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

BÀI THỰC HÀNH LINUX
Bài 03

——————— oOo ———————

Phần 1: Thực hành

1. **Trình soạn thảo Emacs**

   Trong Linux, có một số trình soạn thảo văn bản thường dùng như: nano, vi, gedit... Tuy nhiên, Emacs (*Editor MACroS*) là trình soạn thảo phổ biến nhất. Emacs chạy trên cả giao diện văn bản lẫn đồ họa.

   Các tính năng của Emacs:

   - Soạn thảo trên nhiều cửa sổ (window) và bộ đệm (buffer)
   - Tìm kiếm, thay thế, tự sửa lỗi
   - Soạn thảo đệ quy (recursive edit): cho phép soạn thảo khi một câu lệnh đang thực hiện giữa chừng
   - Nhiều chế độ soạn thảo: văn bản thường, các tệp tin chương trình (tô màu cú pháp và thực hiện từng đoạn mã lệnh), ngôn ngữ đánh dấu (HTML), LaTeX, vẽ hình bằng các kí tự
   - Các macro bàn phím
   - Sửa đổi theo ý thích cá nhân bằng cách chỉnh sửa các biến của chương trình
   - Lập trình bằng ngôn ngữ Emacs Lisp
   - Nhiều chương trình phụ trợ (danh sách thư mục, đọc và soạn e-mail, trò chơi...)

   Để bắt đầu với trình soạn thảo Emacs. Có hai cách sau:

   - *emacs filename*
     Trong đó, filename là tên của tệp tin mà bạn muốn chỉnh sửa. Nếu tệp tin đó chưa có, câu lệnh này sẽ tạo tệp tin filename cho bạn.
   - Nếu tệp tin đã tồn tại, kích chuột phải vào tệp tin và chọn Open with -> GNU Emacs

   Màn hình

   Màn hình Emacs gồm: Vùng soạn thảo, dòng chế độ, tiểu bộ đệm.

   - Dòng chế độ hiển thị các thông tin:
     - Trạng thái bộ đệm: Đã thay đổi (cặp dấu *), chưa thay đổi (cặp dấu -), hay chỉ đọc (cặp dấu %)
     - Tên tập tin đang biên tập
     - Chế độ chính (trong dấu ngoặc)
     - Lượng của tập tin xem trên màn hình
   - Tiểu bộ đệm: Nằm bên dưới dòng chế độ, dùng để hiển thị thông điệp và dấu nhắc yêu cầu gõ lệnh

Một số phím tắt

Ngoài các menu trên giao diện, Emacs còn có rất nhiều những phím tắt. Một số phím tắt cơ bản sau (trong đó, C là Ctrl, M là Alt):

- C-x/C-c: Thoát khỏi Emacs
- C-x/C-f: Tìm và mở tệp tin
- C-x/C-s: Lưu tệp tin
- C-x/C-w: Lưu tệp tin với tên khác
- C-s: Tìm kiếm
- C-x h: Đánh dấu chọn tất cả (Select all)
- C-x u: Khôi phục lại phần vừa xóa (Undo)
- C-w: Cắt (Cut) vùng được đánh dấu
- M-w: Copy vùng được đánh dấu
- C-y: Dán (Paste)
- M-x lệnh: Thực hiện lệnh (lệnh được gõ vào cửa sổ nhỏ ở phía dưới)
  Ví dụ 1: *M-x replace-string* là lệnh thay thế một xâu trong văn bản bằng xâu nhập vào.
  Ví dụ 2: *M-x replace-regexp*: Thay thế một xâu xác định bởi một biểu thức chính quy bằng một xâu khác.

Các chế độ chính trong emacs

Emacs tùy biến lệnh cho các loại văn bản khác nhau thông qua chế độ. Để thiết lập chế độ cho bộ đệm hiện tại, dùng *M-x majormodename-mode*

Ví dụ: Để vào chế độ Java Mode - biên tập mã nguồn Java, gõ *M-x Java-mode*

Các chế độ trong emacs:

- *Fundamental Mode*: Chế độ văn bản, thích hợp cho soạn thảo mọi loại văn bản, chỉ không cung cấp các tính năng đặc biệt.
- *Text Mode*: Soạn thảo văn bản, có các lệnh đặc biệt để kiểm tra chính tả, canh giữa dòng, ..
- *Lisp Mode*: Để biên tập mã nguồn Common Lisp
- *C Mode*: Để biên tập mã nguồn C, có canh lề đặc biệt, ..
- Ngoài ra còn các chế độ: TCL mode, HTML mode, TeX mode, ...

Tập tin, bộ đệm và cửa sổ

Emacs có ba cấu trúc dữ liệu liên quan mật thiết với nhau:

(a) Tập tin là tập tin Linux thực sự trên đĩa. Bạn không bao giờ biên tập trực tiếp trên tập tin này. *Emacs* đọc tập tin vào bộ đệm và viết một bộ đệm vào tập tin để lưu nó

(b) Bộ đệm là cấu trúc dữ liệu nội tại giữ văn bản bạn thực sự biên tập. Emacs có thể có con số bộ đệm bất kì tại bất kì thời điểm nào. Bộ đệm có tên; một bộ đệm xuất phát từ một tập tin gần như luôn luôn mang tên của tập tin đó, và chúng ta nói rằng bộ đệm đang viếng thăm tập tin. Điều này có nghĩa là khi bạn lưu bộ đệm, nó sẽ được lưu vào tập tin đúng. Vào

bất cứ lúc nào cũng chỉ có đúng một bộ đệm được chọn: đây là bộ đệm mà con trỏ phần cứng đang hoạt động và là nơi mà lệnh sẽ có tác dụng. Bộ đệm có thể được xoá theo ý muốn

(c) Cửa sổ là nơi bạn xem bộ đệm. Vì giới hạn thực thể của màn hình, có thể bạn không có chỗ để xem tất cả các bộ đệm cùng lúc. Bạn có thể chia nhỏ màn hình, ngang hay dọc, thành nhiều cửa sổ, mỗi cửa sổ xem một bộ đệm khác nhau. Cũng có thể có vài cửa sổ để xem các phần khác nhau của cùng bộ đệm. Có thể tạo và xoá cửa sổ tuỳ thích; xoá cửa sổ không xoá bộ đệm liên hệ với cửa sổ đó.

Lệnh thao tác trên tập tin

- C-x C-f (find-file): Nhắc nhập tên tập tin và đọc tập tin vào bộ đệm để biên tập. Nếu tập tin đang được biên tập trong một bộ đệm nào đó, nó chỉ chuyển sang bộ đệm đó mà không đọc tập tin
- C-x C-s (save-buffer): Lưu tập tin, hay chính xác hơn là viết bộ đệm hiện tại lên đĩa
- C-x s (save-some-buffers): Lưu tất cả các bộ đệm đang viếng thăm tập tin, truy vấn từng cái và đưa ra một số tuỳ chọn

Lệnh thao tác trên bộ đệm

- C-x b (switch-to-buffer): Nhắc nhập tên bộ đệm và chuyển bộ đệm của cửa sổ hiện tại sang bộ đệm đó. Tạo bộ đệm rỗng mới nếu tên mới được nhập vào.
- C-x C-b (list-buffers): Xuất hiện cửa sổ mới liệt kê tất cả bộ đệm với tên, đã thay đổi hay không, kích thước theo byte, chế độ chính và tập tin mà bộ đệm đang viếng thăm.
- C-x k (kill-buffer): Nhắc nhập tên bộ đệm và dỡ bỏ toàn bộ cấu trúc dữ liệu cho bộ đệm đó khỏi Emacs. Nếu bộ đệm đã thay đổi bạn sẽ có cơ hội lưu nó. Lệnh này không xoá tập tin liên hệ, nếu có.
- C-x C-q (vc-toggle-read-only): Đặt thuộc tính chỉ-đọc hoặc đặt thuộc tính đọc-viết nếu nó đang là chỉ-đọc.

Lệnh thao tác trên cửa sổ

- C-v (scroll-up): Cuộn tới (về cuối tập tin) một màn hình. Theo mặc định, Emacs chừa 2 hàng từ màn hình trước.
- M-v (scroll-down): Như C-v, nhưng cuộn ngược.
- C-x o (other-window): Chuyển sang cửa sổ khác. Lặp đi lặp lại lệnh này sẽ di chuyển qua tất cả các cửa sổ, từ trái sang phải và trên xuống dưới.
- C-x 1 (delete-other-windows): Xoá tất cả các cửa sổ khác, trừ cửa sổ hiện tại. Lệnh này không xoá các bộ đệm và tập tin liên hệ với cửa sổ.
- C-x 0 (delete-window): Xoá cửa sổ hiện tại, thay đổi kích thước các cửa sổ khác cho thích hợp.
- C-x 2 (split-window-vertically): Chia cửa sổ hiện tại thành hai, theo chiều dọc. Lệnh này tạo một cửa sổ mới, nhưng không tạo bộ đệm mới: cùng một bộ đệm sẽ được xem trong 2 cửa sổ. Điều này giúp xem các phần khác nhau của bộ đệm đồng thời.

- C-x 3 (split-window-horizontally): Tương tự như C-x 2 nhưng chia cửa sổ theo chiều ngang.
- C-M-v (scroll-other-window): Tương tự như C-v, nhưng cuộn các cửa sổ kế tiếp (là cửa sổ mà lệnh C-o sẽ chuyển sang).

2. **Trình soạn thảo vi**

vi là trình soạn thảo các tệp tin văn bản trong các hệ thống Linux. Có các ưu điểm sau:

- Màn hình được xem như một cửa sổ mở trên tập tin
- Có khả năng di chuyển con trỏ đến bất kì vị trí nào trên màn hình
- Cửa sổ có thể di chuyển tự do trên tập tin

Phần lớn các phím sử dụng độc lập hoặc kết hợp với các phím *Shift* và *Ctrl* để tạo ra các lệnh của **vi**. Các lệnh của **vi** được gọi khi có dấu **:** ở cuối màn hình.

Các chế độ vi

Để thực hiện các thao tác trong trình soạn thảo vi, có thể thực hiện bằng hai chế độ chính:

- Chế độ soạn thảo: Văn bản được đưa vào trong tài liệu, bạn có thể chèn hoặc bổ sung văn bản.
- Chế độ dòng lệnh: Trong chế độ này, bạn có thể thực hiện các thao tác đơn giản như tìm kiếm, ghi, thoát chương trình hay trộn tài liệu,... ngoại trừ việc nhập văn bản.

Khởi động vi

Khởi động vi dùng cú pháp: *vi filename*

Ví dụ: *student@linux ~ $ vi bai1.txt*

Kết quả là một màn hình soạn thảo sẽ hiện lên. Các dấu ~ trước mỗi dòng cho biết dòng đó còn trống. Dòng dưới cùng cho biết tên tệp tin đang được mở. Nếu như là tệp tin mới thì trạng thái của tệp tin là "[new file]". Nếu tệp tin cũ thì sẽ hiển thị số dòng, số ký tự trong tệp tin.

Chế độ dòng lệnh là chế độ mặc định khi bạn mở hoặc tạo tệp tin mới. Dùng phím **i** hoặc **a** để hiển thị chế độ soạn thảo. Từ chế độ soạn thảo, dùng phím ESC để hiển thị chế độ dòng lệnh.

Lưu và thoát

Để thoát khỏi chế độ của vi, sử dụng lệnh: **:q**.

Để lưu tệp tin, sử dụng lệnh: **:w**.

Khi muốn lưu và thoát, dùng cả hai lệnh: **:wq**.

Khi muốn thoát mà không lưu thay đổi, dùng lệnh: **:q!**

Di chuyển con trỏ

Có thể dùng các phím đơn để di chuyển con trỏ trong vi, ngoài ra có thể dùng các phím mũi tên trên bàn phím.

- h - sang trái 1 ký tự
- l - sang phải 1 ký tự

- k - lên 1 dòng
- j - xuống dưới một dòng
- w - sang phải 1 từ
- b - sang trái 1 từ

Chèn văn bản Trong chế độ dòng lệnh, **i** hoặc **a** cho phép bạn chèn thêm văn bản vào tài liệu. Một số đặc tính khác của việc chèn văn bản trong **vi**:

- a chèn văn bản vào sau vị trí con trỏ hiện tại
- A chèn văn bản vào cuối dòng
- i chèn văn bản vào trước vị trí con trỏ hiện tại
- o chèn văn bản vào dòng mới phía dưới con trỏ hiện tại
- O chèn văn bản vào dòng mới phía trên con trỏ hiện tại
- s xóa kí tự hiện thời và chèn văn bản
- S xóa dòng hiện thời và chèn văn bản

Sao chép/dán

Để sao chép một dòng dùng lệnh **yy**. Để sao chép N dòng dùng lệnh **Nyy**
Ví dụ: Sao chép toàn bộ dòng hiện thời: *yy*
Để dán ta dùng lệnh **p**.

Ghi/mở văn bản

Để ghi văn bản, dùng câu lệnh **:w filename**.
Ví dụ, ghi tài liệu hiện tại ra tệp có tên là newfile: *:w newfile*
Để mở tệp tin ra đọc, dùng lệnh: *:r filename*

Tìm kiếm trong văn bản

Để tìm kiếm, ta dùng các lệnh sau:

- ? tìm trở lên
- / tìm trở xuống

Nhấn 'n' để di chuyển đến kết quả tìm kiếm tiếp theo.
Ví dụ:
/hu là tìm các xâu 'hu' trong văn bản.

Xóa văn bản

Sử dụng các lựa chọn sau:

- x - xóa ký tự tại vị trí con trỏ
- X - xóa ký tự bên trái con trỏ
- dd - xóa 1 dòng
- dw - xóa 1 từ
- 3dw - xóa 3 từ
- 5dd - xóa 5 dòng

Undo/redo

Sử dụng các lựa chọn:

- r - redo

- u - undo

3. **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**
$\hat{}$	Đánh dấu đầu dòng
$	Đánh dấu cuối dòng
.	Biểu diễn ký tự bất kỳ
[]	Biểu diễn một ký tự bất kỳ trong cặp dấu []
[$\hat{}$]	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:
     - [$\hat{}$09] ≡ Tập các ký tự khác 0 và 9
     - $[a - z45]$ ≡ {a, b, c, .., z, 4, 5}
     - $0|([1 - 9][0 - 9]^*)$ ≡ Biểu diễn tập các số tự nhiên
     - $\hat{}tux$ Biểu diễn các dòng bắt đầu bằng *tux*
     - $03(6)(24)\backslash 1 \equiv 036246$

– *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 *ouputFile*)

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

1. Viết các biểu thức chính quy biểu diễn:
   - Tập các số nguyên
   - Tập các số tự nhiên chia hết cho 5
   - Tập các số tự nhiên chia hết cho 2 và có độ dài chia hết cho 3
   - Tập các số nhị phân độ dài 6 và chia hết cho 4
   - Tập các dòng bắt đầu bằng chữ cái 'T' và chứa ít nhất 2 xâu "this is an exercise"
2. Kết hợp *ls* với *grep* để thực hiện:
   - 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*
   - 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. Sử dụng trình soạn thảo *vi* để soạn thảo văn bản có nội dung trong tệp tin *vd01.pdf*. Sau đó tiến hành thực hiện các nội dung sau:
   - Xóa đi các dòng trống trong văn bản
   - Sao chép đoạn đầu tiên và dán vào cuối văn bản
   - Chèn thêm 1 dòng ở đầu văn bản có nội dung: "Bài thực hành VI"
   - Lưu lại văn bản đã sửa
4. Sử dụng trình soạn thảo *emacs* để soạn thảo một đoạn chương trình java có nội dung trong tệp tin *vd02.pdf*. Áp dụng biểu thức chính quy, thay thế tất cả các xâu chú thích *// comment* bằng */* comment */*
5. Sử dụng trình soạn thảo *emacs* soạn thảo đoạn văn bản *html* có nội dung trong tệp tin *vd03.pdf*. Áp dụng biểu thức chính quy, thay thế tất cả các thẻ *<meta content />* bằng thẻ *<meta content > </meta>*

Phần 3: Liên lạc

STT	Họ và tên	Email	ĐT
1	Nguyễn Thị Tâm	nguyenthitam.hus@gmail.com	
2	Trần Thị Hương	tranthihuong.hus@gmail.com	