



iNFO

IUT

GRAND OUEST
NORMANDIE

R 1.04

2024 - 2025

Introduction aux systèmes d'exploitation et à leur fonctionnement

TD N°3

« L'arborescence du système de fichiers d'UNIX »



ANNE Jean-François
D'après le TD de F. BOURDON

Le but de ce TD est de se familiariser avec les systèmes d'exploitation et avec leur fonctionnement.

« L'arborescence du système de fichiers d'UNIX »

Notions vues dans ce TD :

Noms génériques de fichiers, arborescence (./..) et méta-caractères.

Nombre de séance de 2h00 prévu pour faire ce TD : 1.

PS : Les parties correspondant à du travail à faire sont toutes en italiques ; le restant étant du complément au cours.

1. Gestion des utilisateurs

Un utilisateur Linux est caractérisé en particulier par :

- un nom (ou login) ;
- un numéro unique d'utilisateur (user id) ;
- un mot de passe (password) ;
- l'appartenance à un ou plusieurs groupes.

La plupart des caractéristiques des utilisateurs sont recensées dans le fichier `/etc/passwd`.

 *A l'aide de la commande `cat`, afficher le fichier `/etc/passwd`. Décrire les informations présentes dans ce fichier (vous pouvez pour cela chercher des informations sur internet).*

 *De la même manière, afficher le fichier `/etc/group` et décrire les informations qui y sont présentes.*

Les deux fichiers `/etc/group` et `/etc/passwd` décrivent les groupes et les utilisateurs sur la machine locale. Dans le cas de systèmes distribués, ces fichiers ne se trouvent pas sur la machine locale mais sur un serveur. L'affichage des fichiers adéquats peut alors se faire grâce à la commande `ypcat`, cependant, ces commandes réseau sont parfois bloquées par les administrateurs.

 *Utiliser maintenant la commande `groups`. que se passe-t-il ? En utilisant le `man`, écrire une variante de l'appel à `groups` qui n'affiche des données que de l'utilisateur voulu.*

Certaines commandes permettent à l'administrateur de gérer les utilisateurs et les groupes.

 *En utilisant le `man` ou en cherchant sur internet, expliquer ce que font les commandes suivantes :*

`groupadd -g 1001 users`

`useradd -d /home/pba -m -p toto -g users pba`

```
useradd -d /home/aka -g users -u 5002 aka  
groupdel users  
userdel -r aka
```

Expliquer chaque commande indépendamment.

La plupart des commandes de gestion des utilisateurs sont réservées aux administrateurs.

 Entrer les commandes de la question 4 dans un terminal. Que se passe-t'il ? Quelle en est l'explication ?

2. Exploration de l'arborescence Linux

1. Indiquez par une commande dans quel répertoire vous vous trouvez.
2. Allez dans le répertoire `/usr/share/doc`, puis vérifiez le chemin de votre répertoire courant.
3. Remonter dans le répertoire parent puis vérifier.
4. Allez dans votre répertoire personnel sans taper son chemin.
5. Retournez dans votre répertoire précédent sans taper son chemin.
6. Retourner dans votre répertoire personnel et listez les fichiers présents.
7. Listez maintenant tous les fichiers (même ceux cachés).
8. Affichez de façon détaillée le contenu du répertoire `/usr` sans changer le répertoire de travail.
9. Affichez l'arborescence de fichiers contenue dans `/var` sans changer le répertoire de travail.
10. Affichez de façon détaillée le contenu du répertoire `/var/log` en classant les fichiers du plus vieux au plus récent.
11. Afficher les informations détaillées du répertoire `/home` sans lister son contenu.

3. Répertoires et consultation de fichiers

1. En tant qu'utilisateur, allez dans votre répertoire personnel.
2. Créez un répertoire portant le nom de **CommandesLinux**.
3. Allez dans votre répertoire **CommandesLinux**.
4. Créez l'arborescence **cours1/cours2/cours3/cours4** pour que ce soit fait **en une seule commande** ?
5. Listez le contenu du répertoire courant de façon récursive.
6. Supprimez le répertoire `cours1`. Est-ce possible ?
7. Supprimez l'arborescence de répertoire **cours1/cours2/cours3/cours4**, puis vérifiez en listant le répertoire de façon récursive.
8. Créez maintenant l'arborescence suivante dans votre répertoire courant.
(**/home/nomUtilisateur/CommandesLinux**)

```
--couleur
|--froide
|--forme
  |--angle
  |--courbe
```

1. Copiez-le fichier /etc/services dans votre répertoire CommandesLinux.
2. A qui appartient le fichier que vous venez de copier ? Quelle est sa date de sa dernière modification ?
3. Créez les fichiers ne contenant aucune donnée et dont les noms sont les suivants :
rond.txt, triangle.txt, carre.txt, rectangle.txt, vert.txt et bleu.txt
4. Déplacez le fichier rond.txt dans le répertoire courbe et les fichiers **triangle.txt, carré.txt, rectangle.txt** dans le répertoire **angle**.
5. Déplacez les fichiers **vert.txt et bleu.txt** dans le répertoire **froide**.
6. Allez dans le répertoire couleur et affichez le contenu du répertoire de façon récursive.
7. Copier le répertoire sous le nom **chaude**. Est-ce possible ? Comment ?
8. Allez dans le répertoire **chaude** et renommez le fichier bleu.txt en rouge.txt et vert.txt en jaune.txt.
9. Effectuez un lien dur (« hardlink ») nommé orange.txt vers le fichier rouge.txt.
10. Effectuez un second lien symbolique nommé orangeAbsolu vers le fichier rouge.txt en spécifiant cette fois-ci le fichier rouge avec un chemin absolu.
11. Remontez dans le répertoire CommandesLinux et renommez le répertoire couleur en peinture. Est-il besoin de spécifier une option particulière à la commande **mv**.
12. Essayez d'afficher le contenu des fichier orange et orangeAbsolu. Cela fonctionne-t-il ?
13. Listez la totalité de l'arborescence contenue dans le répertoire CommandeLinux.
14. Affichez le contenu du fichier /etc/issue. Que contient-il ?
15. Affichez page par page le contenu du fichier /etc/services. Que contient-il ?

4. L'arborescence du système de fichiers d'UNIX

Une arborescence est une façon d'organiser le rangement et par conséquent l'accès aux fichiers d'UNIX. Chaque système de fichiers possède donc une et une seule arborescence qui évolue sans cesse. Ceci se concrétise par la présence d'une seule racine appelée par convention « / », de nœuds et de feuilles. Chaque nœud correspond à un répertoire de l'arborescence dans lequel on peut trouver des nœuds et/ou des feuilles. Une feuille correspond à un fichier ordinaire (différent des répertoires).

On appelle répertoire d'accueil (« home directory », ou plus simplement home) l'endroit où l'interprète de commande (Shell) est lancé en début de session (connexion « login/passwd »).

On appelle répertoire courant ou répertoire de travail (« pwd ») l'endroit de l'arborescence où l'on se trouve lorsqu'on lance des commandes.

Un fichier (ordinaire ou répertoire) est donc nécessairement placé au moins à un endroit de l'arborescence. Pour manipuler ces différents fichiers il faut les nommer. On peut inclure dans leur nom, le chemin permettant de les retrouver dans l'arborescence. Par exemple le fichier « passwd » qui contient les informations concernant les différentes sessions autorisées sur la machine, se trouve dans le répertoire « etc », qui lui-même se trouve dans le répertoire racine « / ». On pourra alors directement parler du fichier « passwd » en utilisant « /etc/passwd ».

✍ Créer à l'aide de la commande « `mkdir` » un répertoire de nom « `rep` » dans votre répertoire de travail. Placez-vous dans ce nouveau répertoire (« `cd` ») et examiner le contenu de ce dernier, en utilisant successivement les commandes « `ls` » puis « `ls -al` » enfin « `ls -ali` ». Qu'observez-vous ? Ce répertoire contient-il des fichiers ? Si oui, quels sont-ils ?

✍ Préciser la fonction de l'option `-i` pour la commande `ls`.

La racine est un élément particulier du système de fichier. elle est le point qui ne peut être dépassé.

✍ En utilisant la commande `cd /..` et la commande `pwd`, confirmer le rôle particulier de la racine. Que se passe-t-il quand on tape ces deux commandes plusieurs fois d'affilée ?

5. Fichiers cachés

Il est possible sous linux de créer des fichiers cachés. Par convention, le nom d'un fichier ou dossier caché débute toujours par un point `..`. Le fait qu'il soit caché permet à un fichier/dossier de ne pas être pris en compte par certaines commandes.

✍ Après être retourné dans le répertoire `rep` via la commande `cd ~/rep`, créer le fichier `.cache` à l'aide de la commande `touch .cache`.

✍ Remonter ensuite dans le répertoire parent grâce à la commande `cd ..` puis tenter d'effacer le répertoire `rep` à l'aide de la commande `rmdir rep`.

✍ Que se passe-t'il ?

Avant d'effacer un répertoire avec la commande `rmdir`, il faut s'assurer qu'il est bien vide.

✍ Lister les fichiers présents dans le répertoire `rep` à l'aide de la commande `ls rep`. Le répertoire contient-il des fichiers ?

✍ Refaire la question précédente en utilisant cette fois la commande `ls -a rep`. Quel sont les changements ?

Dans le cas d'un effacement de répertoire sans devoir au préalable s'assurer qu'il est bien vide, la commande `rm` est très utile.

✍ Effacer le répertoire `rep` grâce à la commande `rm -r rep`. Quel est le résultat ?

✍ Pour vous aider à répondre à ces questions, lancer les commandes suivantes : « `$ cat .` », « `$ rm .` » et « `$ cd ..` » ; puis pour chacune d'elles, expliquer le message renvoyé par l'interprète de commande (Shell).

✍️ Retourner dans votre répertoire d'accueil (« home directory »), lancer la commande « `ls -ali | more` » (lire « pipe more »). En observant les deux premières lignes obtenues, pouvez-vous compléter la réponse à la question précédente (nature des fichiers observés) ?

La gestion des fichiers et dossiers cachés peut être illustrée simplement avec l'utilisation de la commande `ls` et ses paramètres.

En généralisant cette observation à tous les répertoires (nœuds) de l'arborescence du système de fichiers d'UNIX, on peut comprendre comment est construite cette arborescence.

✍️ Expliquez comment cela se passe pour la racine ? Faites la commande « `cd /` », puis la commande « `cd ..` ». La deuxième commande a-t-elle eu un impact sur votre Shell ?

Dans UNIX la notion de liens « hard » correspond à deux ou davantage d'entrées (noms de fichiers symboliques) qui désignent un même fichier de données. Pour une entrée donnée on connaîtra le nombre de liens qu'elle a en regardant les informations fournies par l'option « `-l` » de la commande « `ls` ». Un exemple que vous connaissez maintenant est justement celui des entrées « `.` » et « `..` » qui sont des liens hard. Pour visualiser ces liens hard vous devez utiliser l'option « `a` » (affichage des fichiers cachés) de la commande « `ls` », en plus de l'option « `l` » : « `ls -al ..` ».

✍️ Créez un nouveau répertoire « `nouv_rep` » dans lequel vous ferez la commande « `touch .fich1` ». Remontez d'un niveau dans l'arborescence (« `cd ..` ») puis effectuez les deux commandes « `rmdir` » et « `ls` » sur ce répertoire nouvellement créé, qui produiront les résultats suivants :

```
$ rmdir rep
rmdir : rep not empty
$ ls -l rep
total 0
$
```

✍️ Expliquez ce qui se passe en utilisant la commande « `ls` » et son option « `-a` ». Faites ensuite la commande « `ls -ali` » et observez le résultat. A quoi correspond le numéro affiché en début de chaque ligne (cf. le manuel « `man` ») ? La troisième colonne affiche le nombre de liens hard du fichier courant. Expliquez les valeurs de ces nombres (liens) en vous appuyant sur les nombres de la première colonne obtenu en faisant « `ls -ali` » sur le sous-répertoire courant et sur son répertoire parent.

✍️ En utilisant les liens hard « `.` » et « `..` », expliquer en général combien un répertoire contient de sous-répertoires. Il existe une formule simple qui permet de faire ce calcul. Pour trouver cette formule vous pouvez prendre votre répertoire « `rep` » sous votre répertoire d'accueil (« `$HOME/` ») et ajouter progressivement des sous-répertoires. Vous pourrez vérifier votre formule sur le répertoire « `/etc` » en utilisant la commande « `ls -ld` » et en comptant à la main le nombre de ses sous-répertoires. Pour comprendre le rôle de l'option « `d` » de la commande « `ls` », reportez-vous au manuel « `man` ».

Pour définir le chemin d'accès à un fichier et par conséquent son nom, nous disposons de deux techniques complémentaires qui sont le « nommage absolu » et le « nommage relatif ».

 Expliquer ces deux notions que vous illustrerez en utilisant la commande « ls » en donnant deux chemins différents qui visualisent le contenu du répertoire d'accueil d'un autre étudiant.

6. Noms génériques et principe de fonctionnement de la commande « ls »

La commande « **ls -a** » affiche la liste triée de tous les noms de fichiers du répertoire courant. On ne considère que les fichiers ordinaires et les répertoires ; la commande « **ls fich** » fonctionne de la façon suivante :

Si « fich » n'existe pas **Alors** « erreur » est affiché

Sinon Si « fich » est ordinaire **Alors** « fich » est affiché

Sinon le contenu de « fich » est affiché

Fsi

Fsi

La commande utilisée (« ls », « cd » ...) ne voit pas les noms génériques (méta-caractères ...). Le Shell réalise donc les étapes suivantes :

1. Substitution des méta-caractères par la définition associée
2. Création d'une liste de noms de fichiers
3. Lancement de la commande sur chaque nom de la liste triée (ordre ASCII)

On appelle nom générique de fichier un nom de fichier qui comprend des caractères spéciaux appelés « méta-caractères » (incluant les opérateurs associés). Ces « méta-caractères » pourront être substitués par des caractères classiques suivant des règles précises de substitution. Ainsi un nom générique de fichier est une définition d'un ensemble de noms de fichiers. L'un des méta-caractères le plus employé est « * ». Il correspond à n'importe quelle chaîne de caractères (type de chaque caractère et nombre de caractères).

 Construisez la sous-arborescence suivante sous « rep » :

1. Dans « rep » vous créez le répertoire « rep1 » et les fichiers « .c », « a » et « b ».
2. Dans « rep1 » vous créez le répertoire « rep2 » et les fichiers « a1 », « b1 » et « .c1 ».
3. Dans « rep2 » vous créez le répertoire « rep3 » et les fichiers « a2 », « b2 » et « .c2 ».
4. Enfin dans « rep3 » vous créez le fichier « a3 ».

 Donner les résultats de :

- ls
- ls .
- ls ..
- ls -a

- ls -a .
- ls -a ..

Il est possible de visualiser une arborescence de fichiers de façon textuelle. Pour cela, la commande ls peut être complétée du paramètre -R.

✍ Dans le répertoire rep créé lors des questions précédentes, utiliser la commande ls -R. Quel en est le résultat ?

7. Noms génériques et méta-caractères

On appelle nom générique de fichier un nom de fichier qui comprend des caractères spéciaux appelés méta-caractères. Ceux-ci pourront être substitués par des caractères classiques suivant des règles précises de substitution. Ainsi un nom générique de fichier peut être est une définition d'un ensemble de noms de fichiers.

L'un des méta-caractères le plus employé est l'étoile *. Il correspond à n'importe quelle chaîne de caractères.

Afin de voir comment le système gère les noms génériques, il est possible d'utiliser la commande echo. Celle-ci affiche son argument après traitement par le shell.

✍ Pour chacun des noms génériques ci-dessous, expliquez, en utilisant les commandes « echo », « ls » et « ls -a », comment le Shell réalise l'expansion des noms (passage d'un nom générique à une liste de noms) de fichiers correspondants :

- *
- .*
- */*
- */.*
- .*/*
- .*/.*

Les noms génériques de fichiers utilisent des méta-caractères (« * », « ? » ...) et des opérateurs spécifiques (« [] », « - », « ! » ...). Par exemple le nom générique « fich[123] .* » correspond à la liste des fichiers dont le nom commence par « fic », puis dont le 5^{ème} caractère est soit « 1 », soit « 2 », soit « 3 », le 6^{ème} est un « . » et la suite est une chaîne de caractères dont chacun d'eux et leur nombre peut prendre toutes les valeurs possibles.

✍ Placez-vous sous « rep » et créer à l'aide de la commande « touch » les quatre fichiers suivants : « 21 », « 27 », « 2a » et « 39 ». En utilisant la commande « ls » et les noms génériques de fichiers, afficher tous les fichiers de votre répertoire courant (« rep ») dont les noms sont un nombre compris entre 20 et 40.

✍ Sous le même répertoire courant, créer les fichiers « b.truc », « chose », « s.f » et « sous ». Trouver une commande qui permet d'afficher tous les fichiers dont le nom ne commence pas par « s. ».

✍ La solution trouvée précédemment fonctionne-t-elle toujours si l'on ajoute le fichier « sauf » à la liste des fichiers proposée dans la question précédente ? Proposez une

nouvelle solution qui réponde encore à la question (affichage des fichiers dont le nom ne commence pas par « s. »).

- ✍ En ajoutant le fichier « a.b » à la liste précédente, la nouvelle solution trouvée à la question précédente fonctionne-t-elle encore ? Si non, proposez une solution.
- ✍ En utilisant l'éditeur « vi » construire un fichier dont le contenu contient sur plusieurs lignes, plusieurs occurrences de la chaîne de caractères « le système Unix », écrites sous les formes suivantes : « le système Unix », « Le système Unix », « Le Système Unix » ou encore « le système Unix ». Utiliser les commandes « grep », « | » et « wc » pour compter le nombre d'occurrences de cette chaîne, toute orthographe considérée (majuscule/minuscule).

Complément d'exercices sur la notion de liens « hard » et de liens « symbolique ».

- ✍ Comme nous l'avons vu précédemment le système crée des liens hard pour la gestion de l'arborescence du système de fichier. C'est le cas des fichiers « . » et « .. ». Nous pouvons créer des liens hard sur les fichiers ordinaires. Pour cela il faut utiliser la commande « ln » avec comme premier paramètre le fichier original et comme second paramètre son lien hard. Placez-vous dans un nouveau répertoire (vide) et créez un premier fichier ordinaire, avec la commande « touch », puis un second fichier ordinaire comme étant un lien « hard » sur le premier, via la commande « ln ». Regardez les caractéristiques des deux fichiers en particulier leur i-noeud. Détruisez le premier fichier. Qu'observez-vous sur le lien (deuxième fichier créé) ? On ne peut pas utiliser la commande « ln » pour créer un lien « hard » sur un fichier répertoire. Expliquez pourquoi.
- ✍ De la même façon nous pouvons construire des liens « symbolique » avec la commande « ln » et en ajoutant l'option « -s ». Créez un lien « symbolique » sur un fichier ordinaire existant avec la commande « ln -s ». Faites « ls -li » pour observer les informations concernant ce lien. Regardez en particulier les différents i-noeuds des fichiers et les droits d'accès. Le lien « symbolique » possède-t-il le même i-noeud que le fichier cible ? Vous pouvez détruire le fichier cible avec la commande « rm ». Qu'en est-il du lien « symbolique » ? Pouvez-vous accéder à ce dernier ? Pouvez-vous faire un « rm » sur ce dernier ? Expliquez la différence entre un lien « symbolique » et un lien « hard ».

8. Jeu

- <http://luffah.xyz/bidules/Terminus/>

II. Webographie :

- <http://www.seinturier.fr/joomla/index.php/enseignements/9-ens-iut-se>
- <https://bourdon.users.info.unicaen.fr/cours/IUT-1A/tdtp/td3/TD3-1A.html>
- <https://linuxtrack.net/viewtopic.php?id=611>
- <https://overthewire.org/wargames/bandit/bandit0.html>
- <http://luffah.xyz/bidules/Terminus/>
-