

# Hệ điều hành mã nguồn mở

GV: Ths. Lê Ngọc Kim Khánh



# Lập trình Shell

- 🍎 Đặc điểm của Shell Scripting
- 🍎 Tầm quan trọng của Shell scripting
- 🍎 Các loại Shell trong Linux
- 🍎 Tham biến Shell
- 🍎 Các lệnh kiểm tra điều kiện
- 🍎 Cấu trúc điều khiển trong Shell
- 🍎 Thông dịch

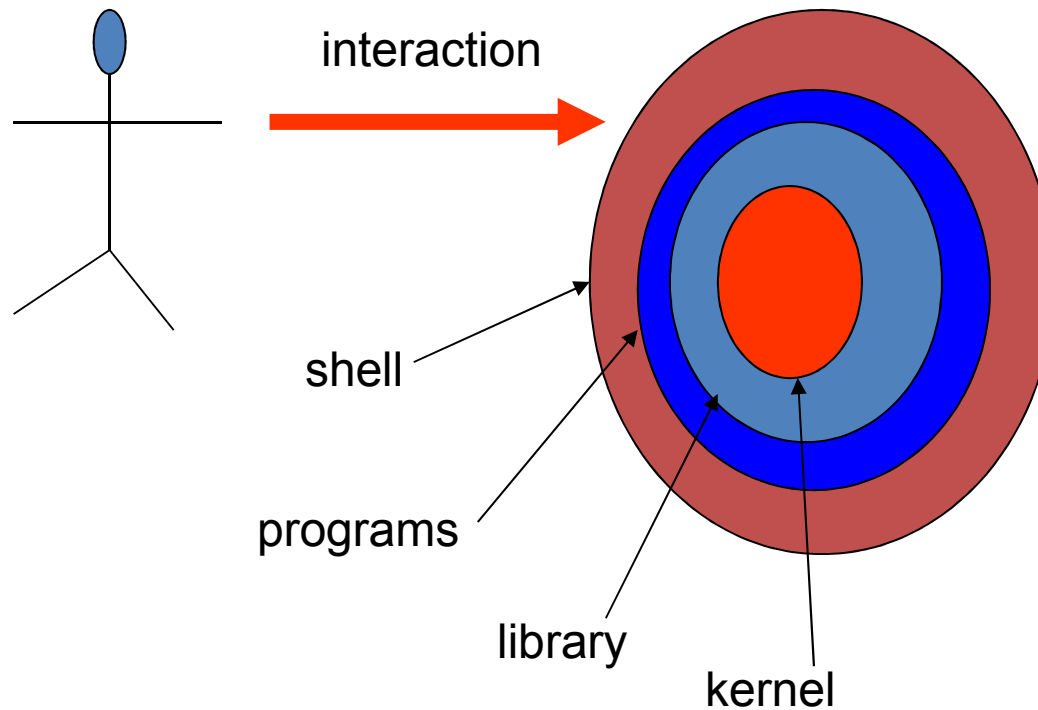


# Đặc điểm của Shell scripting

- 🍎 Automating commonly used commands
- 🍎 Performing system administration and troubleshooting
- 🍎 Creating simple application
- 🍎 Manipulation of text or files



# Tâm quan trọng của Shell scripting



# Ví dụ

🍏 `#!/bin/bash`  
🍏 `clear`  
🍏 `echo "Hello $USER"`  
🍏 `echo -n "Today is"`  
🍏 `date "+%A %d %B %Y"`  
🍏 `echo -n "There is/are"`  
🍏 `who | wc -l`  
🍏 `echo connection on $HOSTNAME`  
🍏 `echo "Calendar" ; cal`  
🍏 `exit 0`



# Các loại Shell trong Linux

- 🍎 Bash shell (Bourne Again Shell), thường được biết những shell có đuôi .sh
- 🍎 ksh (Korn Shell) của David Korn
- 🍎 csh (C Shell), của đại học Berkeley, và tcsh là version cải tiến của csh



# Tham biến Shell

- 🍏 Biến thông thường
- 🍏 Biến môi trường
- 🍏 Tham số



# Tham biến Shell

## 🍎 Biến thông thường

- Không cần phải khởi tạo biến trước khi sử dụng
- Mặc định, giá trị trong biến luôn luôn là kiểu chuỗi
- Tên biến phân biệt hoa thường.
- Xuất biến ra màn hình, dùng lệnh **echo**
- Lấy giá trị của biến: **echo \$test**
- Gán giá trị cho biến:  
✓ **test** = "wellcome"





# Tham biến Shell

## 🍎 Biến thông thường

- Phân biệt các VD sau:
  - ✓ `text="Monday"`
  - ✓ `echo $text` → Monday
  - ✓ `echo "Today is $text"` → Today is Monday
  - ✓ `echo `Today is $text`` → Today is \$text
  - ✓ `echo "Today is \ $text"` → Today is \$text



# Tham biến Shell

## 🍎 Biến môi trường:

- Các biến được khai báo sẵn và gán giá trị mặc định khi shell được khởi động
- Thường được viết hoa
- Xem danh sách biến môi trường:  
**env** hay **printenv**
- Tạo biến môi trường:  
**export para\_name=para\_value**



# Tham biến Shell

## 🍎 Tham số:

- `bash my-script Hanoi Paris Bordeaux "Ho Chi Minh City"`
  - ✓ `$0="my-script"`
  - ✓ `$1="Hanoi",`
  - ✓ `$2="Paris",`
  - ✓ `$3="Bordeaux",`
  - ✓ `$4="Ho Chi Minh City"`
  - ✓ `$*='Hanoi Paris Bordeaux "Ho Chi Minh City"'`
  - ✓ `$#=4` → số lượng tham số



# Các lệnh kiểm tra điều kiện



## Lệnh test:

- Return 0 for true
- Return 1 for false
- Sử dụng:
  - ✓ test -f name : Kiểm tra name có phải là tập tin hay không?
  - ✓ test -d name : Kiểm tra name có phải là thư mục hay không?
  - ✓ test String1=String2 : so sánh chuỗi
  - ✓ test String1 != String2 : so sánh chuỗi
  - ✓ test EXPR1 op EXPR2 : so sánh biểu thức với operation (op): -eq (equal) -ne (not equal) -lt (lesser than) -le (lesser or equal) -gt (greater than) -ge (greater or equal).



# Các lệnh kiểm tra điều kiện

🍏 Có thể thay thế lệnh **test** bằng **[]**

🍏 **VD:**

▪ test -f abc.txt      ↔      [ -f abc.txt ]



# Cấu trúc điều khiển trong Shell



```
if condition
then
    statement
fi
```

```
if condition; then
    statement
fi
```



# Cấu trúc điều khiển trong Shell

**VD:**

```
if [ -d $r ]  
then  
    echo "The directory $r exists!"  
fi
```



# Cấu trúc điều khiển trong Shell



```
if condition
then
    statement
elif condition
then
    statement
else
    statement
fi
```





# Cấu trúc điều khiển trong Shell

**VD:**

```
if [ -r $r ]  
then  
    echo "You can read file $r "  
elif [ -x $r ]  
then  
    echo "File $r can be executed"  
else  
    echo "Anything else"  
fi
```



# Cấu trúc điều khiển trong Shell

🍏 **for**

```
for variables in  
    values  
do  
    statements  
done
```

🍏 VD:

```
for a in {1..20}  
do  
    host= 192.168.100.$a  
    ping -c2 $host  
done
```



# Cấu trúc điều khiển trong Shell

## 🍏 While

```
while condition  
do  
    statements  
done
```

## 🍏 VD:

```
a=0  
while [ $a -lt 10 ]  
do  
    echo $a  
    let a=$a+1  
done
```



# Cấu trúc điều khiển trong Shell

🍏 **until**

```
until condition  
do  
    statements  
done
```

🍏 VD:

```
a=0  
until [$a -gt 10]  
do  
    echo $a  
    let a=a+1  
done
```



# Cấu trúc điều khiển trong Shell

## **case**

```
case variables in  
    pattern) statements;;  
    pattern) statements;;  
    ...  
esac
```

## VD:

```
n=0  
read n  
case $n in  
    1) echo "Spring";;  
    2) echo "Summer";;  
    3) echo "Autumn";;  
    4) echo "Winter";;  
esac
```



# Thông dịch

🍏 Một bash shell luôn luôn bắt đầu bằng:

**#! /bin/bash** hay **#! /bin/sh**

🍏 Chạy một bash shell

- Cách 1:

- ✓ Gán quyền execute cho file:

- ```
chmod +x my-script
```

- ✓ Chạy bằng 1 trong 2 câu lệnh sau :

- ```
./my-script
```

- Cách 2:

- ```
bash my-script
```

