网络C/S文件传输工具设计文档

目 录

[1. 简介 1](#_Toc9362600)

[1.1 编写目的 1](#_Toc9362601)

[1.2 背景 2](#_Toc9362602)

[1.3 术语和缩写 2](#_Toc9362603)

[1.4 参考资料 3](#_Toc9362604)

[2. 模块说明 3](#_Toc9362605)

[2.1 文件服务器探测模块 3](#_Toc9362606)

[2.1.1 广播实现 3](#_Toc9362607)

[2.1.2具体实现步骤 3](#_Toc9362608)

[2.2 TCP文件传输模块 4](#_Toc9362609)

[2.2.1 TCP建立连接流程 4](#_Toc9362610)

[2.2.2 TCP并发服务实现 4](#_Toc9362611)

[2.2.3 TCP具体实现 5](#_Toc9362612)

[2.3 UDP文件传输模块 6](#_Toc9362613)

[2.3.1 UDP传输报文流程 6](#_Toc9362614)

[2.3.2 传输实现协议 6](#_Toc9362615)

[2.4 文件完整性校验服务 7](#_Toc9362616)

[2.4.1 SHA56信息摘要的生成 7](#_Toc9362617)

[2.4.2 TCP文件校验实现 7](#_Toc9362618)

[2.4.3 UDP文件校验实现 7](#_Toc9362619)

[3. 修订记录 8](#_Toc9362620)

**1. 简介**

**1.1 编写目的**

本文档的目的是详细描述网络C/S文件传输工具软件实现方式，为软件实现和软件维护提供基础。

本文档的预期读者是项目经理、软件开发人员、测试人员、后期软件维护人员和项目管理人员等。

**1.2 背景**



图1-1 文件传输工具整体设计架构

本项目来源于19-01期校招新员工(嵌入式编程训练)，本文所设计的文件传输工具是基于linux环境gcc工具链，实现相应的软件开发，支持TCP以及UDP协议的传输。项目整体架构如图1-1所示，其中有关文件服务器探测部分主要是基于UDP的广播功能实现，客户端可以实现探测当前局域网中的可用服务器；文件传输支持TCP和UDP两种传输协议，支持文件的上传和下载两种功能，可以实现同名文件直接覆盖，支持文件校验机制，确保文件传输内容正确完整。具体操作过程会显示在客户端界面上，用户通过客户端的相关提示输入想要执行的命令，最终实现文件的上传和下载的相关操作。

**1.3** **术语和缩写**

表1-1 术语和缩写对应解释

|  |  |
| --- | --- |
| 术语/缩写 | 含 义 |
| TCP | TCP（Transmission Control Protocol）是一种面向连接的、可靠的、基于字节流的传输层通信协议 |
| UDP | UDP (User Datagram Protocol）是一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务 |
| 广播 | 广播是由一个主机向一个网络上所有主机发送消息的方式，广播是一种一对多的通信方式，所以TCP不支持广播。一般在局域网中，广播用来探测服务器地址或客户端的发现。 |
| 文件完整性校验 | 文件完整性校验就是对文件“验明正身”，就是检查下载到的文件内容是否被篡改过，文件完整性校验的常用方法是使用基于“散列函数”的校验算法。 |

**1.4 参考资料**

* 《TCP/IP详解》
* 《C程序设计语言》
* 《UNIX环境高级编程》
* 《UNIX网络编程》

**2. 模块说明**

文件传输工具主要分为以下四个模块：文件服务器探测模块；TCP文件传输/下载模块；UDP文件传输/下载模块；文件校验模块。

**2.1 文件服务器探测模块**

**2.1.1 广播实现**

常见的TCP和UDP通信大都使用的是点对点的单播方式，这种方式可以很方便进行交互处理，在多个交互同时进行时，网络带宽占的比较多。广播是由一个主机向一个网络上所有主机发送消息的方式，需要的网络带宽相对单播来说，降低很多。

广播是一种允许一台主机发送单一[数据包](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%8C%85)到局域网内所有主机的[网络技术](https://baike.baidu.com/item/IP%E7%BD%91%E7%BB%9C%E6%8A%80%E6%9C%AF)。广播作为一点对多点的通信，是节省[网络带宽](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B8%A6%E5%AE%BD)的有效方法之一。广播的地址是将IP地址中主机部分全部置为1，即xxx.xxx.xxx.255。255.255.255.255这一特殊的广播地址可以向全世界进行广播，但一般路由器都会屏蔽。广播发送的目的MAC地址为FF:FF:FF:FF:FF:FF。

**2.1.2具体实现步骤**

1、由于服务器IP地址未知，客户端发送广播包进行服务器探测，其中广播地址设为：“255.255.255.255”，服务器的端口号设为： 6464（**可改**，目的是为了唯一标记服务器），定义在头文件protocol2.h中：

2、运行客户端，首先利用setsockopt函数设置SO\_BROADCAST参数标志实现广播，然后发送DISCOVERY命令。

3、服务器接收到广播命令，确认后单播发送ACK包。

4、客户端接收到ACK后，即完成服务器的探测功能，客户端输出探测到的服务器IP地址。

5、用户选择要交互的服务器编号或刷新。

**2.2 TCP文件传输模块**

**2.2.1 TCP建立连接流程**

服务器通过TCP协议与客户端建立连接的步骤如下：

1. 通过socket函数创建套接字
2. 通过bind函数绑定套接字和地址信息
3. 通过listen函数监听连接请求
4. 通过accept函数连接客户端
5. 通过send和recv函数实现文件的上传和下载操作
6. 通过close函数关闭套接字，结束连接

客户端通过TCP协议与服务器建立连接的步骤如下：

1. 通过socket创建套接字
2. 通过connect函数与服务器连接
3. 通过send和recv函数实现文件的上传下载操作
4. 通过close函数关闭套接字，结束连接

**2.2.2** **TCP并发服务实现**

文中采用多线程实现TCP连接来实现同时响应多个客户端的功能。具体当服务器每监听到一个客户端请求时，便会创建一个线程实现连接，同时根据上传网络包中的packet.data\_type 的数值判断是上传还是下载，进而处理相应的线程函数。为了防止高并发的出现，服务器设置了监听客户端数量LISTEN\_SIZE（默认设置20个），在LISTEN\_SIZE范围内，服务器每接收到一个客户端的连接，便创建一个线程去处理相关实现；超过LISTEN\_SIZE值，就会返回错误打印。



图2-2 TCP并发服务实现流程图

**2.2.3 TCP具体实现**

**（1）TCP上传：**

* 客户端先文件校验得到校验码
* 客户端发送网络文件包，主要包含上传命令，文件路径名，文件大小，文件校验码
* 服务器端接收文件包信息
* 服务器根据文件路径解析文件名，创建文件，同名文件直接覆盖
* 客户端发送数据
* 服务器端接收数据
* 服务器接收完成，服务器端文件校验，输出文件校验码，并对比

**（2）TCP下载：**

* 客户端发送文件头信息，主要包含下载命令，文件路径名
* 服务器端接收文件头信息，判断请求下载文件是否存在，如果存在，则文件校验得到文件校验码
* 服务器回复客户端文件头信息，主要包含请求文件是否存在标志，文件路径名，文件大小，文件校验码
* 客户端接收文件头信息
* 客户端根据文件路径名解析文件名，创建，同名文件直接覆盖
* 服务器发送数据
* 客户端接收数据
* 客户端数据接收完，客户端文件校验，输出文件校验码，并对比

**2.3 UDP文件传输模块**

**2.3.1 UDP传输报文流程**

UDP传输协议主要是面向无连接的，UDP服务器执行步骤如下：

1. 通过socket函数创建套接字
2. 通过bind函数绑定套接字和地址信息
3. 通过sendto和recvfrom函数是实现文件的上传和下载操作
4. 通过close函数关闭套接字，结束传输

UDP客户端执行步骤如下：

1. 通过socket函数创建套接字
2. 通过sendto和recvfrom函数进行文件的上传下载操作
3. 通过close函数关闭套接字，结束传输

**2.3.2 传输实现协议**

**（1）UDP上传：**

* 客户端先文件校验得到文件校验码
* 客户端发送文件头数据，包含文件路径名，文件大小，文件检验码和上传命令。客户端等待服务器端响应，如果超过了设定的时间没有收到服务器的响应，便会重新发送这个数据包，直到收到服务器的响应，才会开始发送下一个数据包
* 服务器端接收文件头数据，回复客户端，其中包含应答命令
* 服务器根据文件路径名解析文件名，创建文件，同名文件直接覆盖
* 客户端收到服务器端的应答后，开始传输文件数据
* 服务器端每收到一个数据报，都会回复客户端一个应答
* 服务器接收完文件数据，服务器端文件校验，计算接收到的文件校验码，并对比

**（2）UDP下载：**

* 客户端发送文件头数据，客户端等待服务器端响应，如果在设置的时间内没有收到服务器的响应，便会重新发此数据包，直到收到服务器的响应
* 服务器端接收文件头数据，回复客户端，包含应答命令，判断请求下载的文件是否存在 ，如果存在则文件校验得到文件校验码
* 客户端应答,包含应答命令
* 服务器端发送文件头数据，包括文件大小，文件路径名，文件校验码，等待客户端应答
* 客户端应答,包含应答命令，并根据文件路径名解析文件名，创建，同名文件直接覆盖
* 服务器端开始传输文件数据
* 客户端每收到一个数据报，都回复服务器端一个应答
* 客户端文件校验，计算下载文件校验码，并对比

**2.4 文件完整性校验服务**

**2.4.1 SHA56信息摘要的生成**

* 常量初始化
* 信息的预处理，即附加填充比特和附加长度
* 逻辑运算
* 计算消息摘要

**2.4.2 TCP文件校验实现**

利用sha\_256校验函数得出需要上传或下载的文件校验码，对比两次获得的文件校验码是否相同。

**2.4.3 UDP文件校验实现**

利用sha\_256校验函数得出需要上传或下载的文件校验码，对比两次获得的文件校验码是否相同。

**3. 修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 变更时间 | 版本 | 变更人 | 变更说明 |
| 1 | 2019-5-21 | V1.0.0 |  | 创建设计文档 |
|  |  |  |  |  |
|  |  |  |  |  |