Introduction au Système d'Exploitation Unix/Linux

Expressions régulières

B. Jacob

IC2/LIUM

12 septembre 2017

Plan

1. Expressions régulières "simples"
2. Expressions régulières "étendues"
3. Commande `grep`
4. Commande `find`

Plan

1 Expressions régulières "simples"

Expressions régulières simples

"regexp" simples → sélection de noms de fichiers et/ou répertoires

Elles utilisent des métacaractères

- `?` : "joker" pour un car
- `*` : n car. ($n \geqslant 0$)
- `[ch]` : 'c' ou 'h'
- `{c,pl}` : 'c' ou 'pl'
- `[a-e]` : un car. $\in$ [a,b,c,d,e]
- `[^d-f]` : un car. tous sauf d,e,f
- `[a-zA-Z]` : une lettre minuscule ou majuscule
- `[a-z][0-9]` : lettre minuscule suivie d'un chiffre

Expressions régulières simples

Les regexp sont utilisées

$\Rightarrow$ pour une sélection multiple de fichiers

$\Rightarrow$ par des commandes utilisant plusieurs fichiers (comme `ls`)
(donc plusieurs noms de fichiers)

Exemples regexp simples avec "ls"

```
prompt% ls
File1    File2    File3
fichier  prog1.c  prog2.c

prompt% ls File?
File1 File2 File3

prompt% ls ?i*
File1 File2 File3 fichier

prompt% ls *.c
prog1.c prog2.c

prompt% ls [^Ff]*
prog1.c prog2.c
```

Exercices

Exercices sur les regexp simples . . .

Plan

2 Expressions régulières "étendues"

Expressions régulières "étendues"

"regexp" étendues → sélection de contenus de fichiers

Fonctions des regexps étendues

Fonctions des regexps simples +

- \n : retour à la ligne
- ^ : début de ligne
- $: fin de ligne

Expressions régulières étendues

Liste des Métacaractères ($\neq$ simples) :

- `.`(dot) : un caractère quelconque
- `*` : opérateur de répétition
- [xyz] : x ou y ou z
- [A – G] : intervale
- [^ xyz] : sauf x, y ou z
- ^: début de ligne
- $ fin de ligne
- \{m,n\} : répétitions entre m et n fois
- \ : échappement (enlève l'interprétation d'un car. spécial)

Exemples de regexp étendues

- ^$: représente une ligne vide
- ^[A-Z] : une majuscule en début de ligne
- [^A-Z] : tout sauf une majuscule
- [a-z][a-z]$: 2 minuscules en fin de ligne
- [ABCD]\{2,10\}$: entre 2 et 10 car. A,B,C ou D en fin de ligne

Plan

3 Commande grep

Recherche de motifs dans un fichier

`grep` : Recherche de contenus dans des fichiers

- sélection de contenu → regexp étendues
- sélections des fichiers → regexp simples

Format

```
grep [vin]  [regexp étendue] [regexp simple]
     options motif           fichiers
```

Options

- -v : complémentaire
- -i : maj. et min. indifférentes
- -n : affiche les numéros de lignes

Exemple grep sur 1 fichier

```
$ more texte
il fait beau
il fait chaud
beau temps n'est ce pas ?

$ grep "^il" texte
il fait beau
il fait chaud

$ grep "aud$" texte
il fait chaud

$ grep "[ph]a" texte
il fait chaud
beau temps n'est ce pas ?
```

Exemple grep sur plusieurs fichiers

```
$ more fich11.txt
il fait beau
mais pas trop
il est midi

$ more fich22.txt
il fait chaud
et humide

$ grep "^il" fich*.txt
fich11.txt:il fait beau
fich11.txt:il est midi
fich22.txt:il fait chaud
```

Exercice

Exercices sur `grep` et les regexp étendues . . .

Plan

4 Commande find

find

`find` : Recherche de fichiers

- dans une arborescence
- sur un ou plusieurs critères

Si critère porte

- sur le nom des fichiers alors
  → utilisation des regexp simples
- sur le contenu des fichiers alors
  → utilisation des regexp étendues

Recherche de fichier avec find

`find dir expr command`

- Recherche dans une arborescence de racine `dir`
- Des fichiers satisfaisant le(s) critère(s) `expr`

  Exemples

  - `( ) -o -a` opérateurs logiques
  - `-name` *regexp simple*
  - `-size n / -size +/-n` (taille n $\times$ 512 octets)
  - `-user` *nom utilisateur*
  - ...autres possibilités : `man find`

Recherche de fichier avec find

`find dir expr command`

- Avec l'application de `command`

  2 formats pour `command` :

  - `-print`
    affiche les noms des fichiers sélectionnés
  - `-exec CmdUnix {} \;`
    applique la commande Unix `CmdUnix` sur les fichiers sélectionnés *avec*
    - `{}` = variable prenant tour à tour chacun des noms des fichiers sélectionnés
    - `\;` = marque de fin de la commande Unix

  Si `CmdUnix` = `grep` alors → utlisation des regexp étendues

Exemples de find

- affichage des fichiers .c

```
% find . -name '*.c' -print
```

- affichage de mes fichiers

```
% find / -user $USER -print
```

- suppression des fichiers .o

```
% find $HOME -name '*.o' -exec rm {} \;
```

- recherche de lignes commençant par "toto" dans les fichiers .txt

```
% find . -name "*.txt" -exec grep "^toto" {} \;
```

Exercice

- Exercices sur `find` ...