

-----oOo-----

Phần 1: Thực hành

a. Biến môi trường

Linux Shell xác định các biến để điều khiển môi trường của người sử dụng đối với mỗi phiên sử dụng. Việc đặt các biến này sẽ xác định với hệ thống những tham số như thư mục nào được sử dụng làm thư mục chính, nơi đặt mail, những thư mục nào được sử dụng mặc định khi bạn gọi đến các lệnh Linux,... Một số biến hệ thống có thể được đặt trong tệp khởi động (startup file) và được đọc khi bạn đăng nhập. Trong tệp khởi động có thể đặt những câu lệnh Linux, và những lệnh này sẽ được thực hiện khi bạn login

Xem tất cả biến môi trường hiện có

env

Xem giá trị của 1 biến môi trường cụ thể

echo \$ten_bien_moi_truong

- Ví dụ:
student@linux ~\$ echo \$HOSTNAME -- Hiển thị tên máy tính
Kết quả: linux
- *student@linux ~\$ echo \$HOME* -- Hiển thị thư mục mặc định của người dùng hiện tại
Kết quả: /home/student

Đặt giá trị cho biến môi trường

Cú pháp: *export ten_bien_moi_truong=gia_tri_cua_bien_moi_truong*

- Ví dụ
student@linux ~\$ export TAILIEU=/home/student/tailieu
student@linux ~\$ echo \$TAILIEU
Kết quả: /home/student/tailieu

b. Biến trong Bash Shell

Giống như các ngôn ngữ lập trình khác, Bash shell có sử dụng biến, nhưng khác là không cần phải khai báo kiểu.

Cách gán giá trị cho biến

Cú pháp: *ten_bien=gia_tri*

(Chú ý: Không có dấu cách trong cú pháp này)

- Ví dụ: File test.sh
#!/bin/bash
var=100
echo \$var

Sử dụng Bash Shell rất linh hoạt, không những có thể gán 1 giá trị cụ thể vào biến và nó còn cho phép gán kết quả của một câu lệnh cho biến

Cú pháp: *ten_bien=`cau_lenh`* (Chú ý đây là ký tự ``` nằm ở dưới phím ESC chứ không phải ký tự nháy đơn `'`)

- Ví dụ: File test.sh
#!/bin/bash
var=`pwd`
echo \$var

Để tăng tính tương tác, bạn có thể viết shell script cho phép yêu cầu nhập giá trị cho biến ngay khi chạy nó. Sử dụng câu lệnh `read` để đạt được điều này.

Cú pháp: *read ten_bien*

- Ví dụ:
#!/bin/bash
echo "Ban ten gi: "
read ten
echo "Chao ban \$ten"

Bash shell không cho phép chúng ta thực hiện tính toán trực tiếp như những ngôn ngữ khác, do nó không phân biệt kiểu của biến. Thay vì đó, nó cung cấp lệnh `expr` khi chúng muốn thực hiện việc tính toán trên biến.

Cú pháp: *expr \$bien_1 + \$bien_2*

- Ví dụ:

student@linux ~\$ expr \$a + \$b

Đặt trong file shell script test.sh

#!/bin/bash

var=`expr 1+3`

echo The result is \$var

Lưu ý: Phải có khoảng trắng giữa toán tử và toán hạng. Nếu ở trên đổi thành 1+3 hoặc 1+3 hoặc 1 +3, thì đều không cho kết quả hoặc có thông báo sai.

Đối dòng lệnh

Đối dòng lệnh là các giá trị được đưa vào như là 1 tùy chọn (option) ở câu lệnh chạy.

Ví dụ: *./test.sh a b c*

a,b,c là các đối dòng lệnh. a là đối dòng lệnh thứ nhất, b là đối dòng lệnh thứ 2, c là đối dòng lệnh thứ 3,...

Trong bash shell, chúng ta có thể lấy giá trị đối dòng lệnh theo cách sau:

#!/bin/sh

echo "Ten tep [\$0]"

echo "Doi dong lenh thu nhat [\$1]"

echo "Doi dong lenh thu hai [\$2]"

echo "Doi dong lenh thu ba [\$3]"

Ở ví dụ trên, \$0 là 1 biến đặc biệt, nó cho phép ta lấy ra tên tệp chứa biến này. \$1, \$2,\$3 tương ứng là giá trị của các đối dòng lệnh thứ 1,2,3.

c. Các toán tử trong lập trình shell

- So sánh số

-eq	==
-ne	!=
-lt	<
-le	<=
-gt	>
-ge	>=

- So sánh chuỗi

string1 = string2	Chuỗi string1 bằng chuỗi string2
string1 != string2	Chuỗi string1 khác chuỗi string2
string1	Chuỗi string1 không null hoặc không xác định
-n string1	Chuỗi string1 không null và đã tồn tại
-z string	Chuỗi string1 là null và đã tồn tại

- Kiểm tra file hoặc thư mục

-s file	Không phải là file trống
-f file	Là file đã tồn tại hoặc là một file bình thường, không phải là thư mục
-d dir	Là một thư mục đã tồn tại và không phải là một file
-w file	Là file cho phép ghi
-r file	Là file chỉ cho phép đọc
-x file	Là file cho phép thực thi

- Các phép toán logic

!	NOT
-a	AND
-o	OR

d. Cấu trúc rẽ nhánh

- **if - else**

Cú pháp của if-else trong bash shell cũng tương tự như những ngôn ngữ khác

```
if [ điều kiện ]
then
    <lệnh_cần_chạy>
elif [ điều_kiện ] then <lệnh_cần_chạy>...
else <lệnh_cần_chạy>
fi
```

Ví dụ: So sánh 2 số a và b

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

```
echo "Nhập số a:"
```

```

read a
echo "Nhap so b:"
read b
if [ $a -lt $b ]
then
    echo "so a nho hon so b."
elif [ $a -eq $b ]
then
    echo "so a bang so b."
else
    echo "so a lon hon so b."
fi

```

- **case**

Với 1 bài toán có nhiều trường hợp, ta có thể sử dụng cấu trúc case để giải quyết. Cấu trúc rẽ nhánh này tương tự như cấu trúc switch ... case trong C

Cú pháp:

```

case <biến>
in
    giá_trị_1)
        <lệnh>
    giá_trị_2)
        <lệnh>
    giá_trị_3)
        <lệnh>
    ...
    *) #còn lại
    exit
esac

```

Ví dụ:

```

#!/bin/bash
case $1
in
    1) echo "Ban vua chon option 1"
        exit;;
    2) echo "Ban vua chon option 2"

```

```
exit;;  
3) echo "Ban vua chon option 3"  
exit;;  
*) echo "Exit";  
exit;;
```

esac

e. Vòng lặp for

Vòng lặp for có thể được sử dụng theo 2 dạng sau:

Dạng 1: Sử dụng mệnh đề “in” để chỉ ra danh sách các giá trị của biến

Cú pháp:

```
for <biến> in <danh sách>  
do  
    <các lệnh cần thực hiện>  
done
```

Ở đây, giá trị của biến sẽ lần lượt được gán bằng các giá trị có trong danh sách.

Ví dụ:

```
#!/bin/bash  
i=1  
for day in Mon Tue Wed Thu Fri  
do  
    echo "Weekday $((i++)): $day"  
done
```

Kết quả:

```
./for_loop.sh  
Weekday 1: Mon  
Weekday 2: Tue  
Weekday 3: Wed  
Weekday 4: Thu  
Weekday 5: Fri
```

Ví dụ:

```
#!/bin/bash  
for i in 1 2 3 4 5 6  
do  
    echo "$i"
```

done

Kết quả:

\$/for_loop.sh

1

2

3

4

5

6

Chú ý: Danh sách không được đặt trong dấu nháy kép “”, nếu đặt trong dấu nháy kép thì tất cả các giá trị trong danh sách sẽ được hiểu là một giá trị của biến. Mỗi giá trị trong danh sách cách nhau một dấu cách.

Hoặc

#!/bin/bash

i=1

weekday="Mon Tue Wed Thu Fri"

for day in \$weekday

do

echo "Weekday \$((i++)): \$day"

done

Kết quả:

\$/for_loop.sh

Weekday 1: Mon

Weekday 2: Tue

Weekday 3: Wed

Weekday 4: Thu

Weekday 5: Fri

Dạng 2: Sử dụng vòng lặp for tương tự như trong Java

Cú pháp:

for ((giá_trị_khởi_tạo; điều_kiện_dừng; bước_nhảy))

do

<các_lệnh_cần_thực_hiện>

done

Ví dụ:

```
#!/bin/bash
for (( i = 1; i <= 5; i++ ))
do
    echo "In ra lan thu $i"
done
```

Kết quả:

```
./for_loop2.sh
In ra lan thu 1
In ra lan thu 2
In ra lan thu 3
In ra lan thu 4
In ra lan thu 5
```

f. Vòng lặp While**Cú pháp:**

```
while [ điều kiện ]
do
    <câu lệnh cần thực hiện>
done
```

Ví dụ:

```
#!/bin/bash
i=1
while [ $i != 5 ]
do
    echo "In ra lan $i"
    i=`expr $i + 1`
done
```

Phần 2: Bài tập thực hành

- Bài 1

Viết 1 shell script in ra màn hình các thông tin sau:

- Xâu: "I am a student"
- Thư mục hiện hành

- Tập tin và thư mục, kể cả các thư mục ẩn trong thư mục hiện hành
- Ngày và giờ hiện tại

- Bài 2:

Viết 1 shell script khi chạy cho phép người dùng nhập tên file, nếu file đó tồn tại thì sẽ in ra nội dung của file, nếu không hiển thị thông báo “File ten_file không tồn tại”

- Bài 3:

In ra màn hình câu: hoặc Chào buổi sáng , hoặc chào buổi trưa, hoặc chào buổi tối, tùy thuộc vào thời điểm hiện tại

Gợi ý: Lệnh date +%H sẽ in ra giờ hiện tại.

- Bài 4:

Viết 1 shell script in ra màn hình nội dung:

Xin moi ban chon hanh dong?

1. Xem dung luong su dung cua may tinh
2. Xem noi dung thu muc hien hanh
3. Xem cac tien trinh dang chay tren may tinh duoi dang cay
4. Xem ten nguoi dung dang nhap he thong

Khi người dùng nhập vào lựa chọn thì màn hình sẽ hiển thị kết quả tương ứng.

Gợi ý: Sử dụng câu lệnh free để xem hiện tại máy tính đang sử dụng RAM như thế nào

	total	used	free	shared	buffers	cached
Mem:	1027672	939560	88112	43652	132460	500160
-/+ buffers/cache:		306940	720732			
Swap:	0	0	0			

- Bài 5:

Viết 1 shell liệt kê tất cả các file và thư mục trong /Desktop. Liệt kê tất cả các file có đuôi .sh?

- Bài 6:

Viết 1 shell tạo ra file có tên tailieu.txt có nội dung: “Tuan 07: Cac toan tu trong lap trinh shell va Vong lap”. Sau đó, copy nội dung của tailieu.txt thành 3 file khác nhau có kèm số thứ tự ở cuối file.

VD: Sau khi chạy script sẽ có thêm 3 file giống hệt lần lượt là : tailieu_1.txt, tailieu_2.txt, tailieu_3.txt.

- Bài 7:

Viết 1 shell tạo ra 4 file như sau:

- file1.txt có nội dung: Toi la sinh vien 1
- file2.txt có nội dung: Toi la sinh vien 2
- file3.txt có nội dung: Toi la sinh vien 3
- file4.txt có nội dung: Toi la sinh vien 4

- Bài 8:

Viết 1 shell để đổi tên tất cả các file có phần mở rộng là .doc sang .txt trong thư mục /home/student

Phần 3: Liên lạc

STT	Họ và tên	Email
1	Nguyễn Minh Hải	nguyenminhhai06@gmail.com
2	Nguyễn Thị Tâm	nguyenthitam.hus@gmail.com
3	Trần Thị Hương	tranthihuong.hus@gmail.com