**设计说明书**

1. 总体功能概览

程序启动时，将S3上的某个bucket中的内容同步到本地某个目录，出现文件冲突时，以文件最新更改时间为准。后再将本地目录同步到S3bucket中。执行完毕后，对本地目录进行监听，当本地目录中文件发生增删改时，S3bucket发生相应变化。同时对大文件采用多线程分片上传，下载，具有端点续传功能。

2.端点续传实现：将上传部分的uploadId 存储在缓存中，当上传文件时，先在缓存中查找，看是否存在端点的上传任务。当存在时，先取出该id，进行查询，然后接着完成接下来

分片传输。并采用线程池的方式，对文件进行分片传输

3.基本同步功能实现：先列举出bucket中的所有文件和目录，同时列举出本地目录下的所有文件和目录。先将bucket中的目录同步到本地目录下，再根据文件最后修改时间进行文件同步。此后对本地目录的进行监听，当本地目录中文件发生增删改时，S3bucket发生相应变化。

1. 程序架构设计

**FileSychronizer**

**-MultiPart : Runnable**

**-dir:String**

**-bucketName:String**

**-accessKey:String**

**-secretKey:String**

**+FileSychronizer(String dir, String bucketName, String accessKey, String serectKey);**

**+start()**

**+listen()**

**+upload(filename: String, date :Date)**

**+download(filename: String, date :Date)**

**- multiPartUpload(filename: String)**

**-multiPartDownLoad(filename: String)**

**-listSuccessfulPart(filename: String)**

**-createFolder(filename: String)**

**-deleteFolder(filename: String)**

JedisCache

**+setCache(keyName :String, cache :Cache)**

**+getCache(keyName :String)**

**+removeCache(keyName :String)**

**FileListener**

**+onFileCreate(file:File)**

**+onFileChange(file:File)**

**+onFileDelete(file:File)**

**+onDirectoryCreate(directory:File)**

**+onDirectoryChange(directory:File)**

**+onDirectoryDelete(directory:File)**