



Nhập môn hệ điều hành Unix

Nguyễn Hải Châu
Khoa Công nghệ Thông tin
Trường Đại học Công nghệ
Đại học Quốc gia Hà Nội
(Bài giảng tuần 4)



Nội dung

- Lập trình shell trên Unix-Linux
 - shell là gì?
 - Cơ chế pipe
 - Các yếu tố cơ bản để lập trình shell (bash)



shell là gì?

- Là một chương trình thông dịch lệnh cho phép người sử dụng tương tác với hệ điều hành (tương tự command.com trên DOS)
- Có nhiều loại shell trên Unix-Linux:
 - sh (/bin/sh): shell chuẩn, hay Borne shell
 - bash (/bin/bash): GNU Bourne Again Shell
 - tcsh (/bin/tcsh): C shell
 - ksh (/bin/ksh): Korn shell
 - ssh (/bin/ssh): Secure shell
 - ...



Các yếu tố cơ bản của shell



Đặc điểm của shell

- Là chương trình thông dịch lệnh
- Chú thích trong shell có giá trị trên từng dòng lệnh
- Chú thích bắt đầu bằng dấu #
- Chú thích đặc biệt, tại dòng 1 của một chương trình shell chỉ ra chương trình đó sẽ sử dụng thông dịch lệnh sh:
#!/bin/bash



Thực hiện chương trình với shell

- Có nhiều cách:
 - sh** < tên chương trình >
 - sh** < tên chương trình >
 - chmod u+x** < tên chương trình > # Chỉ làm 1 lần
./ < tên chương trình >



Câu lệnh trong shell

- Trên một dòng lệnh shell có thể có một hoặc nhiều câu lệnh
- Một câu lệnh: <tên lệnh> [<tham số>...]
- Nhiều câu lệnh được ghép từ một câu lệnh cách nhau bởi các dấu phân cách ";" hoặc "&&" hoặc "||" hoặc "&"
- Ví dụ: ls -l ; date ; cal



Biến trong shell

- Biến trong shell: Mang giá trị và giá trị có thể thay đổi khi chương trình thực hiện
- Có 3 loại biến:
 - Biến môi trường
 - Biến do người sử dụng tạo ra
 - Biến tự động
- Biến được xác định qua tên của biến đó



Sử dụng biến trong shell

- Tên biến trong shell là một chuỗi ký tự bắt đầu bằng chữ cái hoặc dấu "_": myvar, _x
- Gán giá trị cho biến:
 - <tên biến>=<giá trị>
 - Ví dụ: myuniversity="Đại học công nghệ"
 - Trước và sau dấu = không có khoảng trống
- Sử dụng giá trị của biến:
 - \$<tên biến>
 - Ví dụ: **echo** \$myuniversity
 - echo** -n \$myuniversity # Không xuống dòng



Đọc giá trị biến từ bàn phím

- Cú pháp:
 - read** <tên biến>
- Ví dụ:
 - read** myvar # Đọc giá trị từ bàn phím



Biến môi trường (1)

- Biến môi trường (liên hệ với biến toàn cục trong C/C++)
 - Một số biến đặc biệt do hệ thống tạo ra như \$HOME, \$PATH, \$SHELL, \$PS1, \$PS2
 - Một số khác do người sử dụng tạo ra, được đặt trong tệp \$HOME/.profile
 - Cách tạo biến môi trường của người sử dụng: **export** <tên biến không có \$>=<giá trị biến>
 - Ví dụ: **export** MYNAME="Tux Linux"



Biến môi trường (2)

- Để xem các tên và giá trị các biến môi trường đang có: **env**
- Để xem giá trị của một biến môi trường: **echo** <tên biến môi trường>
- Ví dụ:
 - echo** \$PATH
 - echo** \$HOME
- Lệnh **echo** có thể áp dụng cho tất cả các loại biến của shell



Biến do người sử dụng tạo (1)

- Để tạo một biến, ta dùng lệnh gán giá trị cho biến đó và không cần khai báo biến: < tên biến > = < giá trị >
- Ví dụ: myprog="/home/ab123/hello"
- Để sử dụng giá trị biến: \$ < tên biến >
\$myprog
- Có thể gán giá trị của một biến cho biến khác:
newprog=\$myprog



Biến tự động (1)

- Là các biến do hệ thống tự động tạo ra
- Biến tự động là biến chỉ đọc, tức là chúng ta chỉ được đọc giá trị của biến tự động và không được gán giá trị cho biến tự động
 - Đúng: **echo** \$2
 - Sai: 2="gán giá trị cho biến tự động"
- Các biến tự động: \$0, \$1, ..., \$9, \$#, \$*, \$?



Biến tự động (2)

- Các biến \$0, ..., \$9: \$0 là tên lệnh, \$1, ..., \$9 là giá trị các tham số dòng lệnh
- Biến \$# chứa tổng số các tham số dòng lệnh không tính biến \$0
- Biến \$* là toàn bộ các tham số dòng lệnh được ghép thành 1 chuỗi
- Biến \$? Chứa giá trị kết quả trả lại của câu lệnh trước



Lệnh shift

- Khi ta có hơn 10 tham số dòng lệnh: Sử dụng shift để lấy các tham số từ 10 trở lên
- Cú pháp: **shift** [< số nguyên từ 1..9 >]
- shift** 1 tương đương với **shift**
- Sau khi thực hiện **shift** 3:
 - Giá trị của \$1 được thay bởi giá trị của \$4
 - Giá trị của \$2 được thay bởi giá trị của \$5
 - ...
 - Giá trị của \$9 được thay bởi giá trị của tham số dòng lệnh thứ 13



Ví dụ về lệnh shift

```
chau@home [~]# cat tong
echo `expr $1 + $2`
echo -n "Tham so 1: " $1 ; echo " Tham so 2: " $2
shift 2
echo -n "Tham so 1: " $1 ; echo " Tham so 2: " $2
chau@home [~]# ./tong 1 2 3 4
3
Tham so 1: 1 Tham so 2: 2
Tham so 1: 3 Tham so 2: 4
```



Ví dụ về biến tự động (1)

```
chau@home [~/shell]# ls -l test
-rwxr-xr-x 1 chau chau 146 Oct 1 21:15 test*
chau@home [~/shell]# cat test
echo -n "So luong tham so: "; echo $#
echo -n "Cac tham so: "; echo $*
echo -n "Tham so thu 2: "; echo $2
echo -n "Ten chuong trinh: "; echo $0
chau@home [~/shell]# ./test -l ts1 ts2 "tham so 3"
So luong tham so: 4
Cac tham so: -l ts1 ts2 tham so 3
Tham so thu 2: ts1
Ten chuong trinh: ./test
```



Ví dụ về biến tự động (2)

```

chau@home [~/shell]# ls -l test
-rwxr-xr-x 1 chau chau 146 Oct 1 21:15 test*
chau@home [~/shell]# echo $?
0 ← Kết quả trả lại là 0: Tốt
chau@home [~/shell]# ls -l test123
/bin/ls: test123: No such file or directory
chau@home [~/shell]# echo $?
1 ← Kết quả trả lại là 1: Có lỗi thực hiện lệnh

```



Lấy giá trị cho các biến từ output của một lệnh

- Để lấy giá trị cho biến tự động \$1, ..., \$9: **set** <lệnh>, ví dụ: **set `date`**
- Trong ví dụ trên:
 - **date** in ra Wed Oct 3 19:16:32 MDT 2007
 - Sau khi thực hiện **set `date`**, ta có giá trị các biến tự động: \$1: Web, \$2: Oct, ...
- Lấy giá trị cho biến của người sử dụng: <tên biến>=<lệnh>, ví dụ:



Lấy giá trị cho biến tự động

```

chau@home [~]# date
Wed Oct 3 19:18:32 MDT 2007
chau@home [~]# set `date`
chau@home [~]# echo $1
Wed
chau@home [~]# echo $2
Oct
chau@home [~]# echo $3
3
chau@home [~]# echo $4
19:18:46
chau@home [~]# echo $5
MDT
chau@home [~]# echo $6
2007

```



Lấy giá trị cho biến NSD

```

chau@home [~]# ngaythang=`date`
chau@home [~]# echo $ngaythang
Wed Oct 3 19:22:51 MDT 2007

```



Phép toán với các biến

- Các tính toán trong shell được thực hiện với các đối số nguyên
- Các phép toán gồm có: cộng (+), trừ (-), nhân (*), chia (/), mod (%)
- Tính toán trên shell có dạng: **expr** <biểu thức>



Ví dụ phép toán với biến

```

chau@home [~]# a=1
chau@home [~]# b=4
chau@home [~]# echo `expr $a + $b`
5
chau@home [~]# echo `expr $a+$b`
1+4

```

- **Chú ý:** Giữa các toán hạng \$a, \$b và phép toán + phải có dấu cách



Bài tập

- Sử dụng lệnh **env** để xem giá trị các biến môi trường
- Viết một chương trình shell in ra các thông tin sau: Tên chương trình, số tham số và các tham số dòng lệnh do NSD đưa vào
- Hãy tạo một biến **soluong**, có giá trị là số các tệp/thư mục trong thư mục /etc (Sử dụng lệnh **ls** kết hợp với các lệnh **wc** và **set**)
- Hãy tạo một biến **thumuc** chứa giá trị của thư mục hiện hành



Bài tập

- Hãy lập chương trình **nhan** với các yêu cầu sau: (tương tự với **cong**, **tru**, **chia**)
 - Người sử dụng gõ lệnh
 - **nhan** <số 1> <số 2>
 - Chương trình in ra kết quả phép toán <số 1>*<số 2>
- Hãy xóa giá trị của biến môi trường PATH, sau đó thực hiện một số lệnh như ls, mkdir,... và nhận xét kết quả



Bài tập

- Hãy gán lại giá trị cho biến môi trường PS1 và nhận xét kết quả thu được
- Xem giá trị của biến môi trường HOME, thực hiện lệnh **cd** sau đó thực hiện **pwd**. So sánh kết quả in ra của **pwd** với giá trị của **HOME**. Thay đổi giá trị của **HOME** và lặp lại quá trình trên. Nhận xét kết quả thu được.