

# Chapitre 04 : Utilitaires de texte et Expressions régulières

## 1. Utilitaires de texte

Le texte est la principale forme d'entrée et de sortie pour les commandes Linux et il existe donc de nombreuses commandes dont l'objectif principal est de traiter des fichiers texte.

La commande **cat** (dérivée du mot concatenate) accepte plusieurs fichiers en entrée et génère un fichier fusionné.

Pour afficher le contenu des fichiers `/etc/hosts` et `/etc/hostname`, exécutez les commandes suivantes :

```
sysadmin@localhost:~$ cat /etc/hosts
127.0.0.1      localhost
::1          localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.1.2    localhost
sysadmin@localhost:~$ cat /etc/hostname
localhost
```

Pour concaténer le contenu des fichiers `/etc/hosts` et `/etc/hostname`, exécutez la commande suivante :

```
sysadmin@localhost:~$ cat /etc/hosts /etc/hostname
127.0.0.1      localhost
::1          localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.1.2    localhost
localhost
```

La sortie standard de la commande `cat` peut être envoyée vers un autre fichier en utilisant la redirection. La redirection de sortie standard est obtenue en suivant une commande avec le caractère **supérieur à** `>` et un fichier de destination.

```
sysadmin@localhost:~$ cat /etc/hosts /etc/hostname > result
```

Pour afficher le fichier de résultats créé ci-dessus, exécutez la commande suivante :

```
sysadmin@localhost:~$ cat result
127.0.0.1      localhost
::1          localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.1.2    localhost
localhost
```

Pour afficher le fichier de résultats ainsi que les numéros de ligne préfixés à chaque ligne, exécutez la commande suivante :

```
sysadmin@localhost:~$ cat -n result
 1 127.0.0.1      localhost
 2 ::1          localhost ip6-localhost ip6-loopback
 3 fe00::0 ip6-localnet
 4 ff00::0 ip6-mcastprefix
 5 ff02::1 ip6-allnodes
 6 ff02::2 ip6-allrouters
 7 192.168.1.2    localhost
 8 localhost
```

**NB :** La commande cat peut être utilisée uniquement pour afficher des fichiers texte.

Bien que l'affichage de petits fichiers avec la commande cat ne pose aucun problème, ce n'est pas un choix idéal pour les gros fichiers. Pour les fichiers plus volumineux, une commande pager peut être utilisée pour afficher le contenu. Les commandes du pager affichent une page de données à la fois, vous permettant d'avancer et de reculer dans le fichier à l'aide des touches de déplacement.

La commande **less** offre une capacité de pagination très avancée. Il s'agit généralement du pager par défaut utilisé par des commandes comme la commande man.

Pour afficher un fichier avec la commande less, passez le nom du fichier en argument :

```
sysadmin@localhost:~$ less /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd/network:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd/resolved:/usr/sbin/nologin
/etc/passwd
```

Pendant que vous êtes dans la sortie de la commande less, vous pouvez afficher l'écran d'aide en appuyant sur la touche **h** :

Lorsque vous êtes dans l'écran d'aide, vous pouvez revenir au fichier en appuyant sur la touche **q**.

Il existe deux manières de rechercher le résultat de la commande less : en effectuant une recherche vers l'avant ou vers l'arrière à partir de votre position actuelle.

Pour lancer une recherche pour regarder vers l'avant à partir de votre position actuelle, utilisez la **barre oblique /** la touche. Ensuite, tapez le texte ou le motif correspondant et appuyez sur la touche Entrée.

Pour rechercher le mot sbin dans le fichier /etc/passwd, utilisez la barre oblique / la touche :

```
sysadmin@localhost:~$ less /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd/network:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd/resolved:/usr/sbin/nologin
```

Appuyez sur n pour avancer et sur N pour reculer dans le résultat de la recherche.

Enfin, quittez le less pager en tapant la lettre **q** :

La commande **split** est utilisée pour diviser le fichier d'entrée en deux ou plusieurs fichiers.

Pour diviser un gros fichier, tel que le fichier de mots en morceaux plus petits, copiez d'abord le fichier /usr/share/dict/words dans le répertoire actuel :

```
sysadmin@localhost:~$ cp /usr/share/dict/words .
```

Pour diviser le fichier de mots en morceaux plus petits avec la sortie du préfixe, puis vérifier le résultat de la commande split, exécutez les commandes suivantes :

```
sysadmin@localhost:~$ split words output
sysadmin@localhost:~$ ls output*
outputaa  outputan  outputba  outputbn  outputca  outputcn  outp
outputab  outputao  outputbb  outputbo  outputcb  outputco  outp
outputac  outputap  outputbc  outputbp  outputcc  outputcp  outp
outputad  outputaq  outputbd  outputbq  outputcd  outputcq  outp
outputae  outputar  outputbe  outputbr  outputce  outputcr  outp
outputaf  outputas  outputbf  outputbs  outputcf  outputcs  outp
outputag  outputat  outputbg  outputbt  outputcg  outputct  outp
outputah  outputau  outputbh  outputbu  outputch  outputcu  outp
outputai  outputav  outputbi  outputbv  outputci  outputcv  outp
outputaj  outputaw  outputbj  outputbw  outputcj  outputcw  outp
outputak  outputax  outputbk  outputbx  outputck  outputcx  outp
outputal  outputay  outputbl  outputby  outputcl  outputcy  outp
outputam  outputaz  outputbm  outputbz  outputcm  outputcz  outp
```

Par défaut, les nouveaux fichiers créés avec la commande split seront nommés avec un préfixe x et un suffixe alphabétique aa, ab, etc.

La commande split permet aux fichiers d'avoir un suffixe numérique au lieu d'un suffixe alphabétique par défaut. Pour diviser le fichier de mots en fichiers préfixés par le nom result et suffixés par des chiffres, exécutez les commandes suivantes :

```
sysadmin@localhost:~$ split words -d result
sysadmin@localhost:~$ ls result*
result    result14  result29  result44  result59  result74  resu
result00  result15  result30  result45  result60  result75  resu
result01  result16  result31  result46  result61  result76  resu
result02  result17  result32  result47  result62  result77  resu
result03  result18  result33  result48  result63  result78  resu
result04  result19  result34  result49  result64  result79  resu
result05  result20  result35  result50  result65  result80  resu
result06  result21  result36  result51  result66  result81  resu
result07  result22  result37  result52  result67  result82  resu
result08  result23  result38  result53  result68  result83  resu
result09  result24  result39  result54  result69  result84  resu
result10  result25  result40  result55  result70  result85  resu
result11  result26  result41  result56  result71  result86  resu
result12  result27  result42  result57  result72  result87  resu
result13  result28  result43  result58  result73  result88
```

Par défaut, la commande split divisera un fichier en morceaux de 1 000 lignes. Les 1 000 premières lignes du fichier d'origine iront dans le premier fichier, les 1 000 secondes lignes iront

dans le deuxième fichier, etc. L'option -l peut être utilisée pour modifier le nombre par défaut de lignes à diviser.

La commande **nl** numérotera les lignes de sa sortie.

Pour afficher le fichier newhome.txt dans le répertoire Documents avec les numéros de ligne, exécutez les commandes suivantes :

```
sysadmin@localhost:~$ cd Documents/  
sysadmin@localhost:~/Documents$ nl newhome.txt  
 1 Thanks for purchasing your new home!!  
  
 2 **Warning** it may be haunted.  
  
 3 There are three bathrooms.  
  
 4 **Beware** of the ghost in the bedroom.  
  
 5 The kitchen is open for entertaining.  
  
 6 **Caution** the spirits don't like guests.  
  
 7 Good luck!!!
```

Par défaut, la commande **nl** est utilisée pour numéroté les lignes non vides dans la sortie. Pour compter le nombre de lignes dans le fichier newhome.txt, y compris les lignes vides, exécutez la commande suivante :

```
sysadmin@localhost:~/Documents$ nl -ba newhome.txt  
 1 Thanks for purchasing your new home!!  
 2  
 3 **Warning** it may be haunted.  
 4  
 5 There are three bathrooms.  
 6  
 7 **Beware** of the ghost in the bedroom.  
 8  
 9 The kitchen is open for entertaining.  
10  
11 **Caution** the spirits don't like guests.  
12  
13 Good luck!!!  
sysadmin@localhost:~/Documents$ cd  
sysadmin@localhost:~$
```

Le but de la commande **head** est d'afficher le début d'un fichier ou d'une sortie

Pour afficher le début du fichier `/usr/share/dict/words`, exécutez la commande suivante :

```
sysadmin@localhost:~$ head /usr/share/dict/words
A
A's
AMD
AMD's
AOL
AOL's
Aachen
Aachen's
Aaliyah
Aaliyah's
```

Si le nombre de lignes n'est pas précisé, la commande `head` affichera les 10 premières lignes. Pour afficher les 15 premières lignes de la sortie du fichier `/usr/share/dict/words`, exécutez la commande suivante :

```
sysadmin@localhost:~$ head -15 /usr/share/dict/words
A
A's
AMD
AMD's
AOL
AOL's
Aachen
Aachen's
Aaliyah
Aaliyah's
Aaron
Aaron's
Abbas
Abbas's
Abbasid
```

Pour afficher les 5 premières lignes du fichier `/usr/share/dict/words`, exécutez la commande suivante :

```
sysadmin@localhost:~$ head -n 5 /usr/share/dict/words
A
A's
AMD
AMD's
AOL
```

Pour afficher les 20 premières lignes du résultat de la commande `man ls`, exécutez la commande suivante :

```
sysadmin@localhost:~$ man ls | head -20
LS(1)                                User Commands

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory
    Sort entries alphabetically if none of -cftuvSUX nor --sort
    fied.

    Mandatory arguments to long options are mandatory for
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
```

La commande **tail** affiche le contenu à partir de la fin du fichier et non du début.

Pour afficher les 10 dernières lignes du fichier `/usr/share/dict/words`, exécutez la commande suivante :



```
sysadmin@localhost:~$ tail /usr/share/dict/words
élan's
émigré
émigré's
émigrés
épée
épée's
épées
étude
étude's
études
```

Pour afficher les 5 dernières lignes du fichier /usr/share/dict/words, exécutez la commande suivante :

```
sysadmin@localhost:~$ tail -5 /usr/share/dict/words
épée's
épées
étude
étude's
études
```

Pour afficher les 5 dernières lignes du résultat de la commande **ls -ltr**, exécutez la commande suivante :

```
sysadmin@localhost:~$ ls -ltr /etc | tail -n 5
-rw-r--r-- 1 root root 3611 Jul 3 14:42 mailcap
lrwxrwxrwx 1 root root 12 Apr 12 16:01 mtab -> /proc/ mounts
-rw-r--r-- 1 root root 172 Apr 12 16:01 hosts
-rw-r--r-- 1 root root 10 Apr 12 16:01 hostname
-rw-r--r-- 1 root root 38 Apr 12 16:01 resolv.conf
```

Pour afficher les 4 dernières lignes du fichier /etc/hosts (le nombre total de lignes

```
sysadmin@localhost:~$ tail -n -4 /etc/hosts
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.1.2 localhost
```

Pour afficher le contenu du fichier `/etc/hosts` à partir de la 3ème ligne jusqu'à la fin du fichier, exécutez la commande suivante :

```
sysadmin@localhost:~$ tail -n +3 /etc/hosts
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.1.2    localhost
```

La commande **paste** fusionnera les lignes d'un ou plusieurs fichiers, ligne par ligne, en les séparant par défaut par une tabulation comme délimiteur (séparateur). La commande **Coller** est particulièrement utile pour les fichiers contenant des données au format colonne.

Exécutez les commandes suivantes pour créer deux fichiers nommés `head` et `tail`, puis fusionnez-les dans un fichier nommé `total` :

```
sysadmin@localhost:~$ head -5 words > head
sysadmin@localhost:~$ tail -5 words > tail
sysadmin@localhost:~$ paste head tail > total
sysadmin@localhost:~$ cat total
A      épée's
A's    épées
AMD    étude
AMD's  étude's
AOL    études
```

Pour fusionner à nouveau les deux fichiers en utilisant le caractère deux-points : comme délimiteur au lieu du délimiteur de tabulation par défaut, exécutez la commande suivante :

```
sysadmin@localhost:~$ paste -d : head tail > total
sysadmin@localhost:~$ cat total
A:épée's
A's:épées
AMD:étude
AMD's:étude's
AOL:études
```

La commande **cut** est utilisée pour extraire les champs d'un fichier texte. L'espace et la tabulation sont les délimiteurs par défaut et peuvent être modifiés à l'aide de l'option `-d`.

Pour extraire les 1er, 3ème et 4ème champs du fichier /etc/passwd, exécutez la commande suivante :

```
sysadmin@localhost:~$ cut -d: -f1,3,4 /etc/passwd | head -n 4
root:0:0
daemon:1:1
bin:2:2
sys:3:3
```

**NB :** Dans l'exemple ci-dessus, la commande head est utilisée pour limiter la sortie de la commande cut.

Une gamme de champs peut également être fournie à la commande cut. Pour extraire les champs 1 à 4 du fichier /etc/passwd, exécutez la commande suivante :

```
sysadmin@localhost:~$ cut -d: -f1-4 /etc/passwd | head -n 4
root:x:0:0
daemon:x:1:1
bin:x:2:2
sys:x:3:3
```

À l'aide de la commande cut, un champ spécifique ou un ensemble de champs peut également être extrait de données contenant des colonnes de largeur fixe. Pour extraire le contenu des colonnes 31 à 43 de la sortie de la commande ls -l, exécutez la commande suivante :

```
sysadmin@localhost:~$ ls -l /etc | head | cut -c31-43

Jul  1  2019
Mar  7  2019
Jul  1  2019
Mar 22  2019
Mar 22  2019
Apr 17  2019
Mar 22  2019
Apr  4  2018
Apr  2  2018
```

La commande **sort** est utile pour travailler avec des données organisées en colonnes. Il permet d'afficher un fichier trié sur un champ de données spécifique.

Pour spécifier les champs à trier du premier au dernier, utilisez la commande sort avec une ou plusieurs options **-k**.

Tout d'abord, affichez le fichier `adjectives.txt` dans le répertoire Documents à l'aide de la commande `cat` :

```
sysadmin@localhost:~$ cd Documents/  
sysadmin@localhost:~/Documents$ cat adjectives.txt  
1 golden  
2 honey  
3 fruit  
4 grey  
5 bald
```

Pour trier le fichier `adjectives.txt` en utilisant le deuxième champ (c'est-à-dire l'adjectif : 1 doré) comme clé, exécutez la commande suivante :

```
sysadmin@localhost:~/Documents$ sort -k2 adjectives.txt  
5 bald  
3 fruit  
1 golden  
4 grey  
2 honey
```

Dans la commande ci-dessus, l'option **-k** a un argument : le 2, qui indique le deuxième champ à trier.

Par défaut, le tri divise chaque ligne d'un fichier en champs en utilisant des espaces (tabulations ou espaces) comme délimiteurs. Pour spécifier un autre délimiteur, utilisez l'option **-t**.

Pour trier le fichier `/etc/passwd` en utilisant le premier champ (c'est-à-dire le nom d'utilisateur : `bin:x:2:2:bin:/bin:/usr/sbin/nologin`) comme clé et en utilisant le caractère deux-points : comme délimiteur, exécutez la commande suivante :

```

sysadmin@localhost:~/Documents$ cd
sysadmin@localhost:~$ sort -t: -k1 /etc/passwd | head -n 4
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
bind:x:106:111::/var/cache/bind:/usr/sbin/nologin

```

**NB :** La commande head est utilisée pour limiter la sortie à quatre lignes.

Utilisez l'option **-u** de la commande de tri pour supprimer les entrées en double, comme illustré ci-dessous :

```

sysadmin@localhost:~$ cut -f7 -d: /etc/passwd | head > output
sysadmin@localhost:~$ more output
/bin/bash
/usr/sbin/nologin
/usr/sbin/nologin
/usr/sbin/nologin
/bin/sync
/usr/sbin/nologin
/usr/sbin/nologin
/usr/sbin/nologin
/usr/sbin/nologin
/usr/sbin/nologin
sysadmin@localhost:~$ sort -u output
/bin/bash
/bin/sync
/usr/sbin/nologin

```

La commande **tr** peut être utilisée pour traduire d'un jeu de caractères à un autre.

Pour mettre en majuscule le fichier `/etc/hosts` avec la commande **tr**, exécutez la commande suivante :

```

sysadmin@localhost:~$ cat /etc/hosts | tr 'a-z' 'A-Z'
127.0.0.1      LOCALHOST
::1          LOCALHOST IP6-LOCALHOST IP6-LOOPBACK
FE00::0      IP6-LOCALNET
FF00::0      IP6-MCASTPREFIX
FF02::1      IP6-ALLNODES
FF02::2      IP6-ALLROUTERS
192.168.1.2   LOCALHOST

```

## 2. Expression régulières

Les expressions régulières sont des modèles interprétés par certaines commandes. Les modèles sont placés entre guillemets simples pour éviter que le shell ne les interprète mal. Les expressions régulières diffèrent du globbing, qui est une fonction exécutée par le shell et donc non associée directement à une commande.

La commande **grep** est utilisée pour démontrer l'utilisation des expressions régulières.

Il existe deux types d'expressions régulières : **de base** et **étendues**. Alors que les expressions de base sont interprétées par la plupart des commandes, les expressions étendues peuvent être utilisées avec une option dans les commandes prenant en charge leur interprétation.

La plus simple de toutes les expressions régulières utilise simplement des caractères littéraux, tels que des caractères alphanumériques. Bien que très utiles pour la commande **grep**, les caractères littéraux à eux seuls n'offrent aucun caractère générique ou capacité de correspondance spéciale. Pour cela, vous devrez utiliser les opérateurs disponibles dans les expressions régulières de base.

Métacaractère	Description
.	Représente n'importe quel caractère en dehors d'une nouvelle ligne.
^	Représente un début d'une expression.
\$	Représente la fin d'une expression.
*	Représente zéro ou plusieurs occurrences du caractère précédent.
+	Représente une ou plusieurs occurrences du caractère précédent.
?	Représente zéro ou une occurrence du caractère précédent.
[ ]	Permet de définir une série de caractères potentiels.
[ ^ ]	Permet d'exclure une série de caractères potentiels.
( )	Permet de créer un groupe de capture pour extraire une sous-chaîne de caractères.
\	Permet d'échapper un caractère pour qu'il soit traité comme du texte et non comme un expression

L'opérateur **d'ancrage avant** **^** peut être utilisé pour garantir qu'un motif apparaisse au début de la ligne. Pour rechercher les occurrences de la racine du motif en début de ligne dans le fichier `/etc/passwd`, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep '^root' /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

L'opérateur **d'ancrage arrière** `$` peut être utilisé pour garantir qu'un motif apparaisse à la fin de la ligne. Pour rechercher les occurrences du motif `bash` en fin de ligne dans le fichier `/etc/passwd`, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep 'bash$' /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:1000:37::/root:/bin/bash
sysadmin:x:1001:1001:System Administrator,,,:/home/sysadmin:/bin/bash
alice:x:1002:1005:Alice White:/home/alice:/bin/bash
maya:x:1003:1006:Maya Brown:/home/maya:/bin/bash
joe:x:1004:1007:Joe Green:/home/joe:/bin/bash
```

L'opérateur de **période** `.` est utilisé pour faire correspondre n'importe quel caractère à l'exception du caractère de nouvelle ligne. Pour faire correspondre le modèle `r..t` (c'est-à-dire tout modèle contenant `r`, suivi d'exactly deux caractères, puis suivi de `t`) dans le fichier `/etc/passwd`, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep 'r..t' /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:1000:37::/root:/bin/bash
```

L'**opérateur de liste** `[]` (notation entre crochets) utilisé dans le regroupement de fichiers a une implémentation similaire dans les expressions régulières, qui consiste à définir une liste ou une plage de caractères littéraux pouvant correspondre à un caractère. Par exemple, pour faire correspondre les caractères simples `0`, `1`, `2`, `3` et `4` entre les caractères deux-points : dans le fichier `/etc/passwd`, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep ':[01234]:' /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
Output Omitted...
```

Lorsqu'il est placé au début de l'opérateur de liste `[]`, l'opérateur d'ancrage `^` désigne la négation du modèle à rechercher. Ce qui suit correspondra à un seul caractère entre les deux points : caractères qui ne sont pas un `0`, `1`, `2`, `3`, `4` ou `x` :

```
sysadmin@localhost:~$ grep ':[^01234x]:' /etc/passwd
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

Dans les expressions régulières de base, placer une barre oblique inverse \ devant un autre caractère signifie faire correspondre ce caractère littéralement. Par exemple, en utilisant le \. Le modèle est un moyen approprié de faire correspondre le caractère (.).

Pour afficher les noms de fichiers du répertoire /etc commençant par les lettres rc, suivis d'un chiffre compris entre 3 et 6 et se terminant par l'extension de fichier .d, exécutez la commande suivante avec l'expression régulière rc[3-6]\*\.d :

```
sysadmin@localhost:~$ ls /etc | grep 'rc[3-6]*\.d'
rc3.d
rc4.d
rc5.d
rc6.d
```

Dans la commande ci-dessus, l'opérateur **astérisque** \* est utilisé pour rechercher zéro ou plusieurs instances de caractères numériques compris entre 3 et 6.

La commande grep fournit l'option -i pour rendre la correspondance de modèle insensible à la casse. Pour faire correspondre les majuscules et les minuscules du modèle abid dans le fichier /usr/share/dict/words, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep -i "abid" /usr/share/dict/words
Abidjan
Abidjan's
abide
abided
abides
abiding
rabid
```



L'option `-v` de la commande `grep` entraînera l'affichage de toutes les lignes qui ne correspondent pas au modèle. Pour afficher toutes les lignes qui ne contiennent pas le modèle `local` dans le fichier `/etc/hosts`, exécutez les commandes suivantes :

```
sysadmin@localhost:~$ more /etc/hosts
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
192.168.1.2    localhost
sysadmin@localhost:~$ grep -v "local" /etc/hosts
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

Dans les exemples vus jusqu'à présent, la commande `grep` a été utilisée pour rechercher des modèles dans un seul fichier. La commande `grep` peut également être utilisée pour rechercher dans plusieurs fichiers. Pour faire correspondre le modèle de test dans les fichiers qui correspondent à `m*` dans le répertoire `/etc`, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep "test" /etc/m*
/etc/mailcap:application/x-troff-man; /usr/bin/man -X100 -l '%s'
$DISPLAY" -a -e /usr/bin/gxditview; description=Man page
/etc/mailcap:text/troff; /usr/bin/man -X100 -l '%s'; test=test -
-e /usr/bin/gxditview; description=Man page
/etc/mailcap:text/plain; view %s; edit=vim %s; compose=vim %s; t
/bin/vim; needsterminal
/etc/mailcap::text/*; view %s; edit=vim %s; compose=vim %s; test
/vim; needsterminal
grep: /etc/mc: Is a directory
grep: /etc/modprobe.d: Is a directory
grep: /etc/modules-load.d: Is a directory
```

Pour rechercher plusieurs fichiers mais afficher uniquement les noms de fichiers au lieu de chaque ligne correspondante, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep -l "script" /etc/mime*
/etc/mime.types
```

L'utilisation de l'option **-r** de la commande **grep** permettra aux recherches dans les répertoires d'être récursives. Pour faire correspondre le modèle de recherche en recherchant de manière récursive tous les fichiers du répertoire **/etc**, puis en affichant les noms de fichiers correspondants, exécutez la commande suivante :

```
sysadmin@localhost:~$ grep -rl 'find' /etc/* 2>/dev/null
/etc/apparmor.d/abstractions/apparmor_api/is_enabled
/etc/apparmor.d/abstractions/apparmor_api/find_mountpoint
/etc/apparmor.d/usr.sbin.tcpdump
/etc/init.d/screen-cleanup
/etc/init.d/ufw
/etc/locale.gen
/etc/rmt
/etc/vim/vimrc
/etc/wgetrc
```

**NB :** La commande **2>/dev/null** envoie la sortie d'erreur (**stderr**) au périphérique nul afin qu'elle ne soit pas affichée à l'écran.