

Systemes d'exploitation

Chapitre 7-6

Filtres UNIX

Filtres UNIX (1)

- Les filtres UNIX sont des commandes qui acceptent des données de l'entrée standard, effectuent des transformations sur ces données et les dirigent vers la sortie standard.



Filtres UNIX (2)

- Pour les besoins de ce cours:
 - ◆ Un filtre transforme les données présentées à son entrée (et de d'autres sources).
 - ◆ Un filtre achemine les données transformées à sa sortie standard (et vers d'autres destinations).
 - ◆ Un filtre affiche ses messages d'erreur vers la sortie erreur standard.
- Donc, pas toutes les commandes UNIX sont des filtres (dans le sens défini ci-dessus).

Filtre comm(1)

- comm(1) compare le contenu de deux ensembles de données.
 - ◆ L'un des deux ensembles peut provenir de l'entrée standard.
 - ◆ Synopsis:
comm [options] fichier1 fichier2
 - 1 Ne pas afficher la première colonne (les lignes présentent uniquement dans le fichier1)
 - 2 Ne pas afficher la deuxième colonne (les lignes présentent uniquement dans le fichier2)
 - 3 Ne pas afficher dans la troisième colonne (les lignes présentent dans les deux fichiers)

Filtre comm(1)

- ♦ Note: si le nom de l'un des fichiers est le caractère - alors son contenu est lu de l'entrée standard.
- ♦ Statut de sortie:
 - 0 pour indiquer qu'il n'y a pas eu d'erreur;
 - >0 pour indiquer qu'une erreur est rencontrée.
- ♦ Exemple:

Fichier1

```
Je suis ici vous voir  
cher Monsieur Jean.  
Comment ça va mon cher Watson ?  
Non, Pas le Dr. Watson de Windows!
```

Fichier2

```
Je suis ici vous voir  
cher Monsieur Jean  
Comment ça va mon cher Watson ?  
Non, Pas le Dr. Watson de Holms!
```

Filtre comm(1)

◆ Exemple (suite):

```
centi 15> comm comm1.dat comm2.dat
          Je suis ici vous voir
      cher Monsieur Jean
cher Monsieur Jean.
          Comment ça va mon cher Watson ?
      Non, Pas le Dr. Watson de Holms!
Non, Pas le Dr. Watson de Windows!
centi 16>
```

Première colonne

Deuxième colonne

Troisième colonne

Filtre diff(1)

➤ diff(1) donne les différences entre deux ensembles de données.

- ◆ Ce filtre accepte des noms de répertoire.

- ◆ Synopsis:

 - diff [options] fichier1 fichier2

 - diff [options] fichier1 répertoire

 - diff [options] répertoire fichier1

 - diff [options] répertoire1 répertoire2

Filtre diff(1)

- ◆ Synopsis (suite):

- b Ignorer les blancs (Espace et Tab) à la fin d'une ligne
- c[n] Afficher le contexte où les lignes différentes ont été trouvées

- ◆ Statut de sortie:

- 0 Pas de différence rencontrée
- 1 Des différences ont été relevées
- >1 Une erreur est rencontrée

Filtre diff(1)

- ♦ Interprétation des résultats de sortie:
addition (a), élimination (d) et changement (c)

Résultat	Signification
L1 a L2,L3 > Lignes du fichier2	Ajouter lignes du fichier2 après ligne L1 du fichier1.
L1,L2 d L3 < Lignes du fichier1	Effacer les lignes L1 à L2 du fichier1.
L1,L2 c L3,L4 < Lignes du fichier1 --- > Lignes du fichier2	Changer les lignes L1 à L2 du fichier1 en L3 à L4 du fichier2.

- ♦ En effectuant ces opérations, les deux ensembles de données deviendront identiques. 9

Filtre diff(1)

◆ Exemple:

```
centi 17> diff comm1.dat comm2.dat
2c2
< cher Monsieur Jean.
---
> cher Monsieur Jean
4c4
< Non, Pas le Dr. Watson de Windows!
---
> Non, Pas le Dr. Watson de Holms!
centi 18>
```

?

Filtre uniq(1)

➤ Montrer les lignes uniques d'un ensemble de données.

- ◆ Ce filtre accepte des données de l'entrée standard ou d'un fichier.

- ◆ Synopsis:

uniq [options] [-n] [+c] [fentree] [fsortie]

-c Afficher le nombre d'apparition de la ligne

-d Afficher uniquement les lignes répétées

-u Afficher les lignes qui ne sont pas répétées

-n Ignorer les n premiers champs des lignes

+c Ignorer les c premiers caractères des lignes

Filtre uniq(1)

- ♦ Synopsis (suite):

fentree fichier d'entrée (rien=entrée std)

fsortie fichier de sortie (rien=sortie std)

- ♦ Statut de sortie:

0 Pas d'erreur

>0 erreur rencontrée

- ♦ Note: Un champ est un ensemble de caractères entourés de caractères espaces ou Tab. Il peut exister 1 ou plusieurs champs dans une ligne.

Filtre uniq(1)

◆ Exemple:

```
Michelle est de retour
Minou à Moi
Minou à Moi
Minou à Moi
Michelle est de retour
Michelle est de retour
Michelle est de retour
Michelle est de retour
```

uniq.dat

```
centi 42> cat uniq.dat | uniq -c
```

```
1 Michelle est de retour
```

```
3 Minou à Moi
```

```
2 Michelle est de retour
```

```
2 Michelle est de retour
```

```
centi 43> uniq -d uniq.dat
```

```
Minou à Moi
```

```
Michelle est de retour
```

```
Michelle est de retour
```

```
centi 44> uniq -u < uniq.dat
```

```
Michelle est de retour
```

```
centi 45>
```

Lignes qui ne sont pas uniques

Lignes qui sont uniques

Filtre tr(1)

➤ Remplacer un ensemble de données en un autre ensemble de données.

- ◆ Ce filtre accepte certains méta-caractères de regex.

- ◆ Synopsis:

tr [options] [chaine1 [chaine2]]

- c Complément de l'ensemble des caractères représentés par la chaine1

- d Éliminer de l'entrée standard les caractères qui correspondent à ceux de chaine1

- s Réduire une séquence de caractères identiques en un seul caractère. La spécification de chaine2 est facultative

Filtre tr(1)

- ♦ Statut de sortie:
 - 0 Pas d'erreur
 - >0 erreur rencontrée
- ♦ Note: Pour éliminer les caractères contenus dans chaine1 du flux d'entrée, simplement ne pas spécifier chaine2 et utiliser l'option -d.
- ♦ Note: le remplacement s'effectue caractère par caractère. Donc, l'ordre des caractères dans chaine1 et chaine2 est important.

Filtre tr(1)

◆ Exemple:

comm1.dat

```
Je suis ici vous voir  
cher Monsieur Jean.  
Comment ça va mon cher Watson ?  
Non, Pas le Dr. Watson de Windows!
```

Pourquoi ce résultat ?



```
centi 49> tr -cs "[A-Z][a-z]" "[\n*]" < comm1.dat  
Je  
suis  
ici  
vous  
voir  
cher  
Monsieur  
Jean  
Comment  
a  
va  
mon  
cher  
Watson  
Non  
Pas  
le  
Dr  
Watson  
de  
Windows
```


Filtre tr(1)

- ◆ Exemple:

Observer bien le rôle de l'option -s.

```
centi 49> ls -l
total 6
-rw-r--r--  1 wong      recherche    109 mars 20 23:40 comm1.dat
-rw-r--r--  1 wong      recherche    106 mars 20 23:41 comm2.dat
-rw-r--r--  1 wong      recherche    153 mars 21 01:12 uniq.dat
centi 50> ls -l | tr -s ' '
total 6
-rw-r--r-- 1 wong recherche 109 mars 20 23:40 comm1.dat
-rw-r--r-- 1 wong recherche 106 mars 20 23:41 comm2.dat
-rw-r--r-- 1 wong recherche 153 mars 21 01:12 uniq.dat
```

La chaîne2 n'est pas spécifiée. Alors remplace les multiples apparitions du caractère de chaîne1 par le même caractère de chaîne1 (le caractère₁₇ espace).

Filtre cut(1)

- Sélectionner des champs des lignes d'entrées.
 - ◆ Ce filtre accepte des lignes de l'entrée standard ou des fichiers.
 - ◆ Synopsis:
`cut -c liste [fichier ...]`
`cut -f liste [-d car] [-s] [fichier ...]`
-c liste indique les positions en termes de caractères.
Par exemple, -c1-72 signifie les 72 premiers caractères de chaque ligne

Filtre cut(1)

- f liste indique les positions en termes de champs entourés par des caractères spécifiques. Par défaut, ce sont les blancs d'une ligne (Espace ou Tab). Par exemple, -f1,7 signifie les 7 premiers champs de chaque ligne
- dcar Le caractère car suivant -d est le séparateur de champ. Par exemple, -d: utilise le symbole : comme séparateur de champ
- s Suppression des lignes ayant aucun séparateur de champ
- ♦ Statut de sortie:
 - 0 Pas d'erreur
 - >0 erreur rencontrée

Filtre cut(1)

- ♦ Note: Les champs, dans ce contexte, sont spécifiés par l'option `-c` en termes de position de caractères ou par l'option `-f` en termes de caractères délimités par le séparateur spécifié par `-d`. L'argument liste représente les champs affectés.
- ♦ Note: La spécification des champs est séparée par le symbole `,` (virgule) et doit être en ordre croissant (ex : 1,4,7). Il est permis de spécifier une plage de champs (ex : 1-5,9).

Filtre cut(1)

◆ Exemple:

```
centi 75> ls -l
total 8
-rw-r--r--  1 wong      recherche      59 mars 21 01:14 #
-rw-r--r--  1 wong      recherche    109 mars 20 23:40 comm1.dat
-rw-r--r--  1 wong      recherche    106 mars 20 23:41 comm2.dat
-rw-r--r--  1 wong      recherche    153 mars 21 01:12 uniq.dat
centi 76> ls -l | cut -c1-10,55-
total 8
-rw-r--r-- #
-rw-r--r-- comm1.dat
-rw-r--r-- comm2.dat
-rw-r--r-- uniq.dat
centi 77> ls -l | tr -s ' ' ':'
total:8
-rw-r--r--:1:wong:recherche:59:mars:21:01:14:#
-rw-r--r--:1:wong:recherche:109:mars:20:23:40:comm1.dat
-rw-r--r--:1:wong:recherche:106:mars:20:23:41:comm2.dat
-rw-r--r--:1:wong:recherche:153:mars:21:01:12:uniq.dat
centi 78> ls -l | tr -s ' ' ':' | cut -f1,3,10 -d: | tr ':' ' '
total
-rw-r--r-- wong #
-rw-r--r-- wong comm1.dat
-rw-r--r-- wong comm2.dat
-rw-r--r-- wong uniq.dat
```

Filtre egrep(1)

- Trouver des caractères dans un ensemble de données.
 - ◆ Ce filtre fait partie de la famille des filtres grep (*global regular expression print*). La chaîne de caractères recherchée est décrite par une regex.
 - ◆ Synopsis:
egrep [options] [-e expr] [expression] [-f fich]
[fichiers ...]
-b Afficher au début de chaque ligne trouvée, le numéro de bloc sur le disque

Filtre egrep(1)

- c Afficher uniquement le nombre de lignes contenant la chaîne
- h N'afficher pas les noms de fichier
- i Ne pas faire de distinction entre les majuscules et les minuscules
- s Aucune sortie sauf les messages d'erreur
- e Nécessaire pour les expressions régulières commençant par un tiret
- f fich Expression régulière est contenue dans le fichier fich
- ◆ Statut de sortie:
 - 0 la chaîne est trouvée
 - 1 la chaîne n'est pas trouvée
 - 2 erreur(s) rencontrée(s)

Filtre egrep(1)

- ◆ Note: egrep(1) recherche les lignes qui contiennent la ou les chaînes de caractères indiquées par expr ou expression. Par défaut, les lignes trouvées sont envoyées à la sortie standard.
- ◆ Note: Lire l'aide en-ligne de egrep(1) pour connaître les expressions génériques acceptées.

Filtre egrep(1)

◆ Exemple:

```
centi 4> ypcat passwd | egrep ':Pierre.*'
parp2808:cRuT6m7jPRJWM:30502:112:Pierre Parent:/export/home/exa/email/parp2808:/bin/csh
doup2609:S3i0oM68.AuVQ:30251:112:Pierre Doucet:/export/home/exa/email/doup2609:/bin/csh
robcad22:Z.VGpH9oanp3M:11423:114:Pierre St-Denis:/usr/people/exa/robcad/robcad22:/bin/csh
gaup2806:8fZaJLAA4ybOY:30848:112:Pierre-Luc Gauthier:/export/home/exa/email/gaup2806:/bin/csh
dupp0911:LBXJgdV.Zl8lg:30267:112:Pierre Duplessis:/export/home/exa/email/dupp0911:/bin/csh
gerp1209:JLiqJTddvRdJY:30307:112:Pierre Hugues Gervaud:/export/home/exa/email/gerp1209:/bin/csh
gefp0904:zHizZC2VwOjlQ:30304:112:Pierre Olivier Geffroy:/export/home/exa/email/gefp0904:/bin/csh
boup0806:aX.t0b2BGmZcM:30804:112:Pierre-Etienne Bouchard:/export/home/exa/email/boup0806:/bin/csh
grop2001:jzWiOWxd.2iPQ:30966:112:Pierre Grondin:/export/home/exa/email/grop2001:/bin/csh
duvp0101:4jttBQNX/rHcg:30271:112:Pierre-Yves Duval:/export/home/exa/email/duvp0101:/bin/csh
habp0410:ytzP1NEqscex2:30857:112:Pierre-Luc Habel:/export/home/exa/email/habp0410:/bin/csh
: : :
```

Filtre egrep(1)

◆ Exemple:

```
centi 5> ypcat passwd | egrep ':Pierre .*'
parp2808:cRuT6m7jPRJWM:30502:112:Pierre Parent:/export/home/exa/email/parp2808:/bin/csh
doup2609:S3i0oM68.AuVQ:30251:112:Pierre Doucet:/export/home/exa/email/doup2609:/bin/csh
robcad22:Z.VGpH9oanp3M:11423:114:Pierre St-Denis:/usr/people/exa/robcad/robcad22:/bin/csh
dupp0911:LBXJgdV.Zl8lg:30267:112:Pierre Duplessis:/export/home/exa/email/dupp0911:/bin/csh

gerp1209:JLiqJTddvRdJY:30307:112:Pierre Hugues Gervaud:/export/home/exa/email/gerp1209:/bin/csh
gefp0904:zHizZC2VwOjlQ:30304:112:Pierre Olivier Geffroy:/export/home/exa/email/gefp0904:/bin/csh
grop2001:jzWiOWxd.2iPQ:30966:112:Pierre Grondin:/export/home/exa/email/grop2001:/bin/csh
stdp2705:srllvkqzhiJho:30582:112:Pierre St-Denis:/export/home/exa/email/stdp2705:/bin/csh
berp0604:mOT.Bp4f9rQ3I:30138:112:Pierre Berube:/export/home/exa/email/berp0604:/bin/csh
pinp2309:lkVu0qUAYsMmg:30519:112:Pierre Pinard:/export/home/exa/email/pinp2309:/bin/csh
daup2612:6qqRrxMOkIbkA:30220:112:Pierre Noel Dautrey:/export/home/exa/email/daup2612:/bin/csh
: : :
```

Filtre egrep(1)

◆ Exemple:

```
$ egrep '[.,!]' *  
comm1.dat :cher Monsieur Jean.  
comm1.dat :Non, Pas le Dr. Watson de Windows!  
comm2.dat :Non, Pas le Dr. Watson de Holms!
```

Trouver le caractère point ou le caractère virgule ou le caractère point exclamation dans tous les fichiers du répertoire courant.

Le filtre egrep (grep, fgrep, etc.) est très utile dans la recherche des patrons dans un fichier ou un ensemble de fichiers. De plus, la vitesse d'exécution de egrep est très très grande!