

TP n° 1

Consignes les exercices ou questions marqués d'un \star devront être d'abord rédigés sur papier (afin de se préparer aux épreuves écrites du partiel et de l'examen). En particulier, il est recommandé d'être dans les mêmes conditions qu'en examen : pas de document ni de calculatrice. Tous les TPs se font sous Linux.

1 TP sous Linux avec la session de secours

En mode « dégradé » les salles de TP ne possèdent qu'un utilisateur local à la machine dont le login est « **secours** ». Le mot de passe vous sera donné en début de séance. Il convient de sauvegarder régulièrement votre travail, soit sur une clé usb, soit sur un espace en ligne.

Conseil : Les deux premiers TPs vont vous permettre de découvrir le shell Unix et de vous familiariser avec des concepts et commandes de base. Il est **vivement recommandé** de garder une trace des commandes que vous avez testées, soit sur une feuille, soit en les copiant dans un fichier texte que vous sauvegarderez. Vous pourrez ainsi réviser efficacement en ne vous contentant pas que du corrigé.

2 Commandes de base

\star On considère la séquence de commandes ci-après, rentrées les unes après les autres dans un terminal. Décrire l'effet de chaque commande (création de répertoire, changement de répertoire, affichage dans le terminal, erreur, ...). Les commandes sont décrites page 26 et suivantes du support de cours 1.

- | | |
|-------------------------------------|-------------------------------|
| 1. <code>cd ~</code> | 5. <code>cd ..</code> |
| 2. <code>mkdir IntroInfo</code> | 6. <code>ls</code> |
| 3. <code>mkdir IntroInfo/TP1</code> | 7. <code>chmod 700 TP1</code> |
| 4. <code>cd IntroInfo/TP1</code> | |

Ouvrir un terminal et effectuer les commandes ci-dessus. Constatez que tout se déroule comme prévu. **Attention** : il est possible que la toute première commande échoue si vous utilisez un poste sur lequel l'exercice a déjà été fait (les sessions de secours sont nettoyées régulièrement, mais pas à chaque déconnexion). Dans ce cas, vous commencerez par exécuter la commande `mv IntroInfo IntroInfo_old` (en vérifiant avant que le répertoire `IntroInfo_old` n'existe pas déjà et en adaptant le suffixe (`_old2`, `_old3`, ...) si c'est le cas).

3 Motifs *glob*

\star pour chacun des motifs *glob* ci-dessous, donner une suite de caractères de longueur au moins 1 reconnue par le motif.

- | | |
|--------------------------|--|
| 1. <code>*txt</code> | 5. <code>@(a.txt b.txt c.txt)</code> |
| 2. <code>+(txt)</code> | 6. <code>+([\^0-9])+([0-9])+([\^0-9]).bak</code> |
| 3. <code>[0-9]*</code> | 7. <code>????</code> |
| 4. <code>+([0-9])</code> | 8. <code>?*?</code> |

Tester les réponses dans le terminal. Pour cela, se placer dans le répertoire `TP1` créé lors de l'exercice précédent, créer des fichiers vides au moyen de la commande « `touch nomdefichier` » et tester l'expression au moyen de « `ls expression` ». Le fichier que vous venez de créer doit être listé (au moins lui). Exemple :

```
$ touch toto.txt
$ ls *txt
```

Affiche `toto.txt`.

Si les expressions de type `@(...)`, `*(...)`, `+(...)`, ou `?(...)`, ne fonctionnent pas, il faut activer le support pour les motifs globs étendus en entrant la commande :

```
$ shopt -s extglob
```

dans le terminal où sont fait les tests.

4 Permissions

★ On suppose pour cet exercice que le répertoire courant est le répertoire personnel. De plus, on suppose que les répertoires `IntroInfo` et `IntroInfo/TP1` existent (cf. exercice 1).

Donner la commande permettant de mettre les permissions demandées. On utilisera des permissions numériques et on justifiera en montrant la représentation binaire.

1. le répertoire personnel possède tous les droits pour l'utilisateur et uniquement le droit d'exécution pour le groupe et les autres
2. les répertoires `IntroInfo` et `IntroInfo/TP1` possèdent tous les droits pour l'utilisateur et les droits de lecture et d'exécution pour le groupe et les autres (une commande par répertoire)
3. le fichier `lisible.txt` du répertoire `IntroInfo/TP1` possède les droits de lecture/écriture pour l'utilisateur et uniquement les droits de lecture pour le groupe et les autres
4. le fichier `secret.txt` du répertoire `IntroInfo/TP1` possède les droits de lecture/écriture pour l'utilisateur et aucun droit pour le groupe et les autres

Mettre les permissions sur tous les fichiers et répertoires comme indiqué ci-dessus (vous pouvez créer 2 fichiers `lisible.txt` et `secret.txt` au moyen d'un éditeur de texte). Demander à un voisin (sur une autre machine) de tester les permissions de vos répertoires et fichiers (en essayant de rentrer dans les répertoires et de lire les fichiers). En particulier constater la différence entre droit en exécution et droit en lecture sur un répertoire. Enfin, supprimer les droits pour les autres et le groupe sur le répertoire personnel. Constater que plus personne n'a accès à ce répertoire à part le propriétaire.

5 Permissions 2

Le découpage des permissions entre l'utilisateur, le groupe et les autres n'est parfois pas assez fin. Par exemple, on souhaiterait laisser l'accès à un fichier à tous les utilisateurs connaissant un certain mot de passe, quel que soit leur groupe.

1. ★ rappeler brièvement ce qu'autorisent les droits en lecture et en exécution sur des *répertoires*.
2. En se basant la dessus, imaginer quel droits mettre sur votre repertoire personnel pour autoriser tous les utilisateurs connaissant « le mot de passe » `0CtHTp22` à accéder à un fichier `partage.txt`.

Remarque : en pratique cette technique n'est pas vraiment utilisée et permet d'illustrer les différences de permissions sur les répertoire. Un système moderne utilise les ACL (*access control list*), un mécanisme permettant de donner des droits différents à des ensembles arbitraires d'utilisateurs (mais ça dépasse le cadre du cours).

6 Arborescence et chemins

On suppose que l'on se trouve dans un répertoire `TEST`, que ce dernier est vide et que l'on exécute les sept commandes suivantes. Dessiner l'arborescence finale des fichiers et répertoires (on utilisera `TEST` comme racine de l'arborescence).

1. `mkdir a b c d`
2. `touch a/t.txt d/foo.txt`
3. `cd c`
4. `mkdir ../b/e f g`
5. `cd ..`
6. `cp */*.txt c/g`
7. `rm -rf d`

7 Recherche de fichiers (bonus)

Si vous avez fini votre feuille en avance, vous pouvez tenter l'exercice suivant, en lisant la fin des supports du cours 1

Se placer dans le répertoire `/usr/share/doc`. Ce dernier contient les fichiers de documentation de tous les paquets logiciels installés sur la machine. Utiliser la commande `find` pour trouver les fichiers correspondant aux critères ci-dessous. En cas de doute, consulter le cours ou la page de manuel au moyen de la commande `man find`.

1. les fichiers dont le nom est `README.txt`
2. les fichiers d'une taille supérieure à 3 Mo
3. les fichiers d'une taille inférieure à 4 Mo (attention, lire attentivement l'aide de l'option `-size` de la page de manuel)
4. les fichiers dont la taille est comprise entre 3 et 4 M
5. les répertoires dont le nom commence par `a` ou `A`
6. les fichiers étant des liens symboliques et, pour chacun de ces fichiers, appeler la commande `ls -l` sur le fichier.