

# TP UNIX.

## TP 1 : Commandes de base et shell

1. Connectez-vous sur votre machine virtuelle à l'aide de ssh (ou Putty) et affichez la liste des utilisateurs connectés.
2. Affichez le nom de votre répertoire de travail (le répertoire courant).
3. Listez tous les attributs du fichier `.bashrc` (et uniquement de ce fichier là).
4. Affichez la liste de TOUS vos processus en cours d'exécution.
5. Exécutez la commande `sleep 6000` en arrière plan et vérifiez qu'elle est bien en cours d'exécution.
6. Tuez la commande lancée précédemment.
7. Créez un fichier appelé `numtel` avec l'éditeur `Vi` contenant le texte suivant:  

```
MARIE 04-91-85-96-34
ODILE 04-91-56-92-35
ALAIN 04-42-46-87-12
```
8. Affichez le contenu de ce fichier trié par ordre alphabétique
9. Affichez le nombre de lignes du fichier `numtel`.
10. Créez un répertoire `rep1` et faites-en votre répertoire de travail (répertoire courant), copiez le fichier `numtel` dans ce répertoire, remontez dans le répertoire parent puis créez un répertoire `rep2` et déplacez `numtel` dedans (sans changer de répertoire, évidemment).
11. Supprimez en une seule commande le répertoire `rep2` et son contenu.
12. Modifiez les droits du répertoire `rep1` de façon à ce qu'il n'y ait que vous qui puissiez accéder (avec tous les droits).
13. Créez un lien symbolique nommé `reper` vers le fichier `numtel`, dans le répertoire `rep1`.
14. Donnez un nom supplémentaire au fichier `numtel`. Vérifiez qu'il s'agit bien du même fichier physique.
15. Cherchez quelle est la variable d'environnement qui contient l'invite de commande. Changez-la définitivement (changement effectif même si vous fermez votre session).
16. Affichez le contenu du fichier `/usr/share/dict/words`, page par page.
17. Affichez les 10 premières puis les 10 dernières lignes du fichier `/usr/share/dict/words` (en une seule ligne de commande).
18. Trouvez tous les fichiers `*.conf` appartenant à l'utilisateur `root` contenus dans `/etc` et ses sous-répertoires.

## TP 2 : Les filtres

### Exercice 1

Connectez vous plusieurs fois votre utilisateur. Affichez les connexions de cet utilisateur (et uniquement lui, avec `grep`). Déconnectez-vous.

### Exercice 2

Trouvez dans le fichier `/usr/share/dict/words` tous les mots qui commencent par `acro` ou `Acro` (avec `grep`).

Exemple de résultat :

```
Acropora
acrotic
...
```

### Exercice 3

Créez un fichier nommé `conf` à partir du fichier `/etc/sysctl.conf` dans lequel tous les caractères `'='` ont été remplacés par `:'` (avec `sed`).

### Exercice 4

Affichez le contenu du fichier `numtel` avec les noms en minuscules (avec `tr`).

### Exercice 5

Trouvez dans le fichier `/usr/share/dict/words` tous les mots contenant un palindrome de six lettres (avec `grep`).

Exemples:

```
addresser
barrabora
grammar
...
```

## TP 3 : Les scripts

### Exercice 1

Créez un script qui crée 100 répertoires nommés toto1, toto2, toto3, ...

### Exercice 2

Écrire un script qui permet de faire l'addition de nombres saisis au clavier, de cette façon:

```
# ./somme
nombre: 45
nombre: 37
nombre: 23
nombre: .
Somme = 105
```

### Exercice 3

Écrire un script qui affiche les informations de la ligne de commande, de cette façon:

```
# ./infocmd 22 3
La ligne de commande comporte 3 paramètres
Commande: ./infocmd
Paramètres: 22 3
```

### Exercice 4

Écrire un script qui affiche des informations sur la connexion d'un utilisateur dont le nom est passé en paramètre, de cette façon:

```
# ./infouser ttoto
L'utilisateur ttoto s'est connecté 3049321 fois sur servux.intra.e-cml.org
Ses 3 dernières connexions sont:
Mer  2 oct 09:32 - 17:55
Mar  1 oct 18:13 - 18:14
Mar  1 oct 11:31 - 12:35
```

### Exercice 5

Écrire un script qui, dans le répertoire courant, affiche pour chaque fichier le nom de celui-ci et ses n premières lignes (n est fourni en paramètre). Après l'affichage de chaque fichier, le script attend l'appuie d'une touche avant de passer au suivant.

Exemple:

```
# ./affrep 3
===== fichier: .bashrc
export CVS_RSH=ssh
if test `uname` = "FreeBSD"
then
===== fichier: .profile
export PATH=./sbin:/bin:/usr/sbin:/usr/bin
BLOCKSIZE=K;      export BLOCKSIZE
EDITOR=vim;       export EDITOR
```

## TP 4 : Scripts supplémentaires

### Exercice 1

Trouvez dans le fichier `/usr/share/dict/words` tous les mots contenant les voyelles a, e, i et o exactement dans cet ordre (avec `grep`).

Exemples:

```
alepidote
kaleidophon
latebricole
...
```

### Exercice 2

Écrire un script de gestion des répertoires qui fonctionne avec un menu de ce style:

```
1 - Choisir un répertoire de travail.
2 - Afficher les attributs du répertoire.
3 - Afficher le contenu du répertoire.
4 - Créer un sous répertoire.
5 - Supprimer un sous répertoire.
q - quitter.
```

### Exercice 3

Écrire un script récursif (qui se rappelle lui-même) qui calcule la factorielle d'un nombre.

Exemples:

```
# ./fact -2
erreur: l'argument doit etre positif
# ./fact 0
1
# ./fact 3
6
```

Ce script ne doit avoir qu'un seul argument.

### Exercice 4

Écrire un script qui permet d'éditer un fichier avec l'éditeur par défaut de l'utilisateur (variable d'environnement `EDITOR`, ou `vi` si aucun éditeur par défaut n'est défini) en gérant plusieurs versions successives de ce fichier. Par exemple, si vous voulez éditer un fichier nommé `essai`, le script vérifie si il existe déjà des versions de ce fichier dans le répertoire courant. Si il n'en existe pas, il lance l'éditeur de texte. Si il existe un fichier nommé `essai`, le script le renomme en `essai.1` avant de lancer l'éditeur. Si il existe déjà un `essai.1`, celui-ci est renommé en `essai.2` et `essai` devient `essai.1`, et ainsi de suite. Le nombre de versions pourra être paramétrable avec la variable d'environnement `NVER` (par défaut, trois versions si la variable n'est pas définie).

### Exercice 5

On souhaite écrire une commande similaire à une commande UNIX, de nom `t_rep`, dont le rôle est de donner diverses indications sur le contenu d'un répertoire. Appelée sans paramètre, cette commande doit renvoyer la taille cumulée des fichiers que contient le répertoire courant. Appelée avec l'option `-r`, elle doit indiquer le nombre de sous-répertoires contenus dans le répertoire courant. Appelée avec l'option `-f`, elle doit renvoyer le nombre de fichiers contenus dans le répertoire courant. Appelée avec un nom de répertoire en paramètre (après les éventuelles options), les valeurs retournées concernent le contenu de ce répertoire. On supposera que cette commande ne peut être appelée qu'avec au plus une seule option (soit `-f`, soit `-r`).