

## TD n°1 : Commandes UNIX

**Objectif :** Se familiariser avec les commandes UNIX de base

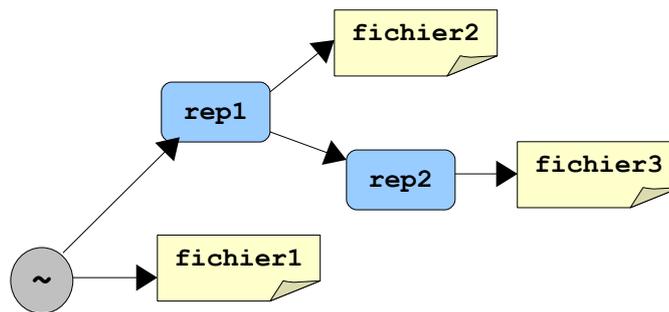
**Exercice 1 – Documentation en ligne**

Unix possède un manuel « en ligne ». La commande `man` permet d'explorer ce manuel.

- 1) Quelle documentation contient ce manuel ? Comment est-il structuré ?
- 2) La commande `write` porte le même nom que l'appel système `write`. Comment accède-t-on à la page du manuel concernant l'appel système `write` ?
- 3) Commande `ls` : précisez les options que vous savez utiliser et celles que vous pourriez éventuellement utiliser.

**Exercice 2 – Création d'arborescence**

Sans bouger du répertoire racine (celui qui est à la base de l'arborescence ; il s'agit ici de `~`), créez l'arborescence suivante :

**Exercice 3 – Navigation dans l'arborescence**

- 1) Dans votre répertoire personnel, créer un répertoire `test`. Créer sous-répertoires `pub` et `bin` du répertoire `test`.
- 2) Modifier les droits d'accès de façon que :
  - le répertoire `test` et son sous-répertoire `pub` soient accessibles en lecture et exécution pour tous, et en écriture pour le propriétaire seul.
  - le répertoire `bin` soit accessible en exécution pour tous, et en lecture et écriture pour le propriétaire et son groupe.
- 3) Aller dans le répertoire `test`. Créer un fichier `doc.txt`. Recopier le fichier dans `pub`.
- 4) Aller dans `bin`. Afficher à l'écran le contenu de `test/pub/doc.txt`.
- 5) Afficher la liste des fichiers contenus dans `test` et ses sous-répertoires.
- 6) Supprimer du répertoire `test` et de ses sous-répertoires tous les fichiers se terminant par le caractère `~`.
- 7) Afficher à l'écran le contenu des fichiers `toto` et `titi` qui se trouvent dans le répertoire d'accueil de l'utilisateur `balev`. Supprimer ces deux fichiers.

### Exercice 4 – Navigation dans l'arborescence (suite)

On désire aller dans le répertoire `/usr/local/games/mariokart`, et le répertoire courant est `/usr/local`. Quelle(s) commande(s) peut-on taper ?

- **A**: `cd /games/mariokart`
- **B**: `cd games/mariokart`
- **C**: `cd local/mariokart`
- **D**: `cd /usr/local/games/mariokart`
- **E**: `cd /usr/local/../../local/games/mariokart`
- **F**: `cd ../games/mariokart`

### Exercice 5 – Visualisation de fichiers

Lister tous les fichiers :

- se terminant par '5',
- commençant par 'annee4',
- commençant par 'annee4' et de 7 lettres maximum,
- commençant par 'annee' avec aucun chiffre numérique,
- contenant la chaîne 'ana',
- commençant par 'a' ou 'A'

### Exercice 6 – Manipulation de fichiers

- 1) Quelle différence y a-t-il entre les commandes `mv toto titi` et `cp toto titi` ?
- 2) Copier les fichiers dont l'avant-dernier caractère est un '4' ou '1' dans le répertoire `/tmp` en une seule commande.

### Exercice 7 – Gestion des droits d'accès

Dans votre répertoire courant, vous créez un répertoire courant `essai_droit`. Par défaut, ce répertoire est à 755 (rwxr-xr-x). Quelles sont les commandes (en notation symbolique et en base 8) pour lui donner les droits suivants (on suppose qu'après chaque commande on remet le répertoire à 755) :

	<i>Propriétaire</i>			<i>Groupe</i>			<i>Les autres</i>		
	Lecture	Ecriture	Accès	Lecture	Ecriture	Accès	Lecture	Ecriture	Accès
<b>Commande 1</b>	oui	oui	oui	oui	non	oui	non	non	oui
<b>Commande 2</b>	oui	non	oui	non	oui	non	non	non	oui
<b>Commande 3</b>	non	oui	non	non	non	oui	oui	non	non
<b>Commande 4</b>	non	non	oui	oui	non	oui	non	non	non

### Exercice 8 – Redirection et tubes

Ecrire une ligne de commande équivalente à :

```
ls -l /usr/bin > tmp
less tmp
rm tmp
```

### Exercice 9 – La commande cat

Supposons :

```
$ cat devinette.txt
devinette numero 4 :
pince mi et pince moi
sont dans un bateau.
pince mi tombe à l'eau.
qui est ce qui reste ?
```

Qu'affichent les commandes suivantes (**A** : 0; **B** : 1; **C** : 2; **D** : 3; **E** : 4; **F** : 5) :

- 1) `cat devinette.txt | grep ce | wc -l ?`
- 2) `cat devinette.txt | grep 4 | wc -l ?`

### Exercice 10 – Les commandes grep et cut

On suppose qu'un fichier `liste.txt` contient des informations sur les étudiants (10 au moins). Chaque ligne représente un étudiant, et contient les informations suivantes : nom, âge et filière. Les champs seront séparés par un « ; ».

**Exemple** : la ligne `Dumont;23;L3` correspond à l'étudiant Dumont, âgé de 23 ans et appartenant à la filière L3.

- 1) Renvoyer toutes les lignes du fichier `liste.txt` qui correspondent à l'étudiant s'appelant 'Sami'.
- 2) Renvoyer toutes les lignes correspondant à des étudiants de la filière L3.
- 3) Renvoyer toutes les lignes des étudiants âgés de 22 ans.
- 4) Renvoyer les lignes des étudiants n'appartenant pas à la filière L3.
- 5) Renvoyer toutes les lignes contenant la chaîne 'mi' sans tenir compte de la casse.
- 6) Afficher le nom et l'âge de chaque étudiant, puis le nom et la filière.
- 7) Afficher les trois premiers caractères de chaque ligne.

### Exercice 11 – La commande find

- 1) Chercher tous les fichiers dont le nom est 'passwd'.
- 2) Chercher tous les fichiers dont la date de la dernière modification remonte à plus de 10 minutes.
- 3) Trouver tous les fichiers du groupe 'root'.
- 4) Chercher tous les fichiers dont la taille est supérieure à 20Mo.
- 5) Chercher tous les répertoires se trouvant sous `/etc`.
- 6) Chercher tous les fichiers de l'utilisateur 'Raimbault'