



Tr et Grep et expressions régulières



La commande tr

- Transpose ou élimine des caractères (**t**ranslate).
- Les caractères entrés sont traités et le résultat est envoyé sur la sortie standard.
- C'est un filtre ne reconnaissant pas les expressions régulières.
- On peut utiliser les intervalles du type a-z et les codes ASCII des caractères en notation octale \0xx
- **tr [options] ch1 ch2 <fich1 >fich2**
- Remplace toutes les occurrences de TOUS les caractères de **ch1** par le caractère de **ch2**, de même rang, dans le flot d'entrée.
- Exemple 1 : transposition.
 - `echo "ABCDE"|tr 'A' 'C'`
 - CBCDE
 - `echo "ABCDE"|tr -d 'A'`
 - BCDE
 - `echo "AABBBAACCDE"|tr -s 'A'`
 - ABBBACCDE



Exemples d'utilisation de tr

- `cat adresse.txt | tr 'A-Z' 'a-z' > adressemin.txt`
créer le fichier en minuscules
- `tr 'a-z' 'n-za-m' < adresse.txt > adressecrypte.txt`
décale de 13 lettres tous les caractères
alphabétiques du fichier `adresse.txt` et copie le
résultat dans `adressecrypte.txt`.
Ce "cryptage" est dit "crible de César" car utilisé
par le célèbre empereur pour dissimuler ses
missives.
Le décalage de 13 lettres permet par application
du même filtre redonne le fichier original.
- `echo abcde | tr -c b d` tout ce qui n'est pas "b"
est remplacé par "d" on a donc:
dbddd

Exemples d'utilisation de tr

- `echo abcde | tr -d b` tous les "b" seront effacés
on a donc:
acde
- `cat adresse.txt | tr -s '\t'` remplace toutes les tabulations (`\t`) multiples par une seule tabulation.
- `$ for i in *\ *; do mv "$i" `echo $i | tr " " "_`; done`
permet de convertir les espaces dans le nom des fichiers en caractère souligné "_"
- `$ for i in *_*; do mv "$i" "`echo $i | tr "_" " "`; done`
pour faire l'inverse
- `$ for i in *\ *; do printf "$i\n" | tr " " "_"; done` si vous voulez tester cette commande



La commande cut

- Découpe des lignes d'un fichier ou de l'entrée standard
- `$ cut -cliste [<nom de fichier>]`
- `$ cut -fliste [-d<caractère>] [-s] [<nom de fichier>]`

liste est une liste d'entiers séparés par des virgules correspondant aux caractères

- Exemples de valeurs de liste :
 - 1,4,7 1 et 4 et 7
 - 1-3, 8 1 à 3 et 8
 - -5,10 1 à 5 et 10
 - 3- 3 à la fin



Les options de la commande cut

- -c découpe selon les caractères
- -f découpe par champ selon le caractère séparateur de champ donné par -d.

Par défaut le caractère séparateur est le caractère tab

- -s supprime les lignes qui ne contiennent pas de séparateur



Exemples d'utilisation de cut

- `$ cut -c -5 /etc/passwd` extrait les 5 premiers caractères de chaque ligne du fichier `/etc/passwd`
- `$ cut -f1,5 -d: /etc/passwd` extrait le premier champ (nom de login) et le cinquième champ (nom réel de l'utilisateur) du fichier `/etc/passwd`



La commande grep

- La commande **grep** permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :
 - ☐ **-i** Ignorer les différences majuscules/minuscules
 - ☐ **-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
 - ☐ **-l** affiche le nom des fichiers qui contiennent la chaîne
 - ☐ **-E, --extended-regexp**
 - Interpréter le *MOTIF* comme une expression rationnelle étendue



Métacaractères

caractère	signification
^	début de la chaîne
\$	fin de la chaîne
.	n'importe quel caractère sauf retour à la ligne
*	matche 0 ou plusieurs fois
+	matche au moins 1 fois
?	matche 0 ou 1 fois
	alternative
()	groupement; mémorisation
[]	jeu de caractères
{ }	répétition



Répétition

caractère	signification
a^*	zéro ou plusieurs a
a^+	un ou plusieurs a
$a^?$	zéro ou un a
$a\{m\}$	exactement m a
$a\{m,\}$	au moins m a
$a\{m,n\}$	au minimum m a et au maximum n a



alias

Alias	Correspondance	Equivalence
\n	Caractère de nouvelle ligne	
\r	Caractère de retour à la ligne	
\t	Caractère de tabulation	
\s	Caractère d'espacement (espace, tabulation, saut de page, etc)	[\\f\\n\\r\\t\\v]
\S	Tout ce qui n'est pas un espacement	[^\\f\\n\\r\\t\\v]
\d	Un chiffre	[0-9]
\D	Tout sauf un chiffre	[^0-9]
\w	Un caractère alphanumérique	[a-zA-Z0-9_]
\W	Tout sauf un caractère alphanumérique	[^a-zA-Z0-9_]
\n	Caractère en octal ex:\001 ==> " 1 "	
\xn	Caractère en hexadécimal ex: \x41 ==> " A "	



Expression régulière dans un script

```
#!/bin/bash
if [ $# -ne 1 ]; then
    echo "Utilisation: $0 <IP>"
    exit
fi
if [[ $1 =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$
]]; then
    echo "$1 est une adresse ip"
else
    echo "L'adresse IP $1 est invalide."
    exit
fi
```



Exercice:

Corriger ce script pour qu'il valide une adresse IP comprise entre 0.0.0.0 – 255.255.255.255

```
#!/bin/bash
if [ $# -ne 1 ]; then
    echo "Utilisation: $0 <IP>"
    exit
fi
if [[ $1 =~ ^((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])$) ]]; then
    echo "$1 est une adresse ip"
else
    echo "L'adresse IP $1 est invalide."
    exit
fi
```

