

# R1.04 – TP 1

Récupérez l'archive `tp1.tgz` et décompactlyz là obligatoirement dans votre répertoire personnel.

## Table des matières

1	Shell	1
2	Système de fichiers	3
3	Utilisateur et droits	4
4	Fichiers texte	6

## 1 Shell

Le système d'exploitation est un logiciel qui joue deux rôles principaux :

1. servir d'intermédiaire entre les programmes et les éléments matériels de l'ordinateur,
2. gérer le partage des différentes ressources.

C'est le premier programme exécuté au démarrage de la machine, et c'est le seul qui reste en permanence en exécution.

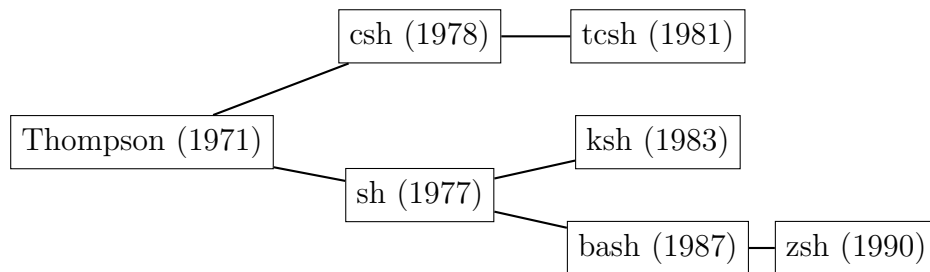
Cette partie fondamentale du système est appelée noyau (*kernel*).

Le noyau s'occupe de la gestion de la mémoire, de l'accès à la CPU, des accès aux périphériques. Il masque l'hétérogénéité du matériel, en fournissant un accès normalisé à celui-ci sous forme de services.

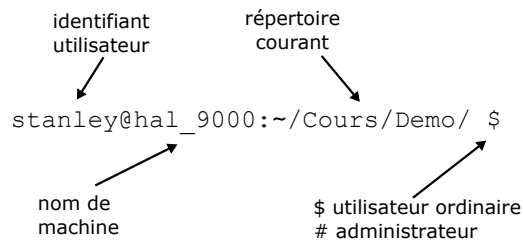
Les utilisateurs et les programmeurs peuvent demander l'accès à ces services par des **appels systèmes**, soit directement (en langage C par exemple), soit indirectement en utilisant une **commande** réalisant un tel appel.

Un interpréteur de commandes est un logiciel système faisant partie des composants de base d'un système d'exploitation. Appelé *shell* (coquille autour du noyau), sa fonction est d'interpréter les commandes qu'un utilisateur tape au clavier dans l'interface en ligne de commande.

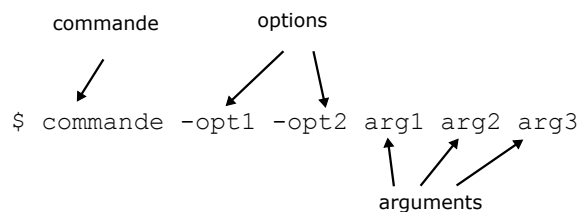
Sous unix, le shell a toujours été le moyen privilégié d'interaction avec le système. Il est apparu dès la première version. Depuis, il en existe de nombreuses variantes, dont la plus populaire est le bash (Bourne Again SHell, basé sur sh ou bourne SHell).



Le *shell* est un interpréteur de commandes. Les commandes sont saisies sur un terminal. Celui-ci affiche d'ordinaire une invite de commande ou *prompt*. Celle-ci est souvent composée d'informations de contexte et finie généralement par un caractère indiquant le statut de l'utilisateur \$ (pour un utilisateur ordinaire) ou # pour l'administrateur du système (super-utilisateur nommé *root*).



Les commandes saisies prennent la forme de plusieurs mots séparés par des espaces (leur nombre importe peu). Le premier mot est la commande elle-même. Les autres sont les arguments. Parmi ces arguments, certains sont des options de la commande. Ces options sont de la forme `-o` (forme courte avec `o` désignant l'option) `--option` (forme longue).



Exemple :

```

$ date -u
mer. 20 oct. 2021 08 :41 :11 UTC
$ date --universal
mer. 20 oct. 2021 08 :41 :19 UTC
  
```

Il existe principalement quatre moyens pour obtenir une aide :

1. aide interne à la commande : erreur dans la commande ou utilisation d'une option : `-h` ou `--help`

2. commande `man` (*MANual*) pour une aide plus structurée : nom/synopsis/description/...
3. commande `help` pour une description des commandes internes au bash
4. `apropos` pour trouver une commande dont on ignore le nom

**Question.** Consulter l'aide interne de `date`. Qu'est-ce-que Epoch ?

**Question.** Consulter la page de man de `date`. Afficher la date du jour sur le modèle suivant : 25 décembre 2021

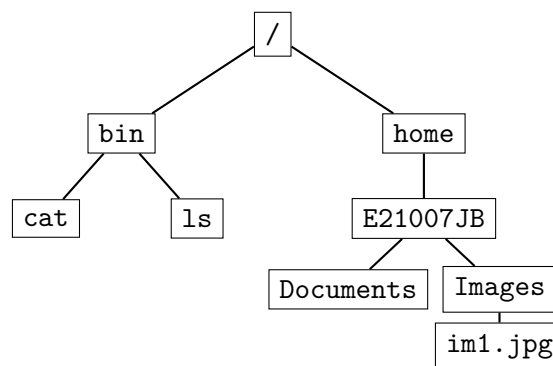
**Question.** Consulter l'aide de la commande `type`. `type` est-elle une commande interne ?

**Question.** Trouver la commande permettant d'afficher un calendrier (*calendar* en anglais). Afficher la date de Pâques.

## 2 Système de fichiers

Les fichiers et les dossiers (ou répertoires) sont organisés sous unix selon une arborescence : la racine (/) est un dossier qui contient tout le reste, les nœuds sont les autres dossiers (qui contiennent d'autres dossiers ou des fichiers), les feuilles sont des fichiers ou des dossiers (vides). La figure montre un exemple d'organisation de dossiers et de fichiers. Cette organisation des fichiers et des dossiers ressemble à ce qui existe sous Windows. La principale différence est qu'ici tous les éléments (même ceux situés sur une clé usb par exemple) sont organisés au sein d'une unique arborescence (on ne retrouve pas les C :, D :, etc de Windows).

Un chemin d'accès ou *path* est une expression pour désigner une ressource (fichier ou répertoire) dans l'arborescence. Il est constitué d'une liste de noms identifiant les branches du chemin séparés par des /.



On distingue deux types de chemins :

- chemin **absolu** qui commence par / (ou ~ qui désigne de manière absolue le répertoire de travail d'un utilisateur). Exemple : `/usr/local/bin/npm` ou `/home/E21007JB/Images/Im1.jpg`
- chemin **relatif** qui ne commence **pas** par / et part du répertoire courant. On peut utiliser les symboles suivants : `..` répertoire parent, `.` répertoire courant. Exemple : `../iut/un.c` ou `Im1.jpg`

Commandes de base de navigation et de modification du système de fichiers :

- `pwd` (*Print Working Directory*) affiche le chemin absolu du répertoire courant

- `cd` (*Change Directory*) change le répertoire courant
- `ls` (*LiSt directory contents*) liste le contenu du répertoire courant
- `mkdir` *MaKe DIRectories* crée un répertoire
- `rmdir` *ReMove DIRectories* supprime un répertoire (obligatoirement vide)
- `rm` (*ReMove*) supprime un objet du système de fichier (fichier ou répertoire)
- `touch` crée un fichier vide (ou mets à jour l'horodatage du fichier).
- `mv` (*Move*) déplace (ou renomme) un objet du système de fichier (fichier ou répertoire)
- `cp` (*CoPy*) copie un objet du système de fichier (fichier ou répertoire)

**Question.** Consultez les pages de manuel des différentes commandes données.

**Question.** Exécutez la commande `cd` sans argument. Quel est votre répertoire courant ?

**Question.** Exécutez la commande `cd Tp1` pour changer de répertoire courant.

**Question.** À l'aide de la commande `pwd`, vérifiez que le répertoire courant est `/home/vous/Tp1`.

**Question.** À l'aide de la commande `ls`, listez le contenu du répertoire courant (qui doit être `/home/vous/Tp1`).

**Question.** Exécutez la commande `cd` sans argument.

Parmi les commandes suivantes, laquelle ne vous donne pas le contenu de votre répertoire personnel ?

```
$ ls vous
$ ls ~
$ ls ~vous
$ ls /home/vous
```

**Question.** Positionnez le répertoire courant dans le répertoire `/home/vous/Tp1`. Créez un répertoire `Trav`.

**Question.** Sans vous déplacer, créez un répertoire `Un` dans le répertoire `Trav`. Positionnez le répertoire courant dans le nouveau répertoire (`Un`). Créez le fichier (vide) `unTexte.txt` dans le répertoire courant.

**Question.** À partir du répertoire courant `/home/vous/Tp1/Trav/Un`, on souhaite ajouter un répertoire `Deux` dans le répertoire `Trav`. Quelle commande ne va pas fonctionner ?

```
$ mkdir ../Deux
$ mkdir Deux
$ mkdir ~/Tp1/Trav/Deux
$ mkdir /home/vous/Tp1/Trav/Deux
```

**Question.** Depuis le répertoire `/home/vous/Tp1/Trav/Deux`, copiez le fichier `unTexte.txt` dans le répertoire `Un`. Changez le nom du fichier `unTexte.txt` en `leTexte.txt` dans le répertoire `Un`.

**Question.** En relatif, positionnez vous dans le répertoire `~/Tp1/Trav`. Supprimez le répertoire `Deux` ainsi que le fichier qu'il contient.

### 3 Utilisateur et droits

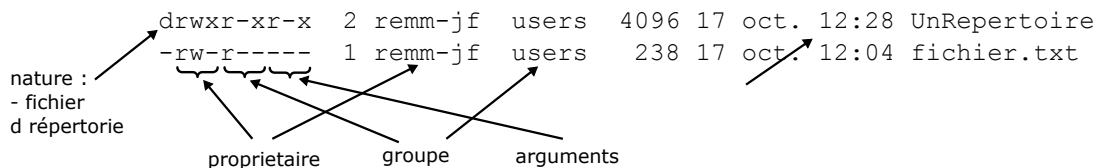
les système unix sont des systèmes multi-utilisateurs. Un utilisateur est donc identifié et se voit attribuer des droits. Il existe deux types d'utilisateurs :

1. un super-utilisateur nommé `root`, qui peut créer des utilisateur, accorder des droits, ...
2. les autres,

Les utilisateurs possèdent un `login`, un identifiant `uid`, un groupe principal `gid` et éventuellement des groupes secondaires.

**Question.** À l'aide de la commande `id`, retrouver votre `uid`, votre `gid` et votre répertoire personnel. Appartenez-vous à des groupes secondaires ?

À un objet du système de fichier sont associés des droits. Ces droits s'appliquent à trois publics différents : le propriétaire, les membres de son groupe et les autres. Pour chacun de ces public est associé un triplet de permission. Pour un fichier, le premier caractère indique l'autorisation (`r`) ou l'interdiction de lecture (`-`), le second l'autorisation (`w`) ou l'interdiction d'écriture (`-`) et le troisième l'autorisation (`x`) ou l'interdiction d'exécution (`-`). Pour un répertoire, le premier caractère indique le listage, le second la modification du contenu de ce répertoire et le troisième le positionnement (par `cd`) dans ce répertoire.



`chmod` permet au propriétaire d'un objet du système de fichier d'en modifier les droits. `chmod` peut s'utiliser de deux manières différentes :

1. symbolique : sous la forme `chmod qui op droits` où :
  - **qui** : `u` *user*, `g` *group*, `o` *other* les autre, `a` *all* tous
  - **op** : `+` ajoute, `-` supprime, `=` positionne
  - **droits** : une combinaison de `rwX`

Exemple :

```
$ ls -l fichier.txt
-rw-r--r-- 1 remm-jf iut 16 15 oct. 10 :50 fichier.txt
$ chmod u+x,g=rx,o-r fichier.txt
$ ls -l fichier.txt
-rwxr-x--- 1 remm-jf iut 16 15 oct. 10 :50 fichier.txt
```

2. numérique : 3 fois la combinaison des droits en considérant que `r=6`, `w=4`, `x=1`. Par exemple

```
$ ls -l fichier.txt
-rw-r--r-- 1 remm-jf iut 16 15 oct. 10 :50 fichier.txt
$ chmod 750 fichier.txt
$ ls -l fichier.txt
-rwxr-x--- 1 remm-jf iut 16 15 oct. 10 :50 fichier.txt
```

Deux commande **chown** et **chgrp** permettent de changer le propriétaire ou le groupe d'un fichier. Elle ne sont accessible que par root, sauf s'il s'agit de modifier le groupe parmi ses groupes d'appartenance.

**Question.** Consultez la page de manuel de **chmod**, **chown** et **chgrp**

**Question.** Positionnez vous dans le répertoire **Tp1/Trois**. Créez un répertoire **Dossier**. Quels sont les droits de ce dossier ?

**Question.** Changez de manière symbolique les droits du fichier **illisible.txt** de manière à ce que tout le monde puisse le visualiser. Vérifiez, en le visualisant (à l'aide d'un éditeur de texte).

**Question.** Changez de manière symbolique les droits du fichier **cmd.sh** de manière à ce que vous puissiez l'exécuter. Vérifiez, en l'exécutant.

**Question.** Changez de manière symbolique les droits du répertoire **Impenetrable** de manière à ce que vous puissiez y accéder. Vérifiez, en vous y positionnant.

**Question.** Changez de manière numérique les droits du fichier **illisible2.txt** de manière à ce que tout le monde puisse le visualiser. Vérifiez, en le visualisant (à l'aide d'un éditeur de texte).

**Question.** Changez de manière numérique les droits du fichier **cmd2.sh** de manière à ce que vous puissiez l'exécuter. Vérifiez, en l'exécutant.

**Question.** Changez de manière numérique les droits du répertoire **Impenetrable2** de manière à ce que vous puissiez y accéder. Vérifiez, en vous y positionnant.

**Question.** Le répertoire **PourTous** doit permettre à tous les étudiants d'info (1 et 2 mais pas aux gea) d'y placer un fichier. Comment vous y prenez-vous ?

## 4 Fichiers texte

Un fichier ordinaire est une série d'octets enregistrés sur un support de stockage et identifié par un chemin dans l'arborescence. Quand un fichier ne contient que des caractères imprimables, il est qualifié de fichier texte, par opposition à fichier binaire. Sa manipulation est alors très facile depuis le shell.

- consultation de tout ou partie du fichier : **cat** concaténer des fichiers et les afficher sur la sortie standard, **tail** afficher la dernière partie de fichiers et **head** afficher le début de fichiers
- navigation : **more** et **less** qui permettent de paginer l'affichage et de rechercher des motifs.
- édition : **nano**, **vi(m)**, **emacs**, ...

Des fichiers csv ont été placés dans le répertoire **Quatre**.

**Question.** Visualisez le fichier **personne10.csv** à l'aide de la commande **cat**

**Question.** À l'aide de la commande **cat**, affichez le fichier **personne.csv**, en numérotant les lignes.

**Question.** Visualisez les 2 premières lignes du fichier **personne10.csv** à l'aide de la commande **head**.

**Question.** Visualisez les 2 dernières lignes du fichier **personne10.csv** à l'aide de la commande

`tail`.

**Question.** Visualisez les dernières lignes (toutes à l'exception de la première) du fichier `personne10.csv` à aide de la commande `head`.

`less` (à préférer à `more`, bien plus limité) comporte quelques commandes internes :

- `h` : pour obtenir l'aide
- `/` recherche un motif (`n` suivant, `N` précédent)
- `&` n'affiche que les lignes contenant le motif
- ...

**Question.** À l'aide de `less` trouver (= compter) le nombre d'occurrences de `Gabrielle` dans le fichier `personne.csv`