|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 深圳市华曦达科技有限公司 | | 文 档 编 号 | 版本号 | 密级 |
|  | | 文档编号 | V1.0 | 密级 |
| 文档名称 | 客户端到服务端的信息推送和简单控制 | | 日期 | 2018年3月23日 |

客户端到服务端的信息推送和简单控制概要设计书

dmc1

|  |  |  |  |
| --- | --- | --- | --- |
| **文档作者：** | **吴桂明** | **日期：** | **2018.3.23** |
| **项目经理：** | **樊东康** | **日期：** | **2018.3.23** |
| **审 核：** |  | **日期：** |  |
| **批 准：** |  | **日期：** |  |

深圳市华曦达科技股份有限公司

**SHENZHEN SDMC TECHNOLOGY CO.,LTD**

文档历史发放及记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 | 初版发行 | 吴桂明 | V 1.0.1 | 2018-3-21 |  |
| 2 | 修改版发行 | 吴桂明 | V 1.0.2 | 2018-3-22 |  |
| 3 | 修改版发行 | 吴桂明 | V 1.0.3 | 2018-3-23 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**文档简要功能及适用范围**

1. **文档的简要功能**

本文档主要介绍运行在Android机顶盒上运行一个服务器，与客户端通过Socket建立连接；客户端与服务器制定一套基于Socket的私有通信协议；客户端可以向服务器推送xml、json信息及发送图片、文件和视频播放链接。服务器主要负责实现对xml、json等的解析以及通过libcurl库来实现对链接文件的下载等基本操作。

1. **文档的适用范围**

本文档主要供项目负责人及项目开发人员阅读，用于了解该项目的设计流程和计划。

目录

[文档历史发放及记录 2](#_Toc509580242)

[文档简要功能及适用范围 3](#_Toc509580243)

[1.文档的简要功能 3](#_Toc509580244)

[2.文档的适用范围 3](#_Toc509580245)

[一、 引言 5](#_Toc509580246)

[1.1 编写目的 5](#_Toc509580247)

[1.2 背景 5](#_Toc509580248)

[1.3 参考资料 5](#_Toc509580249)

[二、 项目概述 6](#_Toc509580250)

[2.1 项目目标 6](#_Toc509580251)

[2.2 使用特点 6](#_Toc509580252)

[2.3 开发环境 6](#_Toc509580253)

[2.4 运行环境 6](#_Toc509580254)

[2.5 条件和限制 6](#_Toc509580255)

[三、 具体需求 7](#_Toc509580256)

[3.1 功能需求 7](#_Toc509580257)

[服务器的主要功能 7](#_Toc509580258)

[3.2 具体需求 9](#_Toc509580259)

[Server模块 9](#_Toc509580260)

[协议解析模块 9](#_Toc509580261)

[Xml解析和Json解析 10](#_Toc509580262)

[文件的存储 11](#_Toc509580263)

[URL的访问 11](#_Toc509580264)

[四、 通信协议 13](#_Toc509580265)

[4.1 协议基本格式： 13](#_Toc509580266)

# **引言**

## **编写目的**

客户端到服务端的信息推送和简单控制，是基于TCP的Socket网络编程；此项目为实习课题项目，所以主要目的是为了让实习生熟悉Socket网络编程及数据包的解析过程， 熟悉XML、JSON等信息的解析方法，熟悉libcurl、CJSON、TinyXML等库的使用方式，且能让实习生能更好的了解程序开发的整个流程。

## **背景**

随着科技的发展，互联网的不断普及，网络已经成为人们不可缺少的一部分，一套好的通信协议不仅能加快客户端与服务器之间的通信速度，也能为两者之间的数据传输提供良好的安全保障。因此学会如何制定一套用于客户端与服务端之间的私有通信协议，是十分重要的。

## **参考资料**

**cJSON的入门与应用：**

[**http://blog.csdn.net/lintax/article/details/50993958**](http://blog.csdn.net/lintax/article/details/50993958)

[**http://blog.csdn.net/u010647471/article/details/52384220**](http://blog.csdn.net/u010647471/article/details/52384220)

**XML的入门与应用：**

[**http://blog.csdn.net/HLW0522/article/details/52370434**](http://blog.csdn.net/HLW0522/article/details/52370434)

**http://www.360doc.com/content/15/0128/15/16685697\_444467285.shtml**

**libcurl库的使用：**

<http://blog.csdn.net/wu110112/article/details/72898630>

**Linux网络编程技巧**

**<https://www.cnblogs.com/jfyl1573/p/6476607.html>**

# **项目概述**

## **项目目标**

客户端与服务端之间要制定一套基于socket的私有通信协议；

服务器可以接收客户端推送的xml、json信息，并能将信息解析出来；

服务器可以存储客户端发送的图片、文件；

服务器能下载对应视频播放链接下的视频到本地；

## **使用特点**

用于实习期C/C++语言项目开发的练习

设计和实现的约束和限制

## **开发环境**

该项目在Visual Studio 2017上进行开发，使用Windows上的ndk对项目进行编译

## **运行环境**

Android机顶盒上运行

## **条件和限制**

由于当前项目仅仅只是用于内部学习和讨论研究，不涉及任何商业性质，所以开发者根据项目需求的说明实现即可。

# **具体需求**

## **功能需求**

系统功能分为：制定私有通信协议、解析XML信息、解析JSON信息、文件存储、URL链接的访问

### 服务器的主要功能

服务器的主要功能流程图如下图所示：

**错误**

**正确**

读取数据包

cmd =2

cmd =3

cmd= 4/5

Cmd =6

检验操作 checkOperation

cmd=1  
断开  
连接

获取数据包中的命令类型getCmd

Cmd=0时：数据包(SDMC+0+1+0/1(0:拒绝/1:接受))

Cmd=1时：客户端断开连接 直接终止本线程 不发数据包

Cmd=2时：数据包(SDMC+2+数据长度+解析后的XML数据)

Cmd=3时：数据包(SDMC+3+数据长度+解析后的json数据)

Cmd=4时：数据包(SDMC+4+1+1/2 (1:成功/2:失败)) 图片

Cmd=5时：数据包(SDMC+5+1+1/2 (1:成功/2:失败)) 文件

Cmd=6时：数据包(SDMC+6+1+1/2 (1:成功/2:失败)) URL

URL访问

文件存储

解析JSON数据

解析XML数据

cmd=0  
请求  
连接

发送反馈数据包给客户端

CheckHead

为每个Client创建一个线程调用数据读取函数

建立连接 accept

开始

监听客户端 listen

绑定服务器 bind

创建服务器Socket

检查数据包是否完整，且头部标识符是否正确

Cmd=2/3时:若数据解析成功则发送解析后的数据给客户端/失败则发2

Cmd=4/5时:若文件存储成功发送数字1给客户端/失败发2

cmd=6时:若URL链接下的文件下载成功发1/失败发2

断开连接终止本线程  
重新连接新的客户端

## **具体需求**

### Server模块

Server类中的各成员函数的定义主要放在ServerModule.h头文件中，函数功能的实现主要放在ServerModule.cpp源文件中，其主要成员函数如下：

1. serverInit()函数主要是创建Socket套接字，绑定服务器IP地址和端口号，监听客户端请求

void serverInit();

2) accpetClient()函数主要是建立与客户端的链接，并为客户端创建一个线程,该线程主要是调用pthreadFunction()函数，并把客户端的socketfd作为参数传过去。

void acceptClient();

pthread\_create(&thread\_read, NULL, pthreadFunction, (void \*)&clientSocket);

3）线程处理函数，主要用来调用协议解析模块的getMessageFromClient函数，参数为：Client的Socketfd

void\* pthreadFunction(void \*arg);

### 协议解析模块

Package类中的各成员函数的定义主要放在 PackageParse.h头文件中，函数功能的实现主要放在PackageParse.cpp源文件中，其主要成员函数如下：

1）getMessageFromClient()函数：接收客户端的数据包：参数为客户端的Socket

void getMessageFromClient(int clientSocketfd);

2）bCheckPacket()函数：验证数据包是否完整，参数为存储整个数据包的buff 和该数据包的总长度，成功返回0，失败返回-1

int bCheckPacket(char packageBuff[], int package\_length);

3） bCheckHead()函数：验证头部标识符是否正确，参数为存储整个数据包的buff,成功返回0，失败返回-1

Int bCheckHead(char packageBuff[]);

4）sendFeedbackDataToClient()函数：发送发聩数据包给客户端，此时用到函数重载，当xml、JSON解析成功用第一个函数，其他情况用第二个函数

int sendFeedbackDataToClient(int cmd, int dataSize, char dataBuff[]);

int sendFeedbackDataToClient(int cmd, int flag);

5）getCmd()函数：获取命令类型cmd，参数为存储整个数据包的buff,成功返回cmd,失败返回-1

int getCmd(char packageBuff[]);

6）getDataSize()函数：获取数据大小，参数为存储整个数据包的buff,成功返回数据的大小，失败返回-1，这里的数据可以是xml数据、JSON数据、文件内容、RUL链接等

int getDataSize(char packageBuff[]);

### Xml解析和Json解析

DataParse类中的各成员函数的定义主要放在 DataParse.h头文件中，函数功能的实现主要放在DataParse.cpp源文件中，其主要成员函数如下：

1）bParseXml () XML数据解析函数：参数为存储XML数据的buff,和存储解析后的xml数据的buff,成功返回0，失败返回-1

int bParseXml(char dataBuff[], char FeedbackBuff[]);

2）bParseJson () JSON数据包解析函数：参数为：存储JSON数据的buff,和存储解析后的JSON数据的buff,成功返回0，失败返回-1

int bParseJson(char dataBuff[], char FeedbackBuff[]);

### 文件的存储

File类中的各成员函数的定义主要放在 SaveFile.h头文件中，函数功能的实现主要放在SaveFile.cpp源文件中，其主要成员函数如下：

bSaveFile()函数主要是文件存储：参数为：cmd（4为图、5为普通文件）、 存储文件内容的buff,成功返回0 失败返回-1

int bSaveFile(int cmd, char dataBuff[]);

### URL的访问

URL类中的各成员函数的定义主要放在 URLDownload.h头文件中，函数功能的实现主要放在URLDownload.cpp源文件中，其主要成员函数如下：

1）bVisitURL()访问RUL：参数为指向URL链接的指针，成功返回0 失败返回-1

int bVisitURL(char \*dataBuff);

2）downloadFileFromUrl()函数主要是下载URL文件,参数：参数1：CURL类型的指针，参数2：URL链接，参数3：本地存储路径，成功返回0，失败返回-1

int downloadFileFromUrl(CURL \*easy\_handle, const char \*remotePath, const char \*localPath);

3）当库函数curl\_easy\_setopt第三个参数为CURLOPT\_HEADERFUNCTION时，会调用下面的函数， 主要实现从HTTP头部获取文件size，

size\_t getContentLengthFunc(void \*ptr, size\_t size, size\_t nmemb, void \*stream);

4）当库函数curl\_easy\_setopt第三个参数为CURLOPT\_WRITEFUNCTION时，会调用下面的函数，主要实现保存下载文件

size\_t wirteFunc(void \*ptr, size\_t size, size\_t nmemb, void \*stream);

各模块之间的大致关系如下图所示：

pthreadFunction()

协议解析模 块

服务器

bCheckPacket()

bCheckHead()

getCmd()

getDataSize()

sendDataToClient()

getMessageFromClient()

URL下载模块

bParseURL()

文件存储模块

bSaveFile()

JSON解析模块

bParseJson()

bParseXML()

Xml解析模块

# **通信协议**

## **协议基本格式：**

服务端接收客户端的推送消息功能使用TCP连接来进行数据传输，传输协议基本格式为：

**标记符 + 命令类型 + 数据长度 + 数据**

**协议格式**：标识符+命令类型+数据长度+数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 长度 | 4 | 4 | 4 | x |

连接

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 0 | 1 | 1 |
| 长度 | 4 | 4 | 4 | 1 |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 0 | 1 | 0拒绝  1不加密  2加密 |
| 长度 | 4 | 4 | 4 | 1 |

断开

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 1 | 1 | 1 |
| 长度 | 4 | 4 | 4 | 1 |

客户端不加密发送xml文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 2 | x | Xml数据 |
| 长度 | 4 | 4 | 4 | x |

客户端加密发送xml文件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 加密前数据大小 | 数据 |
| 参数 | SDMC | 2 | x | N | Xml数据 |
| 长度 | 4 | 4 | 4 | 4 | x |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 2 | X/1 | X/2(失败) |
| 长度 | 4 | 4 | 4 | x/1 |

客户端不加密发送json文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 3 | x | Json数据 |
| 长度 | 4 | 4 | 4 | x |

客户端加密发送JSON文件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 加密前数据大小 | 数据 |
| 参数 | SDMC | 2 | x | N | json数据 |
| 长度 | 4 | 4 | 4 | 4 | x |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 3 | X/1 | x/2(失败) |
| 长度 | 4 | 4 | 4 | x/1 |

客户端不加密发送图片文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 4 | x | x |
| 长度 | 4 | 4 | 4 | x |

客户端加密发送图片文件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 加密前数据大小 | 数据 |
| 参数 | SDMC | 2 | x | N | x |
| 长度 | 4 | 4 | 4 | 4 | x |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 4 | 1 | 1(成功)  2(失败) |
| 长度 | 4 | 4 | 4 | 1 |

客户端不加密发送txt文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 5 | x | x |
| 长度 | 4 | 4 | 4 | x |

客户端加密发送txt文件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 加密前数据大小 | 数据 |
| 参数 | SDMC | 2 | x | N | x |
| 长度 | 4 | 4 | 4 | 4 | x |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 5 | 1 | 1(成功)  2(失败) |
| 长度 | 4 | 4 | 4 | 1 |

发送URL 下载文件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 6 | x | x |
| 长度 | 4 | 4 | 4 | x |

服务端响应

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议字段 | 标识符 | 命令类型 | 数据长度 | 数据 |
| 参数 | SDMC | 6 | 1 | 1(成功)  2(失败) |
| 长度 | 4 | 4 | 4 | 1 |