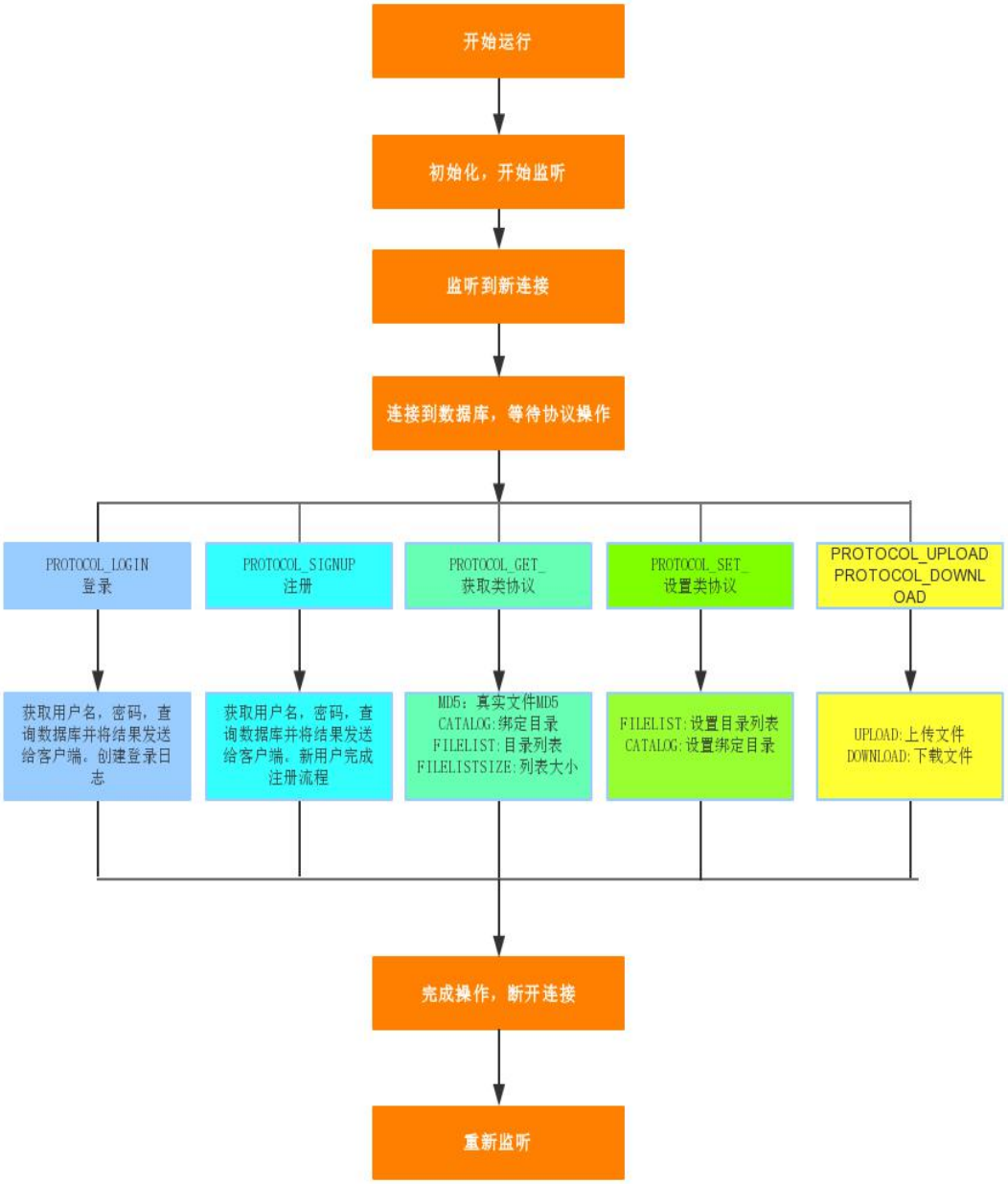
同济大学
计算机科学与技术系

网络同步盘实验报告

小组	G2335
成员	吴议　1452335
专　　业	计算机科学与技术
授课老师	沈坚
日　　期	2016.12.9

一、服务端

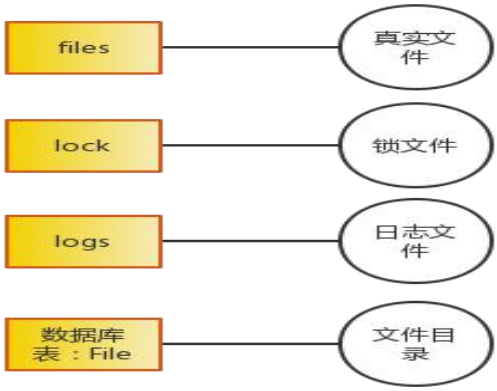
1. 服务端工作流程：



- 连接类型都是短连接，操作完成后即断开连接
- 服务端 fork 子进程完成协议操作
- 采用阻塞 socket 传输

2. 存储方案

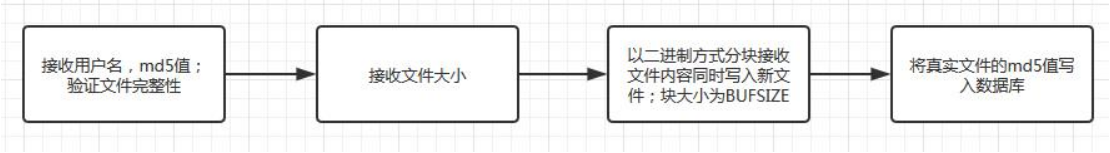
服务端采用如下存储方案：



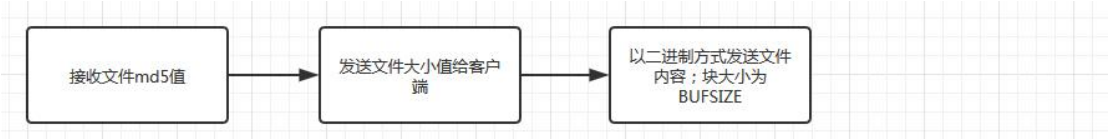
其中数据库表 File 可以看做一个虚拟的文件夹，它存储了网盘目录下的所有文件信息。实际上也可以用一个真实的文件夹代替这张表。

3. 文件传输

上传单个文件流程:



下载单个文件流程：



- 上传和下载过程，send 和 recv 都指定缓冲区大小为 BUFSIZE，客户端亦相同
- 为避免信息丢失，自定义 myrecv 接收到指定大小后才能返回

3. 加锁机制

为了避免多机用户同时操作异步性，在每个用户注册时会生成一个 username.lock 文件。在进行以下操作时会对其上锁：

- Setfilelist：重设目录列表
- Sendfilelist： 发送目录列表
- Sendfilelistsize： 发送目录列表大小

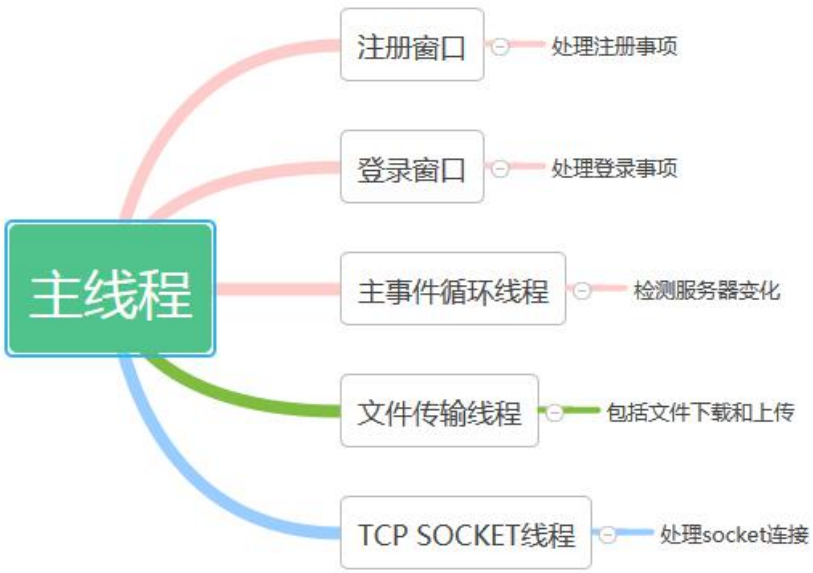
加锁方式采用 flock() 阻塞加锁，如果当前文件已经上锁，即某个需要等待完成的操作正在进行，则会阻塞当前操作，直至其他操作完成并解锁。

二、客户端

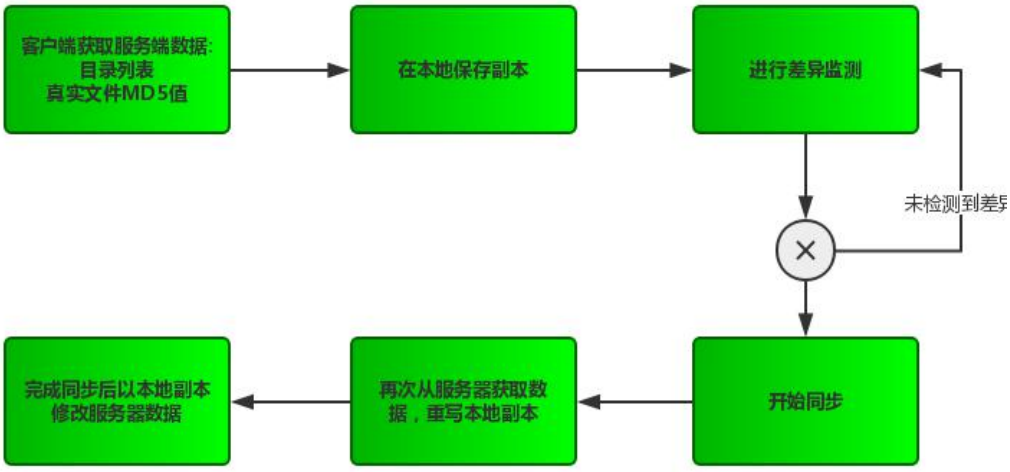
1. 界面概览


Qt 自定义视图；显示网盘目录
用户名标签；断网时显示网络已经断开
显示本地绑定目录
显示网盘绑定目录
Netdisk
Netdisk Catlog
TextLabel
注销
Local bind catalog:
Server bind catalog:
开始同步
更换绑定目录
解除绑定
Info
Name
Synsize
Totalsize
SynType
Progress
Local catalog
Name
Type
Date Modified
55d8c2c3f3d9b31832c4...
txt File
2016/12/31 20:29
config.ini
ini File
2016/12/12 13:25
eula.1028.txt
eula.1031.txt
eula.1033.txt
eula.1036.txt
eula.1040.txt
eula.1041.txt
eula.1042.txt
eula.2052.txt
eula.3082.txt
2007/11/7 8:00
显示本地文件目录
提示消息窗口：如新文件增加，绑定目录改变等
Qt 自定义视图；文件下载进度窗口

2. 程序框架


主线程
注册窗口
处理注册事项
登录窗口
处理登录事项
主事件循环线程
检测服务器变化
文件传输线程
包括文件下载和上传
TCP SOCKET线程
处理socket连接

3. 整体逻辑



4. 关于线程

1) 主事件循环线程 threapLoop

- 该线程发生在用户完成第一次初始同步目录后，即用户点击“初始同步”按钮之后。
- 它以指定的时间频率向服务器发送获取目录列表协议消息（PROTOCOL_GET_FILELIST_SIZE）和当前用户名，获得目录列表的大小。
- 差异的发生表现在两个方面：1.目录列表大小变化 2.本地目录发生改变（当前进度只涉及到文件和目录的增加）
- 程序采用 Qt 自带的文件系统监视类 QFileSystemWatcher 监视本地绑定目录
- 线程中有两个很关键的变量

```
bool            isChangeListDone
bool            isDirChanged
```

`isChangeListDone` 用来控制线程的同步性，它表示当前更改目录的操作，即上传或者下载是否完成，如果未完成线程将不会执行任何事；`isDirChanged` 表示目录是否变化，当 `QFileSystemWatcher` 检测到目录的变化时 `isDirChanged` 会被置为 `true`

- 线程采用信号和槽函数的方式与主线程进行交互。当发生差异需要同步时，会发送信号：emit listChanged()

槽函数 void changlistDone(bool value) 接收主线程的信号，并将 isChangelistDone 赋值为 value ，value 值由主线程的信号指定，由此控制线程的运行。

2) TCP SOCKET 线程

- 该线程处理 tcp socket 的连接，防止用户的 界面因为阻塞连接而被冻结
- 该线程不新建 socket ，它接受一个自定义的 Socket 类参数，然后执行该类的连接函数

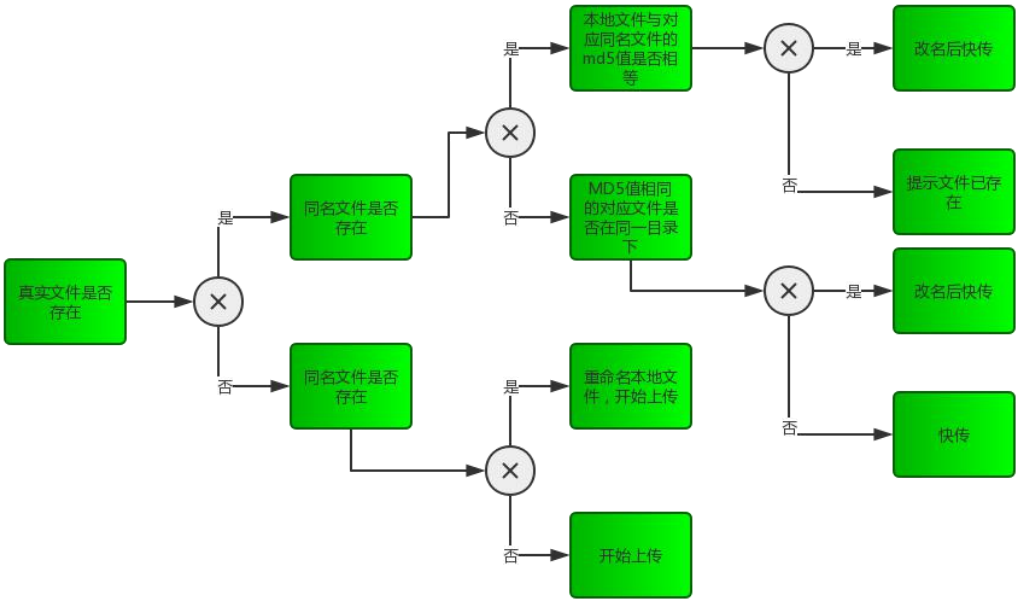
3) 文件传输线程

- 该线程处理文件的传输。开新线程传输文件能有效提高速度和程序稳定性，且能保持用户界面非冻结。它处理文件传输的方式和服务端的方式相同

5. 文件传输

程序中上传和下载文件根据有无真实文件存在、有无同名文件存在和是否需要改名按以下逻辑进行操作

上传：



下载的逻辑过程与上传类似，这里不再赘述

6. 信息的保存和日志文件的输出

程序中有几个变量专门用来保存从服务器获取的数据

- serverMd5s: 保存真实文件 md5 值；字符串链表
- File：模拟一个真实文件；类
- Fielist：保存目录列表；File 链表

程序的本地绑定目录和服务器绑定目录在登录时都是从本地的配置文件获取，而不是从服务端获取。

程序的日志文件包括：

- 登录日志
- 同步日志：同步的 TableWidget 内容全部写入日志

三、总结

1. 存在的问题

- 功能不齐全，未实现异常处理
- 程序结构不清晰

2. 心得体会

- 由于初期设计工作没做好，导致在代码实施的时候还需要一边更改总体结构，有些模块的设计甚至是“临场发挥”，这很大程度上拖慢了总体的进度也影响了最终成果的质量，而代码实施一旦开始很难再更改既有的基础，除非重新开始。

- 没有数据结构的设计使得很多时候都要想尽办法使程序满足数据，程序结构因此在很多地方显得不太合理
- 此次作业提高了自己的动手能力和对 socket 编程的理解，但只完成了基础模块有比较大的遗憾