Licence Informatique 1ère année — Université d'Orléans
Pratique du système Unix — Année 2024-2025

TP n°1 - Commandes Unix

Pour cette séance, il vous est demandé d'ouvrir un `terminal` et de travailler dans cette fenêtre `terminal` en écrivant des commandes UNIX classiques vues au CM n°1.

Il pourra aussi être utile, pour certaines questions (création de répertoires, de fichiers, ...) de visualiser les changements "en direct" via une interface graphique (un explorateur graphique de fichiers).

Mais l'explorateur ne devra pas être utilisé pour les manipulations. Autrement dit, pas de commandes exécutées à la souris !

Pour chaque séance de TP n°i, vous devez créer un répertoire appelé `TPi` et travailler dans ce répertoire.

Pour cette séance, à partir de votre **répertoire personnel**, créez le réperoire `unix` puis le sous-répertoire `unix/TP1` et déplacez-vous dans ce répertoire en utilisant les commandes suivantes :

```
mkdir unix                                          (mkdir : make Directory)
mkdir unix/TP1
cd unix/TP1                                          (cd : change Directory)
```

Exercice 1. Quelques commandes pour commencer :

- Affichez votre nom de login avec la commande `whoami`.
- Affichez votre répertoire courant avec la commande `pwd`.

Dans ce qui suit, vous travaillerez dans le répertoire `TP1`.

1. Créez un fichier `premierTexte` contenant une ou deux phrases de votre choix. Pour cette création, on demande d'utiliser l'éditeur `nano` en lançant la commande `nano premierTexte`.

   utilisation de nano : le prompt affichant le nom de la machine disparaît ; à la place, on a une page blanche où on peut taper du texte. Le nom du fichier que nous éditons et le nom de l'éditeur sont affichés en haut. La liste des commandes disponibles est écrite en bas. Vous pouvez faire `Ctrl-O` pour enregistrer vos changements et `Ctrl-X` pour quitter (avec retour au terminal).

2. Pour visualiser le contenu de `premierTexte`, on peut utiliser la commande `cat` avec comme argument `premierTexte` : `cat premierTexte`.
   Notez que la tabulation permet de compléter le nom du fichier.

3. Affichez les métadonnées du fichier `premierTexte` (notamment sa taille) par la commande `ls -l premierTexte`

4. En examinant les permissions sur ce fichier, qui peut le lire ? Qui peut le modifier ?

NB : Unix est *case sensitive*, ce qui signifie qu'il est sensible à la casse des caractères : minuscules et majuscules sont différentes. Par conséquent, par exemple `premierTexte` et `PremierTexte` sont deux noms de fichiers différents.

Exercice 2. Les commandes `cp, ls, mv`

1. Faites une copie du fichier `premierTexte` appelée `introduction` grâce à la commande `cp` :
   `cp premierTexte introduction` `(cp pour "copy")`

   Comparez leur taille en tapant la commande `ls -l` (sans arguments, `ls` donne la liste de tous les fichiers du répertoire).

   On peut aussi taper la commande `ls -l premierTexte introduction`

   Lancez la commande `ls -li` pour afficher les détailes des fichiers et aussi leur inode. Les fichiers sont-ils identiques ? Préciser votre réponse.

2. Renommez `introduction` en `double` en utilisant la commande `mv` :
   `mv introduction double` `(mv pour "move")`

   Examinez le résultat de la commande `ls -l`

Exercice 3.

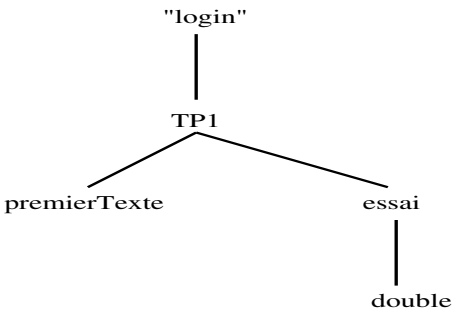
1. Dans le répertoire `TP1`, créez un répertoire appelé `essai`. Vérifiez que `essai` a bien été créé en utilisant la commande `ls`. Vous pouvez aussi constater le résultat par l'explorateur graphique de fichiers.

2. Déplacez le fichier `double` dans le répertoire `essai` avec la commande `mv`, puis déplacez-vous dans le répertoire `essai`.

   ```
   mv double essai/
   cd essai
   ```

   NB : `mv` sert à renommer des fichiers ainsi qu'à les déplacer. Le slash (/) n'est pas obligatoire, c'est un agrément pour identifier plus vite les répertoires dans une liste de fichiers.

   NB2 : On peut déplacer et renommer en même temps, avec la commande
   `mv monFichier essai/monNouveauFichier`.



   Placez-vous de nouveau dans le répertoire `TP1` avec la commande `cd ..`

   NB3 : `cd ..` permet de remonter au répertoire *père*, `cd ../..` remonte au répertoire *grand-père*, etc...

3. Affichez la liste de ce que contient le répertoire `essai` en utilisant `ls` avec le bon répertoire en paramètre.

4. En restant dans le répertoire `TP1`, faites une copie du fichier `premierTexte` dans le répertoire `essai` sous le nom de `copie`, et affichez de nouveau le contenu du répertoire `essai` :

   ```
   cp premierTexte essai/copie
   ls essai
   ```

Exercice 4. On reste dans le répertoire `TP1`

1. Tentez de détruire le répertoire `essai` avec la commande `rm`. Que se passe-t-il ?

   C'est la commande `rmdir` qui sert à supprimer un répertoire.

   Essayez avec la commande `rmdir essai`. De nouveau, c'est un échec. De cette manière, on ne peut pas détruire le répertoire avant de le vider.

   2 solutions possibles :

   a) On commence par vider le répertoire `essai`, et pour cela on peut utiliser

   ```
   rm essai/*                                  (supprimer tout fichier de essai/)
   ```

   Ensuite, on peut détruire le répertoire vide

   ```
   rmdir essai                              (suppression de essai désormais vide)
   ```

   b) Ou alors, on le fait avec une seule commande en tapant la commande

   ```
   rm -r essai
   ```

   L'option `-r` est "récursive". Elle supprime toute l'arborescence à partir du répertoire `essai` et est donc "radicale" voire dangereuse.

   Préférez l'utilisation de la commande `rm` avec l'option interactive `rm -i`. Ainsi, si vous utilisez aussi l'option `-r`, la combinaison `rm -ri essai` demande une confirmation pour supprimer le(s) fichier(s).

   ```
   p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/TP/TP1/TP1$ rm -ri essai
   rm : descendre dans le répertoire 'essai' ? o
   rm : supprimer 'essai/double' du type fichier ? o
   rm : supprimer 'essai' du type répertoire ? o
   p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/TP/TP1/TP1$ ls -l
   total 4
   -rw-rw-r-- 1 p103199 p103199 61 janv.  6 09:25 premierTexte
   p103199@OR-ST-LIFO-SGAGNElinux:~/Documents/univ/L1/unix/TP/TP1/TP1$
   ```

   Testez ces deux solutions en recréant la structure de fichiers entre les deux essais de suppression (reprendre les commandes qui avaient servi dans l'exercice précédent).

Exercice 5. Les caractères magiques

Avec la commande `mkdir`, créez un répertoire `lesSujets` dans le répertoire `TP1`. Il doit toujours y avoir `premierTexte` dans `TP1`.

1. En utilisant la commande `touch` qui permet de créer des fichiers vides, créez dans le répertoire `TP1` des fichiers supplémentaires que vous appelerez `suj1, suj2, suj3, suj12, suj7, suj21, suj4, suj30, suj32, sujet201` et `sujet`. Vous pouvez créer plusieurs fichiers simultanément en donnant plusieurs paramètres à la commande `touch` (séparés par un espace).

2. **Après avoir lu les diapos 25 et 36 du CM 1 sur les "caractères magiques", et sans lancer les commandes suivantes dans le terminal**, quel doit être le résultat de chacune d'elles ?

```
ls s*                ls s?              ls suj?                ls suj[0-9]
ls suj[0-9]*         ls suj[0-9]?       ls [a-z][a-z]?[0-9]    ls suj[^23]*
ls suj[^23]          ls sujet???        ls [^s]*               ls *e*
```

Pour rappel, les caractères magiques sont dans le tableau suivant :

Caractère	Description
*	Correspond à un certain nombre (≥ 0) de caractères (sauf /)
?	Correspond à un seul caractère (sauf /)
[]	Définit une plage ou un ensemble de caractères autorisés
[^]	Correspond à tout caractère sauf ceux mentionnés dans les crochets
{}	Correspond à plusieurs options définies, séparées par des virgules
\	Échappe les métacaractères pour les traiter comme des caractères normaux

De plus, `[0-9]` désigne les nombres entre 0 et 9 et `[a-z]` désigne les lettres entre a et z.

Vérifiez chacune de vos réponses en lançant les commandes dans le terminal.

Pour les questions 2, 3 et 4, vous utiliserez une commande `ls`, `rm` ou `mv` avec en paramètre une seule chaîne de caractères, contenant des caractères magiques. Si votre réponse ne produit pas le résultat demandé, recréez la structure de fichiers et recommencez.

3. Listez en une seule commande les détails (métadonnées) des fichiers `suj21`, `suj30` et `suj32`. Vous proposerez une solution basique sans caractère magique, et une autre plus évoluée avec des caractères magiques.

4. Effacez en une seule commande les fichiers `suj3, suj30` et `suj32`. Même consigne : une commande basique et une autre avec des caractères magiques.

5. Déplacez en une seule commande les fichiers `suj1, suj2, suj12, suj21` dans le répertoire `lesSujets`. Vous donnerez la version "caractères magiques".

Exercice 6. Les droits d'accès

1. Créez dans le répertoire `TP1` un fichier `tp1.txt` en utilisant `nano` et affichez ses droits d'accès.

2. Utilisez la commande `chmod` de manière "symbolique" pour modifier ces droits d'accès et leur donner successivement les valeurs suivantes (commande pour a, puis commande pour b à partir de a, puis commande pour c à partir de b) :

a) `rwxrwxr-x`
b) `rwxr-xr-x`
c) `rwxr--r--`

3. Redonnez les droits de départ au fichier, de façon octale.

4. Reprenez la question 2 en proposant la solution octale pour chaque cas.

Exercice 7. À partir du répertoire `TP1` :

1. Effacez de manière récursive et interactive (permission demandée à chaque suppression) tous les fichiers et répertoires présents dans `TP1`.

2. Créez avec `nano` un fichier nommé `monPremierScript` et qui contient les lignes :

```
echo "Hello world !" > monFichier
echo "Le fichier monFichier a été créé."
echo "Vous pouvez voir son contenu par la commande : cat monFichier"
```

   Conseil : cherchez ce que signifie la commande `echo "Hello world !" > monFichier` et en particulier le sens du `>`.

3. Vérifiez que le fichier `monPremierScript` n'est pas exécutable en affichant ses permissions par une commande `ls -l`.
   Confirmez en essayant de lancer la commande `./monPremierScript` (cette commande est sensée lancer un fichier exécutable).

4. Changez les permissions de `monPremierScript` pour le rendre exécutable au moins par l'utilisateur en utilisant la commande `chmod`.
   Exécutez le script par la commande de la question précédente.

   Les questions suivantes demandent d'avoir lu les slides du cours sur les variables d'environnement (32 à 34).

5. Á quoi correspondent les variables d'environnement `PATH` et `PWD` ?

6. Essayez de lancer le script par la commande `monPremierScript` (sans le précéder de `./` et trouvez pourquoi cela ne fonctionne pas (indication : la commande `echo $PATH`).

7. Modifiez la variable d'environnement `PATH` en y ajoutant le répertoire `TP1` :
   `PATH=$PATH:$PWD`

   NB La commande précédente affecte à la variable `PATH` sa valeur actuelle concaténée avec `" :"` et le chemin vers le répertoire actuel de travail (`TP1`).

   Vérifiez que cette fois la commande `monPremierScript` est bien exécutée (d'ailleurs l'auto-complétion doit maintenant fonctionner).

   NB2 : à l'ouverture d'une nouvelle, la variable `PATH` sera réinitialisée, les changements précédents seront inopérants.