# MyLib框架需求文档

# 引言

## 项目目的

根据个人的对与c++程序的学习设计一个程序框架用于其他程序开发，为了锻炼个人的编程能力。

## 适用范围

Linux和Windows双平台程序框架，解决基础性模块需求

## 定义、缩略词和缩写

暂无缩写

## 参考资料

[sylar: C++高性能分布式服务器框架](https://github.com/sylar-yin/sylar)

# 总体描述

## 产品视角

### 产品定位

该C++程序框架旨在为个人开发者提供一个灵活、高效的开发工具，以简化常见编程任务并提高开发效率。

### 核心价值

框架将提供一系列可重用的组件和模块，帮助开发者快速实现常见功能，如数据管理、网络通信和用户界面设计。

### 市场适应性

框架将设计为可扩展和可定制的，以适应不同的开发需求和不断变化的技术环境。

## 产品功能

### 基本功能

* 提供通用的数据结构和算法实现
* 实现高效的内存管理和错误处理机制
* 支持多线程编程和并发控制
* 提供网络通信和文件操作的简化接口
* 包括日志记录和调试工具
* 提供程序配置接口

### 扩展功能

* 允许自定义插件和模块以扩展框架功能
* 支持与第三方库和工具的集成

## 用户特征

### 目标用户

个人开发者，特别是那些熟悉C++语言和面向对象编程的开发者。

### 用户技能

用户应具备基本的C++编程知识和一定的软件开发经验。

### 用户需求

用户需要一个能够帮助他们快速构建和维护高质量软件的工具，同时减少重复编码工作。

## 一般约束

### 技术约束

框架将遵循C++标准，并确保兼容主流编译器和操作系统。

### 性能约束

框架应尽可能高效，避免不必要的性能开销。

### 资源约束

框架设计时应考虑最小化资源消耗，特别是在内存和处理器使用方面。

## 假设和依赖关系

### 假设

用户具备一定的C++编程背景。

开发环境包括支持C++标准的编译器。

### 依赖关系

框架可能依赖于某些第三方库，如用于日志记录、网络通信或图形界面的库。

框架可能需要特定的操作系统或硬件支持，这应在文档中明确说明。

# 具体需求

## 功能需求

### 框架核心功能

### 模块化设计

### 扩展性和可定制性

### 异常处理机制

### 日志记录

## 性能需求

### 执行效率

### 内存管理

### 资源占用

## 设计约束

### 编译器兼容性

### 平台兼容性

### 语言标准

## 软件系统属性

### 可用性

### 可靠性

### 安全性

### 可维护性

### 文档和注释规范

# 数据描述

## 数据结构和算法

## 内存管理策略

## 数据持久化

# 用户界面需求

## 命令行界面（CLI）

## 图形用户界面（GUI）

# 系统接口需求

## 库依赖

## 操作系统接口

## 网络通信接口

# 安全性需求

## 访问控制

## 数据加密

## 错误处理和恢复

# 测试需求

## 单元测试

## 集成测试

## 性能测试

## 安全测试

# 附录

## 术语表

## 参考文献

## 资源列表

## 历史版本