下载项目

一、项目描述

该项目的核心功能在于实现IOS系统中多线程下载的任务。具体要求主要分为HTTP网络文件的创建、多线程任务的下载以及文本文件的管理，其中，多线程下载为本项目的重点。其它要求包括UI以及非ARC的实现。

二、功能描述

1、HTTP网络文件的创建

本功能比较简单，主要目的在于接受用户下载文件URL的输入，主要任务在于UI的实现及URL路径的判断处理。

2、多线程下载

本模块功能包括多任务创建、删除、下载暂停继续，同时显示下载进度以及文件大小等信息。其它要求包括断点续传、后台下载、重启后保存状态以及批量操作等。

3、文件管理

本模块主要功能包括下载文件的显示及删除等操作。

三、实现方案

1、技术选择

UI方面。本项目UI比较简单，URL输入可以采用UIAlertView即可，下载界面主要采用 UITableView实现，自定义UITableViewCell实现下载任务的状态显示以及下载暂停等操作响应。文件管理可以通过单击UITableViewCell弹出UIActionSheet简单实现。具体效果如下图所示



图一、下载界面

下载方面。本项目为非ARC项目，同时需要实现断点续传功能，结合以上两点考虑，同时为了开发方便，文件下载主要采用第三方开源下载工具ASIHttpRequest。所有下载采用异步下载，下载状态通过数据库SQLite保存在本地。数据库存储采用第三方工具FMDataBase，所有下载文件存储在/Documents/Cache文件夹中。

文件管理方面。文件管理采用系统自带NSFileManager即可，文件查看预设显示图片文件，其它文件暂不支持查看。

2、系统结构

整个项目采用MVC的设计理念。下载界面RootVC和列表DownloadCell负责UI的显示及用户响应；DownManage以全局单例的方式实现下载对象ET\_Download的创建及删除；ET\_Download负责具体文件的下载逻辑。DB\_Download通过数据库FMDatabase 实现ET\_Download状态的本地化存储。

ET\_Download主要属性包括网络URL、已下载文件大小、总文件大小以及当前下载状态。其中，文件下载又分为等待、下载、暂停、完成和失败5种状态，下载流程如下图所示。

等待

下载

并发下载达到最大数量

暂停

完成/失败

下载失败

图二 下载流程图

1. 新建下载任务时，创建 ET\_Download 对象，设置相应的Url并存储在数据库中。
2. 判断当前并发下载数量，如果达到最大数量，设置相应的状态为等待，并进入到下载队列中等待，若没有达到最大数量则直接开始下载。
3. 点击暂停按钮时，设置下载状态暂停并更新数据库。执行步骤5。
4. 点击继续下载时，执行步骤2。
5. 依次扫描所有下载队列，当扫描对象为等待状态时执行下载并更新状态，直到并发下载达到最大上限时停止扫描。
6. 下载完成或下载失败时，更新对象状态和数据库，执行步骤5。

当下载对象状态发生改变时，通知UITableView刷新对应的DownloadCell的显示。删除下载任务时，提示是否同时删除文件，同时，判断下载任务是否处于下载状态，如果处于下载状态，首先停止下载然后再执行任务的删除。

查看文件时首先检查文件是否存在，如果存在，判断文件后缀是否为jpg或png，如果符合，push图片浏览界面显示图片，如果是其它格式暂不做处理。

应用进入后台时，启动后台下载Required background modes模式。当程序退出时，通知所有下载对象保存最新状态到数据库。当程序重新启动时，恢复退出前的下载状态。

**待定实现**：1、网络状态发生改变时，可能需要重新建立连接，初步方案是注册监听网络状态变化情况，当网络发生变化时根据情况恢复下载。

四、开发进度

设计阶段：1-3天

编码阶段：3-5天

测试阶段：1-3天