

# Introduction aux systèmes d'exploitation (IS1)

## TP 2 – Utilisateurs, groupes et permissions

Dans ce TP, vous vous familiariserez avec le système des droits d'accès aux fichiers. Avant de commencer à travailler, nous vous suggérons de taper dans un terminal la commande suivante (dont la syntaxe sera expliquée dans un prochain TP) :

```
export PS1='\w$ '
```

pour afficher en permanence votre répertoire courant.

### Groupes d'utilisateurs

`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 ainsi que la liste des groupes auxquels vous appartenez.

**Exercice 1** Déterminez votre ou vos groupes d'appartenance.

Chaque fichier est attribué à un utilisateur (son *propriétaire*) et à un groupe. Ces informations jouent un rôle dans la définition des actions autorisées ou interdites.

**Exercice 2** Utilisez `ls -l` depuis votre répertoire personnel, et repérez les symboles décrivant les droits, le propriétaire et le groupe d'appartenance du fichier.

### Droits des fichiers

`chmod` : La commande `chmod liste_droits fichier(s)` permet d'accorder ou de retirer au(x) fichier(s) passés en arguments les permissions correspondant à *liste\_droits*. Cette liste est de la forme *droit<sub>1</sub>, ..., droit<sub>n</sub>* où chaque *droit<sub>i</sub>* peut être par exemple :

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

**Exercice 3 – Un essai.**

1. A l'aide de la commande `echo "une phrase" > fic`, vous pouvez écrire le texte *une phrase* dans le fichier *fic*<sup>1</sup>. Créez un répertoire `test`, et un fichier `essai` dans ce répertoire, et écrivez-y la phrase de votre choix.
2. Notez à l'aide de `ls -l` les permissions actuelles du répertoire `test` et du fichier `essai`.

---

<sup>1</sup>La syntaxe précise de cette commande sera expliquée en détail dans un prochain TP.

3. 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.
4. 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 ?
5. Rétablissez enfin le droit en lecture et tentez à nouveau d'exécuter le fichier.

## Droits des répertoires

Si le sens de chaque droit pour un fichier ordinaire est assez clair, il est parfois moins intuitif pour un répertoire, comme vous le verrez dans l'exercice suivant.

### Exercice 4 – *Un test.*

1. 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`.
2. Créez un fichier `nouveau` dans `test`. Retirez à ce fichier **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 constatez-vous ?
3. Depuis votre répertoire personnel, retirez enfin 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 faire la liste, etc. Qu'en déduisez-vous quant au sens du droit en exécution pour les répertoires ?

## Partager ses fichiers et répertoires

Dans certains cas, par exemple dans le cadre d'un projet à plusieurs ou de la fabrication d'une page web, il peut être intéressant de donner accès à certains de ses fichiers ou répertoires à d'autres utilisateurs. Les exercices suivants explorent deux cas d'utilisation possibles.

### Exercice 5 – *Donner accès à ses fichiers.*

1. Attribuez au fichier `essai` les droits suffisants pour qu'une autre personne de votre groupe UNIX puisse y accéder en lecture (mais pas en écriture).
2. Les fichiers des utilisateurs, et en particulier vos fichiers, sont visibles depuis tous les ordinateurs de la salle. Demandez à votre voisin(e) de tenter de lire votre fichier `essai` depuis sa machine.
3. Essayez d'accéder aux répertoires personnels d'autres étudiants et de consulter leurs fichiers.

### Exercice 6 (Difficile) – Ouvrir un répertoire d'accueil.

Avec l'aide de votre voisin(e), créez un fichier `voisin` dans votre répertoire `test` et fixez les droits nécessaires pour que :

- seul votre voisin puisse écrire dans ce fichier,
- toute personne du groupe puisse lire ce fichier,
- vous seul(e) puissiez effacer ce fichier ou créer de nouveaux fichiers dans `test`.

### Droits par défaut des nouveaux fichiers

Lorsque de nouveaux fichiers ou répertoires sont créés, des droits par défaut leur sont attribués. Ces droits sont calculés à partir d'un ensemble de droits de base en utilisant un *masque des droits par défaut des fichiers utilisateurs* (appelé `umask`).

Les droits de base pour un fichier sont les accès en lecture et écriture pour tout le monde (propriétaire, groupe et autres utilisateurs). Pour un répertoire, les droits de base comprennent en plus le droit à l'exécution.

On peut ensuite choisir de n'accorder qu'une partie de ces droits de base à l'aide de l'`umask`, dont l'accès se fait par la commande `umask`. Pour ajouter ou retirer des droits au masque, on peut utiliser la syntaxe de `chmod`. Pour visualiser le masque courant, utilisez `umask -S`. Par exemple, pour retirer aux utilisateurs ne faisant pas partie de son propre groupe l'accès en lecture aux nouveaux fichiers, on tape `umask o-r`.

### Exercice 7 – Réglage du masque

1. Définissez un `umask` très restrictif qui interdit à quiconque à part vous l'accès en lecture ou en écriture, ainsi que la traversée de vos répertoires. Testez sur un nouveau fichier et un nouveau répertoire.
2. Définissez un `umask` très permissif qui autorise tout le monde à lire vos fichiers et traverser vos répertoires, mais n'autorise que vous à écrire. Testez sur un nouveau fichier et un nouveau répertoire.
3. Définissez un `umask` équilibré qui vous autorise un accès complet et autorise un accès en lecture aux membres de votre groupe Unix. Testez sur un nouveau fichier et un nouveau répertoire.

### Aller un peu plus loin

Ne faites ces exercices que s'il vous reste du temps en fin de séance.

### Exercice 8 (Optionnel) – Écrire les droits en octal

Les commandes `chmod` et `umask` permettent aussi d'utiliser une syntaxe de la forme `chmod nnn`, où chaque `n` est un nombre entre 0 et 7 (notation *octale*). Lisez le manuel de ces deux commandes, testez cette syntaxe et décrivez son fonctionnement.

### Exercice 9 (Optionnel) – Limites du système de permissions

1. Si, dans l'exercice 6, on remplaçait le fichier `voisin` par un sous-répertoire, cela permettrait à votre voisin de vous transférer des fichiers ou d'autres sous-répertoires. Testez cette idée et demandez à votre voisin(e) de créer des fichiers dans ce répertoire.
2. Essayez à présent de supprimer le répertoire `voisin` vous-même. Que remarquez-vous ? Pouvez vous suggérer une solution ?
3. De la même manière, est-il possible de partager un fichier entre deux personnes sans le rendre accessible au reste du groupe ? Commentez.
4. Faites une recherche sur Internet au sujet du mécanisme d'ACL (*Access Control List*), et cherchez une solution aux problèmes évoqués précédemment<sup>2</sup>.

---

<sup>2</sup>Le mécanisme d'ACL n'est pas installé par défaut sur notre version de FreeBSD.