1. Les commandes grep et find

1.1 Les expressions régulières

On a vu auparavant 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`

Exemple l'expression **[a-z][a-z] *** cherche les lignes contenant au minimum un caractère en minuscule. **[a-z]** caractère permis, **[a-z]*** recherche d'occurrence des lettres permises.

L'expression **[0-9]\ {4\}$** a pour signification, du début à la fin du fichier **$**, recherche les nombres**[0-9]** de 4 chiffres **\ {4\}**.

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 avec le fichier **carnet-adresse** :

```
olivier:29:0298333242:Brest
marcel:13:0466342233:Gardagnes
myriam:30:0434214452:Nimes
yvonne:92:013344433:Palaiseau
```

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 n'en a pas, d'où l'absence de sortie.

```
grep Brest carnet-adresse
```

Permet d'obtenir les lignes contenant la chaîne de caractère Brest, soit :

```
olivier:29:0298333242:Brest
```

Il existe aussi les commandes fgrep et egrep équivalentes.

1.3 La commande find

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

```
find <répertoire de recherche> <critères de recherche>
```

Les critères de recherche sont les suivants :

- `-name recherche sur le nom du fichier,`
- `-perm recherche sur les droits d'accès du fichier,`
- `-links recherche sur le nombre de liens du fichier,`
- `-user recherche sur le propriétaire du fichier,`
- `-group recherche sur le groupe auquel appartient le fichier,`
- `-type recherche sur le type (d=rép., c=car., f=fichier normal),`
- `-size recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets),`
- `-atime recherche par date de dernier accès en lecture du fichier,`
- `-mtime recherche par date de dernière modification du fichier,`
- `-ctime recherche par date de création du 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 utilisé 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.

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 tapez :

```
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).

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
```

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
```

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
```

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
```

Recherche en utilisant les opérateurs logiques

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

```
find . ! -user olivier -print
```

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

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
```

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 -exec rm {}\   ;
```

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 {}\   ;
```

Autres subtilités

Une fonction intéressante de **find** est de pouvoir être utilisé avec d'autres commandes UNIX. Par exemple:

```
find . -type f -print | xargs grep toto
```

En tapant cette commande vous allez rechercher dans le répertoire courant tous les fichiers normaux (sans les répertoires, fichiers spéciaux), et rechercher dans ces fichiers tous ceux contenant la chaîne **toto**.