

## Systeme utilisateur :: Systeme UNIX ::

semaine 4 – les expressions rationnelles

## Kézako ?

Décrire une chaîne de caractère pour la rechercher, la remplacer, la supprimer... avec les commandes grep, sed ...

Utilise des jokers, des ensembles, des répétitions pour analyser un texte et détecter des motifs, vérifier des syntaxes ...

## Du déjà vu

Syntaxe proche de l'utilisation des joker utilisés avec la commande ls, find ... avec toutefois quelques différences !

- => Notions d'ensemble
- => Notions de répétitions

## Décrire un caractère – 1 Atome

=> un caractère spécifique : le caractère lui même a, b, c ...

=> n'importe quel caractère : utilisation du joker « . »

=> un caractère choisi une liste : [<<liste>>]

- exemple [abcdefghijklmnopqrstuvwxy] : tous sauf « z »

=> un caractère choisi hors d'une liste : [^<<liste>>]

- exemple [^z] : tous caractères sauf « z »

=> Possibilité d'utiliser des ensembles prédéfinis :

- [:alnum:] : alpha numériques a-zA-Z0-9

- [:digit:] : chiffres ou [0-9] [1-9] ...

- [:lower:] : a-z ou [a-z] [c-z] ...

- [:upper:] : A-Z ou [A-Z] ..

- [:alpha:] : A-Za-z

# Les répétitions d'atomes

Les répétitions suivent les atomes pour indiquer combien de fois l'atome se répète dans le motif :

- $\Rightarrow ?$  - l'atome est présent 0 ou 1 fois
- $\Rightarrow *$  - l'atome est présent 0 ou  $n$  fois
- $\Rightarrow +$  - l'atome est présent 1 fois ou plus
- $\Rightarrow \{n\}$  - l'atome est présent  $n$  fois
- $\Rightarrow \{n,\}$  - l'atome est présent au moins  $n$  fois
- $\Rightarrow \{,m\}$  - l'atome est présent au plus  $m$  fois
- $\Rightarrow \{n,m\}$  - l'atome est présent entre  $n$  et  $m$  fois

Exemple : un mot est une suite d'au moins un caractère alphabétique séparée par des espaces :

$.*[ ]+[ :alpha: ]+[ ]+.*$  ou plus simplement  $.*[ ][:alpha:][ ]+.*$

## Quelques caractères spéciaux

Indiquer le début d'une ligne : « ^ »

Indiquer la fin d'une ligne : « \$ »

Indiquer le début d'un mot : « \< » « \b »

Indiquer la fin d'un mot : « \> »

Protection des caractères : « \ » pour indiquer par exemple le caractère « \* » il faudra le faire précéder de « \ »

## Quelques exemples

=> Décrire une plaque d'immatriculation française : 1234 AB 12

- 1 chiffre entre 1 et 9 suivi d'au plus trois chiffre entre 0 et 9  
(nombres de 1 à 9999)
- 1 à 3 lettre majuscules
- 2 chiffres

=>  $[1-9]\{1\}[0-9]\{,3\}[A-Z]\{1,3\}[0-9]\{2\}$

-> Petit hic : le département 00 n'existe pas ...

## Quelques exemples

=> Décrire un entier relatif : exemple +2.1 , -15 ...

- Un signe suivi d'au moins un chiffre suivi éventuellement d'un point et de chiffres

Première approche :

$[+-]?[0-9]+\backslash\.[0-9]^+$

$[+-]?$  => un signe facultatif et présent au plus une fois

$[0-9]^+$  => au moins un chiffre

$\backslash\.$  => le symbole « . » présent au plus une fois

$[0-9]^+$  => au moins un chiffre partie décimale

Petit hic : « 2 » ne correspond pas .. il nous manque quelque chose...

## Les expressions alternatives

=> Décrire plusieurs possibilités :

( <<possibilité 1>> | <<possibilité 2>> | << trois>> ... )

=> « | » indique un « ou »

=> Les nombres relatifs :

[+-]? ( [0-9]+ | [0-9]+\.[0-9]+ )

=> Remarque .56 et ab2c sont reconnus !! car toutes lignes contenant au moins un chiffre correspondent [0-9]+

=> Il faut donc délimiter notre motif en précisant par exemple qu'il est en début de ligne  $^[+-]? ( [0-9]+ | [0-9]+\.[0-9]+ )$  ou en début d'un mot  $\backslash\mathbf{b}[+-]? ( [0-9]+ | [0-9]+\.[0-9]+ )$

## Les alternatives, les blocs

Exemple de la plaque :

$$[1-9][0-9]\{0,3\}[A-Z]\{1,3\}([1-9][0-9]|0[1-9])$$

Les blocs définis par ( <<bloc>> ) peuvent être utilisés pour indiquer des répétitions :

Décrire que l'on recherche une suite de deux blocs constitué de 3 chiffres suivis de 2 caractères :

$$([0-9]\{3\}[A-Z]\{2\})\{2\}$$

## Plus loin avec les blocs

Décrire une répétition :

il est possible de décrire la répétition d'un bloc en utilisant «  $\backslash n$  » où  $n$  est le numéro du bloc : par exemple indique une répétition d'un même caractère dans une phrase :

$(.)\backslash 1 \Rightarrow$  n'importe quel caractère suivi du même

Le présence dans une ligne d'un même groupe de trois caractères présent 2 fois :

$(...).\backslash 1 \Rightarrow$  n'importe quel groupe de 3 caractères suivi de n'importe quoi puis suivit du même groupe de 3 caractères.

## sed – aller plus loin avec les substitutions

sed permet les substitutions (rechercher/remplacer) au travers de la syntaxe suivante :

```
sed -e 's/expression1/expression2/'
```

expression1 est celle rechercher  
expression2 est celle remplacer

Grâce à l'usage des blocs et expressions régulières il est possible de réaliser des opérations complexes comme transformer une date du type

*ven sep 26 11:40:15 CEST 2003*

en

*Nous sommes le ven. 26 sep. 2003*

## Commande sed

=> l'expression 1 permet de découper en bloc la date dans son format initial :

```
ven sep 26 11:40:15 CEST 2003
```

```
^\([^ ]*\) \([^ ]*\) \([^ ]*\).* \([^ ]*\)$
```

La date est une ligne (^ ... \$) composée de 4 blocs :

- le premier est le premier ensemble de n'importe quels caractères autres que espace. (ici "ven")
  - le second est l'ensemble suivant de caractère autres que espace séparé du premier par un espace. (ici "sep")
  - le troisième est défini de la même façon (ici "26")
  - le quatrième est l'ensemble défini de la même façon terminant la ligne (ici "2003")
- Entre le 3ème et le 4ème bloc il y a n'importe quoi ...

## Commande sed

=> l'expression 2 constitue le résultat en faisant appel aux blocs définis dans l'expression 1 pour donner :

Nous sommes le ven. 26 sep. 2003

Nous sommes le \1. \3 \2. \4

La chaîne « Nous sommes le » est suivi du bloc 1 (le jour) on ajoute un « . » puis le bloc 3 (le numéro du jour) puis le bloc 2 le mois avec un « . » et enfin le bloc 4 (l'année).

La commande complète est la suivante :

```
date | sed -e 's/^\([^ ]*\) \([^ ]*\) \([^ ]*\).* \([^ ]*\)$/Nous sommes le \1. \3 \2. \4/'
```

## Exos ...

=> Afficher les lignes de texte avec 3 voyelles consécutives

=> Utiliser sed pour extraire l'adresse IP dans la ligne suivante :

```
inet adr:192.168.114.113 Bcast:192.168.114.255 Masque:255.255.255.0
```

=> Permuter dans le fichier /etc/passwd des colonnes uid et gid

```
login:password:uid:gid:...:home:shell
```

## Exos ...

=> Afficher les lignes de texte avec 3 voyelles consécutives

```
egrep -e [aeiouy]{3} texte
```

=> Utiliser sed pour extraire l'adresse IP dans la ligne suivante :

```
inet adr:192.168.114.113 Bcast:192.168.114.255 Masque:255.255.255.0
```

```
/sbin/ifconfig | grep 'inet adr' | sed -e 's/^[^:]*:\([^ ]*\) .*/IP=\1/'
```

=> Permuter dans le fichier /etc/passwd des colonnes uid et gid

```
login:password:uid:gid:...:home:shell
```

```
cat /etc/passwd | sed -e 's/^\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):(.*)\1:\2:\4:\3:\5/'
```