

Introduction à Linux (IM713)

TP 1 – Prise en main de l’environnement Linux

Le but de ce premier TP est de commencer à vous familiariser avec l’environnement Linux ainsi qu’avec le système des droits d’accès aux fichiers.

Se connecter est évidemment la première chose à faire. Le système démarre et affiche une fenêtre vous invitant à taper votre identifiant (login) et votre mot de passe (password), ce qui vous permet d’ouvrir une session. Une fois votre nom d’utilisateur et mot de passe vérifiés, le gestionnaire de bureau, le programme qui gère l’affichage des menus et des fenêtres (Gnome ou KDE) apparaît. Le menu principal permet de lancer des applications, d’accéder aux outils de paramétrage du système ou encore de fermer la session ou d’éteindre l’ordinateur.

1 Linux

1.1 Le terminal, la ligne de commande et le shell

Le terminal fait tourner un interpréteur de commande, en particulier bash qui est le plus répandu sur les systèmes Linux. Cet interpréteur exécute le programme suivant :

- afficher l’invite de commande ou prompt
- donner le contrôle du clavier à l’utilisateur, jusqu’à ce qu’il presse la touche Entrée
- analyser le texte fourni par l’utilisateur et le transformer en une commande à exécuter par le système
- demander au système l’exécution de la commande
- afficher le résultat de la commande

1.2 Système de fichiers

Linux travaille sur un système arborescent de fichier dont la racine est /, le / étant le séparateur qui permet de marquer les niveaux hiérarchiques.

Cette arborescence est virtuelle ou logique, car elle ne correspond pas à une hiérarchie physique. Certains de ses éléments sont montés à une adresse précise, grâce aux informations du fichier /etc/fstab.

1.3 Raccourcis clavier

Certaines combinaisons de touches produisent un effet particulier dans l’interpréteur de commandes :

- ^A (ou touche Orig/Home) : curseur en début de ligne
- ^C : interrompt la commande en cours
- ^D : termine le shell (équivalent à exit)
- ^E (ou touche End) : curseur en fin de ligne
- tab : complétion de la commande en cours d’édition
- ^L : efface le terminal
- ↑, ↓ : commande précédente, suivante
- ^R : recherche dans l’historique des commandes
- ^S : gèle l’affichage

- `^Q` : dégèle l'affichage
- `^Z` : stoppe la commande courante. `bg` (background) envoie cette tâche en arrière plan, `fg` (foreground) la remet au premier plan
- `shift-PgUp/PgDown` : scroller dans la fenêtre du terminal

Noter en particulier le fameux `^S` qui gèle l'affichage. Cela peut vous arriver de saisir malencontreusement cette combinaison de touches et de constater que le terminal est inactif : il faut saisir `^Q` pour le réactiver.

La touche de tabulation est la plus utile dans cette liste. L'appui sur cette touche en cours de saisie d'un nom de commande ou de fichier déclenche la complétion par l'interpréteur : s'il n'y a qu'une complétion possible, l'interpréteur la remplit, sinon il n'affiche rien et un second appui sur cette touche affiche la liste des possibilités. Cette touche, outre de faire gagner du temps, évite de nombreuses erreurs, en particulier sur les chemins de fichier. Noter cependant que la complétion dépend de la commande lancée et du type de fichier attendu : on n'obtiendra pas de réponse en complétion pour ouvrir un fichier `.txt` avec `firefox` par exemple.

Exercice 1

1. Explorez un peu le menu principal. Lancez quelques applications, et exercez-vous à agrandir, fermer et déplacer les fenêtres. Parcourez l'ensemble des menus disponibles et essayez de deviner à quoi sert chaque fonction.

Avec le bouton droit de la souris, vous pouvez faire apparaître différentes commandes selon l'endroit cliqué (on parle de menus « contextuels »).

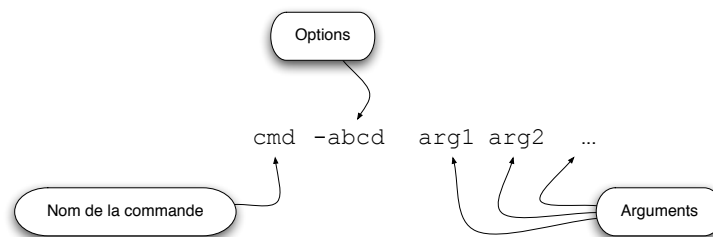
2. Une fois que vous êtes connectés, vous allez devoir lancer votre nouvelle application préférée : le terminal. Allez dans le menu Applications, puis dans le sous-menu Accessoires et vous y trouverez l'application Terminal.

- Dans un *terminal* (ou *console*) on va taper des commandes *shell* (qui ne sont rien d'autres que des programmes). *shell*.

- Le shell se contente d'attendre que vous lui demandiez d'exécuter une commande pour le faire. Vous lui indiquez que vous avez fini de taper une commande en appuyant sur la touche Entrée (`Enter`).

- Le shell vous indique qu'il est prêt en affichant une invite (ou *prompt*), en général le caractère dollar (`$`) ou supérieur (`>`), en début de ligne.

- Pour affiner le comportement d'une commande, on peut lui adjoindre ce que l'on appelle des *options*, représentées par un tiret suivie d'une ou plusieurs lettre(s). On peut également utiliser certaines commandes avec des *arguments*. La forme générale d'une ligne de commande est souvent (mais pas toujours) la suivante :



Les premières commandes

Commençons par la commande la plus utile :

`man` : Le manuel en ligne pour toutes les commandes accessibles depuis le terminal. Il suffit de taper `man cmd` pour accéder à la description complète de la commande `cmd`.

Les pages du manuel sont très complètes, et détaillent chacune des options disponibles pour une commande donnée. Vous pouvez faire défiler le texte avec les flèches haut et bas, ou les touches *page up* et *page down*. Pour sortir, pressez la touche *q*.

`whoami` : Une commande qui vous renvoie votre identifiant.

`ps` : affiche des informations sur les processus actifs dans le système.

`who` : Permet de connaître la liste de personnes connectées sur la machine.

`cat`, `less`, `more` : La commande `cat` monfichier permet d'afficher en une seule fois le contenu entier du fichier texte monfichier . Pour des textes plus longs, on utilise `less` ou `more` de manière à avoir le texte page par page.

`touch` : La commande `touch` monfichier permet de créer un fichier vide. `touch` peut être utilisé aussi pour changer les informations de date des fichiers/dossiers existants.

`pwd` : Utilisée sans argument, vous indique dans quel répertoire vous vous trouvez actuellement (répertoire courant, ou *working directory*).

`cd` : La commande `cd rep` change le répertoire courant en le répertoire `rep`. Elle permet de se déplacer dans l'arborescence. Utilisée seule ou avec l'argument "`~`", elle vous ramène dans votre répertoire personnel. Utilisée avec l'argument "`..`", elle vous fait remonter d'un niveau dans l'arborescence (au répertoire parent).

`ls` : Affiche le contenu du répertoire courant : ses fichiers et ses sous-répertoires.

`mkdir` : La commande `mkdir rep` crée un répertoire vide `rep`. Si un répertoire de même nom existe déjà à cet emplacement, vous obtiendrez bien sûr une erreur.

`cp` : La commande `cp fic1 fic2` crée un nouveau fichier identique à `fic1` appelé `fic2`. On peut aussi écrire `cp fic1 . . . ficn dest` qui copie les fichiers `fic1` à `ficn` dans le répertoire `dest` .

`mv` : La commande `mv fic1 fic2` fonctionne de manière similaire à `cp`, mais elle déplace les fichiers concernés plutôt que de simplement les recopier. Cette commande peut servir pour renommer un fichier ou un repertoire.

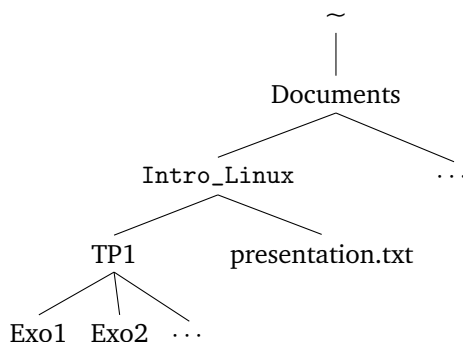
`rm` : La commande `rm fic` efface le fichier `fic` .

`rmdir` : La commande `rmdir rep` efface le répertoire `rep` (même remarque que précédemment, le chemin `rep` est supprimé de l'arborescence).

Testez chacune des commandes ci-dessus et observez les informations affichées.

Exercice 2

1. Tapez `pwd`. Vous voyez s'afficher le nom complet du répertoire dans lequel vous vous trouvez, lequel est votre répertoire personnel.
2. Tapez `cd Documents`, puis `pwd`. Vous êtes maintenant dans le répertoire Documents
3. Tapez `cd` puis `pwd` : dans quel répertoire êtes-vous ?
4. Tapez `cd -` puis `pwd` : dans quel répertoire êtes-vous ?
5. Retournez dans le répertoire `~/Documents`.
6. Tapez `cd ..`, puis `pwd` : dans quel répertoire êtes-vous ?
7. Tapez `cd ..`, puis `pwd` : dans quel répertoire êtes-vous ?
8. Retournez dans le répertoire `~/Documents`.
9. à l'aide de `mkdir`, créer un répertoire `~/Documents/Intro_Linux/TP1`. Vous pouvez procéder en trois temps : depuis le répertoire `~/Documents`, créer un répertoire `Intro_Linux`, s'y rendre à l'aide de `cd` puis y créer un répertoire `TP1`. Vous pouvez aussi procéder en un seul temps, en effectuant `mkdir -p /Documents/Intro_Linux` : vérifiez avec `man mkdir` quel est le rôle de l'option `-p` de `mkdir`
10. Créez un répertoire `Intro_Linux` dans votre *répertoire maison* (aussi appelé *répertoire personnel* ou *répertoire de login*), dans le quel vous allez créer un sous-répertoire `TP1`. Notez que l'on n'utilise ni le caractère " ", ni les accents.
11. Quelle commande pouvez-vous utiliser pour vérifier que le répertoire a bien été créé ?
12. Placez-vous dedans (grâce à la commande `cd`). Comme le nom est un peu long à taper, utilisez la touche `tab` pour le compléter. Créez un fichier `presentation.txt` et rédigez un (petit) texte vous présentant (vos nom, prénom, votre cursus universitaire ...). Sauvegardez le fichier dans le répertoire `Intro_Linux`.
13. En utilisant la commande `cat`, affichez de deux façons différentes le contenu de votre fichier `presentation.txt` (c'est-à-dire avec le chemin absolu du fichier, puis avec un de ses chemins relatifs).
 - Nom absolu du fichier : on part de la racine de l'arborescence
 - Nom relatif : on donne le chemin par rapport à l'endroit de l'arborescence où on se situe
14. À partir de votre répertoire maison reproduisez l'arborescence suivante :



15. Créez un premier fichier `count1.txt` contenant le texte "un" à "dix".
16. Créez avec `cat` un second fichier `count2.txt` contenant le texte "vingt et un" à "trente".
17. Créez à partir de `count1.txt` et `count2.txt` un nouveau fichier `concat.txt`.
18. Afficher le contenu de `concat.txt` avec les numéros des lignes en utilisant la commande `less`

19. En utilisant les commandes (`pwd`, `cd` et `ls`) , promenez-vous un peu dans l'arborescence en affichant le contenu de quelques répertoires. Si vous êtes perdu(e), tapez simplement `cd` pour rentrer chez vous.
20. Vérifier votre travail avec `ls -R` depuis votre répertoire personnel. Pouvez-vous expliquer le rôle de l'option `-R`? Essayer aussi la commande `tree` .
21. Lister que les répertoires sans les fichiers en utilisant `ls`

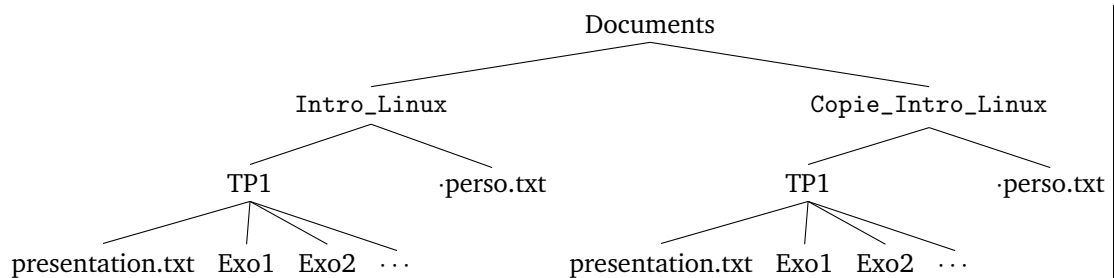
Évidemment, pour l'instant vous êtes seul sur votre machine... À noter que la commande `w` (pour *what*) est plus complète et vous dit plus précisément qui fait quoi sur la machine.

Note :

- Utilisez les flèches "haut" et "bas" pour faire défiler l'historique des commandes déjà tapées. Sous Linux, la souris peut être utilisée pour faire les copier-coller : double-cliquez sur le nom de répertoire que vous venez de faire afficher... puis cliquez avec le bouton du milieu. Le double clic sélectionne un mot (l'espace sert de séparateur) et le triple clic sélectionne toute la ligne
- La commande `history` affiche une liste des commandes récentes, chacune précédée d'un numéro. En tapant `!n` où `n` est un numéro dans l'historique, on rappelle la commande correspondante.

Exercice 3

1. Créer une copie de `presentation.txt` qui est à déplacer sur le répertoire TP1.
2. Changez le chemin de votre fichier initial `presentation.txt` en (le chemin relatif) : `~/ .perso.txt` (sans omettre le point).
3. Affichez le contenu de votre répertoire courant avec `ls`. Que constatez-vous? Réessayez en ajoutant l'option `-a` à `ls`.
4. Par défaut, le résultat de la commande `ls` est un peu aride. Une option permet de colorer l'affichage pour repérer plus rapidement le type de fichiers présents dans un répertoire. Quelle est cette option de `ls` ?
5. Dans une seule commande, créez trois répertoires imbriqués : `dir1/dir2/dir3`.
6. Par défaut `cp` ne copie que des fichiers. L'option `-r` permet de forcer `cp` à copier les répertoires et sous-répertoires de ce qui lui est donné en argument (autrement dit à copier *récurivement*). Utilisez-la pour obtenir l'arborescence suivante :



7. Quelle est l'option de `rm` permettant d'effacer *récurivement* le répertoire `Copie_Intro_Linux` (c'est à dire en descendant dans les sous-répertoires) ?
8. Par défaut, `rm` ne demande pas de confirmation lorsque vous tentez de supprimer un fichier. Ceci peut se révéler assez dangereux. Trouvez l'option qui permet de demander confirmation.

Les alias

Certaines commandes sont plutôt longues à taper, notamment lorsqu'il y a des options. Une commande permet de pallier à cet inconvénient en créant ses propres commandes : `alias`.

Utilisation : `alias ma_commande=commande_complète`. Attention, si `commande_complète` contient des espaces (par exemple, s'il y a des options), il faut alors l'encadrer avec des guillemets. `alias` permet aussi de donner une autre signification à une commande existante. On utilise en général cette caractéristique pour rajouter des options systématiquement à une commande.

À noter que la commande `alias` sans argument liste tous les raccourcis.

Note : Le problème avec la commande `alias` est qu'elle n'agit que dans le terminal dans lequel on est en train d'écrire : dès qu'on se déconnecte, tout est à refaire. Pour stocker ce genre de paramètres une bonne fois pour toutes, on utilise des *fichiers de configurations* liés au shell. Pour le shell `bash`, ces fichiers sont `.bash_profile` et `.bashrc`. Ce sont des fichiers texte contenant des commandes qui sont lues au moment de l'ouverture d'un terminal. Tout changement dans ces fichiers ne sera donc pris en compte que si on relance le shell (via l'ouverture d'un nouveau terminal par exemple).

Exercice 4

1. Créez un alias « `la` » qui liste tous les fichiers d'un répertoire (y compris les fichiers cachés commençant par un point).
2. Changez le fonctionnement de la commande `rm` pour qu'elle demande systématiquement confirmation lors d'une suppression. Changez ensuite le fonctionnement de la commande `ls` afin d'utiliser systématiquement l'affichage coloré.
3. Testez les raccourcis que vous venez de créer. Puis ouvrez un nouveau terminal et testez-les dans celui-ci. Que constatez-vous ?
4. Affichez les alias existants, puis créez un alias, `lspardir`, qui permet d'afficher le contenu du répertoire parent puis de revenir au répertoire fils.
5. Supprimez l'alias créé.
6. Éditez le fichier `.bashrc` situé dans votre répertoire personnel¹ pour rajouter les commandes des exercices précédents (`ls` avec couleur, `rm` avec confirmation et `la`). Pensez à ouvrir un nouveau terminal afin de rendre les changements effectifs. Mais faites attention à ne pas effacer de fichiers importants si vous voulez tester votre nouvelle version de `rm` !
7. Si vous utilisez `bash`, la variable `$PS1` décrit le prompt principal, celui qui s'affiche à chaque ligne de commande. On peut personnaliser ce prompt en modifiant la variable `$PS1`.
8. Rediriger un message d'un terminal vers un autre terminal (`/dev/tty???`)

Manipuler des fichiers

Il existe un certain nombre de symboles spéciaux appelés *Jokers* ou *wildcards*, qui permettent de désigner plusieurs fichiers à la fois. Il en existe plusieurs, mais les principaux sont :

- * : dans un nom de fichier, représente n'importe quelle suite de zéro, un ou plusieurs symboles
- ? : représente exactement un symbole quelconque.

Ainsi, écrire `tp?.ps` désigne `tp1.ps`, `tpi.ps`, et `truc*` désigne `truc`, `trucage`, `trucmuche` mais pas `struck`.

Exercice 5

1. ou bien créez-en un nouveau s'il n'existe pas

1. Affichez la liste de tous les fichiers dans le répertoire `/usr/bin` dont le nom commence par `k` et contient exactement 6 caractères.
2. Affichez la liste de tous les fichiers dont l'extension est `so` dans le répertoire `/usr/lib` (*note culturelle* : ces fichiers sont des bibliothèques).
3. Donner une commande qui retourne le contenu du répertoire trié
 - en ordre alphabétique inversée
 - Par ordre de taille des fichiers avec la plus grande taille répertoriée en premier,
 - Pour afficher d'abord le fichier le plus petit en utilisant l'option inverse.
 - Par date de modifications de fichiers
 - Par extension
4. Afficher les 5 dernières lignes du fichier `concat.txt` avec la commande `tail`.
5. Donner une commande qui affiche l'avant dernière ligne de ce même fichier
6. Créez un fichier `'test.txt'`
7. Créez un lien physique et un lien symbolique vers ce fichier.
8. Modifier le contenu de « lien-phy », puis observez le contenu de « Fichier » et « lien-symbo ». Que constatez-vous ?
9. Modifier le contenu de « lien-symbo », puis observez le contenu de « Fichier » et « lien-phy ». Que constatez-vous ?

`grep` : La commande `grep` permet de rechercher un motif dans un fichier. Mais on peut aussi l'utiliser comme toute commande Linux via un pipe.

Exercice 6

1. Rechercher dans le dossier `Intro_Linux` les occurrences du mot "quatre" ou "QUATRE", puis obtenir le nombre de ces occurrences. La recherche soit être récursive et insensible à la casse en incluant tous les sous-répertoires.
2. `grep` peut aussi afficher la ligne d'un fichier où se trouve l'occurrence.
3. Inverser la recherche pour exclure : chercher toutes occurrences de mots sauf "quatre".
4. Afficher 3 lignes avant (resp. après) le mot "quatre".
5. Rechercher les mots ""deux" et "quatre" en même temps.
6. Rechercher sur votre repertoire les fichiers `.txt`. Faire afficher leur nombre.
7. Quels fichiers du langage C (`.c`) contiennent des références au fichier d'en-tête `stdlib.h` ?
8. Compter le nombre d'occurrence "un" dans le fichier `concat.txt`
9. Combien y a-t'il de processus actifs sur le système ?
10. Combien d'utilisateurs sont connectés sur le système ?
11. En utilisant la commande « `sort` », afficher la liste de ces utilisateurs connectés, triée
 - (a) par ordre alphabétique.
 - (b) selon l'heure de connexion
12. Combien le répertoire `"/etc"` a t'il de fichiers repertoire ?
13. Combien le répertoire `"/etc"` a t'il de sous-répertoires ?
14. Afficher la liste des fichiers du répertoire courant, triée par ordre de taille des fichiers.

Utilisateurs, groupes et permissions

`id` : Vous connaissez déjà la commande `whoami` qui indique sous quel nom de login vous êtes connectés. La commande `id` vous indique en plus vos numéro d'utilisateur, nom et numéro de groupe principal (qui sera votre groupe propriétaire par défaut) ainsi que la liste des groupes auxquels vous appartenez.

Il y a trois types de droits :

— **Droit en lecture** :

- Identifié par la lettre `r` (`read`) et le chiffre 4.
- Donne l'accès à la lecture d'un fichier.
- Permet de lister le contenu d'un répertoire.

— **Droit en écriture** :

- Identifié par la lettre `w` (`write`) et le chiffre 2.
- Donne l'accès à l'écriture dans un fichier.
- Permet d'ajouter ou de supprimer un élément d'un répertoire.

— **Droit en exécution** :

- Identifié par la lettre `x` (`execute`) et le chiffre 1.
- Donne l'accès à l'exécution d'un fichier.
- Permet passer à travers un répertoire.

chmod : La commande `chmod liste_droits fichier(s)` permet d'accorder ou de retirer au(x) fichier(s) (ou répertoire(s)) passés en arguments les permissions correspondant à `liste_droits`. Cette liste est de la forme `droit1, ..., droitn` où chaque `droiti` peut être par exemple :

- `u+r` pour rajouter au propriétaire (`user`) le droit en lecture,
- `g-w` pour retirer aux membres du groupe (`group`) le droit en écriture,
- `o+x` pour donner aux autres utilisateurs (`other`) le droit en exécution,
- ou une combinaison de ces possibilités (ex : `ug-wx`).

Exercice 7

1. Créez un répertoire `test` et un fichier `essai` dans ce répertoire, et écrivez-y la phrase de votre choix. Notez à l'aide de `ls -l` les permissions actuelles du répertoire `test` et du fichier `essai`.
2. En utilisant la commande `chmod`, retirez-vous le droit en lecture et en écriture sur le fichier `essai`. Vérifiez l'effet obtenu en essayant d'afficher le contenu du fichier sur la fenêtre du terminal, puis de remplacer ce contenu par une phrase différente.
3. Un fichier exécutable est simplement un fichier dont vous possédez le droit en exécution. Rétablissez le droit en écriture puis remplacez à l'aide de la commande `echo` le contenu du fichier `essai` par le texte `echo "Ceci est un essai"`. Ajoutez-vous le droit en exécution, et exécutez le fichier `essai` en tapant `./essai` dans le terminal (depuis le répertoire qui le contient). Quel est le problème ?
4. Rétablissez enfin le droit en lecture et tentez à nouveau d'exécuter le fichier. Que se passe-t-il ? Que proposez vous pour obtenir un résultat plus intéressant ?
5. Placez-vous dans le répertoire `test`, et retirez-vous le droit en lecture pour ce répertoire. Listez le contenu du répertoire avec `ls`, puis exécutez ou affichez le contenu du fichier `essai`. Qu'en déduisez-vous ? Rétablissez le droit en lecture sur `test`.
6. Créez dans `test` un fichier `nouveau` ainsi qu'un répertoire `test`. Retirez au fichier `nouveau` et au répertoire `test` le droit en écriture. Tentez de modifier le fichier `nouveau`. Rétablissez ensuite le droit en écriture au répertoire `test`. Tentez de modifier le fichier `nouveau`, puis de le supprimer. Que pouvez-vous déduire de toutes ces manipulations ?

7. Positionnez vous dans votre répertoire personnel, puis retirez le droit en exécution du répertoire `test`. Tentez de créer, supprimer, ou modifier un fichier dans le répertoire `test`, de vous y déplacer, d'en lister le contenu, etc... Qu'en déduisez vous quant au sens du droit en exécution pour le répertoires ?
8. Rétablissez le droit en exécution du répertoire `test`. Positionnez vous dans ce répertoire et retirez lui à nouveau le droit d'exécution. Essayez de créer, supprimer et modifier un fichier dans le répertoire `test`, de vous déplacer dans `rep`, de lister son contenu Qu'en concluez-vous quant à l'influence des droits que l'on possède sur le répertoire courant ?
9. Changer les permissions récursivement sur votre répertoire TP1

Archivage

Il est souvent nécessaire de rassembler plusieurs fichiers, voire une arborescence, en un seul fichier. C'est le rôle de la commande `tar` (tape archive).

Une commande `tar` se déroule comme suit :

```
$ tar <sous-commande> <argument1> [<argument2> ...]
```

Les sous-commandes de `tar` sont (et ne sont pas précédées d'un tiret) :

`c` : création de l'archive

`x` : extraction de l'archive

`f` : indique qu'un nom de fichier d'archive est utilisé (c'est quasiment toujours le cas)

`v` : activer le mode bavard (verbose) qui fournit des informations sur le processus d'archivage

`z` : activer la compression

`t` : afficher le contenu de l'archive

Les conventions veulent qu'un fichier d'archive ait l'extension `.tar`, et `.tar.gz` ou `.tgz` lorsqu'il est compressé.

Exercice 8

Créez une archive que vous appellerez TP1. `tar` contenant votre répertoire TP1.

Déconnexion

À la fin de la séance, vous devrez absolument vous déconnecter (quitter votre compte). Vous êtes responsable de votre compte, si vous le laissez ouvert et que quelqu'un d'autre s'en sert (et efface tout ou envoie des mails en votre nom, par exemple), cela sera votre entière responsabilité. Pour vous déconnecter proprement (vous ne devez pas éteindre votre machine en appuyant sur le bouton de mise en route), allez dans le menu Système et choisissez "Fermer la session" ou "Eteindre" si vous êtes en fin de journée.