

## 高度情報処理演習B

### 入出力とネットワーク

#### 入出力とは

- JavaVMがOSに要求したメモリ以外のデバイスからデータを入力/出力する
  - ファイル, ネットワーク, USBなど
- Java1.4以前
  - java.ioパッケージ
- Java1.4以後
  - java.nioパッケージの追加

#### Javaの入出力を学ぶ

- java.ioパッケージ
  - バイトストリーム
  - 文字ストリーム
- java.nioパッケージ
  - ByteBuffer, Channel
- Fileコピーの性能比較
- ノンブロッキングIO

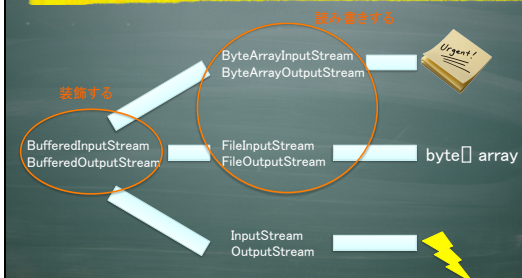
#### java.ioパッケージ

- バイトストリーム
  - バイトデータを直接扱う
  - InputStream/OutputStream
- 文字ストリーム
  - データを文字として扱う
  - Reader/Writer
- お互いに対応するストリームがある

#### 読み書きできるストリームと装飾するストリーム

- 読み書きできるストリーム
  - File(Input/Output)Stream
  - ByteArray(Input/Output)Stream
  - FileReader/FileWriter
  - StringReader/StringWriter
- 装飾するストリーム
  - その他

#### 読み書きできるストリームと装飾するストリーム



## 読み込みの例

- バッファして効率よく読み込む

```
BufferedInputStream in = new BufferedInputStream(
    new FileInputStream("data.dat"));
in.read(buffer, 0, buffer.length);
```

- StreamとReaderの組み合わせ

```
BufferedReader reader = new BufferedReader(
    new InputStreamReader(
        new FileInputStream("test.txt")));
String line = reader.readLine();
```

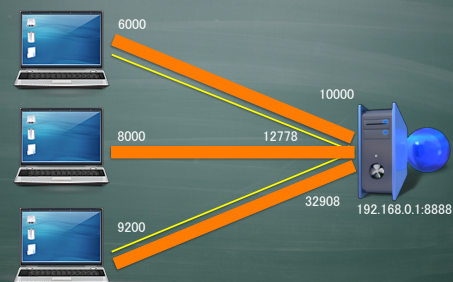
## IOの演習問題

- quiz3rd-1をやってみましょう

## ネットワークプログラミング

- ネットワークプログラム
  - TCP(Socket)とUDP(DatagramSocket)
- 必ず必要な情報
  - IPアドレスとポート番号
- 便利なサービス
  - DNS(Domain Name Service)

## ネットワーク接続



## ネットワークプログラミング実践 - サーバ編

- HelloServer
  - クライアントがサーバに接続すると、Helloという文字を返す(送信する)
- EchoServer
  - クライアントがサーバに接続し、文字列(改行コードまで)を送ると送った文字列がそのまま返される

## サーバのポイント

- ServerSocket
  - アクセスを受け付けるためのソケット
  - 接続を受け付けるポート番号を指定
  - acceptメソッドでコネクションを取得
    - 接続があるまで待ち状態
  - コネクションを受け取ったらデータの送受信
    - 送受信はSocketで行う
    - ServerSocketとSocketは異なる

## ネットワークプログラミング実践 - クライアント編

- HelloClient
  - HelloServerに接続し、文字列を受け取って標準入力に出力する
  - telnetで行ったことを代行する

## クライアントのポイント

- Socket
  - サーバに接続するためのクラス
  - サーバ名(IPアドレス)とポート番号を指定
    - 今回、サーバ名は"localhost"
  - 接続したらストリームを取り出す
    - 入力(InputStream)と出力(OutputStream)
  - サーバの動作にあわせてデータを送受信
    - サーバの送受信方式と合致していなければならない

## HTTPプロトコル

- 多様なメソッド
  - GET/POST/PUT/HEAD/DELETE/TRACE/OPTIONS/CONNECT
- 一般的に利用するメソッド
  - GET
    - クライアントからコンテンツを取得する
  - POST
    - クライアントからサーバにデータを送信し、サーバからレスポンスを取得する

## HTTPプロトコルの例(GETメソッド)

- HTTPリクエスト
 

```
GET /index.html HTTP/1.0
Connection: close
```
  - HTTPレスポンス
 

```
HTTP/1.0 200 OK
Content-Length: 123
Content-Type: text/html
<html><head>xxx</head>...
```
- 改行