# 小企鹅云盘

## 1.可行性研究

### 1.1技术可行性分析

现在云盘普及的十分广泛，经过时间的检验说明技术是成熟的，现在有各个级别的云盘，从个人到企业再到商用，现在要编写一个校园级别的云盘。在通讯上使用socket套接字就可以实现，使用linux的epoll即可已进行IO的复用可以满足在一台性能不是很高的服务器上同时为尽可能多的用户服务，使用IO复用也考虑到用户对云盘的操作不是频繁的。云盘对网络使用较多，所以使用linux系统进行服务器的开发，并且在linux上运行的服务器稳定性高。服务器由c语言编写，有相应的线程，套接字库和API，技术上是可行的。

### 1.2数据流图



## 2.需求分析

在校园中需要发布很多文档，有资料，有作业等类型，为了更好的学习也需要共享学习资料，网盘中所有人都可以在校园云盘中上传文件，如果不管什么文档都需要下载才可以查看，这样就会浪费一些时间，那么降低了用户的体验，提供预览功能是需要的。在下载大型文件时为了考虑到时间可以提供下载暂停功能，并显示下载预计时间。

### 2.1功能需求

上传文件

下载文件

下载暂停

预览文件

### 2.2性能需求

容量满足日常需求

用户账号密码加密

校园网内下载速度8Mbit以上

响应速度小于0.6秒

### 2.3可靠性

一个月内不出现二次故障。

### 2.4将来可能提出的需求

该系统可运行在互联网中。

## 3总体设计



### 3.1功能分解

客户端：

（1）用户登录：

用户输入用户名、密码、验证码信息，信息验证通过后登录成功。

（2）用户注册：

用户可以注册为本系统用户以来获取一定容量的空间，用户输入用户名、密码、Email等信息，信息验证通过后即注册成功。

（3）账号管理：

用户还可以进行修改密码、安全退出等功能。

（4）加密解密

对用户的账号信息在网络中传输时加密。

（5）预览：

用户可以在线无需下载查看较小的文档内容。

（6）下载上传：

对用户指定的文档进行下载和上传

（7）文档结构：

文档分析器：客户端会获得自己的文件结构，这个文件结构是服务器发来的。

文档生成器：客户端会由文档分析器生成文件结构。

json：文档分析器是由json技术提供支持的。

（8）命令：

用户的操作涉及文件时，实际上是对自己在服务器中的文件操作，因此要产生命令，对服务器进行操作，这个模块就是生成命令和对应的信息。

命令生成器：生成命令，给服务器

命令解释器：接受命令，做出相应操作。

（9）文件操作：

用户还可以在相应的文件夹下上传文件、下载文件、删除文件和查看文件信息。

服务器：

1. 加密解密：

对用户的账号信息在网络中传输时加密。

1. 命令：

服务器解析命令，做出相应操作，再给出命令给客户端，可以是结果，可以是要求客户端做出操作。

命令生成器：生成命令，给客户端。

命令解释器：接受命令，做出相应操作。

1. 用户账号处理：

登陆验证，注册。

1. 监听：

对信息的监听和登陆信息的记录。

1. 响应：

对用户的对应操作进行响应。

接受用户：将用户加入管理列表中。

服务用户：对用户命令解析并进行操作。

1. 用户管理

对用户进行管理，如文件、登陆状态。

### 3.2数据库设计

在进行数据库的架构时，一般要遵循以下的规则：

 (1) 规范化设计

在设计和操作维护数据库时，关键的步骤就是要确保数据正确地分布到数据库的表中。使用正确的数据结构，不仅便于对数据库进行相应的存取操作，而且可以极大地简化应用程序的其他内容（查询、窗体、报表、代码等）。正确进行表设计的正式名称就是"数据库规范化"。

(2) 尽量减少数据的冗余性和重复。

数据应该尽可能少地冗余，这意味着重复数据应该减少到最少。比如说，一个部门雇员的电话不应该被存储在不同的表中， 因为这里的电话号码是雇员的一个属性。如果存在过多的冗余数据，这就意味着要占用了更多的物理空间，同时也对数据的维护和一致性检查带来了问题，当这个员工的电话号码变化时，冗余数据会导致对多个表的更新动作，如果有一个表不幸被忽略了，那么就可能导致数据的不一致性。

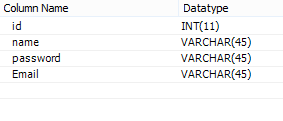
 (3) 结构设计与操作设计相结合。 在设计数据库时，有时为了编程实现的简洁和思路的清晰，往往故意增加一些数据的冗余。这虽然有悖于传统的关系数据库理论，但是考虑到查询与统计的方便，对编程的实现有很大的好处，往往采用这种方法。

(4) 数据结构具有相对的稳定性。

数据结构的相对稳定性，可以作为新，旧信息系统转换的依据。根据系统功能设计的要求及功能模块的划分，对于系统信息数据库，可列出以下数据项和数据结构：

本系统采用mySQL作为数据库服务器。

表user\_login，用于用户登录验证



## 4.详细设计

### 4.1服务端

1. 用户账号处理：有登陆请求时，获得用户名和密码，连接数据库验证，返回数据库处理信息，返回的信息为提示信息。有注册请求时，连接数据库，如果有重名返回信息，如果正确写入数据库，返回信息。
2. 加密解密：
3. 命令生成器：获得命令后经过处理，要给客户端回馈，命令生成器生成命令，给出相应的命令头，其后跟处理信息或结果。
4. 命令解释器：获得命令，分析命令，得出结果后转入相应的处理。
5. 监听：使用epoll函数，当出现用户socket活动后，获得该socket，读出所带信息，转入命令解释器。
6. 接受用户：当是登陆或注册请求时，只将socket加入IO复用的管道中，并将相应的信息记录。
7. 服务用户：命令解释器所转入的模块，这个模块使用了命令生成器，文件推送，用户文件结构生成。当登陆时首先是调用用户文件结构生成，给客户端文件结构，用户操作时会，客户端产生命令，经过解释器解释后，服务用户模块执行相应操作，产生的结果送入命令生成器，生成的命令给客户端。当用户要下载和上传时，另开一个线程，进行上传和下载功能。
8. 用户管理：

### 4.2客户端