

# Introduction aux systèmes d'exploitation (IS1)

## TP n° 3 : liens, i-nœuds et droits

Le but de ce TP est de vous familiariser avec les notions de *lien* et d'*i-nœud*, et de commencer à explorer le système des droits d'accès aux fichiers.

**Modalités de rendu** Au cours du TP, collectez dans un fichier `reponses_TP3.txt` les réponses aux questions marquées par le symbole 🚩 (et seulement à celles-ci!), ou les commandes utilisées pour y répondre (mais pas les résultats qu'elles produisent). Indiquez dans ce fichier vos nom et prénom (ou noms et prénoms si vous travaillez à plusieurs), et déposez le fichier en fin de TP sur Moodle.

### Liens et i-nœuds

Un système UNIX n'identifie pas un fichier par son nom, mais par une structure appelée *i-nœud* (*inode* en anglais pour *index node*). Cet i-nœud contient toutes les caractéristiques concernant le fichier correspondant : son type (fichier « ordinaire » ou répertoire), l'adresse du ou des emplacement(s) sur le disque où se trouve son contenu, les droits accordés, la date de dernière modification, etc. Les i-nœuds sont stockés dans une table, et identifiés dans cette table par un numéro.

Un répertoire est un fichier qui contient une liste de couples (nom, numéro d'i-nœud), comme dans le tableau ci-contre. Lorsqu'on dit qu'un fichier `fic` « se trouve » dans un répertoire `rep`, cela signifie en fait que `rep` contient un couple associant le nom `fic` au numéro d'i-nœud d'un fichier. On appelle cela un *lien* de nom `fic` vers cet i-nœud.

.	13767923
..	13766888
tp3.sh	13768284
tp3.tex	13768275

### Exercice 1 – liens et i-nœuds

Depuis Moodle, télécharger le fichier `tp3.sh` dans votre répertoire `~/Cours/2022/IS1`, puis exécuter dans ce répertoire la commande « `bash tp3.sh` ».

Dans la suite, « IS1 » désigne le répertoire `~/Cours/2022/IS1`.

1. 🚩 À l'aide d'une option de « `ls` », afficher les numéros d'i-nœud des fichiers contenus dans TP3.
2. 🚩 Renommer `IS1/TP3/grosminet` en `sylvestre` dans le même répertoire. Qu'est devenu le numéro d'i-nœud ?
3. 🚩 Copier `IS1/TP3/sylvestre` sous le nom `chat_sauvage` dans le même répertoire. Les numéros d'i-nœud des deux fichiers sont-ils les mêmes ?
4. Afficher les numéros d'i-nœud des fichiers contenus dans `IS1/TP3/Canards`. Que remarquez-vous, en particulier au sujet des liens `fifi` et `riri` ?

« 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*.

5. 🚧 Créer dans IS1/TP3/Canards un lien loulou vers le même i-nœud que fifi.
6. 🚧 Copier riri sous le nom miss\_tick dans le même répertoire, puis afficher le contenu des fichiers riri, fifi, loulou et miss\_tick. Les quatre fichiers ont-ils tous le même contenu ? Ont-ils tous le même numéro d'i-nœud ?

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

```
echo nouveau_contenu > fic
```

7. 🚧 Remplacer le contenu du fichier miss\_tick par le texte « *Je suis parfaitement déguisée.* ». Ensuite, afficher à nouveau le contenu des fichiers riri, fifi, loulou et miss\_tick.
8. Toujours de cette manière, remplacer maintenant le contenu du fichier fifi par le texte « *Hélas, le Manuel des Castors Juniors dit exactement la même chose.* » 🚧 Ensuite, afficher à nouveau le contenu des fichiers riri, fifi, loulou et miss\_tick. Expliquer.
9. Supprimer fifi et lister le contenu de IS1/TP3/Canards. Expliquer. Pouvez-vous afficher le contenu de fifi ? De riri ? De loulou ? De miss\_tick ?  
🚧 Comment recréer un lien de nom fifi avec le même contenu et numéro d'i-nœud d'origine ?

## 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 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

**Exercice 2 – connaître ses groupes**

✎ Déterminez votre (ou vos) groupe(s) d'appartenance. Faire de même pour le super-utilisateur root.

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

**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 (d, -, ...);
- les droits accordés sur le fichier (trois triplets correspondant aux droits rwx des trois types d'utilisateurs : *user, group, other*);
- le nombre de liens vers l'i-nœud;
- son propriétaire et son groupe propriétaire;
- sa taille;
- sa date de dernière modification.

**Exercice 3 – afficher les caractéristiques d'un i-nœud**

1. À l'aide de « `ls -l` », repérer les symboles décrivant les droits, le propriétaire et la taille de chaque fichier de votre répertoire IS1/TP3.

✎ Quel est la taille du plus gros fichier *ordinaire* de ce répertoire? Copier dans la réponse aussi la commande utilisée.

2. Copier le fichier `programme_mystere` qui se trouve à la racine de l'arborescence personnelle de l'utilisateur `is1p`<sup>1</sup> dans votre répertoire IS1/TP3, sous le nom `programme_pirate`. Comparer le propriétaire, le groupe propriétaire et les droits des deux fichiers. Essayer d'exécuter<sup>2</sup> chacun de ces deux fichiers (en utilisant une référence valide, par exemple une référence relative commençant par « `./` » si votre répertoire courant est le répertoire contenant le fichier).

✎ Que constatez-vous? Comment l'expliquez-vous?

3. ✎ Déterminer combien de liens pointent vers l'i-nœud du fichier IS1/TP3/Canards/fifi. Comment cela s'explique-t-il?

4. ✎ Déterminer combien de liens pointent vers l'i-nœud du répertoire IS1/TP3/Canards/CoffreDePicsou. Comment cela s'explique-t-il?

1. c'est-à-dire le fichier de référence `~is1p/programme_mystere`

2. Pas de panique, il ne va rien se passer de grave!

**Exercice 4 – effets des droits des fichiers ordinaires**

Déplacez-vous dans le répertoire 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 le contenu de ce fichier avec celui de `tata`. Utilisez « `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 ?