

---

## TD n° 2 - Système de fichiers, utilisation des entrées / sorties

---

Les objectifs de ce TD sont :

- étudier le système de fichiers, son organisation et ses éléments;
- découvrir les méta-caractère du **shell**;
- étudier les concepts liés aux utilisateurs;
- étudier les droits d'accès;
- découvrir les entrées / sorties de base et leurs redirections.

### 1 Quelques astuces de la ligne de commande

#### Exercices

- 1 Avant d'ouvrir le terminal, que se passe-t-il lorsque, dans votre espace de travail Linux, vous appuyez simultanément sur **Ctrl** + **Alt** + **T** ?
- 2 Tapez les lettres `sl` puis appuyez deux fois sur la touche **↩**. Que constatez-vous ?
- 3 Rajoutez un `e` pour obtenir `sl e` puis appuyez deux fois sur la touche **↩**. Que constatez-vous ?
- 4 Que se passe-t-il si vous appuyez simultanément sur **Ctrl** + **A** ? Puis sur **Ctrl** + **E** ?
- 5 Renseignez-vous sur la commande `sleep`
- 6 Exécutez la commande `sleep 3200000`. Que se passe-t-il quand vous tapez **Ctrl** + **C** ?
- 7 Que se passe-t-il lorsque vous pressez la touche **↑** ?
- 8 Que se passe-t-il si vous faites alors **Ctrl** + **U** ?
- 9 Que se passe-t-il si vous tapez `!sl` puis **↵** ?
- 10 Que se passe-t-il si vous tapez `!!` puis **↵** ?
- 11 Que fait la combinaison de touche **Ctrl** + **L** ?
- 12 Que fait la commande `clear` ?
- 13 Que se passe-t-il lorsqu'on tape **Ctrl** + **D** ?

### 2 Le système de fichiers

Le système de fichiers est vu comme une arborescence avec une racine unique `/`. Les nœuds de cet arbre sont appelés répertoires (*directories*), les feuilles sont des fichiers (*files*).

Nous donnons ici quelques nouvelles commandes utiles.

**pwd** (*print working directory*) affiche le répertoire courant.

**df** (*disk free*) est une commande qui permet d'afficher l'espace utilisé et l'espace disponible sur les différents supports montés. L'unité par défaut est un *bloc* de 1 kilo octet. L'option `-h` permet d'afficher les tailles en unités plus courantes (multiples de l'octet);

**du** (*disk usage*) permet d'afficher la taille d'un répertoire et de tous les sous répertoires récursifs qu'il contient (ici encore, l'unité par défaut est le bloc, l'option `-h` affiche des valeurs plus faciles à lire).

**file** donne des informations sur le contenu d'un fichier (le format des données, l'encodage, etc.).

**touch** modifie la date de la dernière utilisation de son argument si c'est un fichier existant. Autrement, il permet de créer un fichier vide (le nom du fichier à créer est passé en argument).

Quelques raccourcis :

- `~` désigne le répertoire principale de l'utilisateur courant (*home directory*);
- `~user` désigne le répertoire principale de l'utilisateur *user*;
- `.` désigne le répertoire courant;
- `..` désigne le père du répertoire courant;
- `/` désigne le répertoire racine du système.

## Méta-caractères

Les méta-caractères ou caractères génériques (*wildcard characters*) permettent de désigner un ensemble d'objets et notamment un ensemble de noms de fichiers<sup>1</sup>. Ils s'appliquent donc aux paramètres de commandes désignant des noms de fichiers.

- `*` représente une chaîne de caractères quelconque (éventuellement vide);
- `?` représente un caractère;
- `[xyz...]` représente un caractère de la liste `x`, `y`, `z`, etc. On peut également utiliser des intervalles : `[a-z]` représente un caractère compris entre `a` et `z`;
- `[^xyz...]` représente un caractère ne faisant pas partie de la liste `x`, `y`, `z`, etc.;
- `{abc,xyz,...}` représente une chaîne de caractères parmi la liste `abc`, `xyz`, etc.

## Exercices

- 14 Ouvrez un terminal et affichez le contenu du répertoire courant suite au *login* ou à l'ouverture d'une fenêtre terminal. Vérifier que c'est votre répertoire principal (*home*).
- 15 Déplacez-vous à la *racine* de l'arborescence du système. Vérifiez. Affichez le contenu de ce répertoire. Dessinez l'arbre des sous-répertoires sur papier.
- 16 Déplacez-vous dans le répertoire `usr`. Affichez le contenu de ce répertoire. Dessinez l'arbre des sous-répertoires sur papier.
- 17 Déplacez-vous dans le répertoire `/usr/local/bin` depuis votre répertoire de travail `usr` en utilisant son nom relatif.
- 18 Déplacez-vous dans le répertoire `/etc` depuis votre répertoire de travail `/usr/local/bin` en utilisant le nom relatif du cible.
- 19 Retournez dans votre répertoire principal. Affichez le contenu du répertoire `/etc` sans vous déplacer.
- 20 Affichez les fichiers et répertoires de `/usr/bin` dont le nom
  - se termine par `"td"`
  - contient `"td"`
  - se termine par `"td"` ou `"cd"`
  - commence par `"a"`, `"b"` ou `"c"` et se termine par un chiffre.
- 21 Listez le contenu du répertoire `/dev`.
- 22 Trouvez l'emplacement du fichier `stdio.h` dans le système de fichier.
- 23 Combien de lignes, de mots et de caractères comporte le fichier `/usr/include/string.h`?
- 24 Vérifiez l'espace disque libre que vous possédez.

## 3 Les utilisateurs

Les utilisateurs du système sont identifiés par un numéro unique appelé *user ID* (UID). De plus, chaque utilisateur peut appartenir à un ou plusieurs groupes, identifiés par un *group ID* (GID). Chaque fichier du système *appartient* à un utilisateur et à un groupe enregistrés. Il existe un utilisateur spécial appelé `root` qui permet d'administrer le système (cet utilisateur a toutes les permissions sur le système).

Quelques commandes utiles :

**id** donne l'UID d'un utilisateur ainsi que les groupes auxquels il appartient;

**chown** (*change owner*) permet de changer le propriétaire d'un fichier (seul `root` peut faire);

**chgrp** (*change group*) permet de changer l'appartenance d'un fichier à un groupe (le propriétaire peut le faire);

**usermod** permet à l'administrateur de changer les paramètres d'un utilisateur).

---

1. Attention : ces méta-caractères sont interprétés dans les paramètres des commandes et par le shell

## Exercices

- 25 Affichez le contenu du fichier `/etc/passwd`.
- 26 Consultez les pages du manuel concernant ce fichier en faisant `man 5 passwd`. Quelle est la différence avec `man passwd`?
- 27 Affichez le contenu du fichier `/etc/shadow`.
- 28 Consultez les pages de manuel concernant le fichier `/etc/shadow`.
- 29 Cherchez les données de votre compte d'utilisateur défini dans le fichier `/etc/passwd`.
- 30 Quel est le répertoire de connexion et le shell de l'utilisateur `root`?
- 31 Quels sont les utilisateurs définis dans `/etc/passwd` qui font partie du même groupe que l'administrateur?

## 4 Droits d'accès

### Exercices

- 32 Regardez les droits pour les fichiers dont le nom commence par "pa" dans `/etc`. Vous est-il possible de faire une copie du fichier `/etc/passwd`? Vous est-il possible de supprimer ou de modifier ce fichier?
- 33 À l'aide de la commande `id`, vérifiez votre identité et les groupes auxquels vous appartenez.
- 34 Allez dans votre répertoire personnel.  
Créez un fichier texte `exemple.txt` de contenu quelconque, qui soit lisible par tout le monde, mais non modifiable (même pas par vous).
- 35 Créez un répertoire nommé "prive", dont le contenu n'est visible que par vous-même.
- 36 Vérifiez l'accès au répertoire `prive` de quelques membres de votre groupe.
- 37 Créez un répertoire nommé "divers" tel que les autres utilisateurs ne puissent pas lister son contenu mais puissent lire les fichiers qui y sont placés. Par exemple, par un utilisateur :

```
$ ls divers
ls: divers: Permission denied
$ cat divers/toto.txt
ceci est le contenu
du fichier toto.txt dans divers
```
- 38 Cherchez dans le répertoire `/usr/bin` les droits sur le fichier de la commande `passwd`. Il y a une permission SUID. De quelle genre de commande s'agit-il? Comment elle est exécutée?
- 39 Cherchez dans le répertoire `/usr/bin` des commandes similaires. Indice : la commande `grep` et des expressions régulières peuvent aider...

## 5 Liens

### Exercices

- 40 Créez un fichier `salade.txt` dans le répertoire `divers`. Enregistrez un texte simple dans ce fichier.
- 41 Dans le répertoire `prive`, créez un lien dur `recette.txt` équivalent à `salade.txt`.
- 42 Listez les deux fichiers en affichant leur numéro d'inode. Que remarquez vous?
- 43 Supprimez `recette.txt`. Le fichier `salade.txt` a-t-il disparu?
- 44 Dans le répertoire `prive`, créez un lien symbolique `astuces.txt` sur `salade.txt`.
- 45 Listez les deux fichiers. Analysez la situation.
- 46 Supprimez `salade.txt`. Le fichier `astuces.txt` a-t-il disparu?
- 47 Effacez tous les fichiers créés.

## 6 Entrée, sortie, redirection

Rappel : chaque commande peut utiliser l'entrée standard (le clavier, son "nom" est 0) et la sortie standard (le terminal, son "nom" est 1) pour affichage. De plus, elle peut afficher des messages d'erreur (sur la sortie prévue pour cela avec le nom 2). Les entrées et les sorties peuvent être redirigées vers des fichiers :

**entrée** avec le caractère <

**sortie** avec le caractère >

Exemple :

```
> cat < toto.txt > titi.txt
```

Deux commandes peuvent être enchaînées par le caractère |. On dit qu'elles forment un tube ; la sortie de la première commande est redirigée vers l'entrée de la deuxième.

Exemple :

```
> ls -l | grep sept
```

### Exercices

- 48 Testez les commandes `cat /etc/passwd` et `cat < /etc/passwd` et `cat < /etc/passwd > /tmp/pwd`. Expliquez les différences.
- 49 Nous reprenons la question pour trouver l'emplacement du fichier `stdio.h` dans le système de fichier. En exécutant la commande, on ne veut pas afficher des messages d'erreur. Proposez une solution.
- 50 Affichez les fichiers de `/bin/passwd` qui possèdent tous les droits pour tous le monde.
- 51 Dans la réponse précédente, cherchez les lignes qui sont liées à un shell : les lignes qui contiennent "sh".

## 7 Éditeurs de texte en ligne de commande

Lorsqu'on administre des systèmes ou des réseaux, on est souvent amené à modifier des fichiers de configuration sur des machines distantes auxquelles on accède via `ssh` par exemple. Des éditeurs de texte accessibles directement dans la console, sans interface graphique, sont alors essentiels. Parmi ces éditeurs, les plus classiques sont `nano`, `emacs` et `vi` ou `vim`.

Voici quelques commandes pour vous familiariser avec `nano` (plus *user-friendly* que `vi`).

- 52 Ouvrez `nano` et écrivez quelques lignes.
- 53 Sauvegardez avec `Ctrl+O`, quittez `nano` (`Ctrl+X`), puis ouvrez le fichier que vous avez créé depuis la console.
- 54 Pour copier et coller une partie de texte, commencer par placer une marque avec `Alt+A`, puis sélectionnez avec la flèche `→` le texte à copier. Vous pourrez copier le texte avec `Alt+AltGr+9` (`Alt+^`). Vous pouvez ensuite coller votre sélection où vous le souhaitez : vous trouverez la combinaison de touche dans la barre du bas.
- 55 Vous pouvez couper, plutôt que coller : comment fait-on ? Vérifiez que cela fonctionne.
- 56 Ouvrez l'un de vos programmes Java. Pour aller à la fin de votre fichier, vous pouvez utiliser `Alt+\`, et pour aller au début `Alt+/`.
- 57 `Alt+G` permet d'aller à une ligne spécifique.
- 58 Que permet de faire `Alt+W` ?
- 59 Et `Alt+R` ?
- 60 Vérifiez que « page down » et « page up » sont équivalents à `Ctrl+V` et `Ctrl+Y`.