



HỆ ĐIỀU HÀNH

LAB 02 | SHELL SCRIPT (Cont.)

Learn how to write shell script to automate your task in Linux OS.



MỤC TIÊU

Sau LAB này, sinh viên sẽ có thể:

1. Áp dụng cấu trúc điều kiện và cấu trúc điều khiển để viết chương trình phức tạp
2. Giải quyết vấn đề máy tính bằng cách sử dụng tập lệnh shell



NỘI DUNG

THỰC HÀNH

7. Cấu trúc điều kiện

8. Cấu trúc vòng lặp



CẤU TRÚC ĐIỀU KIỆN

07.



7. Cấu trúc điều kiện

- Khái niệm cơ bản trong bất kỳ ngôn ngữ lập trình nào là điều khiển chương trình dựa trên các kịch bản hoặc điều kiện khác nhau
- Tập lệnh shell có thể kiểm tra mã lỗi trả về của bất kỳ lệnh hoặc tập lệnh nào khác Chúng ta nên trả về mã lỗi bằng cách sử dụng **exit** ở cuối chương trình.



7. Cấu trúc điều kiện

if condition

then

commands

else

commands

fi

if condition; then

commands

elif condition

commands

fi



7. Cấu trúc điều kiện

test condition

[*condition*]

space

space

```
if test -f count.h
```

```
then
```

```
...
```

```
fi
```

```
if [ -f count.h ]
```

```
then
```

```
...
```

```
fi
```



7. Cấu trúc điều kiện

Điều kiện trên chuỗi

Sự so sánh	Kết quả
[string1 = string2]	true if <i>string1</i> is exactly the same as <i>string2</i>
[string1 != string2]	true if <i>string1</i> is different from <i>string2</i>
[-n string1]	true if length of <i>string1</i> is non-zero
[-z string1]	true if length <i>string1</i> is zero



7. Cấu trúc điều kiện

So sánh toán học

Sự so sánh	Kết quả
[<code>expr1 -eq expr2</code>]	<i>true</i> if <i>expr1</i> equals <i>expr2</i>
[<code>expr1 -ne expr2</code>]	<i>true</i> if <i>expr1</i> does not equal <i>expr2</i>
[<code>expr1 -gt expr2</code>]	<i>true</i> if <i>expr1</i> is greater than <i>expr2</i>
[<code>expr1 -ge expr2</code>]	<i>true</i> if <i>expr1</i> is greater than or equal to <i>expr2</i>



7. Cấu trúc điều kiện

So sánh toán học

Sự so sánh	Kết quả
[<code>expr1</code> -lt <code>expr2</code>]	true if <code>expr1</code> is less than <code>expr2</code>
[<code>expr1</code> -le <code>expr2</code>]	true if <code>expr1</code> is less than or equal to <code>expr2</code>
[! <code>expression</code>]	true if <code>expression</code> is false (operator <i>not</i>)



7. Cấu trúc điều kiện

Điều kiện trong file

Sự so sánh	Kết quả
[-d file]	<i>true</i> if <i>file</i> is a directory
[-f file]	<i>true</i> if <i>file</i> is a file
[-e file]	<i>true</i> if <i>file</i> exists on disk



7. Cấu trúc điều kiện

Điều kiện trên file

Sự so sánh	Kết quả
[-s file]	<i>true</i> if <i>file</i> size is not zero
[-r file]	<i>true</i> if <i>file</i> is readable
[-w file]	<i>true</i> if <i>file</i> is writable
[-x file]	<i>true</i> if <i>file</i> is executable



7. Cấu trúc điều kiện

Bài tập trên lớp: 01

Xác thực sinh viên - Phần 1

Viết một chương trình để yêu cầu người dùng nhập tên và mã số sinh viên của họ, sau đó kiểm tra:

- Nếu đầu vào khớp với thông tin của bạn (tên và ID) In lời chào
- Nếu đầu vào không khớp với thông tin của bạn, In "Access denied " và kết thúc chương trình.

7. Cấu trúc điều kiện

Ví dụ: Viết một chương trình để chuyển đổi điểm dựa trên điểm sang lớp dựa trên chữ cái.

Mẫu: Đầu vào: 700 | Đầu ra: B +

Table 1. Chuyển đổi lớp

Điểm	Tương đương		Mô tả tiếng Việt	Mô tả tiếng Anh
[900, 1000]	A+	4.0	Xuất sắc	Outstanding
[800, 900)	A	3.5	Giỏi	Excellent / Very Good
[700, 800)	B+	3.0	Khá	Good
[600, 700)	B	2.5	Trung Bình	Average
[500, 600)	C	2.0	Yếu	Marginal
< 500	D/F	=<1.0	Kém/Không đạt/Trượt	Fail



7. Cấu trúc điều kiện

Trình bày đa điều kiện

Logic	Sử dụng tùy chọn	Sử dụng toán tử
and	-a	&&
or	-o	



7. Cấu trúc điều kiện

Trình bày đa điều kiện

Mẫu: multi_condition.sh

```
1 #!/bin/sh
2
3 read -p "Please input your VN's grade: " grade
4
5 if [ $grade -lt 500 ]
6 then
7     echo "Your letter grade is F"
8 elif [ $grade -ge 500 -a $grade -lt 600 ]
9 then
10    echo "Your letter grade is C"
11 elif [ $grade -ge 600 ] && [ $grade -lt 700 ]
12 then
13    echo "Your letter grade is B"
14 elif [ ($grade) ]
15 fi
```

```
if [ cond1 -a cond2 ]
then
    ...
fi
```

```
if [ cond1 ] && [ cond2 ]
then
    ...
fi
```




7. Cấu trúc điều kiện

Bài tập trên lớp: 02

Chuyển đổi điểm số

Sử dụng thông tin trong Bảng 1 để viết một chương trình chuyển đổi điểm dựa trên điểm thành điểm dựa trên chữ cái. Đầu vào sẽ được truyền từ dòng lệnh.



CẤU TRÚC VÒNG LẶP

8.1. Vòng lặp for

08.



8.1. Vòng lặp for

for variable in list

do

commands

done

- **biến số**: Chỗ dành sẵn cho mỗi mục trong danh sách.
- **danh sách**: Danh sách các mục (chuỗi, số, tên tệp, v.v.) được phân tách bằng dấu cách.



8.1. Vòng lặp for

Ví dụ

```
for fruit in apple banana cherry  
do  
    echo "I like $fruit"  
done
```

```
I like apple  
I like banana  
I like cherry
```



8.1. Vòng lặp for

Ví dụ

```
for i in $(seq 1 3)
do
    echo "Number $i"
done
```

Number 1

Number 2

Number 3



8.1. Vòng lặp for

Ví dụ

Lặp qua các tệp trong một thư mục

`$(ls dir)`

```
for file in *
```

```
do
```

```
    echo "Processing $file"
```

```
    # process something here
```

```
done
```

Try it yourself!



8.1. Vòng lặp for

Ví dụ

Vòng lặp lồng nhau

```
for i in 1 2 3
do
    for j in a b c
    do
        echo "Combination $i$j"
    done
done
```

Try it yourself!



8.1. Vòng lặp for

Vòng lặp kiểu C (chỉ áp dụng trên Bash shell)

```
#!/bin/bash
```

```
for ((i=1; i<=5; i++))  
do  
    echo "Number $i"  
done
```




CẤU TRÚC VÒNG LẶP

8.2. Vòng lặp while

08.



8.2. Vòng lặp while

while condition

do

commands

done



8.2. Vòng lặp while

Ví dụ

```
counter=1

while [ $counter -le 5 ]
do
    echo "Counter: $counter"
    counter=$((counter + 1))
done
```

```
Counter 1
Counter 2
Counter 3
Counter 4
Counter 5
```



8.2. Vòng lặp while

Ví dụ

Đọc dòng từ tệp

```
filename="examples.txt"
```

```
while IFS= read -r line
```

```
do
```

```
    echo "Read line: $line"
```

```
    # process the line
```

```
done < "$filename"
```

Try it yourself!

IFS (Internal Field Separator) là một biến đặc biệt trong các hệ điều hành giống Unix, bao gồm sh shell (và bash), **xác định ký tự hoặc ký tự được sử dụng để chia đầu vào thành các trường**. Nó chủ yếu được sử dụng trong câu lệnh shell để tách chuỗi và xử lý đầu vào.



8.2. Vòng lặp while

Bài tập trên lớp: 03

Xác thực sinh viên– Phần 2

Thay đổi **Bài tập trên lớp 01**:

- Dùng **while loop** để cho phép người dùng nhập lại cho đến khi tên và ID chính xác.
- Dùng **for loop** để người dùng dùng thử 5 lần trước khi bị từ chối.



BÀI TẬP

03.

3. Bài tập

Bài tập : 01

Cho file [gradebook.csv](#) trong đó có điểm của một lớp. Nghiên cứu cách sử dụng IFS để đọc file csv. Sử dụng shell script để:

- Đọc ID sinh viên từ dòng lệnh
- Sử dụng ID sinh viên để tìm và in tên, lớp gốc và điểm dựa trên chữ cái của học sinh tương ứng



3. Bài tập

Bài tập : 02

Viết một kịch bản shell để cho phép người dùng nhập tuổi của họ, sau đó:

- Kiểm tra xem đầu vào có phải là một số không và giá trị của nó nằm trong khoảng từ 0 đến 100.
- Nếu đầu vào nhỏ hơn 12 => In "Bạn là một đứa trẻ"
- Nếu đầu vào từ 12 – 18 => In "Bạn là một thiếu niên"
- Nếu đầu vào lớn hơn 18 => In "Bạn là người lớn"



3. Bài tập

Bài tập: 03

Viết chương trình đọc 02 đối số từ dòng lệnh: đối số đầu tiên là một **chuỗi** trong khi đối số thứ hai là một **đường dẫn**.

Kiểm tra xem chuỗi đó có tồn tại trong bất kỳ file văn bản nào không (ví dụ: test.txt) trong đường dẫn được cung cấp. In dòng chứa chuỗi.



3. Bài tập

Bonus: 01

Nghiên cứu cách cài đặt lệnh git, sử dụng git để download thư mục image từ đây: https://github.com/locth/OS_LAB2_IMG.git

Viết kịch bản để thực hiện các tác vụ sau::

- Kiểm tra thư mục của người dùng. Nếu các thư mục PNG và JPG không tồn tại, hãy tạo hai thư mục này.
- Di chuyển tất cả các tệp PNG từ thư mục hình ảnh ở trên sang thư mục PNG. Hiển thị số lượng hình ảnh PNG trên màn hình.
- Di chuyển tất cả các tệp JPG từ thư mục hình ảnh ở trên sang thư mục JPG. Hiển thị số lượng hình ảnh JPG trên màn hình.



3. Bài tập

Bonus: 02

Tạo tệp văn bản có tên **subject.txt** Trong đó có một danh sách các mã khóa học cho sinh viên trong học kỳ này, với mỗi mã trên một dòng riêng biệt. Viết kịch bản để thực hiện các tác vụ sau:

- Yêu cầu người dùng nhập họ và tên của họ, sau đó tạo một thư mục được đặt tên theo thông tin mà người dùng vừa nhập.
- Đọc thư mục **subject.txt** trong thư mục vừa tạo trong phần a và đối với mỗi mã khóa học, hãy tạo một thư mục có tên theo mã khóa học đó