TRƯỜNG ĐH KHOA HỌC TỰ NHIÊN HÀ NỘI
KHOA TOÁN – CƠ – TIN HỌC

BÀI THỰC HÀNH LINUX
Bài 04

——————— oOo ———————

Phần 1: Thực hành

1. **Biểu thức chính quy**

   - **Biểu thức chính quy** (*regular expression* - viết tắt *regex*) là một cú pháp đặc biệt được sử dụng để mô tả các mẫu ký tự. Trong Linux, biểu thức chính quy thường được sử dụng cùng với các chương trình: *grep, sed, vi, ...* để hỗ trợ các thao tác tìm kiếm và thay thế trong văn bản.
     Biểu thức chính quy là tập hợp các ký tự điều khiển (*metacharacter*) với những ý nghĩa đặc biệt và các ký tự thông thường.
     Các ký tự điều khiển trong biểu thức chính quy

Ký tự	**Mô tả**
^	Đánh dấu đầu dòng
$	Đánh dấu cuối dòng
.	Biểu diễn ký tự bất kỳ
[]	Biểu diễn một ký tự bất kỳ trong cặp dấu []
[^]	Biểu diễn phép phủ định của nội dung trong []
[−]	Biểu diễn một ký tự bất kỳ bên trong vùng được chỉ rõ bởi []
()	Biểu diễn các ký tự trong () như một nhóm ký tự
$\backslash <$	Đánh dấu đầu từ
$\backslash >$	Đánh dấu cuối từ
$\backslash b$	Đánh dấu biên từ
$\backslash B$	Đánh dấu xâu rỗng không ở biên từ
$\backslash n$	Biểu diễn ký tự/nhóm ký tự nằm giữa cặp () thứ n trước đó
?	Phép lặp: Ký tự/nhóm ký tự đi trước có thể xuất hiện hoặc không
+	Phép lặp: Ký tự/nhóm ký tự đi trước xuất hiện ≥ 1 lần
*	Phép lặp: Ký tự/nhóm ký tự đi trước xuất hiện ≥ 0 lần
$\{n\}$	Phép lặp: Ký tự/nhóm ký tự đi trước xuất hiện đúng n lần
$\{n,\}$	Phép lặp: Ký tự/nhóm ký tự đi trước xuất hiện $\geq$ n lần
$\{n,m\}$	Phép lặp: Ký tự/nhóm ký tự đi trước có thể xuất hiện từ n đến m lần
|	Phép hoặc

     Dùng \ trước các ký tự điều khiển để sử dụng chúng như những ký tự thông thường.
     Ví dụ:

- [^09] ≡ Tập các ký tự khác 0 và 9
- $[a - z45] \equiv$ {a, b, c, .., z, 4, 5}
- ^*tux* Biểu diễn các dòng bắt đầu bằng *tux*
- 03(6)(24)\1 ≡ 036246
- *colou?r* ≡ {color, colour}

Các phiên bản *regex* trong Linux

Có 3 phiên bản để biểu diễn cú pháp của biểu thức chính quy:

- BRE (Basic Regular Expressions)
- ERE (Extended Regular Expressions)
- PCRE (Perl Regular Expressions)

Các phiên bản ra đời sau đưa ra bộ cú pháp mới hơn để hỗ trợ tốt hơn cho việc biểu diễn biểu thức chính quy.

Ví dụ: Xâu chứa ký tự 'i' hoặc 'a' được biểu diễn trong:

- BRE: [i\|a]
- ERE: [i|a]

Có thể bổ sung tùy chọn khi thực hiện một lệnh nào đó để lựa chọn phiên bản *regex* sẽ sử dụng.

Ví dụ: Lệnh *grep* sử dụng các tùy chọn:

-G: Đọc các xâu theo phiên bản BRE

-E: Đọc các xâu theo phiên bản ERE

- **Pipe** (giao tiếp giữa các tiến trình)

Đôi khi các tiến trình cần trao đổi thông tin cho nhau để xử lý. Một cơ chế được sử dụng khá phổ biến trong Linux là *pipe*, sử dụng chỉ thị | cho phép đầu ra của tiến trình bên trái ký hiệu | là đầu vào của tiến trình bên phải.

Ví dụ 1: *student@linux ~ $ ls -R /etc | more*

Kết quả: Nội dung của thư mục */etc* được hiển thị ra màn hình theo từng trang

Ví dụ 2:*student@linux ~ $ echo "hoa phong lan" | cat > hoa.txt*
student@linux ~ $ cat hoa.txt

Kết quả: hoa phong lan

- **grep** là một công cụ hữu hiệu trong Linux để tìm kiếm trong nội dung tệp tin các dòng phù hợp với mẫu tìm kiếm.

Cú pháp: *grep [options] 'rexExp' fileName*
grep [options] 'rexExp' fileName > ouputFile
(ghi nội dung tìm kiếm ra tệp *ouput-File*)

Trong đó [options]:

- *'rexExp'*: là biểu thức chính quy biểu diễn mẫu tìm kiếm.
- [*options*]:
  - -i: Tìm kiếm không phân biệt hoa thường
  - -v: Tìm các kết quả không chứa các xâu biểu diễn bởi '*rexExp*'
  - -w: Đưa ra các dòng có chứa xâu biểu diễn bởi '*rexExp*' là một từ
  - -c: Đếm số dòng chứa xâu '*rexExp*'

* -e: Đếm số lần xuất hiện của xâu '*rexExp*' trong tệp tin
* -A5: Đưa ra 5 dòng sau kết quả
* -B5: In ra 5 dòng trước kết quả
* -C5: In ra 5 dòng trước và sau kết quả

Lệnh *grep* có thể sử dụng kết hợp với các lệnh khác: *ls, cat, echo, ...*
Ví dụ:

- *student@linux ~ $ grep 'nt' /etc/group* (in ra các dòng chứa xâu '*nt*' trong tệp tin */etc/group*)
- *student@linux ~ $ ls /etc | grep a$* (in ra các tệp tin/thư mục nằm trong thư mục */etc* có tên kết thúc bằng chữ cái 'a')
- *student@linux ~ $ ls * | grep -E '[i|a]+'* (in ra các tệp tin/thư mục nằm trong thư mục hiện tại có tên thỏa mãn: chữ cái '*i*' hoặc *'a'* xuất hiện ít nhất 1 lần)
- *student@linux ~ $ cat /etc/passwd | grep '\bover\b'* (in ra các dòng có chứa từ *'over'* trong tệp tin */etc/passwd*)
- *student@linux ~ $ echo 'There is a book' | grep -E '[her]+'*

Phần 2: Bài tập thực hành

Bài 1: Cho tệp tin "dataset.txt". Hãy sử dụng các biểu thức chính quy kết hợp với lệnh **grep** để tìm kiếm dữ liệu là:

1. Tập các số tự nhiên.
2. Tập các số tự nhiên chia hết cho 5.
3. Tập các số nhị phân có độ dài 6 và chia hết cho 4.
4. Tập các dòng bắt đầu bằng chữ cái "**T**" và chứa ít nhất 2 xâu "*This is an exercise*".

Bài 2: Tìm kiếm và lọc dữ liệu với *grep* để thực hiện:

1. Tìm kiếm các tệp tin/thư mục nằm trong thư mục */etc* có chứa ít nhất 2 chữ cái 'a'. Lưu kết quả ra tệp *output1.txt*
2. Tìm kiếm các tệp tin/thư mục nằm trong thư mục */etc* bắt đầu bằng chữ cái 'b' và không chứa chữ cái 'c'. Lưu kết quả ra tệp *output2.txt*
3. Trong tệp "name_list.txt" liệt kê tất cả những người có tên là jiayant đứng đầu mà không phân biệt chữ in hoa in thường ở bất kỳ ký tự nào trong chuỗi "jiyant" . Kết quả in ra tệp "out_jiayant.txt".
4. Trong tệp "phone_list.txt" hãy liệt kê ra những số điện thoại có mã vùng Hà Nội với định dạng "024-xxxxxxx". Kết quả in ra tệp "out_phonehanoi.txt",
5. Trong tệp "data_list.txt" hãy liệt kê ra các địa chỉ email hợp lệ. Kết quả được lưu ra tệp "mail_list.txt" Ví dụ: minhanh2000@gmail.com là một địa chỉ email hợp lệ.