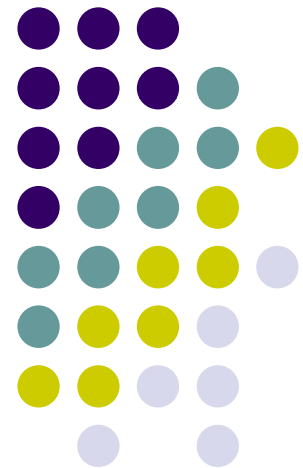


Ổng dẫn

- Khái niệm
- Ổng dẫn trong Java
- Ví dụ về ồng dẫn trong Java

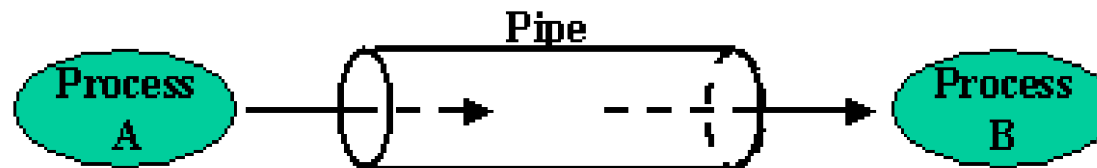


Khái niệm về ống dẫn

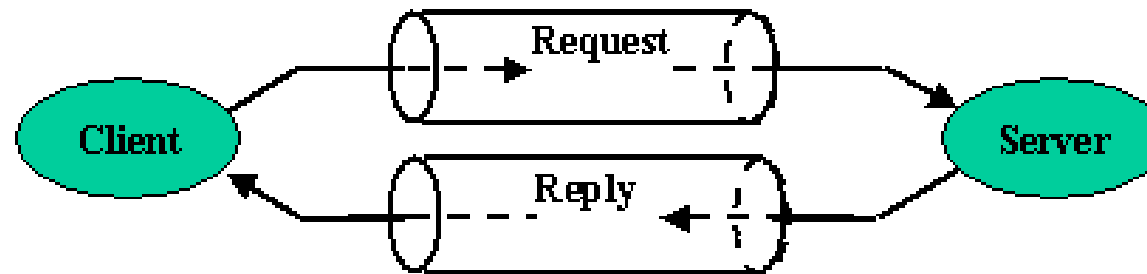


- **Ống dẫn là gì?**

- ❖ Là một tiện ích được hỗ trợ trong hầu hết các hệ điều hành đa nhiệm.
- ❖ Cho phép 2 quá trình **trên cùng 1 máy** có thể trao đổi dữ liệu với nhau.
- ❖ Dữ liệu đi trên ống dẫn theo một chiều nhất định.



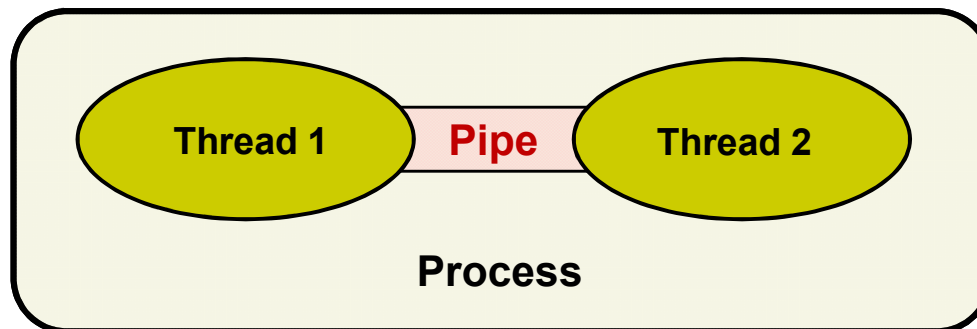
- ❖ Xây dựng ứng dụng Client- Server bằng cách sử dụng 2 ống dẫn:



Phân loại ống dẫn



- Ống dẫn có tên (Named Pipe)
 - ❖ Cho phép hai quá trình có không gian địa chỉ khác nhau nhưng phải **trên cùng một máy** giao tiếp với nhau.
 - ❖ Giống như 1 tập tin: dữ liệu sẽ được lấy ra ở đầu tập tin và được thêm vào ở cuối tập tin.
- Ống dẫn bình thường (Normal Pipe)
 - ❖ Giới hạn trong phạm vi không gian địa chỉ của 1 quá trình.
 - ❖ Giao tiếp giữa các thread trong 1 quá trình.



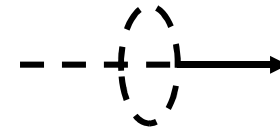
Java hỗ trợ
ống dẫn
bình thường

Ống dẫn trong Java

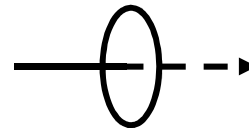


- Java hỗ trợ tiện ích ống dẫn thông qua hai lớp:
 - ❖ `java.io.PipedInputStream`: thừa kế từ lớp `InputStream`.
 - ❖ `java.io.PipedOutputStream`: thừa kế từ lớp `OutputStream`.
- Các bước tạo ống dẫn

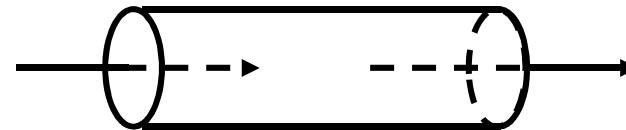
`PipedInputStream` daudoc = new `PipedInputStream`();



`PipedOutputStream` daughi = new `PipedOutputStream`();



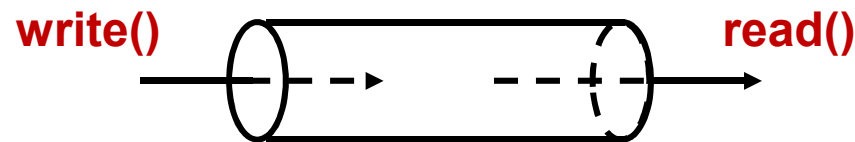
daudoc.**connect**(daughi);



Ổng dẫn trong Java



- Sử dụng Pipe trong Java:
 - ❖ **java.io.PipedInputStream**: thừa kế từ lớp `InputStream`.
 - Có 3 phương thức `read()` : đọc dữ liệu từ Pipe.
 - Phương thức `close()`: đóng pipe và giải phóng tài nguyên.
 - ❖ **java.io.PipedOutputStream**: thừa kế từ lớp `OutputStream`.
 - Có 3 phương thức `write()` : ghi dữ liệu vào Pipe.
 - Phương thức `close()`: đóng pipe và giải phóng tài nguyên.



Ví dụ ống dẫn trong Java



- Dịch vụ phản hồi thông tin (Echo Service)
 - ❖ Xây dựng theo mô hình Client-Server
 - ❖ Server
 - Chờ nhận các byte gửi đến từ Client
 - Khi nhận byte nào sẽ gửi trả lại đúng byte đó cho Client.
 - ❖ Client
 - Gửi từng ký tự đến Server.
 - Nhận kết quả trả về.

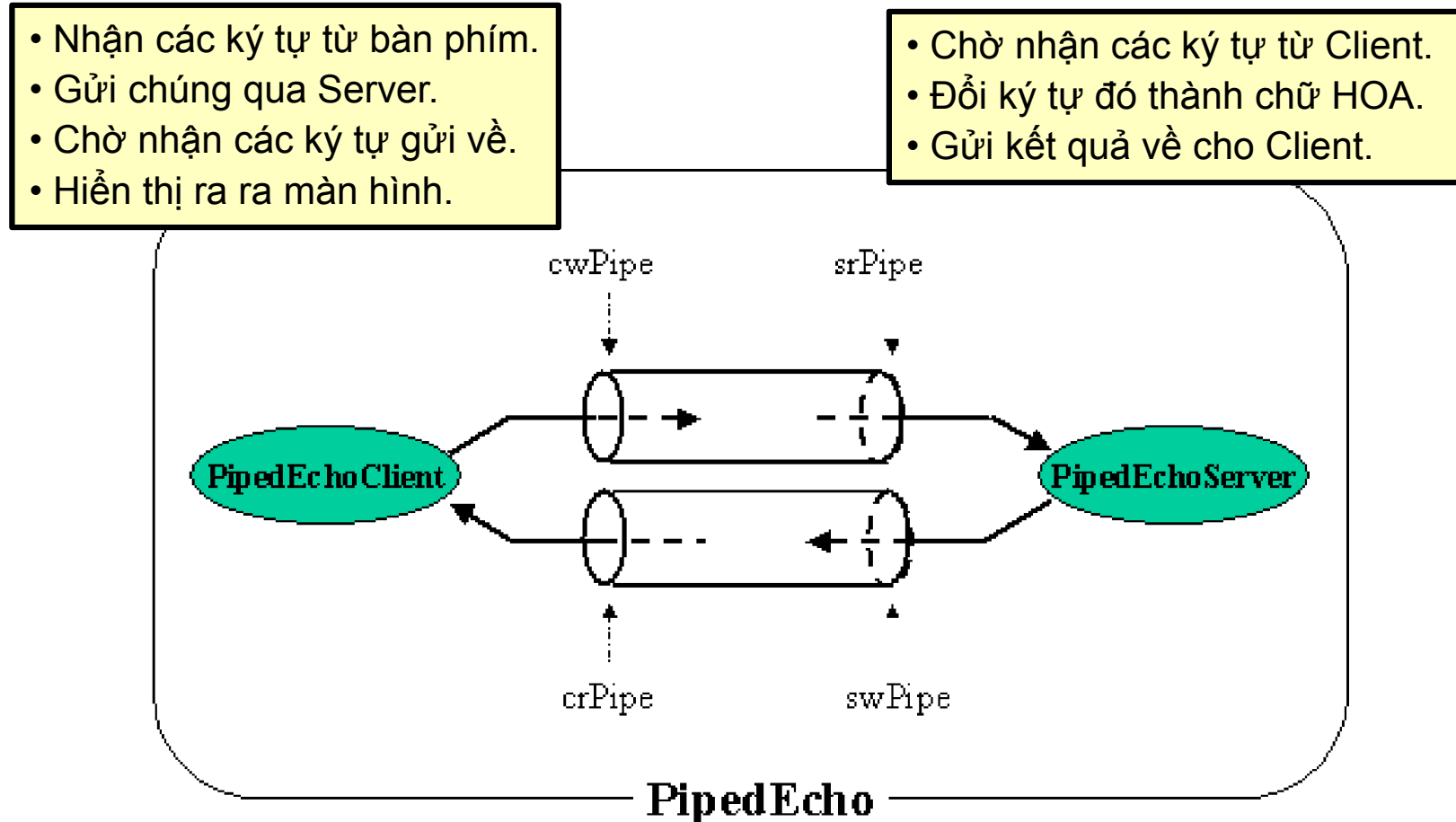


Xây dựng một ứng dụng có tên là PipeEcho mô phỏng dịch vụ phản hồi thông tin sử dụng Pipe làm phương tiện giao tiếp giữa các Thread trong ứng dụng.

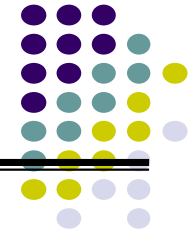
Ví dụ ống dẫn trong Java



- Mô hình ứng dụng PipeEcho



Ví dụ ống dẫn trong Java



```
import java.io.*;
public class PipedEchoServer extends Thread {
    PipedInputStream readPipe;
    PipedOutputStream writePipe;
    PipedEchoServer(PipedInputStream readPipe, PipedOutputStream writePipe){
        this.readPipe = readPipe;
        this.writePipe = writePipe;
        System.out.println("Server is starting . . .");
        start();
    }
    public void run(){
        while(true) {
            try{
                int ch = readPipe.read();
                ch = Character.toUpperCase((char)ch);
                writePipe.write(ch);
            }
            catch (IOException ie) { System.out.println("Echo Server Error: " + ie ); }
        }
    }
}
```

Ví dụ ống dẫn trong Java



```
import java.io.*;
public class PipedEchoClient extends Thread {
    PipedInputStream readPipe;
    PipedOutputStream writePipe;
    PipedEchoClient(PipedInputStream readPipe, PipedOutputStream writePipe){
        this.readPipe = readPipe;
        this.writePipe = writePipe;
        System.out.println("Client creation");
        start();
    }
    public void run(){
        while(true) {
            try {
                int ch=System.in.read();
                writePipe.write(ch);
                ch = readPipe.read();
                System.out.print((char)ch);
            }
            catch(IOException ie){
                System.out.println("Echo Client Error: "+ie );
            }
        }
    }
}
```

Ví dụ ống dẫn trong Java



```
import java.io.*;
public class PipedEcho {
    public static void main(String argv[]){
        try{
            PipedOutputStream cwPipe = new PipedOutputStream();
            PipedInputStream  crPipe = new PipedInputStream();
            PipedOutputStream swPipe = new PipedOutputStream(crPipe);
            PipedInputStream  srPipe = new PipedInputStream(cwPipe);
            PipedEchoServer server = new PipedEchoServer(srPipe,swPipe);
            PipedEchoClient client = new PipedEchoClient(crPipe,cwPipe);
        } catch(IOException ie) {
            System.out.println("Pipe Echo Error:"+ie);
        }
    }
}
```

```
C:\WINNT\system32\cmd.exe - java PipedEcho
D:\UduJava>javac PipedEchoServer.java
D:\UduJava>javac PipedEchoClient.java
D:\UduJava>javac PipedEcho.java
D:\UduJava>java PipedEcho
Server is starting . . .
Client creation
Day la chuoai nhap thu nhap
DAY LA CHUOI NHAP THU NHAT
Dong thu hai nhap tu ban phim ...
DONG THU HAI NHAP TU BAN PHIM ...
```