
Lập trình Hệ thống

Unix Programming

Part 1: Shell Programming

Nguyễn Quốc Tuấn

Network and Communication System Department
Faculty of Electronics and Communications
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Shell Programming

❑ CÁC YẾU TỐ CƠ BẢN CỦA SHELL

- Chương trình Shell được thực hiện dựa trên file script
 - Một chương trình shell dựa trên các lệnh shell
 - Có rất nhiều loại shell
 - + *Bourne shell, C shell, T shell ..*
 - Dùng tiến trình **vi** để soạn thảo
 - + Chú thích bắt đầu bằng dấu# :
 - + Chú thích đặc biệt chỉ thông dịch lệnh sh: ***#!/bin/bash***
- Dùng lệnh **chmod** để chuyển đặc tính file
 - Có nhiều cách: sh < <tên chương trình>

chmod u+x <tên chương trình> # Chỉ cần làm 1 lần
 - Vận hành file

./<tên chương trình>

Shell Programming

❑ CÁC YẾU TỐ CƠ BẢN CỦA SHELL

▪ Các công cụ dùng lập trình

- Các biến
- Các hàm vào / ra
- Các biểu thức điều kiện
- Các cấu trúc lựa chọn
- Các cấu trúc lặp

Ví dụ:

```
while ( expression )  
    block of statements  
end  
foreach variable ( word list )  
    block of statements  
end
```

```
set variable_name = 'command'  
echo $variable_name
```

```
if ( expression ) then  
    block of statements  
else  
    block of statements  
endif
```

Shell Programming

❑ TRÍCH DẪN

▪ Siêu kí tự cần trích dẫn

- Trích dẫn được sử dụng để bảo vệ các siêu ký tự đặc biệt khỏi việc diễn giải. Có ba phương pháp trích dẫn: dấu gạch chéo ngược, dấu nháy đơn và dấu nháy kép

<i>Metacharacter</i>	<i>Meaning</i>
<i>;</i>	<i>Phân tách lệnh</i>
<i>&</i>	<i>Tiến trình nền</i>
<i>()</i>	<i>Nhóm lệnh; tạo subshell</i>
<i>{ }</i>	<i>Nhóm lệnh, không tạo subshell</i>
<i> </i>	<i>Pipe</i>
<i><</i>	<i>Định hướng lại lối vào</i>
<i>></i>	<i>Định hướng lại lối ra</i>
<i>newline</i>	<i>Lệnh chấm dứt</i>
<i>space/tab</i>	<i>Dấu phân cách trường (word)</i>
<i>\$</i>	<i>Kí tự thay biến</i>
<i>* [1 ?</i>	<i>Siêu ký tự shell để mở rộng tên tệp</i>

Shell Programming

❑ TRÍCH DẪN

▪ Dấu gạch chéo

- Để trích dẫn (hoặc thoát) một ký tự duy nhất khỏi diễn giải. Dấu gạch chéo không được diễn giải nếu được đặt trong dấu nháy đơn và nó bảo vệ ký hiệu đôla (\$), dấu backquote (` `)

```
$ echo Where are you going\?
```

```
Where are you going?
```

```
$ echo Start on this line and \ > go to the next line.
```

```
Start on this line and go to the next line.
```

```
$ echo \
```

```
\
```

```
$ echo '\'
```

```
\
```

```
$ echo '\$5.00'
```

```
\$5.00
```

```
$ echo "\$5.00"
```

```
$5.00
```

Shell Programming

❑ TRÍCH DẪN

▪ Dấu nháy đơn

- Các dấu nháy đơn phải được khớp với nhau. Chúng bảo vệ các siêu kí tự khỏi việc diễn giải. Để in một dấu nháy đơn, nó phải được đặt trong dấu nháy kép hoặc thoát ra bằng dấu gạch chéo.

```
$ echo 'hi there
> how are you?
> When will this end?
> When the quote is matched
> oh'
hi there how are you?
When will this end?
When the quote is matched
oh
$ echo 'Don\'t you need $5.00?'
Don't you need $5.00?
$ echo 'Mother yelled, "Time to eat!"'
Mother yelled, "Time to eat!"
```

Shell Programming

❑ TRÍCH DẪN

▪ Dấu nháy kép

- Dấu nháy kép phải được khớp với nhau, sẽ cho phép thay thế biến và lệnh, đồng thời bảo vệ mọi siêu ký tự đặc biệt khác không bị shell thông dịch.

```
$ name=qtuan
```

```
$ echo "Hi $name, I'm glad to meet you!"
```

```
Hi qtuan, I'm glad to meet you!
```

```
$ echo "Hey $name, the time is `date`"
```

```
Hey qtuan, the time is Mon Oct 13 14:04:11 PST 2004
```

Shell Programming

❑ BIẾN CỦA SHELL

- Các loại biến

- *Biến môi trường*
- *Biến người dùng*
- *Biến tự động*

- **Biến môi trường**

(liên quan với biến toàn cục trong C/C++)

- Một số biến đặc biệt do hệ thống tạo ra như \$HOME, \$PATH, , \$PS1,
- 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"

Shell Programming

❑ BIẾN CỦA SHELL

- Biến môi trường đặc biệt

<i>EDITOR</i>	<i>Chỉ đường dẫn mặc định của tiến trình editor</i>
<i>GROUP</i>	<i>Thông báo nhóm login với người dùng</i>
<i>HOME</i>	<i>Đường dẫn tới thư mục login</i>
<i>HOST</i>	<i>Tên trạm đang login</i>
<i>IFS</i>	<i>Tập các kí tự mà shell hiểu là dấu phân cách trường</i>
<i>MAIL</i>	<i>Đường dẫn tới thư mục chứa mail người dùng</i>
<i>LOGNAME</i>	<i>Tên user truy cập</i>

```
#!/bin/sh
```

```
# This is the first Bourne shell program
```

```
echo "Hello SLOGNAME, it's nice talking to you."
```

```
echo "Your present working directory is `pwd` "
```

```
echo "You are working on machine called `uname -n`."
```

```
echo "Here is a list of your files."
```

```
ls
```

```
echo "Bye for now SLOGNAME. The time is `date +%T`!"
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ Biến môi trường

<i>PATH</i>	<i>Đường dẫn tới thư mục</i>
<i>PS1</i>	<i>Dấu nhắc hệ thống 1</i>
<i>PS2</i>	<i>Dấu nhắc hệ thống 2</i>
<i>TERM</i>	<i>Loại term đang sử dụng</i>
<i>USER</i>	<i>Tên người dùng login</i>
<i>PWD</i>	<i>Đường dẫn thư mục đang làm việc</i>

▪ Biến người dùng (biến cục bộ)

\$ biến = giá trị

\$ set biến = giá trị

- Biến người dùng → môi trường

\$ export biến

Ví dụ

```
#!/bin/sh
```

```
# This is the shell program changing prompt
```

```
PS1="$ (uname -n) ! $ "
```

```
export PS1
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ Biến môi trường

Ví dụ:

```
$ env
```

```
LOGNAME= tuannq
```

```
TERMCAP=sun-cmd
```

```
USER=tuannq
```

```
DISPLAY=:0.0
```

```
SHELL=/bin/sh
```

```
HOME=/home/jody/eHie
```

```
TERM=sun-cmd LD_LIBRARY_
```

```
PATH=/usr/local/bin/lib
```

```
PWD=/home//tuannq
```

```
$ set
```

```
DISPLAY=:0.0
```

```
FMHOME=/usr/local/Frame-2.1X
```

```
FONTPATH=/usr/local/OW3/lib/fonts
```

```
HELPPATH=/usr/local/OW3/lib/local:/usr/local/OW3/lib/help
```

```
HOME=/home/qtuan
```

```
HZ=100
```

```
IFS=
```

```
LANG=C
```

```
LD_LIBRARY_PATH=/usr/local/OHS/lib
```

```
LOGNAME=qtuan
```

```
MAICHECK=600
```

```
MANPATH=/usr/local/OW3/share/man:/usr/local/OW3/man  
:/usr/local/man ; /usr/local/doctools/man:/usr/man
```

```
OPTIND=1
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ Biến môi trường

Ví dụ:

Biến user → env

#!/bin/sh

TERM=wyse

export TERM

NAME="qtuan"

export NAME

echo \$NAME

Hủy biến

unset name

unset TERM

```
PATH=/home/qtuan:/usr/local/OW3/bin;  
/usr/ucb:/usr/local/ doctools/bin;  
/usr/bin:/usr/local ; /etc ; /usr/spool/news/bin;  
/home/qtuan/bin; /usr/lo ; /usr/etc;
```

```
PS1= $
```

```
PS2= >
```

```
PWD=/home/qtuan/btap
```

```
SHELL=/bin/sh
```

```
TERM=sun-cmd
```

```
TERMCAP=sun-cmd; te=\E[>4h; ti=\E[>4] ; tc=sun:
```

```
USER= qtuan
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ Modify biến

- Các biến có thể được kiểm tra và sửa đổi bằng công cụ sửa đổi đặc biệt để kiểm tra xem một biến đã được đặt hay chưa, sau đó gán giá trị cho biến dựa trên kết quả của phép thử

<i><code>\${variable:-word}</code></i>	<i>Nếu biến không rỗng thì = giá trị của nó; nếu không, thay word</i>
<i><code>\${variable:=word} }</code></i>	<i>Nếu biến không rỗng thì = giá trị của nó; nếu không = word</i>
<i><code>\${variable:+word}</code></i>	<i>Nếu biến không rỗng thì = giá trị của nó; nếu không thì rỗng.</i>
<i><code>\${variable:?word}</code></i>	<i>Nếu biến không rỗng thì = giá trị của nó; nếu không in ra và thoát</i>

```
$ fruit=peach
$ echo ${fruit:-plum}
    peach
$ echo ${newfruit:-apple}
    apple
$ echo $newfruit
$ echo $EDITOR
$ echo ${EDITOR:-/bin/vi}
    /bin/vi
```

Shell Programming

❑ BIẾN CỦA SHELL

■ Biến tự động

- 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"

- *Biến tự động: \$1, \$2, ..., \$9: Giá trị tại vị trí các đối (là tên lệnh, \$1,...,\$9 hay là giá trị các tham số dòng lệnh*

- *Biến \$0 tham chiếu shell hiện hành*

Ví dụ : \$cat file1 file2 | echo \$1

Shell Programming

❑ BIẾN CỦA SHELL

- Biến tự động: **\$*** *Danh sách các thông số là toàn bộ các tham số dòng lệnh được ghép thành 1 chuỗi*
- Biến tự động : **\$#** *Số lượng các thông số - chứa tổng số các tham số dòng lệnh không tính biến \$0*
- Biến tự động : **\$\$** *Tên tiến trình đính kèm*
- Biến tự động : **\$@** *Danh sách tham số*
- Biến tự động : **\$?** *Mã trả lại của lệnh thực hiện cuối cùng Chứa giá trị kết quả trả lại của câu lệnh trước*
- Biến tự động : **!** *Tên của tiến trình được đưa ra sau cùng*

Shell Programming

❑ BIẾN CỦA SHELL

- Ví dụ biến tự động

```
$ ls -l test
```

```
-rwxr-xr-x 1 qtuan qtuan 146 Oct 1 21:15 test*
```

```
$ 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
```

```
$ ./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
```


Shell Programming

❑ BIẾN CỦA SHELL

▪ Ví dụ biến tự động

```
$ ls -l test
```

```
-rwxr-xr-x 1 qtuan qtuan 146 Oct 1 21:15 test*
```

```
$ echo $?
```

```
0: Tốt
```

```
$ ls -l test123
```

```
/bin/ls:
```

```
test123: No such file or directory
```

```
$ echo $?
```

```
1 : Có lỗi thực hiện lệnh
```

```
$ echo The pid of this shell is $$
```

```
The pid of this shell is 4725
```

```
$ echo The options for this shell are $-
```

```
The options for this shell are s
```

```
$ grep tuannq /etc/passwd
```

```
$ echo $?
```

```
1
```

```
$ sleep 25&
```

```
4736
```

```
$ echo S!
```

```
4736
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ *Biến người dùng*

- Tên biến trong shell là 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

▪ Gán giá trị biến từ bàn phím

- *Cú pháp:*

\$ read answer *Đọc một dòng từ stdin và gán nó cho biến answer*

\$ read first last *Đọc một dòng từ stdin đến khoảng trống đầu tiên hoặc dòng mới, đặt word đầu tiên vào biến **first** và phần còn lại vào biến **last***

Shell Programming

❑ BIẾN CỦA SHELL

▪ Thay thế lệnh

- Khi một lệnh được đặt trong (backquotes) – dấu dưới phím ESC (`), nó sẽ được thực thi và kết quả trả về. Nó được sử dụng khi gán đầu ra của một lệnh cho một biến hoặc khi thay thế đầu ra của một lệnh trong một chuỗi.

```
$ name= `awk -F: '{print $1}' database`
```

```
$ echo $name
```

```
Nam
```

```
$ lich = `date`
```

```
$ echo $lich
```

```
Fri Oct 22 09:35:21 PDF 2020
```

```
$ grep -n Fri lich
```

```
error!!!
```

Shell Programming

❑ BIẾN CỦA SHELL

▪ Ví dụ biến người dùng

(The Script)

```
#!/bin/sh
```

```
# Scriptname: qtuan
```

```
$ echo "Are you happy? \c"
```

```
$ read answer
```

```
$ echo "Sanswer is the right response."
```

```
$ echo "What is your full name? \c"
```

```
$ read first middle last
```

```
$ echo "Hello Sfirst"
```

(The Output)

*Are you happy? **Yes***

Yes is the right response.

*What is your full name? **Tuan Nguyen Quoc***

Hello Tuan

Shell Programming

❑ BIẾN CỦA SHELL

▪ **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

Shell Programming

❑ BIẾN CỦA SHELL

▪ Lệnh shift

Ví dụ:

\$ 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

\$./tong 1 2 3 4 3

Tham so 1: 1

Tham so 2: 2

Tham so 1: 3

Tham so 2: 4

Shell Programming

❑ BIẾN CỦA SHELL

- Đặt biến vị trí cho các biến/đối

- Lệnh `set`<lệnh>``
`set` đối

ví dụ: `$ date`

Wed Oct 3 19:16:32 MDT 2020

`$ set`date``

→

`$ echo $1`

Web,

`$ echo $2`

Oct

`$ echo $#`

6

Shell Programming

❑ BIẾN CỦA SHELL

- Ví dụ:

```
$ set word1 word2 word3  
$ echo $1 $2 $3  
$ set apples peaches plums  
$ shift  
$ echo $1  
$ echo $2  
$ echo $*
```


Shell Programming

❑ CÁC PHÉP TOÁN SHELL

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

Chú ý: Giữa các toán hạng \$a, \$b và phép toán + -/ phải có dấu cách
Sinh viên tự tìm hiểu lệnh **bc***

Shell Programming

❑ CÁC PHÉP TOÁN SHELL

▪ Ví dụ:

```
$ expr 1 + 4
5
$ expr 1+4
1+4
$ expr 5 + 9/3
8
$ expr 5 * 4
expr: syntax error
5 $ expr 5 \ 4 - 2
18
$ expr 11 % 3
2
$ num= 1
$ num= `expr $num + 1`
$ echo $num
2
```

```
$ n = `echo M "scale=3; 13 / 2" | bc `
$ echo $n
6.500
$ n="bc « EOF scale=3 13/2 EOF"
$ echo $n
6.500
$ products = `awk -v x=2.45 -v y=3.123
'BEGIN { printf "%.2f\n", x+t} `
$ echo $product
7.65
```

Shell Programming

❑ CÁC PHÉP TOÁN SHELL

■ Phép toán **kiểm tra chuỗi**

<i>string1 = string2</i>	<i>Xâu1 bằng xâu2 (có khoảng trống quang dấu =)</i>
<i>string1 != string2</i>	<i>Xâu1 khác xâu2</i>
<i>string</i>	<i>Xâu not null</i>
<i>-z string</i>	<i>Độ dài xâu zero</i>
<i>-n string</i>	<i>Độ dài xâu nonzero</i>

■ Phép toán **kiểm tra số nguyên**

<i>int1 -eq int2</i>	<i>int1 bằng int2</i>
<i>int1 -ne int2</i>	<i>int1 không bằng int2</i>
<i>int1 -gt int2</i>	<i>int1 lớn hơn int2</i>
<i>int1 -ge int2</i>	<i>int1 lớn hơn hoặc bằng int2</i>
<i>int1 -lt int2</i>	<i>int1 is bé hơn int2</i>
<i>int1 -le int2</i>	<i>int1 bé hơn hoặc bằng int2</i>

Các phép toán kiểm tra thường dùng là điều kiện cho các lệnh for, while, loop ...

Shell Programming

❑ CÁC PHÉP TOÁN SHELL

- Ví dụ:

```
# Nếu đi với lệnh test
$ name=qtuan
$ grep "$name" /etc/passwd
    qtuan:8ZKX2F:5102:40: Tuan Nguyen Quóc:/home/torn:/bin/ksh 3
$ echo $?
    0      Success!
$ test $name != qtuan
$ echo $?
    1      Failure
$ [ $name = qtuan]  # Brackets replace the test command
$ echo $?
    0      Success
$ [ $name = [Tt]?m ] # Wildcards are not evaluated by the test command
$ echo $?
    1      Failure
```

Shell Programming

❑ CÁC PHÉP TOÁN SHELL

- **Kiểm tra logic**
 - expr1 -a expr2* *Logical AND*
 - expr1 -o expr2* *Logical OR*
 - !expr* *Logical NOT*

- **Kiểm tra file**

- b filename Block special file*
- c filename Character special file*
- d filename Directory existence*
- f filename Regular file existence and not a directory*
- g filename Set-group-ID is set*
- p filename File is a named pipe*
- r filename File is readable*
- s filename File is nonzero size*
- u filename Set-user-ID bit is set*
- w filename File is writable*
- x filename File is executable*

Các phép toán kiểm tra thường dùng là điều kiện cho các lệnh for, while, loop ...

Shell Programming

❑ CÁC PHÉP TOÁN SHELL

- Ví dụ:

```
test -n $word      hoặc  [ -n $word ]  
test torn = qtuan  hoặc  [ torn = qtuan ]
```

```
$ echo "Are you okay (y/n) ?"  
$ read answer  
$ if [ "Sanswer" = Y -o "$answer" = y ] then  
    echo "qtuan to hear it."  
fi
```

```
$ ls > lfile  
$ ls -d *  
$ echo tên file  
$ read ten  
$ [ $ten -d]  
$ echo $?
```

Thanks