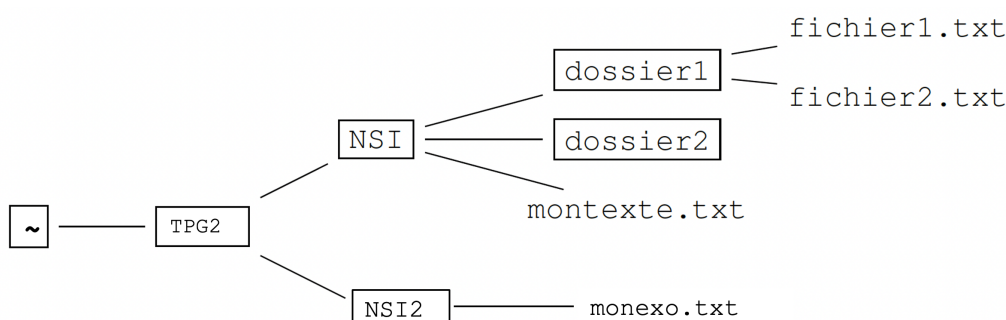
Lycée Fustel de Coulanges, Massy

Marc Biver, mai 2024, *v0.1*

Tout ce qui est décrit dans ce document va être à réaliser dans le terminal de votre machine HP NSI, que vous pouvez lancer depuis la barre de tâches (le logo est un carré noir avec un ">" blanc dessus).

1 Arborescence de fichiers

L'arborescence de fichiers représente l'organisation des dossiers et fichiers dans votre ordinateur. Voici un exemple d'une arborescence extrêmement basique :



- Le répertoire de base dans cette représentation est le répertoire personnel de l'utilisateur de l'ordinateur que par convention on note "~". Sur vos machines il s'agit du répertoire `/home/nsi` et c'est là que s'ouvre votre terminal .
- Dans cet exemple il y a un répertoire dans ~ : TPG2.
- Dans cet exemple il y a deux répertoires dans ~/TPG2, NSI et NSI2.
- Le fichier `montexte.txt` est directement dans le répertoire NSI.
- Le fichier `fichier2.txt` est dans le répertoire `dossier1` qui est un sous-répertoire de NSI.
- Etc...

Voici un certain nombre de commandes utiles pour naviguer dans l'arborescence de fichiers et créer de nouveaux répertoires et fichiers :

<code>pwd</code>	permet de savoir où on se trouve dans l'arborescence de fichiers
<code>mkdir dossier</code>	crée un répertoire dossier
<code>touch fichier.txt</code>	crée le fichier <code>fichier.txt</code> (vide)
<code>mv fichier.txt dossier</code>	déplace le fichier <code>fichier.txt</code> dans le répertoire dossier
<code>mv fichier1.txt fichier2.txt</code>	renomme le fichier <code>fichier1.txt</code> en <code>fichier2.txt</code>
<code>cp fichier1.txt fichier2.txt</code>	crée <code>fichier2.txt</code> identique à <code>fichier1.txt</code> (copie)
<code>ls</code>	affiche le contenu du répertoire dans lequel on se trouve
<code>cd dossier</code>	change de répertoire pour aller dans le répertoire fils dossier
<code>cd ..</code>	change de répertoire pour aller dans le répertoire père
<code>cd ~</code>	change de répertoire pour revenir au répertoire personnel
<code>rm fichier.txt</code>	supprime le fichier <code>fichier.txt</code>
<code>rmdir dossier</code>	supprime le répertoire vide dossier
<code>rm -r dossier</code>	supprime le répertoire non vide dossier (suppression récursive avec l'option <code>-r</code>)

Exercice 1: Manipulations élémentaires de fichiers

- a. Assurez-vous d'abord de bien être dans le répertoire personnel de votre utilisateur : tapez `pwd`, puis *Entrée* : obtenez-vous bien `/home/nsi` ?
- b. A l'aide des commandes ci-dessus, reproduisez l'arborescence du schéma plus haut (TPG2 etc...).
- c. Enrichissez votre arborescence :

- (a) Créez dans `NSI2` deux répertoires : `essai1` et `essai2`.
- (b) Créez dans `essai2` un fichier `text.txt`.
- (c) Dupliquez ce fichier dans le même répertoire en deux fichiers `text1.txt` et `text2.txt`.
- (d) Déplacez `text1.txt` dans le répertoire `dossier2` de l'arborescence.
- (e) Déplacez-vous dans tous les répertoires pour vérifier que le contenu est le bon.
- (f) En une seule commande supprimez toute la sous-arborescence `NSI2`.

2 Gérer l'administration des droits

La commande `ls` possède les options suivantes :

<code>ls -l</code>	affiche le contenu du répertoire de manière détaillée
<code>ls -al</code>	affiche les répertoires et fichiers cachés

Le rendu d'un `ls -l` pourrait ressembler à ceci :

```
localhost:~$ ls -l
total 8
drwxr-xr-x  2 NSI      user2      95 Sep 20 12:27 dossier1
drwxr-xr-x  2 NSI      user2      37 Sep 20 12:27 dossier2
-rw-r--r--  1 NSI      user2        0 Sep 20 12:27 montexte.txt
```

Dans l'ordre, cela se lit, de gauche à droite :

droits - nombre de liens - nom du propriétaire - nom du groupe - taille en octet - date - heure - nom du fichier ou du répertoire

Les systèmes de type UNIX sont des systèmes multi-utilisateurs. Plusieurs utilisateurs peuvent donc partager un même ordinateur. Comme chaque utilisateur possède un environnement de travail qui lui est propre (avec pour répertoire personnel `/home/nsi` dans notre cas ici), chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres. On peut gérer ces droits par le shell.

Il faut distinguer un utilisateur un peu particulier qui est autorisé à modifier tous les droits : ce "super utilisateur" est appelé "administrateur" ou "root".

Au lieu de gérer les utilisateurs un par un, il est possible de créer des groupes d'utilisateurs. L'administrateur peut alors attribuer des droits à un groupe directement. Remarque : dans l'exemple donné ci-dessus, le nom du groupe est `user2` et celui du propriétaire `NSI`.

Ainsi les fichiers et les répertoires possèdent 3 types de droits :

- Les droits en lecture : `r` (Read) signifie que la lecture est autorisée ;
- Les droits en écriture : `w` (Write) signifie que l'écriture est autorisée ;
- Les droits en exécution : `x` (eXecute) signifie que l'exécution est autorisée.
- Un tiret "-" interdit le droit en question.
- Les droits sont listés d'abord pour le propriétaire du fichier, puis pour les membres de son groupe, puis pour tous les autres utilisateurs.

Ainsi, pour le fichier `montexte.txt` dans l'exemple précédent, dont les droits sont écrits "`-rw-r--r--`", les droits sont :

- Le premier caractère, "-" indique que c'est un fichier et non un répertoire (on aurait alors `d` comme Directory) ;
- `rw-` pour le propriétaire : accès en lecture et en écriture ;
- `r--` pour le groupe et tous les autres : accès en lecture seulement ;
- On n'a pas évidemment de `x` ici puisque l'exécution d'un fichier texte n'a pas de sens.

Pour changer les droits d'un fichier ou dossier, on utilise la commande `chmod` suivie d'un nombre composé de 3 chiffres (un pour le propriétaire, un pour le groupe, un pour les autres) puis du nom du fichier concerné. Pour savoir quel nombre on choisit il suffit de savoir compter en binaire. Par exemple :

- `rwX` correspondra au nombre binaire 111 donc au nombre entier 7 ;
- `rw-` correspondra au nombre binaire 110 donc au nombre entier 6 ;
- `r--` correspondra au nombre binaire 100 donc au nombre entier 4.

Donc par exemple si l'on voulait modifier les droits de `montexte.txt` pour donner le droit en écriture au groupe mais sans le donner à tout le monde on taperait la commande : `chmod 664 montexte.txt` (6 pour le propriétaire et le groupe, 4 pour les autres).

Exercice 2: Gestion des droits d'un fichier

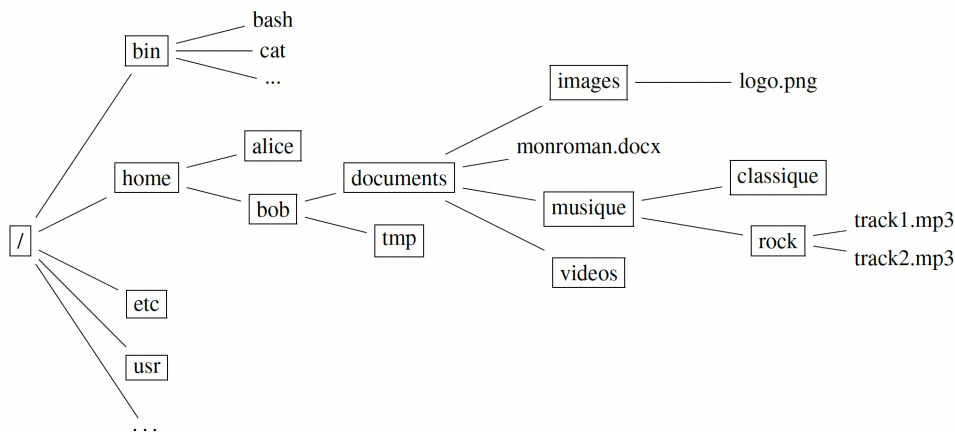
- Quel nombre de 3 chiffres doit-on choisir pour qu'un fichier soit accessible en lecture uniquement au propriétaire et au groupe, accessible en exécution à tous et uniquement accessible en écriture au propriétaire ? Testez votre commande sur le fichier `montexte.txt`
- Changez les droits du répertoire `dossier2` en `rxrw-r--`
- Tapez la commande `chown -help`. A quoi sert la commande `chown` ?

3 Travail à rendre sur papier

Rien de ce qui précède n'est à rendre — mais si tout a été fait, cet exercice, à rendre sur papier, devrait être extrêmement simple et rapide à effectuer.

Exercice 3: A rendre

On considère l'arborescence suivante :



- En supposant que je me trouve dans le dossier `documents`, que sera-t-il affiché dans la console si je tape `pwd` ?
- En supposant que je me trouve dans le dossier `images`, quelle commande dois-je saisir si je veux aller dans le dossier `musique` ?
- Quelle est la différence entre les commandes `cp test1.txt test2.txt` et `mv test1.txt test2.txt` ?
- Si je saisis la commande `rm videos` à partir du dossier `documents` que se passe-t-il ?
- Si je veux détruire le dossier `rock` quelle commande dois-je saisir ?
- Pour connaître les droits du fichier `monroman.docx` que faut-il saisir ?
- Si je veux changer les droits de `monroman.docx` afin d'être la seule personne à pouvoir le lire et le modifier, que dois-je saisir dans le shell ? On supposera également que personne ne pourra exécuter ce fichier.

4 Défi – pour ceux qui le souhaitent

Exercice 4: Quelques défis pour aller plus loin

- Automatiser la création de 100 répertoires nommés `rep1` à `rep100` dans le répertoire `dossier2`.
- Copier tous les fichiers `.txt` du répertoire `dossier1` vers `dossier2` uniquement si leur taille dépasse 1KB.
- Supprimer tous les fichiers temporaires (extension `.tmp`) dans `/home/nsi` qui ont été modifiés il y a plus de 7 jours.