

Introduction aux systèmes d'exploitation (IS1)

TP 2 – Utilisateurs, groupes et permissions

Le but de ce TP est de vous familiariser avec le système des droits d'accès aux fichiers.

Groupes d'utilisateurs

Chaque fichier (et 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 groupe. 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). On distingue alors trois catégories d'utilisateurs pour chaque fichier : le propriétaire, le groupe propriétaire, et les autres. Et les droits du fichier sont définis pour chacune de ces catégories.

`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.

Exercice 1 Déterminez votre ou vos groupes d'appartenance.

Droits des fichiers

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.

Exercice 2 Utilisez `ls -l` depuis votre répertoire personnel, et repérez les symboles décrivant les droits, le propriétaire et le groupe d'appartenance de chaque fichier/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 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`¹. 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`.
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. Que se passe-t-il ? Que proposez vous pour obtenir un résultat plus intéressant ?

Droits des répertoires

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 dans `test` un fichier `nouveau` ainsi qu'un répertoire `ssstest`. 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 ?
3. 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 ?
4. 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 `ssrep`, de lister son contenu Qu'en concluez-vous quant à l'influence des droits que l'on possède sur le répertoire courant ?

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

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. Pensez à lui donner un chemin complet (qui convienne).
3. Essayez d'accéder aux répertoires personnels d'autres étudiants et de consulter leurs fichiers.

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

Alice et Bob sont en TP d'IS1, ils se donnent le but commun de créer, dans le répertoire `test` de Bob, le fichier `exclu` tel que :

- seul Alice puisse écrire dans `exclu` et seul Bob puisse l'effacer.
- toute personne du groupe, y compris Alice mais hormis Bob, puisse lire `exclu`,
- seul Bob puisse créer de nouveaux fichiers dans `test`

Répartissez-vous les rôles d'Alice et Bob avec votre voisin(e) et essayez d'atteindre le but qu'ils se sont fixé.

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 par défaut en utilisant un *masque des droits par défaut des fichiers utilisateurs* (appelé *umask*).

Les droits par défaut pour un fichier sont : droit en lecture et exécution pour tout le monde (propriétaire, groupe et autres utilisateurs), et droit en écriture pour le propriétaire seulement.

On peut ensuite choisir de changer ces droits par défaut à l'aide de l'*umask*, dont l'accès se fait par la commande `umask` (en utilisant la même syntaxe que pour `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.

Droits écrits en octal

Les commandes `chmod` et `umask` peuvent également être utilisées avec une autre syntaxe : `chmod abc fic` où `a`, `b`, et `c` sont des chiffres compris entre 0 et 7. `a` représente la somme des chiffres représentant les droits du propriétaire. De même `b` représente la somme des droits du groupe propriétaire, et `c` représente celle des autres utilisateurs. Par exemple `rwrx-xr--` correspond à $754 = ((4 + 2 + 1)(4 + 1)(4))$.

Exercice 8 –

1. Transcrivez la commandes suivantes de la notation classique à la notation octale ou vice-versa.
 - `chmod u=r,x,g=wx,o=r fic`
 - `chmod uo+w,g-rx fic` en sachant que les droits initiaux de `fic` sont `r-r-x--`
 - `chmod 653 fic` en sachant que les droits initiaux de `fic` sont `711`
 - `chmod u+x,g=w,o-r fic` en sachant que les droits initiaux de `fic` sont `r-r-x--`
2. Remplacez la liste de commandes suivantes par une commande unique ayant le même résultat.
`chmod 653 fic`
`chmod u-r,g+w,o-r fic`

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 précédemment².

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