**Gate详细设计**

文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **文 档 历 史 记 录** | | | |
| 版本号定义规则：  使用阿拉伯数字，并由小数点分割成两部分。  第一部分：整体升级或改造时使用。  第二部分：本文档重大修改时使用，通常需要修改当前生产使用的应用程序。  第三部分：BUG修复时使用 | | | |
| **日 期** | **姓 名** | **版 本** | **更 新 记 录** |
| 2016.05.30 | 白汉升 | 1.0.0 | 基础版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 概述

本文档读者为：系统设计人员，项目组成员，软件工程师，测试工程师。

文档主要介绍模块的详细实现。

### 目的

本详细设计说明书编写的目的是说明程序模块的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础，为代码书写与维护提供帮助。

### 参考资料

《Gate概要设计.docx》

《Gate内部协议说明》

《Gate相关图.vsdx》

# 程序总体设计

程序分为Server、Client两部分。

Server为单独的程序，负责接收各个模块发来的数据并将数据进行转发。

Client为分装好的库，其他模块调用该库中的函数实现数据发送与接收。

# 通讯与协议

Server与Client之间通讯协议参看《Gate内部协议说明.docx》。

协议与各个模块通讯之间无关联，Server只需解析内部协议并按照目的地址进行转发。

其他模块初始化时调用Client接口，将自动生成一直接收数据的线程，并调用传入的回调函数处理传入的数据

其他模块需要想外部模块发送数据时调用Client提供的接口， 传入目的地址、数据、数据长度，将会转发至指定模块。

# Server部分

### 程序主要流程

Server部分是以单进程（线程）处理的方式，在select基础框架上扩展的框架作为程序框架， 通过C++实现的一套代码。

使用unix socket监听外围模块的连接，连接并识别成功后持续接收数据，根据数据包中的目的地址转发数据，实现基本功能。

程序启动后，先初始化LOG，然后实例化Server对象，并调用run()持续监听客户端的连接，外围模块连接并识别后将该模块标识为可用，表明可接收其数据与转发到该模块。若10秒钟未收到该模块数据或识别包，则将该模块置为不可用，直到收到识别包，将重新置为可用。

### 数据结构

* 1. Client状态结构体

|  |
| --- |
| typedef struct  {  bool enable; //是否可用  char flag; //唯一标识  time\_t acceptTime;  time\_t lastTime;  int overTime\_m; //接受超时时间  } \_\_attribute\_\_ ((\_\_packed\_\_))Client\_S; |

* 1. Client映射关系图

|  |
| --- |
| std::map<int,Client\_S> mClient\_m; |

### 子模块设计

* 1. 初始化



* 1. 程序整体流程图



# Client部分

### 项目对外接口

该接口主要指提供给其他模块调用的接口，即Client类

Client::Client(char ID, void (\*handle)(char \*,int));

说明：

Client类构造函数

参数列表：

ID：本模块唯一识别ID，外部模块发来数据通过ID识别

handle：处理外部模块发来数据的回调函数

返回值：

Client实例

bool Client::Forward(char dest\_ID,char\* data, int len);

说明：

数据转发接口函数

参数列表：

dest\_id：数据目的模块的标识ID

data：要发送的数据，与以前的协议无关，不需要添加标识、包头、校验等其他信息，

len：要发送数据的长度

返回值

-1：发送失败，连接重置，重新连接

n：发送数据的长度

### 程序主要流程