

## 1. Partie Théorique

### 1.1 Les expressions régulières

Dans le TP N°2, nous avons vu ce qu'étaient les métacaractères. Les expressions régulières sont aussi des suites de caractères permettant de faire des sélections. Elles fonctionnent avec certaines commandes comme **grep**.

Les différentes expressions régulières sont :

- **^** début de ligne
- **.** un caractère quelconque
- **\$** fin de ligne
- **x\*** zéro ou plus d'occurrences du caractère x
- **x+** une ou plus occurrences du caractère x
- **x?** une occurrence unique du caractère x
- **[...]** plage de caractères permis
- **[^...]** plage de caractères interdits
- **\{n\}** pour définir le nombre de répétition n du caractère placé devant
- **\{n,\}** pour définir au minimum le nombre de répétition n du caractère placé devant
- **\{N1 N2\}** Répétition de N1 à N2.

#### Exemples

1. Les lignes contenant au minimum un caractère en minuscule **[a-z][a-z]\***
2. Recherche les lignes contenant uniquement un nombre à 4 chiffres **^[0-9]\{4\}\$**
3. Cherche toutes les lignes commençant par des nombres de minimum 1 à 5 chiffres suivis par deux points (``:"). **^[0-9]\{1,5\}** :
4. Ligne composée d'un seul caractère **^.\$**
5. Cherche toutes les lignes contenant des nombres de minimum 2 chiffres avant les deux points (``:"). **[0-9]\{2,\}** :
6. Cherche toutes les lignes contenant la chaîne ``une fois" ou ``deux fois". **(une|deux) fois**

### 1.2 La commande grep

La commande **grep** permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :

- **-v** affiche les lignes ne contenant pas la chaîne
- **-c** compte le nombre de lignes contenant la chaîne
- **-n** chaque ligne contenant la chaîne est numérotée

- **-x** ligne correspondant exactement à la chaîne
- **-w** lignes où le mot apparaît tel quel
- **-l** affiche le nom des fichiers qui contiennent la chaîne

### Exemple

Soit le fichier carnet-adresse suivant:

```
mohamed:29:0298334432:Tunis
mehdi:13:0466342233:Sfax
ahmed:75:0144234452:Mahdia
sana:92:0133444335:Sousse
```

On peut utiliser les expressions régulières avec grep.

Si on tape la commande : **grep^[a-d] carnet-adresse** On va obtenir tous les lignes commençant par les caractères compris entre a et d.

Dans notre exemple, on aura :

```
ahmed:75:0144234452:Mahdia
```

## 1.3 La commande find

La commande **find** permet de retrouver des fichiers à partir de certains critères. La syntaxe est la suivante : find

Les critères de recherche sont les suivants :

Critère	Signification
-namemotif	Recherche par nom de fichier. Le motif peut utiliser les caractères génériques du Shell ?, *, [ ]
-type type	Recherche par type de fichier. <b>f</b> pour le fichier ordinaire, <b>d</b> pour les répertoires, <b>b</b> pour les fichiers spéciaux en mode bloc, <b>c</b> pour les fichiers spéciaux en mode caractère. <b>l</b> pour les
-user utilisateur	Recherche par propriétaire. Le propriétaire des fichiers trouvés doit être l'utilisateur référencé.
-group groupe	Recherche par groupe. Les propriétaires des fichiers trouvés doivent être des membres du groupe référencé.
-size [+/-]n	Recherche par taille de fichier. La taille du fichier est exprimée en blocs (512 octets). Si après la taille l'ajoute le caractère c, la taille sera exprimée en octets. Ex
-atime [+/-]n	Recherche par date du dernier accès.
-mtime [+/-]n	Recherche par date de la dernière modification.
-ctime [+/-]n	Recherche par date de création.
-perm mode	Recherche par autorisations d'accès. Recherche les fichiers dont le mode de permission est égal au mode référencé.
-perm-mode	Recherche par autorisations d'accès. Recherche les fichiers qui ont au moins les autorisations décrites le mode.
-links [+/-]n	Recherche par nombre de liens pour les fichiers.
-inum [+/-]n	Recherche par numéro d'index pour les fichiers.
-newer nomfichier	Recherche des fichiers plus récents que le fichier passé en référence (nom fichier).

On peut combiner les critères avec des opérateurs logiques :

- **critère1 critère2** ou **critère1 -a critère2** correspond au et logique,
- **!critère** non logique,
- **\ (critère1 -o critère2\)** ou logique,

La commande find doit être utilisée avec l'option -print.

Sans l'utilisation de cette option, même en cas de réussite dans la recherche, find n'affiche rien à la sortie standard (l'écran, plus précisément le shell).

La commande find est récursive, c'est à dire où que vous tapiez, il va aller scruter dans les répertoires, et les sous répertoires qu'il contient, et ainsi de suite.

### Exemples Pratiques :

#### 1. Recherche par nom de fichier

Pour chercher un fichier dont le nom contient la chaîne de caractères toto à partir du répertoire /usr, vous devez taper : **find /usr -name toto -print**

- En cas de réussite, si le(s) fichier(s) existe(nt), vous aurez comme sortie :  
toto
- En cas d'échec, vous n'avez rien.

Pour rechercher tous les fichiers se terminant par .c dans le répertoire /usr, vous taperez : **find /usr -name "\*.c" -print**

Vous obtenez toute la liste des fichiers se terminant par .c sous les répertoires contenus dans /usr (et dans /usr lui-même).

#### 2. Recherche suivant la date de dernière modification

Pour connaître les derniers fichiers modifiés dans les 3 derniers jours dans toute l'arborescence (/), vous devez taper : **find / -mtime 3 -print**

#### 3. Recherche suivant la taille

Pour connaître dans toute l'arborescence, les fichiers dont la taille dépasse 1Mo (2000 blocs de 512Ko), vous devez taper : **find / -size 2000 -print**

#### 4. Recherche combinée

Vous pouvez chercher dans toute l'arborescence, les fichiers ordinaires appartenant à olivier, dont la permission est fixée à 755, on obtient :

**find / -type f -user olivier -perm 755 -print**

#### 5. Redirection des messages d'erreur

Vous vous rendrez compte assez rapidement qu'en tant que simple utilisateur, vous n'avez pas forcément le droit d'accès à un certain nombre de répertoires, par conséquent, la commande find peut générer beaucoup de messages d'erreur (du genre permission denied), qui pourraient noyer l'information utile.

Pour éviter ceci, vous pouvez rediriger les messages d'erreur dans un fichier poubelle (comme /dev/null), les messages d'erreur sont alors perdus (rien ne vous empêche de les sauvegarder dans un fichier, mais ça n'a aucune utilité avec la commande find).

**find . -name bobo -print 2>/dev/null**

## 6. Recherche en utilisant les opérateurs logiques

Si vous voulez connaître les fichiers n'appartenant pas à l'utilisateur olivier, vous taperez :

**find . ! -user toto -print**

**! -user toto**, est la négation de **-usertoto**, c'est à dire c'est tous les utilisateurs sauf toto.

Recherche des fichiers qui ont pour nom **a.out** et des fichiers se terminant par **.c**. On tape :

**find . \ ( -name a.out -o -name "\*.c" \ ) -print**

On recherche donc les fichiers dont le nom est a.out ou les fichiers se terminant par \*.c, une condition ou l'autre.

Recherche des fichiers qui obéissent à la fois à la condition a pour nom core et à la condition a une taille supérieure à 1Mo.

**find . \ ( -name core -a size +2000 \ ) -print**

## 7. Les commandes en option

L'option **-print** est une commande que l'on passe à find pour afficher les résultats à la sortie standard. En dehors de **print**, on dispose de l'option **-exec**. **find** couplé avec **exec** permet d'exécuter une commande sur les fichiers trouvés d'après les critères de recherche fixés. Cette option attend comme argument une commande, celle ci doit être suivi de **{ }\ ;**.

**Exemple** recherche des fichiers ayant pour nom core, suivi de l'effacement de ces fichiers.

**find . -name core -execrm { }\ ;**

Tous les fichiers ayant pour nom core seront détruits, pour avoir une demande de confirmation avant l'exécution de **rm**, vous pouvez taper :

**find . -name core -ok rm { }\ ;**

# 2. Partie Pratique

## Exercice 1 :

Sachant que le fichier **/etc/passwd** a la structure suivante :

```
(login):(motdepasse):(uid):(gid):(nom et prénom du user):(Home Directory):(Login shell)
```

Et le fichier **/etc/group** a la structure suivante :

**(nom du groupe):(motdepasse):(gid):(membres)**

Donner les commandes permettant de :

1. chercher dans le fichier **/etc/passwd** les lignes commençant par t
2. chercher les lignes ne commençant pas par t dans le fichier **/etc/passwd**
3. chercher les lignes contenant les mots suivant le modèle T.t.
4. chercher les noms des groupes qui commencent par a ou b .. ou j
5. pour lister les sous-répertoires du répertoire **/etc**
6. compter les lignes dans le fichier **/etc/passwd** qui se terminent par a
7. afficher les lignes des fichiers **essai?.txt** qui contiennent a, b ou c
8. compter les lignes des fichiers **essai?.txt** qui contiennent a, b ou c

## Exercice 2 : find

1. Utiliser **find** pour rechercher le fichier ordinaire **cigale.txt** dans l'arborescence de votre répertoire personnel.
2. Utiliser **find** pour rechercher le répertoire **tpunix** dans l'arborescence de votre répertoire personnel.
3. Utiliser **find** pour rechercher tous les fichiers ordinaires vous appartenant dans toute l'arborescence du système de fichiers (c'est à dire depuis la racine).
4. Reprendre la question précédente et détourner les sorties de manière à éliminer les messages d'erreur (les envoyer vers le fichier **poubelle /dev/null**) et paginer l'affichage des fichiers trouvés.
5. Utiliser **find** pour créer le fichier **hebdo.txt** contenant les chemins des fichiers ordinaires vous appartenant contenus dans l'arborescence du système de fichiers et qui ont été modifiés il y a moins d'une semaine, tout en éliminant les messages d'erreur.
6. Utiliser le manuel en ligne pour étudier le fonctionnement de la commande externe **tee**. Reprendre la question précédente pour, en plus de créer le fichier **hebdo.txt**, utiliser **wc** pour faire afficher le nombre de fichiers trouvés, le tout en un seul pipeline.
7. Faire rechercher la liste des terminaux (**/dev/pts**) dont la permission d'écriture est accordée aux autres, ou qui vous appartiennent.
8. Avec une seule commande, afficher les informations détaillées de chacun des fichiers ordinaires qui se trouvent dans l'arborescence de votre répertoire de connexion dont le nom commence par **ts\_** ou par **algo**, Pour cela, utiliser l'option **-exec** de **find**.
9. Reprendre la question précédente mais au lieu d'afficher les informations sur ces fichiers, les supprimer.

10. Rechercher les fichiers ordinaires d'extension .txt contenus dans l'arborescence de votre répertoire personnel, et modifier leurs permissions en supprimant tous les droits aux membres du groupe et aux autres.

### Exercice 3:

Soit un fichier **MOYENNE.TXT** contenant les CIN, Nom, Moyenne, Adresse mail rangés comme suit :

```
01245780#amir bennour#15.2#amir.bennour@isetch.rnu.tn
03287514#Aymen Tounsi#12#aymen.tounsi@isetch.rnu.tn
02587410#Latifa Gharsallah#9.75#latifa.gharsallah@isetch.tn
04578145#AyaTounsi #13.5#tounsi.aya@isetch.rnu.tn
07854123#Samia BenAhmed #17#benahamed.samia@isetch.rnu.tn
```

1. Écrire la commande qui permet de trier le fichier MOYENNE.TXT selon la moyenne.
2. Écrire la commande qui permet d'afficher uniquement les noms 3 meilleurs étudiants.
3. Écrire la commande qui n'affiche que les CIN et les moyennes et les enregistre en même temps dans le fichier **resultat**.
4. Écrire la commande qui permet d'afficher le nom et le prénom de l'étudiant ayant le 5<sup>ième</sup> rang.
5. Écrire la commande qui permet d'afficher toutes les lignes ne contenant pas une adresse mail
6. Écrire la commande qui permet de convertir le contenu du fichier MOYENNE.TXT en Majuscule.