

-----oOo-----

## Phần 1: Thực hành

### a. Makefile

Makefile là gì?

+ Makefile là một file dạng script chứa các thông tin:

- Cấu trúc project (file, sự phụ thuộc)
- Các lệnh để tạo file

+ Lệnh make sẽ đọc nội dung Makefile, hiểu kiến trúc của project và thực thi các lệnh

Ví dụ:

- Project chứa 4 file: Shape.java, Rectangle.java, Square.java, Main.java
- 2 đối tượng Square và Rectangle cùng cài đặt giao diện Shape
- File thực thi là Main.java

Nội dung các file lần lượt như sau:

File Shape.java

```
public interface Shape{ double getArea();}
```

File Square.java

```
public class Square implements Shape{  
    private double a;  
    public Square(double a) {this.a = a;}  
    public double getArea(){return a*a;}  
}
```

File Rectangle.java

```
public class Rectangle implements Shape{  
    private double a, b;  
    public Rectangle(double a, double b) {this.a = a;this.b = b;    }  
    public double getArea(){return a*b;}  
}
```

File Main.java

```
import java.util.ArrayList;  
public class Main{
```

```

        public static void main(String[] args){
            ArrayList<Shape> shapes = new ArrayList<Shape>();
            shapes.add(new Square(5));
            shapes.add(new Rectangle(2, 3));
            System.out.println(shapes.get(0).getArea());
            System.out.println(shapes.get(1).getArea());
        }
    }
}

```

File Makefile có nội dung sau:

```

Main: main.class
    java Main
main.class: Main.java Rectangle.class Square.class
    javac Main.java
Rectangle.class: Rectangle.java Shape.class
    javac Rectangle.java
Square.class: Square.java Shape.class
    javac Square.java
Shape.class: Shape.java
    javac Shape.java

```

### Thứ tự thực hiện:

Khi bạn thực hiện lệnh make, chương trình make sẽ nhảy đến phần đầu tiên là Main với mục đích để tạo ra nó, để làm được điều đó make đi kiểm tra lần lượt nội dung sau dấu: xem tệp tin main.class đã tồn tại chưa. Tệp đầu tiên là main.class chưa có, cần phải tìm câu lệnh nào đó mà ở đó main.class được tạo ra, make tìm ra dòng thứ 3 và nó nhảy đến thực hiện câu lệnh ở dòng này để tạo ra main.class. Sau khi tạo ra main.class, make tiến hành các bước tương tự. Sau khi tất cả các tệp liên quan được tạo ra, make mới có thể tạo ra tệp chạy cuối cùng là Main.

Vậy make thực hiện theo thứ tự như sau:

- + Tạo ra các file object trước (main.class)
- + Tạo ra chương trình nhị phân cuối cùng từ các file object đã được tạo ra trước đó (sum)

Sau khi đã tạo các file với nội dung như trên, vào thư mục chứa các file trên thực hiện lệnh:

```
student@linux ~$make
```

Kết quả:

25

6.0

## b. Mảng trong Shell

- Cú pháp khai báo mảng

*Array\_name = (value1 value2 ... valueN)*

*Array\_name = ([0]=value1 [1]=value2 ... [N]=valueN)*

Hoặc:

*Array\_name[index]=value*

Ví dụ:

*NAME[0]="Zara"*

*NAME[1]="Luca"*

- Hoặc có thể dùng lệnh *read* để nhập tất cả các phần tử trong mảng:

*read -a array*

- Số phần tử của mảng:

- *\${#array\_name[@]}*

- *\${#array\_name[\*]}*

- Truy cập vào phần tử trong mảng

*\${Array\_name[index]}*

Ví dụ: tạo tệp *vd1.sh*

*#!/bin/bash*

*NAME[0]="Zara"*

*NAME[1]="Luca"*

*NAME[2]="Mary"*

*NAME[3]="Tonny"*

*echo "First name in an array: \${NAME[0]}"*

*echo "Second name in an array: \${NAME[1]}"*

- Truy cập tất cả các phần tử từ phần tử thứ n:

- *\${array\_name[@]:n}*

- Truy cập từ phần tử thứ n đến phần tử thứ m trong mảng:

- *\${array\_name[@]:n:m}*

- Truy cập tất cả các phần tử của mảng
  - `${Array_name[$*]}`
  - `${Array_name[@]}`

Ví dụ:

```
for i in "${array[@]}"
do
    #access each element as $i. . .
done
```

## B. Bài tập thực hành

1. Viết chương trình shell nhập một mảng số nguyên từ bàn phím. Thực hiện các yêu cầu sau:
  - a. Tính và in ra tổng các phần tử trong mảng
  - b. Tìm phần tử lớn nhất và nhỏ nhất trong mảng
  - c. Tìm và in ra tất cả các số nguyên tố trong mảng
  - d. Tìm và in ra tất cả các số chính phương trong mảng