– TD 2 – Droits d'accès sous Unix –

Lyon 2 / ICOM / L1 informatique / Environnement Informatique et Internet

Didier Puzenat `didier.puzenat@univ-lyon2.fr`

1 Préambule

Vous êtes plusieurs à avoir demandé comment installer un environnement identique à ce que vous avez en salle de TD, sur votre machine personnelle sous Windows, afin de pouvoir refaire les fiches chez vous. Pensez à récupérer la dernière version de la fiche TD car elles sont améliorées régulièrement, directement sur la page Moodle du CM. À faire chez vous, si vous le souhaitez :

1. ouvrir l'interpréteur de commandes de Windows **en mode administrateur**, pour ce faire taper `cmd` dans la barre de recherche de Windows, cliquer avec le bouton droit de la souris sur la ligne « Invite de commandes », choisir « Exécuter en tant qu'administrateur » dans le menu qui a dû apparaître ;
2. saisir « `wsl --install -d Debian` » dans l'interpréteur de Windows ;
3. lancer « Debian » via les menus ou via la barre de recherche de Windows, comme en salle de TD, mais attention : l'icône peut ne pas être la même qu'en salle de TD ;
4. comme dans la fiche de TD 1, commencer par mettre à jour la base de données sur les paquets disponibles (update) puis mettre à jour la distribution (upgrade).

À noter que vous aurez aussi besoin d'ouvrir l'interpréteur de Windows en mode administrateur si vous avez besoin de réinitialiser la distribution comme indiqué à la fin de la fiche de TD 1 (section « Dépannage »). Enfin, si votre machine personnelle est sous GNU/Linux ou sous MacOS, il suffit d'ouvrir un terminal BASH car vous êtes déjà sous Unix. La machine virtuelle WSL est uniquement nécessaire sous Windows. Éventuellement, sur Mac, il faut préciser explicitement que vous souhaitez un interpréteur BASH selon votre version de MacOS.

2 Introduction

L'objectif du présent TD est de manipuler les droits des fichiers et répertoires sous Unix, à partir de la ligne de commandes, aussi il est nécessaire de connaître le cours sur ce sujet, tant les concepts que l'usage de la commande `chmod`. Pour pouvoir jouer un peu avec les droits, nous allons aussi voir comment créer de nouveaux comptes. À faire :

1. si nécessaire, relire le cours sur les droits sous Unix et sur l'usage de `chmod` ;
2. parcourir la page de man de la commande `chmod`.

3 Exercices sur les droits

3.1 Exercice 1

Commençons par de simples manipulations. À faire :

1. créer un répertoire `TD2` ;
2. créer – dans `TD2` – un fichier texte nommé `blabla.txt` contenant quelques mots ;
3. interdire à tout le monde de lire ou modifier le fichier, y compris son propriétaire ;
4. vérifier les nouveaux droits, par exemple avec `ls`, `cat` et `nano` ;
5. essayer d'effacer le fichier `blabla.txt` par un « `rm blabla.txt` » ;
6. vérifier si le fichier est toujours là (par exemple avec `ls` ou `cat`).

Si vous avez été surpris que le fichier `blabla.txt` ait été effacé, alors que personne n'avait le droit d'écriture (`w`) c'est peut-être que vous aviez oublié que sous Unix, la possibilité d'effacer un fichier ne dépend pas des droits du fichier mais du répertoire où se trouve le ficher. À faire :

1. créer de nouveau le fichier `blabla.txt` ;
2. modifier les droits de `TD2` afin que personne ne puisse supprimer ce qui s'y trouve ;
3. essayer d'effacer le fichier `blabla.txt`.

3.2 Exercice 2

Le but de l'exercice est de créer un répertoire nommé `rapports` à la racine de votre home, ainsi qu'un répertoire `bilans` dans le répertoire `rapports`. Il ne devra être possible que pour vous de savoir ce qui est dans le répertoire `rapports` et d'y créer des fichiers ou encore de détruire des fichiers qui s'y trouveront. Néanmoins, tout le monde devra pouvoir consulter les fichiers contenus dans `bilans` sans pour autant pouvoir supprimer ou modifier ces fichiers. À faire :

1. créer `rapports` et `bilans` sans se préoccuper des droits (commande `mkdir`) ;
2. ajuster et vérifier les droits de `rapports` ; idem pour `bilan`.

Afin de tester, nous avons besoin de plusieurs comptes utilisateurs. Si les postes de travail étaient installés sous Linux, le répertoire `/home` contenant les comptes de tous les étudiants serait sur un « serveur de fichiers » et greffé (ou « monté ») à la racine de l'arborescence de votre poste de travail via le réseau local de l'Université, de même les informations permettant d'authentifier les utilisateurs seraient partagées. Dès lors, il serait facile de tester les accès entre utilisateurs. À défaut, nous allons devoir créer des comptes locaux pour pouvoir tester les droits d'accès.

4 Création de nouveaux comptes

Il existe de nombreux outils pour créer des comptes, tant graphiques qu'en ligne de commandes, pour créer des comptes individuels ou à partir d'une liste, etc. Nous allons utiliser la commandes `adduser` qui est interactive. La commande demande divers informations, seul le mot de passe est indispensable. Souvenez-vous que **rien ne s'affiche lorsqu'on tape un mot de passe** sous Linux en ligne de commandes (c'est volontaire !). Il est également recommandé d'entrer le nom complet ou « *full name* ». À faire :

1. parcourir la page de man de `adduser` ;
2. créer un compte pour Monsieur Jean-Luc Picard, dont le login sera « jlpicard », avec la commande « `sudo adduser jlpicard` » ; laisser vide les informations non indispensables, simplement en utilisant la touche « Entrée » pour passer à la question suivante ;
3. vérifier que le home de Jean-Luc a bien été créé avec « `ls /home` » ou « `tree /home` ».

Comme vu en cours, sous Unix les fichiers de configuration sous essentiellement des fichiers textes stockés dans le répertoire `/etc`. L'usage de fichiers textes est un gage d'ouverture et de pérennité, notamment par rapport à des fichiers binaires a fortiori dans des formats propriétaires non documentés. L'usage de fichiers textes permet également une manipulation aisée depuis le BASH, d'autant qu'il existe de nombreuses commandes pour manipuler les fichiers textes. Nous allons utiliser « `cat` » qui affiche le contenu d'un fichier texte dans la console. À faire :

1. parcourir la page de man de `cat` ;
2. afficher le contenu des fichiers « `/etc/password` », « `/etc/shadow` » et « `/etc/group` ».

5 Tests à partir d'un autre compte

Nous avons déjà utilisé la commande `sudo` pour effectuer des tâches d'administration, en lançant des programmes au nom de l'administrateur – le compte `root` – mais il est possible de spécifier tout utilisateur qui a un compte. À faire :

1. chercher dans la page de man de `sudo` comment lancer une commande au nom d'un autre utilisateur ;
2. essayer de lister les fichiers contenus dans le répertoire `bilans` de l'exercice 2 en exécutant la commande `ls` au nom de l'utilisateur Jean-Luc Picard.

Si la dernière commande a échouée, vous n'avez probablement pas choisi le droits nécessaires, pour rappel il faut que `bilans` soit accessible en lecture (r) et traversable (x) pour l'utilisateur Jean-Luc Picard, mais il faut aussi que les répertoires parents – le répertoire `rapports` et votre répertoire home – soient eux-mêmes traversables (x) pour cet utilisateur.

6 Ouverture d'un interpréteur au nom d'un autre utilisateur

La suite de votre travail de TD consiste à jouer avec les droits, en créant des répertoires et fichiers, en les protégeant selon vos choix, puis en essayant d'y accéder via un autre utilisateur. Dès lors il peut être pratique d'ouvrir une fenêtre au nom d'un autre utilisateur. À faire :

1. ouvrir une seconde fenêtre WSL en choisissant « Debian » dans les menus de Windows ;
2. lancer un interpréteur BASH au nom de Jean-Luc Picard dans la nouvelle fenêtre WSL, grâce à la commande « `sudo -u jlpicard bash` » ;
3. vérifier au nom de qui chacun des deux interpréteurs s'exécute en utilisant « `whoami` » ;
4. jouer avec les commandes `ps`, `top`, `tree`, `kill` ; essayer par exemple de tuer un processus d'un autre utilisateur, par exemple un « `tree /` » pour avoir le temps, etc.
5. jouer avec `chmod`, y compris en spécifiant les droits au format octal.

Pour finir, une petite démonstration avec deux fenêtres :