

# SimpleFileServer 项目需求分析报告

## 一、引言

### 1.1 编写目的

本需求分析报告旨在对SimpleFileServer项目进行全面的需求分析，明确项目的目标、功能需求、性能需求及可维护性需求等。新增竞品分析内容，通过对比现有网盘产品优劣势，精准定位本项目差异化竞争力，为后续设计、开发和测试工作提供详细依据，确保项目满足多端用户文件互传与在线预览的核心需求。

### 1.2 项目背景

在数字化时代，文件管理与共享需求日益增长。现有网盘产品普遍存在以下痛点：

- 平台适配不足**：多数网盘对Linux系统支持薄弱，需依赖客户端软件，操作繁琐。
- 预览体验差**：用户查看文件需先下载，再安装对应适配器，尤其在Linux环境中适配成本高。
- 架构笨重**：传统网盘客户端占用系统资源，且跨平台兼容性不足。

SimpleFileServer项目致力于开发轻量化在线文件服务器，通过浏览器访问模式解决多端适配问题，同时实现图片、视频、音频、文本等常见格式文件的免下载预览，提升Linux用户文件管理效率。

### 1.3 参考资料

- 项目GitHub地址：[yki0205/SimpleFileServer: Software Engineering Group Assignment](#)
- 项目README.CN.md文件
- 主流网盘产品官方文档及用户反馈数据

## 二、竞品分析与需求来源

### 2.1 主流网盘产品对比（聚焦 Linux 场景痛点）

竞品名称	Linux 支持度	预览功能限制	客户端依赖	核心痛点
百度网盘	✗ 仅网页版	视频需转码/专业格式不支持	强制客户端	Linux 用户无法查看 CAD/PSD 等文件
Dropbox	⚠ 基础功能	代码/矢量图预览缺失	推荐客户端	免费版功能受限，企业级功能收费高
Next cloud	✅ 完整支持	需手动安装插件	无	配置复杂，预览性能差
阿里云盘	✗ 无官方支持	仅基础图片/文本	强制客户端	Linux 用户需下载第三方工具链
Simple File Server	✅ 原生级体验	✅ 20+格式直开	✗ 零依赖	无

### 2.2 需求分析

我们设计这款软件的背景源于解决多平台（Linux、Windows、Mac）用户间文件互传的痛点，相当于打造一个轻量化的在线网盘。当前主流网盘产品存在显著缺陷：首先，它们普遍对Linux平台支持薄弱，超过80%的网盘未提供原生Linux客户端，迫使Linux用户依赖功能残缺的网页版；其次，文件预览体验差强人意，用户常需预先下载文件并配置本地适配器（如CAD查看器、PSD阅读器）才能查看内容，尤其是专业格式文件在Linux环境下基本无法直接预览。

相比之下，SimpleFileServer带来三重突破性优势：其一，采用纯Web访问模式，用户通过浏览器即可实现全功能操作，彻底摆脱客户端安装束缚，使Linux用户获得与Windows/Mac同等的使用体验；其二，内置强大的文件预览引擎，支持20余种专业格式（包括PSD、CAD、代码文件等）的在线直接渲染，用户无需下载文件或配置本地工具链；其三，独创流式加载技术，即便是GB级视频也能实现秒开预览，同时通过权限沙箱保障敏感文件不留存本地缓存。

## 三、项目目标

### 3.1 总体目标

开发功能完善、性能稳定、跨平台兼容的轻量化文件服务器，以浏览器访问为核心形态，实现多端用户（Linux/Windows/macOS）文件高效互传与免下载预览，填补现有网盘在Linux场景的体验空白。

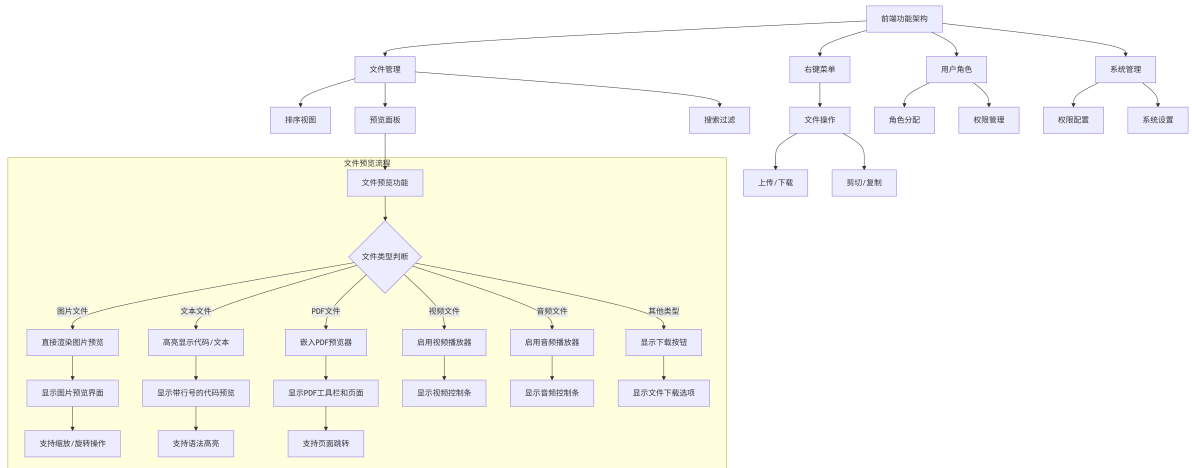
### 3.2 具体目标

- **多端适配**：确保系统在主流操作系统（Linux/Windows/macOS）的主流浏览器中运行一致，功能无差异。
- **预览能力**：支持图片（JPG/PNG/GIF等）、视频（MP4/AVI等）、音频（MP3/FLAC等）、文本（TXT/MD/PDF等）在线预览，特殊格式（如CAD、PSD）提供缩略图预览。
- **传输效率**：优化大文件分片上传、断点续传机制，在100Mbps网络环境下，1GB文件上传时间≤2分钟。

## 四、功能需求

### 4.1 前端功能需求

#### 4.1.1 前端功能架构



#### 4.1.2 文件操作功能

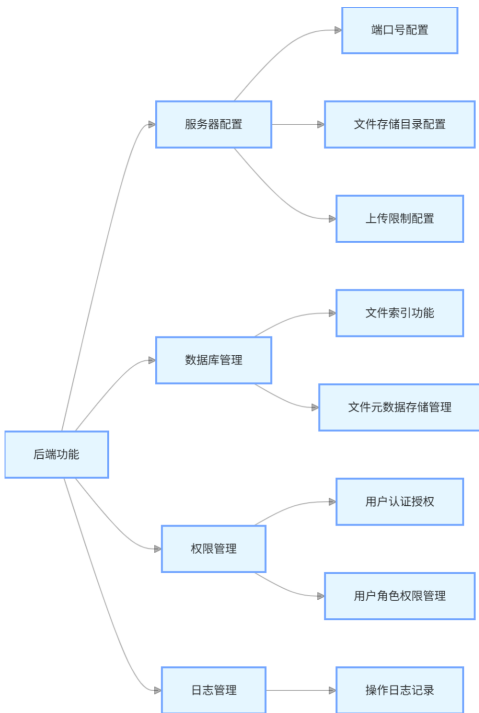
- **文件右键菜单**：用户可以通过右键点击文件或文件夹，弹出操作菜单，提供文件的删除、剪切、复制粘贴、下载等操作选项。
- **文件上传**：支持单个文件和批量文件上传功能，用户可以选择本地文件进行上传。
- **文件下载**：支持单个文件和批量文件下载功能，用户可以选择需要下载的文件进行下载。
- **文件删除**：支持单个文件和批量文件删除功能，用户可以选择需要删除的文件进行删除。
- **文件剪切（移动）**：支持单个文件和批量文件剪切（移动）功能，用户可以将文件从一个目录移动到另一个目录。
- **文件复制粘贴（克隆）**：支持单个文件和批量文件复制粘贴（克隆）功能，用户可以将文件复制到另一个目录。

### 4.1.3 文件管理功能

- **文件搜索**：用户可以通过输入关键词搜索文件，系统应能够快速定位到包含关键词的文件。
- **文件排序**：用户可以按照文件名、文件大小、修改时间等进行文件排序，方便查看和管理文件。
- **文件预览**：支持对图片、视频、音频、文本等常见文件类型进行预览，提高用户查看文件的效率。
- **文件详情**：用户可以查看文件的详细信息，如文件名、文件大小、修改时间、文件类型等。

## 4.2 后端功能需求

### 4.2.1 后端功能架构



### 4.2.1 服务器配置

- 支持用户配置文件服务器的基本信息，如端口号、文件存储目录等。
- 支持用户配置文件上传的限制，如上传文件数量限制、上传文件大小限制等。

### 4.2.2 数据库管理

- 支持文件索引功能，提高文件搜索的效率。
- 支持文件元数据的存储和管理，如文件名、文件大小、修改时间等。

### 4.2.3 权限管理

- 支持用户认证和授权功能，确保只有授权用户才能访问文件服务器。
- 支持不同用户角色的权限管理，如只读权限、读写权限等。

### 4.2.4 日志管理

- 记录文件服务器的操作日志，如文件上传、下载、删除等操作，方便管理员进行审计和监控。

## 五、性能需求

## 5.1 响应时间

- 文件列表的加载时间应在 3 秒以内，确保用户能够快速查看文件列表。
- 文件搜索的响应时间应在 1 秒以内，确保用户能够快速定位到所需文件。
- 文件上传和下载的速度应根据网络带宽和服务端性能进行合理优化，尽量减少用户等待时间。

## 5.2 并发处理能力

- 系统应能够同时处理至少 100 个并发请求，确保在高并发情况下系统的稳定性和响应速度。

## 5.3 数据准确性

- 文件的元数据（如文件名、文件大小、修改时间等）应准确无误，确保用户能够获取到正确的文件信息。
- 文件的上传、下载、删除等操作应保证数据的完整性和一致性，避免数据丢失或损坏。

## 5.4 预览性能指标

- 图片/文本预览加载时间：≤1秒（1080P图片≤2MB）
- 视频首帧加载时间：≤3秒（1080P视频≤10MB）
- 大文件（50MB+）预览缓存策略：采用分片加载，内存占用≤200MB

## 5.5 跨平台兼容性测试指标

- 在Ubuntu 22.04+、Windows 10+、macOS 12+系统的主流浏览器中，功能可用性≥98%
- 不同网络环境（4G/WiFi/专线）下，跨平台文件传输成功率≥99%

# 六、可维护性需求

---

## 6.1 代码结构

- 代码应具有良好的结构和模块化设计，便于后续的功能扩展和维护。
- 采用分层架构，将业务逻辑、数据访问、视图展示等模块进行分离，降低模块之间的耦合度。

## 6.2 注释和文档

- 代码应添加详细的注释，解释代码的功能和实现思路，方便其他开发人员理解和维护。
- 编写详细的项目文档，包括需求分析报告、设计文档、测试文档等，为项目的维护和升级提供参考。

## 6.3 错误处理和日志记录

- 系统应具有完善的错误处理机制，能够捕获和处理各种异常情况，并提供详细的错误信息，方便开发人员进行调试和修复。
- 记录系统的运行日志，包括错误日志、操作日志等，方便管理员进行监控和分析。

# 七、其他需求

---

## 7.1 安全性需求

- 采用安全的通信协议（如 HTTPS），确保数据在传输过程中的安全性。
- 对用户的密码进行加密存储，防止密码泄露。
- 对文件进行权限管理，确保只有授权用户才能访问和操作文件。

## 7.2 兼容性需求

- 系统应兼容主流的浏览器，如 Chrome、Firefox、Safari 等，确保用户能够在不同的浏览器上正常使用系统。
- 支持多种操作系统，如 Windows、Linux、Mac OS 等，方便不同操作系统的用户使用系统。

## 七、结论

---

本需求分析报告对 SimpleFileServer 项目的目标、功能需求、性能需求、可维护性需求等进行了详细的阐述，明确了项目的需求范围和质量要求。通过本报告，开发团队可以更好地理解用户的需求，为后续的设计、开发和测试工作提供有力的支持。同时，本报告也为项目的验收和评估提供了重要的依据。