

Complément de TP Les filtres d'affichage (1)

Exercice 1

Indication : la commande cut

cut : extractions et réorganisation de champs.

Options principales :

-c liste : coupure en mode caractère.
-f liste : coupure en mode champ.
-d c : utiliser le caractère « c » comme séparateur de champs (par défaut, la tabulation).

Format de la liste :

n le n^{ème}.
n,m le n^{ème} et le m^{ème}.
n-m du n^{ème} au m^{ème}.
n- du n^{ème} à la fin.
-n du 1er au n i^{ème}.

Soit le fichier suivant : Un, deux, trois, quatre

Cinq, six, sept, huit

Neuf, dix, onze, douze

Treize, quatorze, quinze, seize

1/ donner la ligne de commande permettant de donner la première colonne.

2/ donner la ligne de commande permettant de donner la première et la troisième colonne.

3/ donner la ligne de commande permettant de donner du 5^{ème} au 9^{ème} caractères.

Exercice 2

A/ consulter le manuel de la commande grep et énoncer les différentes utilisations de la commande.

Les expressions régulières

Les expressions régulières sont utilisées dans de nombreux programmes ou filtres Unix lorsque des chaînes de caractères sont à rechercher.

(Attention, les fonctionnalités ci-dessous dépendent du shell.)

Caractères ou de groupe de caractères

. un caractère, n'importe lequel (saut de ligne compris).

M-R la plage des caractères de M à R.

[abc...] un caractère parmi abc... (ceux spécifiés entre crochets).

[^abc...] un caractère qui n'est pas parmi abc... (aucun de ceux spécifiés entre crochets).

Répétition : on répète une expression en plaçant immédiatement après l'un des opérateurs suivants :

* se répète un nombre quelconque de fois (0 compris).

+	se répète au moins une fois.
?	se répète au plus une fois.
\{n\}	se répète exactement n fois
\{n,\}	se répète n fois ou plus.
\{,m\}	se répète au plus m fois.
\{n,m\}	se répète n fois au moins et m fois au plus.

Groupement : un groupement est une expression régulière « exp » contenant toute combinaison de caractères, d'opérateurs, ou de groupement.

exp1exp2	correspond à exp1 puis exp2 (concaténation).
exp1 exp2	correspond à exp1 ou exp2.
(exp)	groupe le contenu de l'expression exp et la marque.
\n	fait référence à la n ^{ème} expression marquée.

Position

^	début de ligne.
\$	fin de ligne.
\<	début de mot.
\>	fin de mot.

B/ Le fichier « /etc/passwd » contenant des informations concernant les différents utilisateurs de la machine. Les lignes du fichier ont la forme suivante :

Nom de l'utilisateur : x= mot de passe : UID =identifiant de l'utilisateur : GID = identifiant du groupe : répertoire de connexion : shell

Exemple : supposons que stage1, stage2, ..., stage 10 sont les utilisateurs d u système

```
stage1:x:501:501::/home/stage1:/bin/bash
stage2:x:502:502::/home/stage2:/bin/bash
stage3:x:503:503::/home/stage3:/bin/bash
...
stage10:x:510:510::/home/stage10:/bin/bash
```

- 1/ Afficher les lignes et les numéros de lignes contenant la chaîne « st ».
- 2/ Afficher les lignes et les numéros de lignes contenant exactement le mot « st ».
- 3/ Afficher toutes les lignes commençant par « s » ou « S ».
- 4/ Afficher toutes les lignes contenant au moins un chiffre.
- 5/ Afficher toutes les lignes commençant par une majuscule.
- 6/ Afficher toutes les lignes contenant « a », « b » ou « c ».
- 7/ Afficher les lignes contenant « a » comme deuxième caractère.
- 8/ Afficher les lignes qui se terminent par « h ».
- 9/ Afficher les lignes où ne figure pas le mot « stage».
- 10/ Quels sont les utilisateurs dont le nom commence par une voyelle.
- 11/ Quels sont les utilisateurs dont le nom ne commence pas par une voyelle.
- 12/ Afficher le nom de fichiers où ne figure pas le mot « stage».

Exercice 3

Indication : la commande find

recherche et exécution de commandes récursivement sur une arborescence. Toutes les expressions régulières sur les noms de fichier « regexf » sont à donner entre simple cote (')

Syntaxe :

find chemin(s) critère(s) action(s)

chemin est le nom du répertoire initial d'où la recherche va commencer. Au moins un chemin est obligatoire (et en première position).

si aucun critère n'est précisé, tous les fichiers ou répertoires sont traités.

si aucune commande n'est précisée, l'action par défaut est -print.

Arguments numériques :

n : n exactement égal à n.

+n : plus grand que n.

-n : moins grand que n.

Critères de recherche (extrait) :

-name regexf nom du fichier (-iname pour ignorer la casse).

-path regexf nom du chemin (-ipath pour ignorer la casse).

-amin n accès au fichier il y a n minutes (-atime jours).

-mmin n données du fichier modifiées il y a n minutes.

-empty le fichier est vide.

-perm n la permission du fichier est n (en octal).

-size n la taille du fichier est n (en blocks, nc en byte, nk en Kb).

-type c le type du fichier est c (d=répertoire, f=fichier, l=lien).

Tests booléens :

test1 -a test2 test1 et test 2 (le -and est facultatif).

test1 -o test2 test1 ou test2.

! test1 test inverse de test 1.

\(expression\) (expression{exemple : ensemble de tests})

Actions : elles sont exécutées sur tous les fichiers ou les répertoires satisfaisant le critère.

-print affichage simple.

-fprint file comme -print, mais sauvegarde dans file.

-ls affichage comme ls -dils

-fls file comme -ls, mais sauvegarde dans file.

-exec commande \ exécute commande. \ {} représente le nom du fichier ou

répertoire courant.

Etant l'utilisateur MI3, et ayant « / » comme répertoire courant :

[MI3@ localhost /] \$...

1/ Trouver tous les fichiers de l'utilisateur MI3.

2/ Trouver tous les fichiers du groupe de GID = 500.

3/ Trouver tous les sous répertoires du répertoire /etc.

4/ Trouver tous les fichiers qui ont été modifiés pendant les dernières dix minutes.

5/ Chercher tous les fichiers de /bin faisant appel au fichier entête <mah.h>.

6/ Imprimer tous les fichiers d'extension « .log » du répertoire /var.

7/ Effacer tous les fichiers d'extension « .tmp » de votre répertoire de connexion.

8/ Trouver tous les fichiers ordinaires dont les noms commencent par a et se terminent par h

9/ Trouver tous les fichiers ordinaires dont les noms commencent par a et se terminent par h et afficher toutes les informations relatives.

Correction

1/ cut -f1 -d',' fichier
 2/ cut -f1,3 -d',' fichier
 3/ cut -c5-9 fichier

Exercice2

1/ grep -n st /etc/passwd
 2/ grep -nw st /etc/passwd
 3/ grep -I ^s /etc/passwd
 4/ grep 0-9 /etc/passwd
 5/ grep ^A-Z /etc/passwd
 6/ grep [abc] /etc/passwd
 7/ grep ^.a /etc/passwd
 8/ grep h\$ /etc/passwd
 9/ grep -v stage /etc/passwd
 10/ grep -i ^[aeiouy] /etc/passwd
 11/ grep -v ^[aeiouy] /etc/passwd
 12/ grep -vl /etc/passwd

Exercice3

1/ find / -user MI3
 2/ find / -group 500
 3/ find /etc -type d
 4/ find / -mmin -10
 5/ find /bin -exec grep l <math.h>
 Ou grep -l <maths> /bin/*
 6/ find /var -name *.log -print
 7/ find ~ -name *.tmp -exec rm {} \
 find ~ -name *.tmp -exec -OK rm {} \
 8/ find / -type f -name a*h
 9/ find / -type f -name a*h -ls