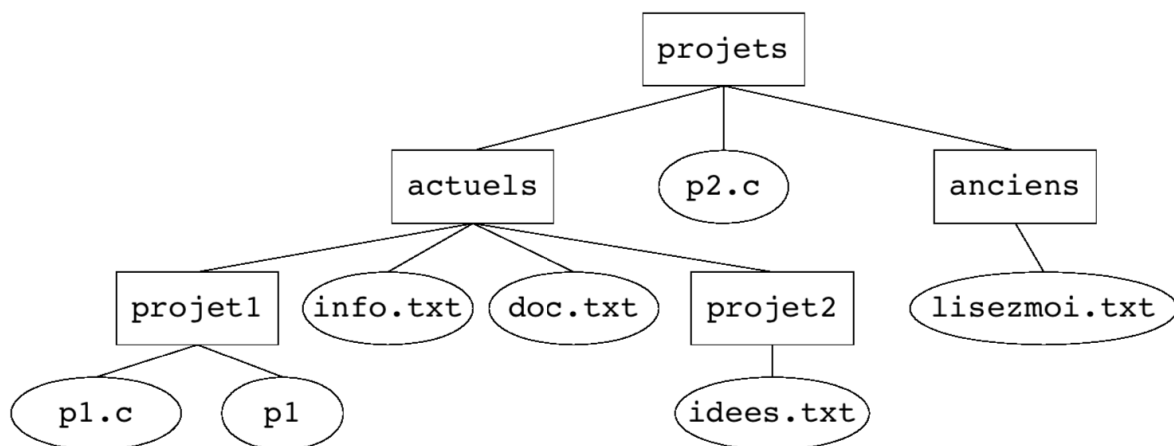


R1.04 - Systèmes - TP 10

Etape 4 : Droits Unix (suite)

Droits des commandes sur les répertoires

Soit l'arborescence de fichiers suivante :



On s'intéresse aux accès **RWX** que font les commandes sur les différents éléments.

On se situe initialement dans le répertoire projets et on ne considère pas les répertoires situés plus haut.

Pour les commandes suivantes, donner les **listes d'accès**, c'est-à-dire expliquer **quelles opérations** sont nécessaires (lire, écrire, modifier, aller dans, lister, exécuter,...etc.) **sur quelles ressources** (fichier ou dossier) puis conclure en donnant les listes de combinaisons **rwX** à positionner sur ces ressources.

Par exemple, pour faire **nano p1.c**, il faut :

- Les droits d'**éditer** (lire/modifier) **p1.c**, donc **rw** sur **p1.c**
- Le droit d'**aller dans** **projets**, **actuels** et **projet1** donc **x** sur **projets**, **actuels** et **projet1**

a) <code>ls</code>	f) <code>rm actuels/*.txt</code>
b) <code>more actuels/doc.txt</code>	g) <code>mv anciens/lisezmoi.txt actuels</code>

c) <code>cd actuels/projet1</code>	h) <code>mv actuels/projet1 anciens</code>
d) <code>ls actuels/projet1</code>	i) <code>ls -l actuels > actuels/doc.txt</code>
e) <code>cp actuels/doc.txt anciens</code>	j) <code>cc p2.c -o p2</code>

Configuration des droits

On reprend le même arbre de fichiers que précédemment. On a différents comptes et groupes d'utilisateurs. Voici les noms et les groupes des utilisateurs :

Utilisateur	Groupe
Ada	tech
Ben	tech
Carl	biz
Dora	guest

Le but de l'exercice est de réfléchir à une attribution des propriétaires, des groupes et des droits sur certains répertoires de manière à y autoriser ou interdire certaines opérations.

Dans chacune des situations suivantes, il faut proposer un utilisateur qui sera **U**, un nom de groupe qui sera **G**, proposer des droits à **U**, **G** et **O** de manière à respecter les droits envisagés.

- 1) Tout le monde peut entrer dans le répertoire **actuels** et lister les fichiers. Seul **Ben** peut créer un autre sous-répertoire.
- 2) **Ada** peut tout faire dans **projet1**, **Ben** ne peut rien y faire, **Carl** peut y aller et lister les fichiers mais pas en créer, **Dora** ne peut rien y faire.
- 3) Le répertoire **projet2** peut être listé par **Ben**, **Carl** et **Ada**. Tout le monde peut y aller, mais seul **Carl** peut effacer le fichier **infos**.
- 4) **Carl** et **Dora** peuvent aller dans **anciens** mais sans pouvoir lister de fichiers. **Ben** et **Ada** peuvent copier des fichiers dedans. Eux seuls peuvent lister les fichiers.

Propriétaires, groupes et utilisateurs

Voici quelques éléments complémentaires au modèle de base qui a été présenté jusqu'à maintenant.

setgid pour les répertoires

Nous avons vu qu'en principe vos fichiers (ceux que vous avez créés) vous appartiennent et sont de votre groupe mais ce n'est pas toujours le cas.

En effet, il y a un moyen sous Linux pour que les fichiers que crée un utilisateur ne soient pas placés dans son groupe, c'est-à-dire qu'ils ne soient pas la propriété de son groupe principal ou d'un de ses groupes secondaires.

Faites l'essai suivant :

```
cd /tmp
touch votre_login 1
ls -l votre_login
```

Cela doit vous montrer que vous êtes le propriétaire et que le fichier est de votre groupe. Maintenant faites :

```
cd /FILER/ano*/SYS2017/
```

Créez un fichier dans le dossier **tmps** :

```
touch tmps/votre_login 2
ls -l tmps/
```

Cela doit vous montrer que vous êtes le propriétaire du fichier **votre_login**, mais que le fichier n'est pas de votre groupe, il est du groupe **INFO-Enseignants** !

La différence entre ces deux fichiers est que le répertoire **tmps** fait hériter de son groupe tous les fichiers et répertoires qu'on crée dedans : ce n'est pas à votre groupe, c'est à celui de **tmps** qu'ils sont associés.

¹ A vous de remplacer **votre_login** par la bonne valeur !!! On va voir qui est réveillé et qui ne l'est pas...

² Toujours pas réveillé ?

Pourquoi ?

En étant dans **/FILER/ano*/SYS2017/**, (vous y êtes normalement), faites :

```
ls -ld tmps/
```

l'option **-d** permet de voir les caractéristiques des répertoires sans en afficher leur contenu.

Remarquez le **rw****s** pour ce dossier **tmps** et pas pour **/tmp**. Ce **s** signifie **setgid** (**set** groupe **id**) et indique de faire l'héritage du groupe : quand vous créez un fichier dans ce dossier **tmps**, il hérite du groupe du répertoire **tmps**, c'est à dire **INFO-Enseignants**.

Vérifiez que, quand vous créez un fichier dans **/tmp**, il vous appartient entièrement.

Le sticky bit

Le droit d'écriture sur un répertoire signifie que l'on peut créer et supprimer des fichiers de ce répertoire. Le sticky bit (traduit en *bit collant*) permet de faire la différence entre les deux droits.

Lorsque ce droit est positionné sur un répertoire, il apparaît sous la forme de la lettre **t** à la place du **x** : il interdit la suppression des fichiers qu'il contient à tout utilisateur autre que le propriétaire du fichier lui-même.

Faites :

```
ls -ld /tmp
```

La présence du bit **t** dans le triplet **rw****t** que vous observez pour **0** (dans **UGO**) indique que, dans **/tmp**, seul le propriétaire d'un fichier peut le supprimer.

Comparez avec la situation de **/FILER/ano*/SYS2017/tmps**

Pouvez supprimer les fichiers des autres dans ce dossier ? Pourquoi ?