

## 1 环境

### 1.1 开发环境

本次实验在 Linux 下进行，发行版 Manjaro 21.1.6，CPU i5-1135G7，内存 16GB。

考虑到本次实验主题是套接字编程，我选择了 Java 语言进行开发，其 socket api 更加清晰通用，易于开发。IDE 是 IDEA，JDK 版本 13。

### 1.2 运行环境

因为使用的 Java 语言，所以编译生成的.class 文件在安装了 JRE 的机器上都可以运行。

## 2 系统功能需求

总的来说，本次实验需要开发一个 Web 服务器，类似 Nginx、Apache Web。结合现有的 Web 服务器的功能，将本次实验的功能需求细化如下：

### 2.1 基本需求

1. 将 Web 服务器监听的 IP 地址、端口，Web 服务器的 base 路径都写到一个配置文件中，修改配置的时候不用重新编译程序；
2. 能够监听给定的地址，当浏览器（或者其他客户端）向给定地址发起形如 `http://192.168.1.17:5678/` 的请求时，能够处理请求、根据请求定位文件、构建响应报文、返回报文给请求方；
3. 能够识别请求文件的 MIME 类型，使浏览器能够正确显示请求结果；
4. 具备日志功能，能够打印每个请求的来源 IP、端口号、HTTP 命令行等信息和请求文件的结果到控制台

### 2.2 进阶需求

1. 抵御路径遍历攻击；
2. 提供良好、完整的异常处理机制。

## 3 系统设计

应该设计一个 `HttpServer` 类作为主类，做一些初始化、善后的工作，并在给定的地址上监听，每收到一个请求，就交由 `Receiver` 处理。

应该设计一个 `ServerUtils` 类来读取配置文件、提供配置信息。

应该设计一个 `Receiver` 类处理收到的请求，包括将请求的读写流抽出来，分别交给 `Request` 类和 `Response` 类。

应该设计一个 `Request` 类根据请求的 `InputStream` 来分析请求体，并保存下来分析结果。

应该设计一个 `Response` 类根据 `Request` 分析的结果定位文件、构造回复、返回回复。

各模块之类的关系如图 1 所示：

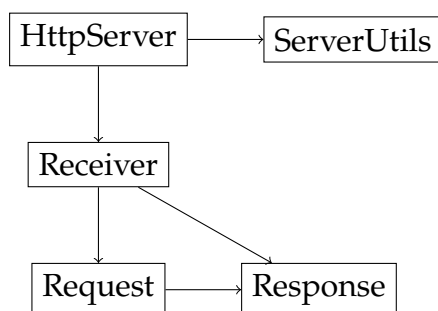


图 1: 模块关系图

## 4 系统实现

## 5 系统测试与结果说明

## 6 其他需要说明的问题

## 7 参考文献