TD1 - Étude des permissions et droits des systèmes Unix

Vincent Danjean

L3 M&I — Systèmes

Exercice 1 : QCM

Question n^{o}1

Comment cache-t-on un fichier dans un système Unix ?

- **a.** on positionne l'attribut `hidden`
- **b.** on fait commencer le nom du fichier par le caractère `.`
- **c.** on fait terminer le nom du fichier par le caractère `#`
- **d.** on fait terminer le nom du fichier par le caractère `~`

Question n^{o}2

Pour le système d'exploitation, un lien symbolique est un fichier

- **a.** vrai
- **b.** faux

Question n^{o}3

Un fichier a pour droits d'accès `---r-xrwx` (057). Toutes les personnes qui peuvent écrire dans ce fichier sont :

- **a.** uniquement le propriétaire
- **b.** tous les utilisateurs du système
- **c.** le propriétaire ou un membre du groupe de ce fichier
- **d.** aucune des propositions précédentes n'est juste

Question n^{o}4

```
chmod ugo-x toto.txt
```

- **a.** Permet aux utilisateurs du même groupe d'exécuter le fichier
- **b.** Enlève le droit d'exécution à tout le monde sur le fichier
- **c.** Donne le droit d'écriture à tout le monde sur le fichier
- **d.** Enlève le droit d'exécution pour l'utilisateur `ugo` sur le fichier

L'utilisateur `alice` exécute les commandes suivantes sur la machine `jpp` :

```
alice@jpp:~> ls /u/HOME/b/bob
/u/HOME/b/bob: Permission denied
alice@jpp:~> cd /u/HOME/b/bob
alice@jpp:/u/HOME/b/bob> pwd
/u/HOME/b/bob
alice@jpp:/u/HOME/b/bob>
```

Question n°5

À partir des résultats des commandes présentés ci-dessus, quels pourraient être les permissions du répertoire `/u/HOME/b/bob` ?

a. `drwx------`

b. `drwx--x--x`

c. `drwxr--r--`

d. `drwxr-xr-x`

Question n°6

Si l'utilisateur `alice` tape maintenant la commande `ls`, que voit-il apparaître ?

a. un message d'erreur

b. la liste des fichiers du répertoire courant

c. la liste des fichiers de son HOME

d. aucune des propositions précédentes n'est juste

Question n°7

Alice revient dans son HOME puis tape la commande : `ln -s /u/HOME/b/bob access`
Que se passe-t-il ?

a. la commande `ln` fait apparaître un message d'erreur

b. Alice peut désormais utiliser la commande "`ls access`" pour voir le contenu de `/u/HOME/b/bob`

c. la commande "`ls access`" renverra un message d'erreur

d. aucune des propositions précédentes n'est juste

Exercice 2 : Travail en binôme

Alice et Bob, deux étudiants veulent travailler sur leur compte Unix sur un projet commun. Pour cela, ils ont besoin d'un répertoire (`TPIUP`) où ils pourront tous les deux modifier les sources de leur logiciel. Il est strictement interdit de prêter son compte à une autre personne.

▷ **Question 1.** *Quelles sont les commandes que doit taper Alice pour créer un tel répertoire dans son HOME ?*

Bob crée un nouveau fichier nommé `solution.c` dans ce répertoire avec les droits suivants :

```
alice@jpp:~> ls -l TPIUP
[...]
-rw-r--r-- 1 bob      users 121409 2007-12-18 00:04 solution.c
alice@jpp:~>
```

Alice peut donc voir le contenu du fichier mais pas le modifier. Or, il y a une erreur dedans.

▷ **Question 2.** *Que peut faire Alice pour corriger le bogue de Bob dans le fichier avant de le compiler, sachant que Bob est parti en vacance, injoignable ? (le nom* `solution.c` *ne doit pas être changé car le* `Makefile` *fourni dans le projet fait référence à ce nom)*

Les permissions spéciales Unix

Outre les permissions classiques `rwx` pour le propriétaire du fichier, pour un membre du groupe du fichier ou les pour les autres, Unix définit quelques permissions spéciales pour certains types de fichiers.

Exécutables

Les exécutables peuvent avoir deux permissions supplémentaires :

SUID (Set UID) : quand cette permission est présente, cela signifie que, lorsque le programme est exécuté, il s'exécute avec les droits du propriétaire du fichier au lieu des droits de l'utilisateur qui le lance ;

SGID (Set GID) : quand cette permission est présente, cela signifie que, lorsque le programme est exécuté, il s'exécute avec les droits associés au groupe du fichier en plus des groupes de l'utilisateur qui le lance ;

Répertoires

Les répertoires peuvent avoir une permission supplémentaire :

sticky bit (t) : quand cette permission est présente, il faut être le propriétaire d'un fichier dans ce répertoire pour pouvoir le renommer ou l'effacer.

Exercice 3 : Questions générales

▷ **Question 1.** *Discutez des usages possibles de ces nouvelles permissions.*

▷ **Question 2.** *Discutez des implications en terme de sécurité de ces nouvelles permissions.*

Exercice 4 : Accès limité au matériel

Dans les systèmes Unix, l'accès direct au matériel est généralement possible en utilisant un des fichiers spéciaux dans le répertoire `/dev`. On peut y trouver, par exemple, les fichiers `/dev/hda`, `/dev/sda`, etc. permettant d'accéder directement [1] au matériel. Pour un disque, on préfère généralement y accéder à travers un système de fichiers, mais on peut vouloir accéder directement à un disque pour faire une sauvegarde complète du disque même si le système de fichiers est endommagé, par exemple.

La plupart des fichiers du répertoire `/dev` ont des permissions restrictives (root/root, mode 600). Mais, si nécessaire, l'administrateur peut décider de mettre d'autres permissions.

L'objectif ici est de permettre à un ensemble d'utilisateurs d'accéder à une clé USB (`/dev/sda`) pour écriture.

1. Les fichiers dans `/dev` permettent de *voir* le périphérique comme une suite d'octets. Pour un disque, par exemple, il n'y aura aucune notion de fichier ou dossier si on y accède à travers ces fichiers.

▷ **Question 1.** *Décrivez précisément ce que peut faire l'administrateur pour permettre à* `alice`*,* `bob` *et* `carine` *(et uniquement eux) d'accéder directement au dispositif.*

En réalité, accéder directement au dispositif peut être dangereux. L'administrateur souhaite rendre obligatoire le passage par un tel programme (appelé `/usr/bin/usbwrite`) afin de sécuriser le matériel.

▷ **Question 2.** *Décrivez précisément ce que peut faire l'administrateur pour permettre à n'importe quel utilisateur d'accéder au dispositif USB à l'aide du programme* `/usr/bin/usbwrite`*. Discutez des implications au niveau de la sécurité de ce que vous proposez.*

▷ **Question 3.** *Décrivez précisément ce que peut faire l'administrateur pour permettre à* `alice`*,* `bob` *et* `carine` *(et uniquement eux) d'accéder au dispositif USB avec l'aide du programme* `/usr/bin/usbwrite`*. Discutez des implications au niveau de la sécurité de ce que vous proposez.*