

用 python 实现 web 服务器

[目录](#)

一、可行性研究报告

- 1、报告编写目的
- 2、项目概述-
 - 2.1、项目背景
 - 2.2、项目目标
 - 2.3、项目内容
- 3、经济可行性
- 4、技术可行性
 - 4.1、开发风险
 - 4.2、资源有效性
 - 4.3、技术水平
 - 4.4、工作基础

二、详细设计

- 1、设计步骤梗概
- 2、相关准备
 - 2.1、http 协议
 - 2.2、web 服务器
 - 2.3、socket 套接字
- 3、设计步骤
 - 2.1、实现静态页面
 - 2.2、实现 CGI 协议与脚本
 - 2.3、代码整理与重构

一、可行性研究报告

1、报告编写目的

可行性研究的目的是为了对问题进行研究，以最小的代价在最短的时间内确定问题是否可解。

项目进行详细调查研究，初拟系统实现报告，对软件开发中将要面临的问题及其解决方案进行初步设计及合理安排。明确开发风险及其所带来的经济效益。本报告经审核后，交软件经理审查。

2、项目概述

2.1、项目背景

web 服务器介绍(百度百科):

Web 服务器是指驻留于因特网上某种类型计算机的程序。当 Web 浏览器（客户端）连到服务器上并请求文件时，服务器将处理该请求并将文件发送到该浏览器上，附带的信息会告诉浏览器如何查看该文件（即文件类型）。服务器使用 HTTP（超文本传输协议）进行信息交流，这就是人们常把它们称为 HTTPD 服务器的原因。

Web 服务器不仅能够存储信息，还能在用户通过 Web 浏览器提供的信息的基础上运行脚本和程序。

本次主要以学习 python 和软件工程为目的用 python 实现一个简单的 web 服务器。并在此过程中，了解学习：

1.1.HTTP 协议基本原理

2.简单的 Web 服务器框架

3.Python 语言的 网络开发

4.Web 服务请求，响应及错误处理的实现

5.CGI 协议的 Python 实现

6.使用 Python 面向对象思想重构代码

2.2、项目目标

1.可以解析 HTTP 协议。当 Web 服务器接收到一个 HTTP 请求，会返回一个 HTTP 响应，送回一个 HTML 页面。

2.为了处理一个请求，Web 服务器可以响应一个静态页面或图片，进行页面跳转，或者把动态响应的产生委托给一些其它的程序例如 CGI 脚本，JSP 脚本，servlets，ASP 脚本，服务器端 JavaScript，或者一些其它的服务器端技术。无论脚本的目的如何，服务器端的程序产生一个 HTML 的响应来让浏览器可以浏览。

3.Web 服务器仅提供一个可以执行服务器端程序和返回响应的环境，而不会超出职能范围。

2.3、项目内容

此简单的 web 服务器主要包含以下功能：

- 1.等待某个人连接我们的服务器并向我们发送一个 HTTP 请求
- 2.解析该请求
- 3.了解该请求希望请求的内容
- 4.服务器根据请求抓取需要的数据（从服务器本地文件中读或者程序动态生成）
- 5.将数据格式化为请求需要的格式
- 6.送回 HTTP 响应

3、经济可行性

使用自己的电脑进行开发，且程序不需要其他设备支持，以及 anaconda 也不需要费用，所以不需要购置软硬件及相关设备。

由于是为课程学习的目的自己进行开发自己进行维护，所以不需要系统开发费用、系统安装和维护费用以及人员培训费用。

综上，项目的开发成本为零。

4、技术可行性

4.1、开发风险

目标中包含的功能都较为简单，且 python 的 web 服务器开发也较为普遍，网络上已有很多先例，所以几乎没有技术上的不可能性。

4.2、资源有效性

使用自己的电脑进行开发，目前来看自己的电脑工作状况良好，可以胜任此任务的开发，硬件资源无问题。电脑中已配备 anaconda 及其自带或其他的 python 模块，软件资源也无问题。人员的技术水平方面，我已有一些 python 的编程经验和 c++ 程序开发经验，若在开发过程遇到问题可以求助 google 等，人员技术水平方面无太大问题。

4.3、技术水平

- 1.使用 python
 - 2.在 anaconda 开发环境下进行
 - 3.使用 python 标准库中的 BaseHTTPServer 模块可以帮助处理
- 可以确定相关技术的发展完全可以支持此系统。

4.4、工作基础

c、c++、python 基础

在程序设计实验课上做过酒店预订系统、MOD 游戏、五子棋 AI、三角函数计算器，在 python 课程上做过新闻标题挖掘、文档相似性比较、豆瓣电影数据分析。

二、项目开发计划

1、工作内容

为完成本项目，需要按照需求分析、设计、评审、编码、测试、安装和维护等不同的阶段来进行，其中，本计划不考虑维护阶段所做的工作。

需求分析明确本项目所开发产品的特性，并对不同的功能组进行划分，得到用户方的

确认。

设计阶段将需求转换为计算机的模型，并且对实现的功能进行分配，详细设计还提供各模块、任务、功能点的详细规划。

评审分布在项目的阶段点，是质量保证的一部分，评审活动用于确定和调整与需求的偏差，以确保最后的产品符合初始提出的要求。

编码实现将按照软件产品设计所描述的内容，编写代码实现软件各部分的功能。

测试部分包括对实现过程的错误的修改、功能的改进的一些活动，同时各包括了各子系统、模块、功能点的组合和连调。

以上的过程中，包含了不同阶段的文档输出工作，并且上一阶段的输出，通常作为下一阶段的输入而存在。

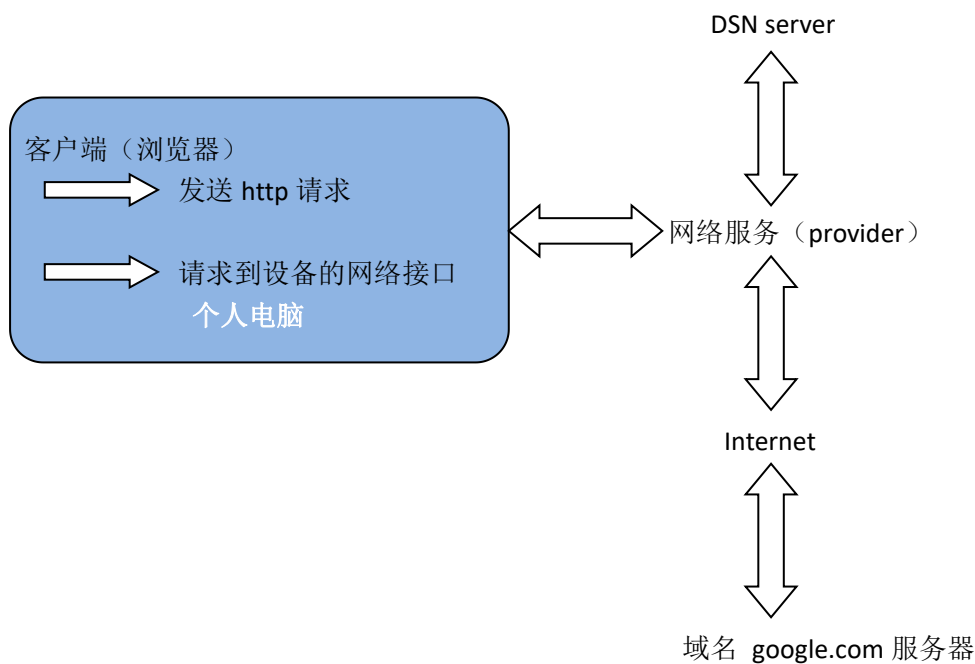
二、详细设计

1、设计步骤梗概

- 1、实现静态页面
- 2、当可以响应静态页面之后，接着实现 CGI 协议与脚本
- 3、代码整理与重构

2、相关准备

2.1、http 协议



2.2、web 服务器

web服务器:

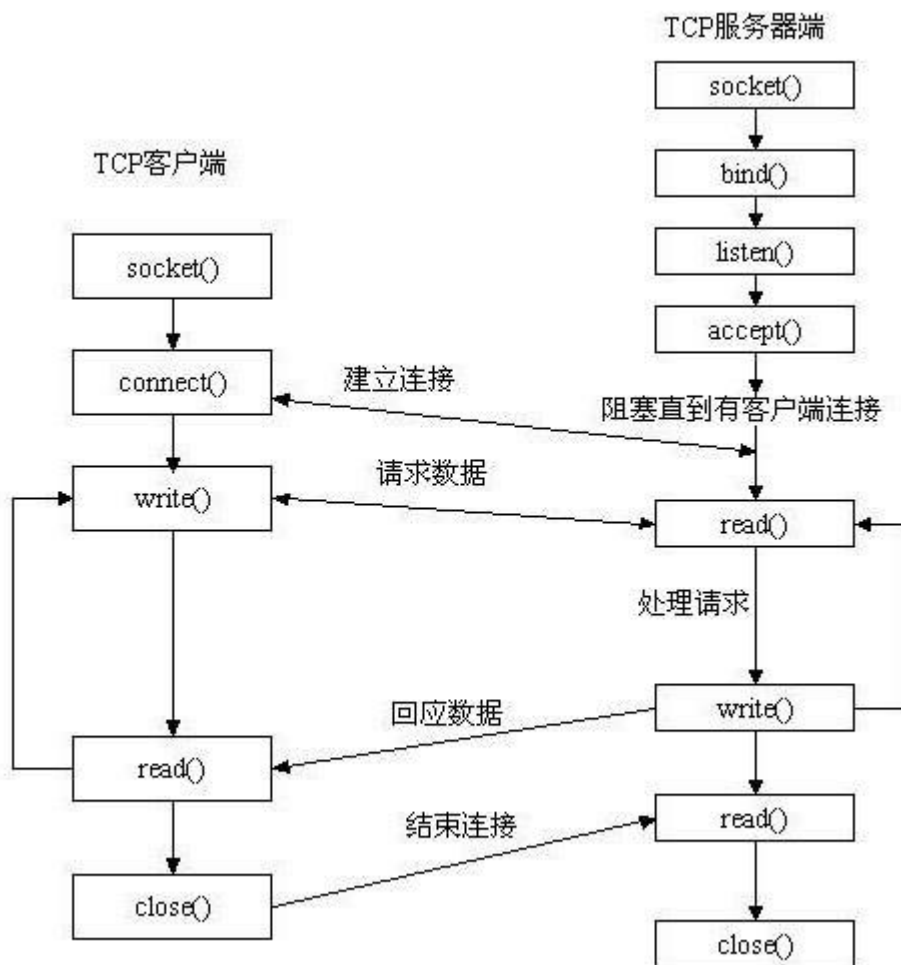
1. 等待某人连接我们的服务器并向我们发送http请求
2. 解析该请求
3. 了解该请求希望请求的内容
4. 服务器根据请求抓取需要的数据
(从服务器本地文件中读取或者程序动态生成)
5. 将数据格式化为请求需要的格式
6. 送回http响应

传输过程:

客户端发起TCP连接
↓
服务器接受连接
↓
客户端发送含url的http请求
↓
服务器接受请求并发送含index、html文本的响应
↓
客户端受到响应并解析
↓
服务器关闭连接

一个socket连接

2.3、socket 套接字



3、设计步骤

1、实现静态页面

步骤 1. 首先建立一个简单 web 服务器，能够响应静态页面

首先在主函数中，固定的使用以下语句即可：

```
1.if __name__ == '__main__':  
2.  
3.    httpAddress = ('', 8030)  
4.  
5.    httpd = hs.HTTPServer(httpAddress, RequestHandler)  
6.  
7.    httpd.serve_forever()
```

使用 `BaseHTTPRequestHandler`，则需要定义一个子类，并在子类中给出 `do_GET()`，页面设计等内容。在该类中给出依照 http 响应的格式写出的内容，再在 `do_GET()` 函数中将该内容作为响应返回。相当于在 `RequestHandler` 类中给出了 http 的响应。

步骤 2. 判断在什么样的情况下给出上述响应，同时处理不合理的请求和异常。

将输入存在三种情况：路径不存在、路径不是一个文件、路径是一个文件。

步骤 3. 测试

用 cmd 打开终端，运行以上 python 代码（命令为 `python httpserver_plain.html`），之后在浏览器中输入设置的端口号/plain.html. 查看效果。

或者 pip 安装 httpie，终端输入设置的端口号/plain.html 来查看调用效果。

2. 实现 CGI 协议与脚本

步骤 1. 第一个内容就是如何实现执行外部命令获得该输出

使用 subprocess 库，具体代码是：

```
1.subprocess.check_output(['cmd', 'arg1', 'arg2'])  
2.data = subprocess.check_output(['python', fullpath])
```

步骤 2. 在 if 语句中判断是否路径中文件以指定后缀结尾

可用字符串方法 `endswith()` 判断是否以 ".py" 结尾。（该方法以 ".py" 作为参数，输出 bool 值）

步骤 3. 填补代码逻辑

3. 代码整理与重构

整理代码逻辑，去掉冗余代码，提升代码鲁棒性。