目录

[1引言 1](#_Toc445474578)

[1.1编写目的 1](#_Toc445474579)

[1.2 背景 1](#_Toc445474580)

[1.3定义 1](#_Toc445474581)

[2总体设计 2](#_Toc445474582)

[2.1系统描述 2](#_Toc445474583)

[2.2系统设计 2](#_Toc445474584)

[2.2.1设计思想 2](#_Toc445474585)

[2.2.2系统数据流图 3](#_Toc445474586)

[2.2.3系统结构 4](#_Toc445474587)

[3接口设计 4](#_Toc445474588)

[3.1外部接口 4](#_Toc445474589)

[3.2硬件接口 5](#_Toc445474590)

[3.3内部接口 5](#_Toc445474591)

[4系统异常处理 5](#_Toc445474592)

[4.1 异常情况 6](#_Toc445474593)

[4.2异常处理 6](#_Toc445474594)

[5 调试测试 6](#_Toc445474595)

# 1引言

## 1.1编写目的

本概要设计主要是利用比较抽象的语言对整个项目需求进行概括，确定对系统的物理配置，环境要求，系统的处理流程和系统的数据结构，接口设计，界面设计等，实现对系统的初步构想与设计。根据需求分析，得到数据流图，将其转化为软件结构和数据结构，建立目标系统的软件逻辑模型，是开发人员对系统有一定深度的了解。

## 1.2 背景

待开发软件的名称：java web http server，轻量级web服务器

项目的任务提出者：succez

项目开发者：[zhyz@succez.com](mailto:zhyz@succez.com)

用户：

## 1.3定义

**http协议：**http（超文本传输协议）是一个基于请求与响应模式的、无状态的、应用层的协议，常基于TCP的连接方式，HTTP1.1版本中给出一种持续连接的机制，绝大多数的Web开发，都是构建在HTTP协议之上的Web应用。

**网络资源定位 URL：** URL是一种特殊类型的URI，包含了用于查找某个资源的足够的信息，其格式如 http://host[“：”port][path],其中http表示要通过HTTP协议来定位网络资源；host表示合法的Internet主机域名或者IP地址；port指定一个端口号，为空则使用缺省端口 80；path指定请求资源的URI。

**NIO：**一种非阻塞的读写数据的方法

**Socket：**用于网络通信的套接字

**其他相关工具包：**java.util.concurrent.\* 包含许多线程安全、测试良好、高性能的并发构建块；实现了 Collection 框架对数据结构所执行的并发操作。通过提供一组可靠的、高性能并发构建块，开发人员可以提高并发类的线程安全、可伸缩性、性能、可读性和可靠性。 org.apache.commons.\*包含许多优秀的封装，如beanutils提供对javaBean的各种良好操作，collections是java的集合操作,httpClient提供了对http客服端与服务器的通信操作，loggin提供java的日志接口

# 2总体设计

## 2.1系统描述

2.1.1需求描述

系统根据用户的不同输入，对http请求做出分析判断，然后完成相应的任务，返回结果给浏览器，供用户操作——下载或者预览，出现异常时返回错误代码与信息。同时后台管理员有权关闭服务器。

2.1.2运行环境

Windows + 浏览器 +java环境 +elipse

## 2.2系统设计

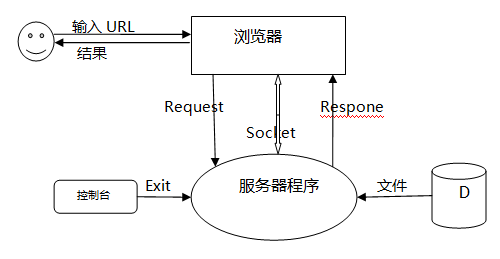
### 2.2.1设计思想

我们都知道Tomcat，它是一个J2ee规范的参考实现，我们这里要做的这个服务器只需要处理静态文件即可，不支持servlet、jsp、java等动态内容，我们要做到和apache服务器的最基本的功能一致就行了。因此本系统建立在简单的基础之上，完成用户的请求资源，下载资源的功能，但是一定要具有良好的可扩展性，正因为简单，后期用户可能仍会提出新的需求，为可能的隐性需求做好更新的准备。

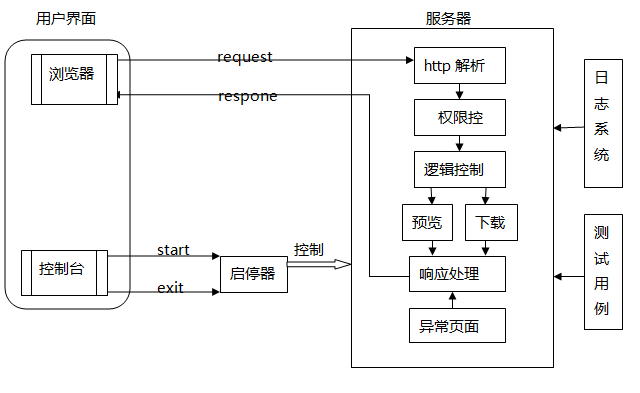
麻雀虽小,五脏俱全，虽然只是轻量级的web服务器，也要做到数据流程清晰，模块功能明确，相互之间协调工作，测试用例覆盖率高，能够应对各种可能的异常。同时考安全性，限制用户的权限，比如只能访问d盘，是否允许上传文件，上传文件的类型，考虑多用户高并发时可能给程序带来的后果，采用线程池，NIO等多线程处理技术提高系统的稳定性，考虑用户下载文件时可能的异常，增加断点下载功能，提高系统的可靠性。

### 

### 2.2.2系统数据流图



### 2.2.3系统结构



# 3接口设计

## 3.1外部接口

浏览器作为用户的与服务器交互的接口，在浏览器地址栏中输入url回车，浏览器便会通过网络向服务器发送http请求，如下图1：

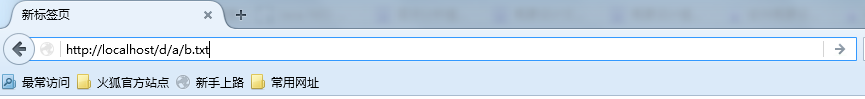
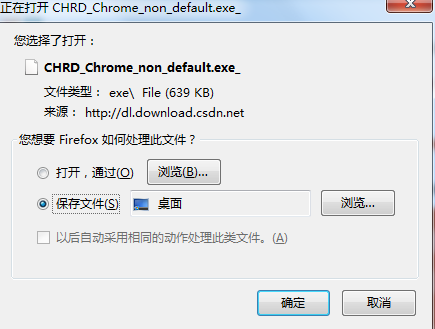
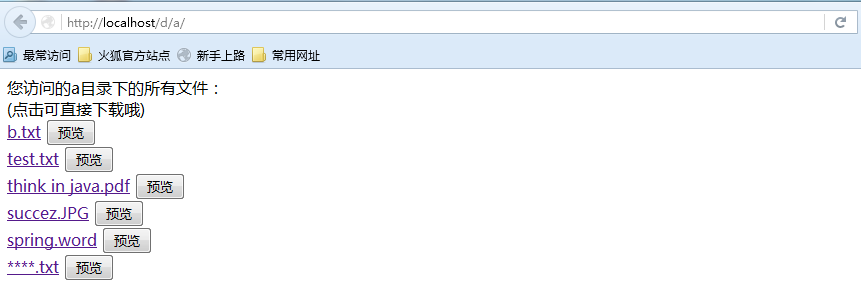


图1：浏览器地址栏

当用户输入的是http://localhost/d/a/b.txt，浏览器回弹出下载界面，供用户选择将资源下载保存在本地，或者下载并通过已有的应用程序打开资源，如下图2：



当用户输入的是http://localhost/d/a/，浏览器会显示目录下文件列表界面，用户可点击直接下载或者预览想要的资源，如下图3：



## 3.2硬件接口

无

## 3.3内部接口

//

# 4系统异常处理

## 4.1 异常情况

1，请求方法不对：http请求有很多中方法，常用的get,post等，系统只处理get请求，其他方法一律抛出异常，

2 ，请求资源越权：用户通过系统能够访问到D盘下的所有文件，但是其他资源将无法访问

3，请求的资源不存在：当用户请求的文件不存在时，用户将不能得到想要的资源

## 4.2异常处理

1，针对用户输入的不同情况的异常，系统应该给予用户相关的提示信息，或者请求方法不对，或者无权访问，或者请求的文件不存在等信息，通过http响应携带静态html，将提示信息反映给用户。

2，服务器自身异常，如响应超时，服务器关闭，等，应当返回给用户相应状态码以及页面

# 5 调试测试

系统应当有详细的日志记录，通过使用slf4j去输入调试日志，同时编写良好的测试用例，使用junit测试相关的代码，尽量提高代码测试覆盖率