北京邮电大学

开题报告

**支持Markdown的静态 Web服务器**

成员: 戴鑫旺 程相睿 杨乐乐

2022年3月20日

|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目 | 支持Markdown的静态WEB服务器 | | |
| 选题来源 | 个人博客、社区公告板 | 论文类型 |  |
| 开题日期 | 2022-3-20 | 开题地点 |  |
| **一、立题依据（包括研究目的、意义、国内外研究现状和发展趋势，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）**  1.1研究背景与意义  自从Markdown诞生以来，越来越多的写作者被这种轻量简洁却又样式丰富的写作方式所吸引。各类博客网站与讨论社区也都提供了基于Markdown的博文编辑器。本团队受此启发，计划开发一个具备Markdown解析功能的静态WEB服务器，使得用户只需编写简单的Markdown文件即可生成网页进行博文记录或信息发布。在此过程中，我们可以加强对Markdown语法的理解和对语法解析流程的认识，提前了解编译原理的基础概念，以及加深对Java服务器端网络编程特别是HTTP协议的理解。  1.2国内外研究现状  1.2.1研究历史  在PC机诞生初期，软件开始主要运行在桌面上，而数据库这样的软件运行在服务器端，这种Client/Server模式简称CS架构。随着互联网的兴起，人们发现CS架构不适合大多数网络应用程序，最大的原因是网络应用程序的修改和升级非常迅速，而CS架构需要每个客户端逐个升级桌面App，因此，基于Web服务器的Browser/Server模式开始流行，简称BS架构。在BS架构下，客户端只需要浏览器，应用程序的逻辑和数据都存储在服务器端。浏览器只需要请求Web服务器，获取Web页面，并把Web页面展示给用户即可。常用的Web服务器包括Nginx、Apache以及IIS等。  18年前，美国科技博主John Gruber希望“用一种易读易写的纯文本格式写作，并可将其转换为有效的 XHTML (或 HTML)”，于是在 2004 年 3 月 19 日推出初代 Markdown。随后，在Aaron Swartz的帮助下，Markdown的语法不断完善，并逐渐走进大众的视野。随着Web2.0时代的到来，Markdown凭借其在写作和预览上的便利性，受到越来越多程序开发者和博文写作者的欢迎。近年来针对 Markdown 开发的编辑器越来越多，支持 Markdown 也成了 GitHub、简书、CSDN等平台的标配。  1.2.2 应用现状  Github与2008推出了名为GitHub Pages的网页寄存服务，可以用于存放静态网页来作为免费的个人静态Web服务器。Jekyll与Hugo等软件可以用于将文档转换成静态网页，而且该软件提供了将网页上传到GitHub Pages的功能。因此将Jekyll或Hugo与GitHub Pages结合起来就可以得到支持Markdown的静态Web服务器。  **二、研究内容和目标（说明课题的具体研究内容，研究目标和效果，以及拟解决的关键科学问题。此部分为重点阐述内容）**  本课题的主要研究内容分为两部分：   1. Markdown文件处理：    1. 识别Markdown文件与HTML文件，对于HTML文件不做处理；    2. 遍历Markdown文件并进行语法解析，得到抽象语法树；    3. 基于生成的抽象语法树，通过增加HTML标签对文本进行渲染，得到原始HTML文件；    4. 提供CSS样式文件与JavaScript脚本文件对原始HTML进行美化。 2. 静态服务器：    1. 接受用户请求，并对用户请求按照HTTP请求格式进行解析，得到请求参数；    2. 根据用户的请求参数对HTML文件进行检索；    3. 将检索得到的HTML文件封装成HTTP响应返回给用户；    4. 处理用户的文件上传操作，将用户上传的Markdown文件保存到服务器端；    5. 提供实时解析能力，将用户上传的Markdown文件交由Markdown文件处理模块实时解析成HTML进行展示。   **三、研究方案设计及可行性分析（包括：研究方法，技术路线，理论分析、计算、实验方法和步骤及其可行性等）**  3.1研究方法：  深入学习Markdown的语法与语法解析原理，全面掌握Markdown语法与HTML标签间的映射关系。借鉴学习现有语法解析器的工作原理。学习计算机网络中网络层与应用层的相关知识，学习并理解HTTP请求响应格式。学习总结并应用多线程与非阻塞IO等Java服务器端编程技术。  3.2技术路线：  对于第一个研究内容，我们需要学习Markdown语法解析相关的知识。首先通过Java的IO库进行本地文件的读取，随后通过状态机的方式进行语法分析，提取Markdown的文本信息与样式信息，最后根据样式信息对原始文本增加特定的HTML标签。同时需要注意的是Markdown文本中可能会使用HTML标签来表达一些Markdown自身语法所不能表达的样式，这些标签需要保留。  对于第二个研究内容，为了深入学习与应用Java网络编程相关的知识，我们计划不采用现成的Web服务器，而是自己从头开始编写一个Web服务器。首先通过ServerSocket建立服务器端端口监听，然后对于每一个用户请求都生成新的线程进行处理（后续可能通过线程池进行优化）。每个线程都需要根据HTTP请求头的格式解析用户请求，然后进行逻辑处理，最后将返回的HTML文件包装成HTTP响应返回给浏览器。  最后我们会将项目部署在本地电脑上，编写一些Markdown文件与HTML文件，并通过Chrome浏览器进行访问，看是否能成功查看。后续会不断完善扩展相关功能，并将其做我们的博客服务器。  3.3理论分析  对于第一个研究内容，将Markdown文件转化为HTML文件其实是用到了编译原理的思想。需要进行语法提取、词法分析、语法分析、代码生成一系列步骤。可以通过正则表达式匹配的方式进行语法元素抽取，通过状态机的编程思想进行语法解析，最后通过字符串拼接的方式进行HTML代码生成。  对于第二个研究内容，编写一个简单的静态Web服务器需要明确一次网页访问流程中涉及到的计算机网络知识，以及如何通过Java网络编程的方式进行实现。因为Java通过ServerSocket提供了传输层的抽象，所以其重点是在此基础上处理HTTP的请求解析与响应构造。  3.4开发工具  本课题所用到的开发工具有：   1. 项目构建工具：Apache Maven（用于项目构建与jar包管理） 2. 集成开发工具：IntelliJ IDEA（用于Java语言开发） 3. 代码编辑器：VSCode（用于HTML、CSS以及JavaScript开发和Markdown预览） 4. 版本控制工具：Git（用于项目备份与团队成员合作） 5. 网络请求模拟工具：Postman（用于模拟网络请求进行功能测试与代码调试） 6. 浏览器：Chrome（浏览器用于测试预览与页面调试）   3.5技术架构  该项目整体上分为两个模块：Markdown文件处理模块与用户请求处理模块。一次完整的项目运行流程是：首先服务器启动，Markdown文件处理模块根据配置读取本地特定目录下的Markdown源文件，进行解析并缓存到特定目录。随后用户通过浏览器发起访问，用户请求处理模块接收用户请求并进行参数解析，得到用户的请求类别。如果是页面访问请求则根据访问参数返回特定的HTML页面，如果是上传请求，则将用户上传的Markdown文件进行本地保存，并调用Markdown文件处理模块进行解析，最后将处理完成后的HTML文件返回给浏览器。   1. **本研究课题可能的创新之处** 2. 除了支持基础的Markdown语法以外，可以提供一些自定义的语法元素来扩展Markdown的功能。例如可以通过自定义语法设置某篇Markdown文章的展示优先级，通过自定义语法设置文章的作者、摘要、标签等元信息。 3. 利用多线程提高服务器对并发访问的处理能力。 4. 增加Markdown文件解析后的HTML缓存，不用每次访问都重新解析，并通过哈希摘要判断文件是否被修改，并只对修改的文件进行重新解析。 5. 增加在线Markdown编辑器，可以在线使用Markdown语法发布评论与新的博文。 6. 通过将图片编码为base64格式来直接把图片编码成字符串写入css文件，减少用户发送HTTP请求的数量。 | | | |