

# Các hệ thống nguồn mở

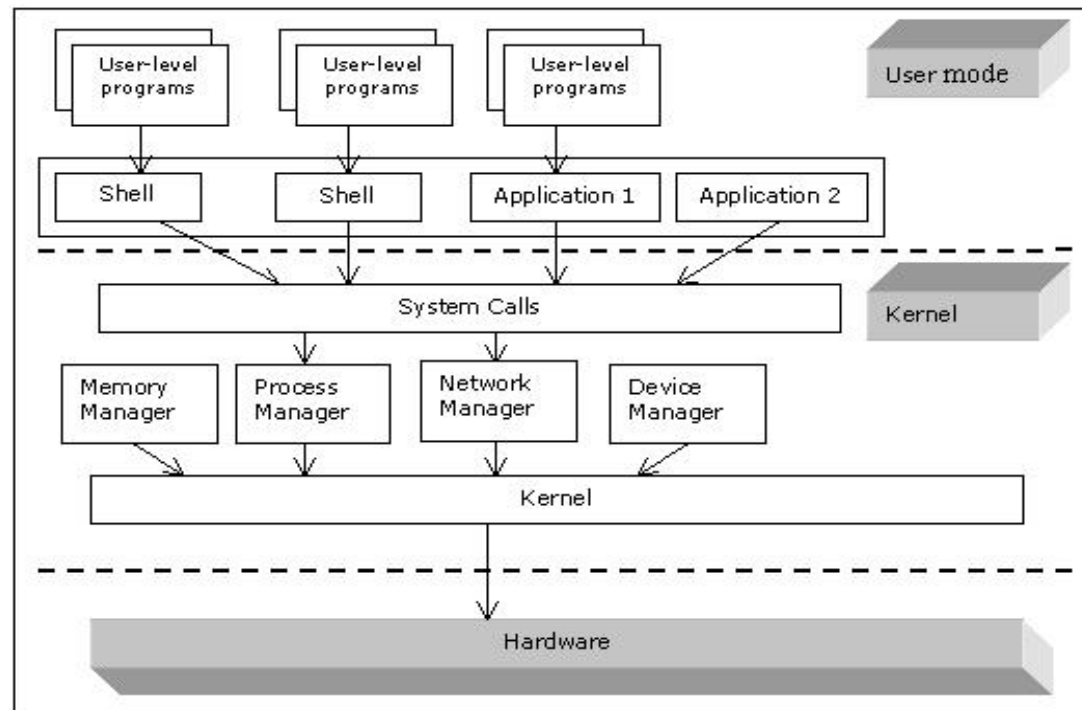
Shell Script

# Mục tiêu

- Vai trò của Linux Shell
- Quá trình thông dịch lệnh Shell
- Khởi tạo biến Shell
- Biến môi trường Shell
- Viết Shell Script

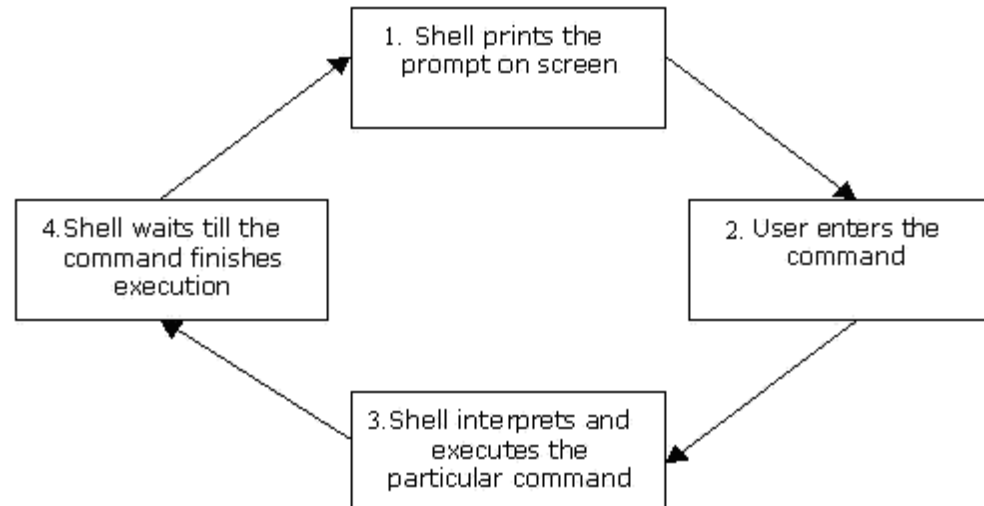
# Đại cương về Shell

- Hệ Shell:
  - Giúp tương tác giữa người dùng và hệ thống
  - Kiến trúc tương tác với hệ Shell như hình dưới



# Thông dịch lệnh Shell

1. Đọc lệnh
2. Tìm file thực thi và tham số
3. Load file
4. Thực thi



# Shell Script

- Là một file text
- Lưu các lệnh Shell tuần tự
- Cho phép hệ Shell đọc và thực thi các lệnh
- Cho phép quản lý biến, quản lý các cấu trúc lập trình.

# Lệnh echo

- Hiển thị một chuỗi lên màn hình.
- Chuỗi có thể đặt trong nháy đơn hoặc kép.
- Mặc định lệnh echo sẽ tự xuống dòng.
- Không xuống dòng: tham số **-n**

```
$ echo "This is an example of the echo command" ↵  
    This is an example of the echo command  
$ _
```

# Thực thi Shell Script

- File script cần có quyền thực thi.

Ví dụ với file script.sh

```
chmod u+x script.sh
```

```
. script.sh
```

# Khởi tạo biến

- Biến Shell Script:
  - Không cần khai báo kiểu.
  - Được xem nghiêm ngặt là chuỗi
  - Có thể tính toán được.
  - Không cần khai báo tường minh.
  - Tự cấp phát sau lần truy xuất đầu tiên.
- Phép gán: <Tên Biến>=GiáTrị
- Khi script kết thúc: Biến được hủy



# Đọc giá trị từ biến và vào biến

- Dùng ký hiệu \$

Bien1=\${Bien2}

- Ký hiệu ngoặc để giới hạn tên biến
- Đọc giá trị từ bàn phím và biến: **read**  
\$ read <Tên biến>

# Biến cục bộ và toàn cục

```
$ continent=Africa
```

```
$ echo "$continent"
```

```
Africa
```

```
$ bash
```

[Creates a new shell]

```
$ echo "$continent"
```

[There is no response]

```
$ continent=Asia
```

[Gives new value Asia to

```
continent]
```

```
$ echo "$continent"
```

```
Asia
```

```
Press <Ctrl> d
```

```
$ exit
```

# Các biến môi trường

- HOME: Thư mục home của user
- PATH: Chứa các biến đường dẫn
- PS1: Ký hiệu dấu nhắc (\$)
- PS2: Ký hiệu dấu nhắc (>)
- LOGNAME: Tên login
- SHLVL: Cấp độ shell
- SHELL: Hệ shell mặc định

# Tính toán trên biến

- Lệnh **expr** dùng để tính toán số học
- Ví dụ:

**\$ expr 4 + 5**

- Lệnh **test** và **[]** dùng để ước lượng biểu thức logic
- Cú pháp:  
**test expression** hoặc **[ expression ]**
- Kết hợp nhiều biểu thức bằng **-a** hay **-o**
- Phép nhân sử dụng dấu **\\***

# Câu lệnh if

- Câu lệnh if

```
if <condition>
then <command(s)>
[else <command(s)>]
fi
```
- Lệnh if...elif

```
if condition(s)
then command(s)
elif condition
then command(s)
else command(s)
fi
```

# Lệnh exit

- Dừng thực thi Script và quay về dấu nhắc
- Ví dụ về lệnh exit:

```
echo "Do you wish to quit?"  
read ans  
if [ $ans = "y" ]  
then exit  
fi
```

# Câu lệnh case

- Cú pháp:

```
case $variable-name in
    value1) command
        ...
        command;;
    value2) command
        ...
        command;;
    *) command;;
esac
```

# Câu lệnh while

- Câu lệnh while

```
while <condition>  
do  
    <command (s)>  
done
```

- Ví dụ:

```
reply=y  
while test "$reply" != "n"  
do  
    echo -n "Enter file name?"  
    read fname  
    cat ${fname}  
    echo -n "wish to see more files :"  
    read reply  
done
```



# Câu lệnh for

- Lặp lại trên từng giá trị trong dãy

- Cú pháp:

```
for variable name in  
    <list_of_values>  
do  
    ...  
done
```

- Ví dụ:

```
for name in Ruby Samuel  
do  
    echo "${name}"  
done
```

# Câu lệnh break và continue

- Ví dụ:

```
while true
do
    echo "Enter choice"
    echo "(press 'q' to exit)"
    echo "1 date          2 who"
    echo "3 ls           4 pwd"
    read choice
    case $choice in
        1) date;;
        2) who;;
        3) ls;;
        4) pwd;;
        q) break;;
        *) echo "That was not one of the
            choices";;
    esac
done
```

# Câu lệnh test

- **Câu lệnh test dùng kiểm tra file và thư mục**

```
echo -n "Enter file name: "  
read fname  
if test -f $fname  
    then echo "$fname is an ordinary  
           file"  
elif test -d $fname  
    then echo "$fname is a directory  
           file"  
elif test -s $fname  
    then echo "$fname is a not an empty  
           file"  
elif test ! -r "$fname"  
    then echo "No readable file called  
           $fname exists"  
fi
```

# Hàm trong Shell Script

- An example of a function is shown below:

```
function ftype
{
    if test -f $fname
    then echo "$fname is an ordinary file."
    elif test -d $fname
    then echo "$fname is a directory file."
    elif test ! -r "$fname"
    then echo "No readable file called $fname
        exists"
    fi
}
echo "Enter a file name"
read fname
ftype
```

- Shell scripting also allows you to pass arguments to function.
- The syntax to pass arguments to a function is:  
`<function name> [ arg1 arg2 ... ]`

# Ví dụ Shell Script

## Kiểm tra chuỗi đối xứng

```
echo "Enter string"
read str
len=`echo $str | wc -c`
l=`expr $len / 2`
ctr=1
flag=0
while test $ctr -le $l
do
    a=`echo $str | cut -c$ctr`
    b=`echo $str | cut -c$len`
    if test $a != $b
    then flag=1
        break
    fi
    ctr=`expr $ctr + 1`
    len=`expr $len - 1`
done
if test $flag -eq 0
then echo "String is palindrome"
else echo "String not a palindrome"
fi
```

# Ví dụ Shell Script

## Đảo chuỗi

```
echo "Enter string"
read str
len=`echo $str | wc -c`
while test $len -ne 0
do
    temp_char=`echo $str | cut -c$len`
    rev_str=${rev_str}${temp_char}
    len=`expr $len - 1`
done
echo $rev_str
```

# Mảng trong shell script

- Mảng:
  - Tập hợp các giá trị dữ liệu được lưu trữ qua một tên chung
  - Giống như biến: không cần khai báo trước khi sử dụng
  - Được tự động tạo ra qua lần dùng đầu tiên
  - Giá trị có thể rỗng
  - Có thể gán nhiều giá trị thông qua 1 câu lệnh
  - Có thể thực hiện các phép toán và xử lý chuỗi trên nó
- Khai báo thông qua lần gán đầu tiên:
  - `<array_name>[xx]=<value>`

# Mảng ...

- Ví dụ:
  - Newarr=(giatri1 giatri2 giatri3)
  - echo \${Newarr[1]}
  - Newarr[2]=giatri2moi
  - echo Newarr[2]
  - Newarr[4]=15
  - Newarr[6]=5
  - Newarr[7]=`expr \${Newarr[4]} + \${Newarr[6]}`
  - echo \${Newarr[@]:2:2}



# BT

- Viết shellscript nhập vào một số  $n$ , sau đó in ra các số nguyên tố từ 1 đến  $n$