


## Introduction aux systèmes d'exploitation Math-Info

### TP n° 3 : environnement, i-nœuds et droits

Comme d'habitude, il vous sera demandé de déposer un fichier sur Moodle en fin de TP, dans les documents du cours IS1. Pour cela, vous copierez dans un fichier appelé `reponses_TP3.txt` les commandes utilisées pour répondre aux questions marquées par le symbole .


#### Personnaliser son environnement

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

« alias »

- avec un argument de la forme `ma_commande=commande_complète`, définit (ou redéfinit) la commande «`ma_commande`». Si `commande_complète` contient des espaces (par exemple, s'il y a des options), il faut alors la délimiter avec des guillemets.
- sans argument, liste tous les raccourcis qui ont été définis.

#### Exercice 1 – définir ses propres commandes et changer une commande

1.  Créez une commande «`ll`» qui invoque «`ls`» avec l'option `-l`. Testez-la.
2. Fermez votre terminal (par exemple à l'aide de la commande «`exit`»), puis ouvrez-en un nouveau. La commande «`ll`» existe-t-elle toujours ?

Si `ma_commande` existe déjà, son comportement est redéfini et remplacé par celui décrit ; cela permet en particulier de rajouter des options à une commande.

3. Modifiez la commande «`ls`» pour qu'elle utilise systématiquement l'option `--color=auto`. Testez-la.

#### Exercice 2 – fichiers de configuration

Les commandes contenues dans le fichier «`.bashrc`» sont exécutées à chaque lancement du *shell*, ce qui permet entre autre d'avoir des alias persistants. Ouvrez à l'aide de *Emacs* le fichier `.bashrc` qui se trouve dans votre répertoire personnel, et ajoutez-y la ligne suivante :

```
alias ls="ls --color=auto"
```

Fermez votre terminal, puis ouvrez-en un nouveau. Testez la commande «`ls`».

## Groupes d'utilisateurs, droits et leurs effets

Chaque fichier (ou répertoire) est la propriété d'un utilisateur particulier. Par défaut, celui-ci appartient à l'utilisateur qui a créé le fichier. Les utilisateurs sont réunis en groupes (par exemple : le groupe des étudiants de L1 mathématiques-informatique, celui des enseignants, celui des administrateurs systèmes). Un utilisateur pouvant faire partie de plusieurs groupes, pour chaque fichier est spécifié le *groupe propriétaire* (c'est-à-dire en tant que membre de quel groupe le propriétaire détient le fichier).

« id » indique vos numéro d'utilisateur, nom et numéro de groupe principal (qui sera par défaut le groupe propriétaire de vos fichiers) ainsi que la liste des groupes auxquels vous appartenez (plus complet que « whoami », donc).

### Exercice 3

🚧 Quel est votre nom d'utilisateur ? Les groupes auxquels vous appartenez ? Même question pour le super-utilisateur root.

Les droits accordés sur un fichier sont définis en séparant les utilisateurs en trois types : le propriétaire, les membres du groupe propriétaire, et les autres. Il y a trois types de droits : le *droit en lecture*, le *droit en écriture* et le *droit en exécution*.

**l'option « -l » de « ls »** permet d'afficher au format dit *long* les méta-données de chaque fichier listé – plus précisément, le contenu de la ligne correspondante de la table des i-nœuds, à l'exception des adresses des blocs utilisés sur le disque.

Pour chaque nom de fichier, on obtient les informations suivantes :

- son type ;
- les droits accordés sur le fichier ;
- le nombre de liens vers l'i-nœud ;
- son propriétaire et son groupe propriétaire ;
- sa taille ;
- sa date de dernière modification.

### Exercice 4 – afficher les caractéristiques d'un i-nœud

1. À l'aide de « ls -l », déterminez la taille, les droits, le propriétaire et le groupe propriétaire de votre fichier .bashrc. Même question pour le fichier « /usr/bin/emacs ».
2. 🚧 Même question pour votre répertoire « Cours », pour le répertoire « /usr/bin » et pour le répertoire « /tmp ». Que constatez-vous ?
3. Quand est-ce que le fichier « /usr/bin/emacs » a-t-il été modifié pour la dernière fois ?

4. Créez le répertoire `~/Cours/IS1/TP3`, puis dans ce répertoire, créez un fichier nommé `foo` à l'aide d'*Emacs* contenant la chaîne de caractères « contenu vide ». Vérifiez sa date de dernière modification. Renommez `foo` en `bar`, et vérifiez sa date de dernière modification. Copiez `bar` en `baz` et vérifiez sa date de dernière modification. Que constatez-vous ?

## Liens et i-nœuds

### Exercice 5 – liens et i-nœuds

1. ➤ À l'aide des bonnes options de « `ls` », afficher le numéro d'i-nœud du répertoire `~/Cours/IS1/TP3`, puis des fichiers qu'il contient.
2. Renommer `~/Cours/IS1/TP3/bar` en `antigone` dans le même répertoire. ➤ Qu'est devenu le numéro d'i-nœud ?
3. Copier `~/Cours/IS1/TP3/antigone` sous le nom `iphigenie` dans le même répertoire. ➤ Comparer les numéros d'inœud. Conclure.

« `ln` » (**link**) permet de créer un nouveau nom (lien) vers un i-nœud existant : si *ancien* est un nom valide et *nouveau* un nouveau nom, `ln ancien nouveau` crée un nouveau lien de nom *nouveau* vers le même i-nœud que *ancien*.

4. Créer dans `~/Cours/IS1/TP3` un lien `circe` vers le même i-nœud que `antigone`. ➤ Comparer les numéros d'inœud. Conclure.
5. Afficher le contenu des fichiers `antigone`, `iphigenie` et `circe`.

On peut remplacer le contenu d'un fichier *fic* pour y placer le texte *nouveau\_contenu* par la commande suivante :

```
echo nouveau_contenu > fic
```

6. Remplacer le contenu du fichier `iphigenie` par le texte « la guerre de Troie n'aura pas lieu » de cette manière. Afficher à nouveau le contenu des fichiers `antigone`, `iphigenie` et `circe`. ➤ Expliquer.
7. Remplacer le contenu du fichier `circe` par le texte « Ulysse ». Afficher à nouveau le contenu des fichiers `antigone`, `iphigenie` et `circe`. ➤ Expliquer.
8. Supprimer `antigone` et lister le contenu de `~/Cours/IS1/TP3/`. Expliquer. Pouvez-vous afficher le contenu de `antigone` ? De `circe` ? ➤ Comment recréer `antigone` exactement à l'identique ?

**Exercice 6 – effets des droits des fichiers ordinaires**

Depuis Moodle, télécharger le fichier `tp3.sh` dans votre répertoire `~/Cours/IS1/TP3`, puis exécuter dans ce répertoire la commande `«bash tp3.sh»`.

Déplacez-vous dans le répertoire `~/Cours/IS1/TP3/Toutou`.

1. Quel(s) droit(s) avez-vous sur le fichier `tata`? Essayer d'afficher son contenu avec `« cat »`. Essayer ensuite de le modifier grâce à un éditeur de texte (`« emacs »` par exemple), puis à l'aide de la ligne de commande : `echo "nouveau contenu" > tata`. 🐞 Qu'en concluez-vous ?
2. Quel(s) droit(s) avez-vous sur le fichier `titi`? Faire les mêmes expériences. 🐞 Qu'en concluez-vous ?
3. Quel(s) droit(s) avez-vous sur le fichier `toto`? Comparer ce fichier avec `tata`. Essayer d'utiliser `« tata »` comme une commande (ou `« ./tata »` pour être parfaitement sûr que `bash` n'interprètera pas `tata` autrement que comme *« le fichier de nom tata dans le répertoire courant »*), puis essayer avec `« toto »`. 🐞 Qu'en concluez-vous ?
4. Quel(s) droit(s) avez-vous sur le fichier `tutu`? Essayer de l'utiliser comme une commande. 🐞 Qu'en concluez-vous ?