# Web服务器详细设计说明书

## 1.引言

### 1.1编写目的

本文档主要为构架web服务器软件的构建而编写。对该项目的详细设计进行说明，在概要设计的基础上进一步确定系统的结构框架，详细的介绍系统的每个模块、以及模块之间的关系为之后的编码实现以及测试提供指导。本详细说明书的预期读者为项目的开发测试人员以及对本项目有兴趣的相关人员，以及之后对软件进行维护或者功能拓展人员。

### 1.2背景

软件名称：轻量级静态文件web服务器。

软件开发者：施忠平。

软件用户：有通过web浏览器对远程服务器的文件进行浏览、下载操作需求的人群。

#### 1.3定义（专门术语）

http，线程池，

#### 1.4参考资料

http：

<http://baike.baidu.com/view/9472.htm>

线程池：

<http://baike.baidu.com/view/262150.htm>

## 2.设计概述

### 2.1任务及目标

完成一个web服务器，该web服务器要能够满足用户在线查看服务器中的内容，预览以及下载服务器的文件。

#### 2.1.1需求概述

用户需求主要有三个方面：

1. 查看服务器的文件
2. 预览文本文件以及图片
3. 下载服务器端的文件

#### 2.1.2运行环境概述

要能够支持不同操作系统的不同的浏览器的不同版本与服务器进行连接并具有一定的兼容性，使用户在不同的平台上有着相同的操作体验。

#### 2.1.3条件与限制

在保证用户的基本功能需求下要防止用户的不当操作，要提高web服务器的鲁棒性。例如：给定用户能够访问的默认根目录，若用户输入..尝试访问设定根目录的上一级目录，要防止用户这种情况的发生，以防止用户的不当操作给服务器的文件造成损坏。用户只能在设定的根目录下进行访问以及其他操作。当然在服务器端要能够更改供用户访问的根目录。

#### 2.1.4详细设计方法和工具

使用java编写web服务器的程序，使用eclipse编译工具。主要涉及思路是将总程序分为web服务模块以及应用程序处理模块。Web服务模块主要负责处理浏览器发送过来的get请求，将请求解析处理之后交付给应用处理程序进行相应的处理。应用处理模块处理完请求后将应答交付给web服务模块，web服务模块再将处理结果返回给浏览器。

## 3.系统详细需求分析

### 3.1详细需求分析

##### 3.1.1在线预览需求

在线预览需求分析：用户能够通过浏览器输入服务器的域名，在线预览服务器端的资源，点击文件夹选项，能够进入到文件夹中，并预览文件夹中的内容，并且能够返回上一级菜单。用户可以选择退出按钮退出在线预览。

支持在线预览的文件：txt文本文件，jpg等图像文件，gif文件。

其它类型的文件不支持在线预览，如rar压缩文件、doc等需要特殊软件打开的文档或文件。

##### 3.1.2下载功能的需求

根据用户的需要，提供出除了文件夹外所有文件的下载，所有支持在线预览或者不支持在线预览的文件都能够下载到本地。

要注意下载功能的注意事项：

1．文件的命名问题，要支持下载时自定义文件名，下载时的路径选择问题，规定默认的下载路径并支持路径修改。

2. 遇到网络问题或者其他异常，下载中断，暂不支持断点重传。

##### 3.1.3修改功能的需求

用户有对服务器端文件的修改的需求，例如重命名、删除文件、上传文件等。服务器端要明确哪些文件可以读写，哪些文件只能读。

##### 3.1.4多用户处理需求

软件程序要能够实现多个用户同时进行预览和下载操作，在用户对服务器端文件的修改时，要保证不同用户在修改时不产生冲突。

### 3.2详细系统运行环境及限制条件分析接口需求分析

## 4.总体方案确认

### 4.1系统总体结构确认

### 4.2系统详细界面划分

#### 4.2.1应用系统与支撑系统详细界面划分

#### 4.2.2系统内部详细界面划分

## 5．系统详细设计

根据http协议原理，以及java中socket类，设计web服务器程序设计流程：

1. 创建ServerSocket类对象，监听我们设定的客户端口，取为8000，和http的TCP/IP端口区别开；
2. 等待，接受客户端连接到端口8000，获取客户端发送过来的socket；
3. 创建与客户端发送过来的socket相关的输入流InputStream和OutputStream；
4. 从socket输入流中读取用户请求信息；
5. 从请求信息中获取请求类型。如果请求类型是GET，则从请求信息中获取用户要访问的路径或文件名；
6. 根据请求类型创建HTML文件，并打开HTML文件，把HTTP头信息和HTML文件内容通过socket传回给Web浏览器，然后关闭文件。否则发送错误信息给Web浏览器；
7. 关闭与相应Web浏览器连接的socket字。

### 5.1系统结构设计及子系统划分



图5-1

系统主要包括web服务模块和应用处理模块，web服务功能主要实现与浏览器的通讯，即接受用户浏览器的请求，并将请求信息解析出来，将请求信息交付应用处理模块处理。之后将应用处理模块处理请求后动态生成的页面回复给用户浏览器。

### 5.2系统功能模块详细设计

#### 5.2.1 web服务模块

Web服务模块主要包括请求处理以及应答模块。

请求处理模块主要对浏览器的请求进行处理，分析浏览器的请求类型，请求提交给上层的应用处理模块进行处理。

Web服务器的应答模块将应用处理模块的处理结果返回给浏览器。

##### 5.2.1.1 Request模块

Request类主要由消息解析方法组成，该方法主要负责从Socket套接字的输入流解析出相关信息，并从相关信息中提取用户浏览器的请求。如图5-2所示，即为浏览器的请求信息输入流的所有数据。

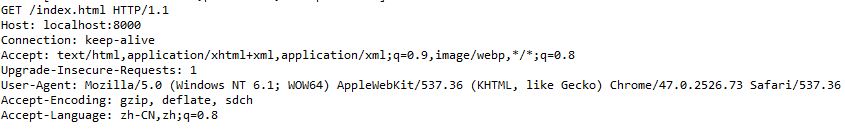


图5-2

消息解析方法要能将浏览器的请求直接解析出来，再交付给应用处理程序进行相关处理。

##### 5.2.1.2 Response模块

Response模块主要由返回数据资源方法组成，该方法将应用程序根据浏览器请求后对本地文件进行提取后动态生成网页文件，并将网页文件交由Response模块回复给用户浏览器。

#### 5.2.2 应用处理模块

应用处理模块接受web服务器提交的请求，根据请求的不同对本地文件服务进行不同的操作，并将操作结果交由给web服务器的Response模块。

应用程序处理模块主要负责对消息的处理，用户发送过来的请求主要可以分成以下几类：文件浏览请求、文件预览请求、文件下载请求、文件修改请求。如图5-3所示：



图5-3

##### 5.2.2.1文件浏览请求处理模块

该模块主要处理用户浏览服务器文件的请求。

##### 5.2.2.2 文件预览请求处理模块

该模块主要处理用户预览服务器文件的请求。

##### 5.2.2.3 文件下载请求处理请求

该模块主要处理用户下载服务器文件的请求。

##### 5.2.2.4 文件修改请求处理模块

该模块主要处理用户修改服务器文件的请求。

### 5.3系统界面详细设计

#### 5.3.1用户界面设计



图5-4

用户在浏览器的地址栏中输入服务器的地址如图5-1所示，确认后，会将服务器上的数据以文件夹的形式在浏览器中向用户展示，形如图5-2的表示

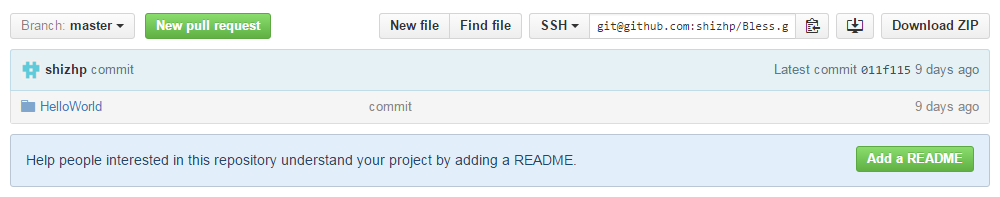


图5-5

当用户点击文件夹时，将会进入到文件夹子目录中，页面将会刷新，并将子目录中的文件以同样的方式展示出来。如图5-3所示

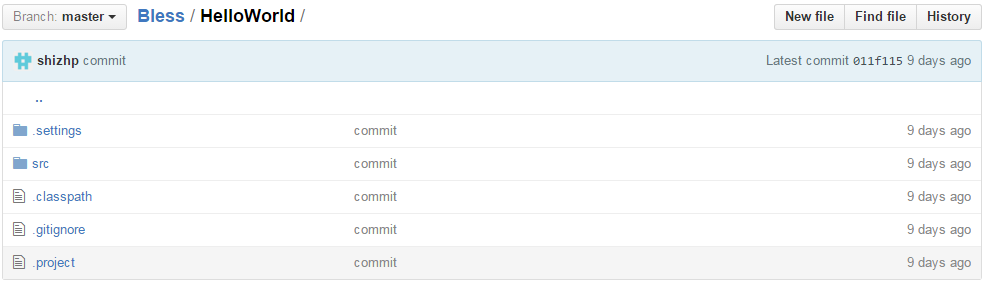


图5-6

当用户点击了txt或者其他类型的文本文件时，将在浏览器中预览该文件的内容，

刷新页面，并将该文件的内容在该页面中展示出来。如图5-4所示



图5-7

用户可以选择关闭当前页面或者返回上一个页面。

当用户选择下载一个文件时，会弹出类似于图5-5所示的界面，用户可以自定义下载的路径以及文件名。

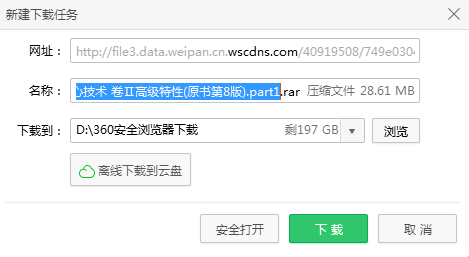


图5-8

#### 5.3.2内部界面设计

服务器端能够修改服务器的端口号，以及设置用户能够访问的根目录，并显示出与用户连接的状态信息。

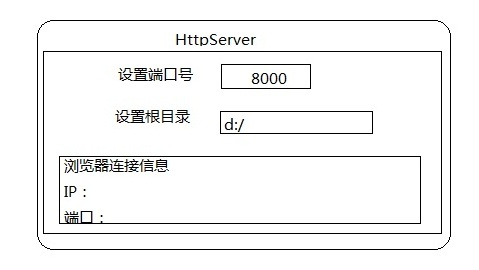


图5-9