

作业四 网络学堂文件系统

马晓彬 2012011402

一、程序概况

基于 Linux 系统下 FUSE 软件，本次作业完成了将清华大学网络学堂映射为一个文件夹的工作，访问课程文件、查看课程信息等操作都转换成为了对文件下目录和文件的操作。并且支持将文件拷贝至课程作业栏目进行作业的提交。经过优化能够在点击时才下载课程文件，节省流量。

由于纯 C 语言对网络操作的劣势，本程序采用了 C 和 Python 混合编程的方法，使用 Python-C API 用 C 语言编译的 FUSE 调用 Python。C 语言完成数据保存和请求处理的功能；Python 进行和网络学堂的交互，下载文件、获得学堂中课程信息。混合 Python 的方法也是我在本次作业中确定的方法，也向我们班很多同学提供了思路，解决了 C 语言和网络学堂交互的矛盾。

二、使用介绍

1. 系统搭建

- 1)
- 本系统在 Linux 系统环境下运行依赖以下两个除了 Python 自带的和 FUSE-2.9.4 之外的第三方库：

名称	类型	用途	安装方法
Python-C API	C 库	与 Python 进行交互	控制台下执行 <code>sudo apt-get install python-dev</code>
httplib2	Python 库	发送处理 http 请求	解压附带的 <code>httplib2.tar.gz</code> 并执行 <code>sudo python setup.py install</code>

2) 路径和文件设置

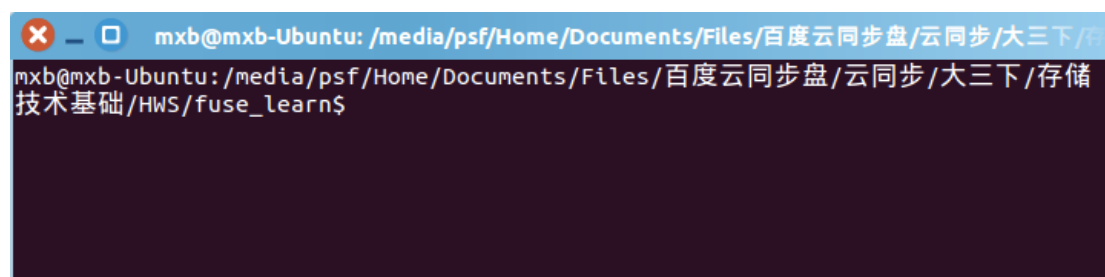
本程序共有四个文件和两个路径：

名称	作用	设置方法
<code>learn.config</code>	设置 Python 脚本调用路径、文件缓存路径及用户名、密码	格式为[缓存文件夹路径]\n[python脚本路径]\n[用户个数]\n([用户学号]\n[密码]\n)*n，需要在程序运行前设置，和编译好的 learn 程序放在同一路径下

learn.c	实现了 FUSE 的 main 函数和系统调用的文件操作，是程序的主函数	需要使用自定义的便已命令 gcc -Wall learn.c `pkg-config fuse --cflags --libs` -L/usr/lib -lpython2.7 -o learn 编译，之后使用 ./learn [挂载路径] -d 即可使用
url.py	实现网络学堂课程相关信息的抓取处理	将其放在 learn.config 定义的 Python 脚本路径下，需要调用
upload.py	实现网络学堂作业的提交上传	将其放在 learn.config 定义的 Python 脚本路径下，需要调用
缓存文件夹路径	存放缓存的学堂文件和公告文本	是一个文件夹，需要提前建立
Python 脚本路径	存放 Python 脚本	是一个文件夹，需要提前建立

2. 运行方法

- 1) 安装 python-c API(python-dev)和 httplib2
- 2) 从终端中打开附件中的 fuse_learn 文件夹

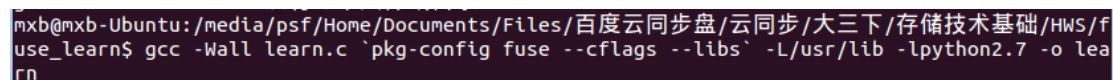


```

mxsb@mxsb-Ubuntu: /media/psf/Home/Documents/Files/百度云同步盘/云同步/大三下/存储技术基础/HWS/fuse_learn$

```

- 3) 输入 gcc -Wall learn.c `pkg-config fuse --cflags --libs` -L/usr/lib -lpython2.7 -o learn 进行编译



```

mxsb@mxsb-Ubuntu: /media/psf/Home/Documents/Files/百度云同步盘/云同步/大三下/存储技术基础/HWS/fuse_learn$ gcc -Wall learn.c `pkg-config fuse --cflags --libs` -L/usr/lib -lpython2.7 -o learn

```

- 4) 输入 ./learn ./fuse -d 进行挂载



```

mxsb@mxsb-Ubuntu: /media/psf/Home/Documents/Files/百度云同步盘/云同步/大三下/存储技术基础/HWS/fuse_learn$ ./learn ./fuse -d
user num : 1
用户ma-xb12 Tsinghua2012
开始获取ma-xb12的课程数据!
course num : 9
搜索引擎技术基础(0)(2014-2015春季学期) 124095
网络编程技术(0)(2014-2015春季学期) 124546
存储技术基础(0)(2014-2015春季学期) 124093
计算机系统结构(0)(2014-2015春季学期) 124090
数值分析(0)(2014-2015春季学期) 123306
三年级男生羽毛球(1)(2014-2015春季学期) 123119
计算机网络专题训练(0)(2014-2015春季学期) 124097
多媒体技术基础及应用(0)(2014-2015春季学期) 124086
文化素质教育讲座(1)(90)(2014-2015春季学期) 122360

```

进行以上三步之后可以看见当前目录下的 fuse 文件夹变成了磁盘，进入后已浏览我的学号下的作业，打开相应课程可以浏览信息，查看课程文件。课程信息较小，会在挂载之前自动读取，课程公告和课程作业会在访问时自动下载。

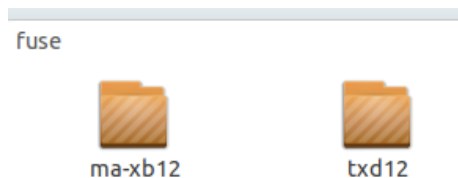
使用提交作业功能时，进入相应作业的二级目录，将要提交的文本复制到该文件夹下，查看网络学堂，发现作业已经提交。此处可以测试搜索引擎技术基础课程的课程设计这个作业，这是一个还未提交的作业。

三、程序实现

1. 数据结构

整个程序分为四级结构，第一级为用户列表，第二级为课程列表，第三级为课程中课程公告、课程信息、课程作业和课程文件的文件夹，第四级为文件夹中的相关内容。如下所示：

```
struct User
{
    char userid[100];    //网络学堂用户名
    char userpass[100]; //网络学堂密码
    Int course_num;     //课程数目
    struct Course *course_list; //课程链表
    struct User *next;  //用户链表
};
```



```
struct Course
{
    char course_id[20];    //课程号
    char course_name[100]; //课程名
    char info[15000];     //课程信息
    Int infoLen;          //课程信息长度
    struct cFile *downList; //课程文件列表
    Int downNum;          //课程文件数
    struct cFile *noteList; //课程公告列表
    Int noteNum;          //课程公告数
    struct cFile *homeworkList; //课程作业列表
    Int homeNum;          //课程作业数
    struct Course *next;  //链表
};
```



```
struct cFile
{
    char name[200];    //文件名
    char link[1500];   //文件网络路径
    char type[50];     //文件类型
    char submit_id[20]; //用于课程作业提交的作业号
    char bufPath[1000]; //本地缓存文件绝对路径
    Int size;          //文件大小
    Int isReady;       //文件是否已经下载标识
    struct cFile *next; //链表
    struct cFile *homework; //若为作业，指向作业的提交文件
};
```



2. 函数架构

1) 辅助函数

函数名	功能
struct cFile * findFile(const char *path, struct User * user, struct Course * course, struct cFile* fileList, char *fFolder)	在 user 下地 course 课程的 fFolder 文件夹下寻找路径 path 定义的文件，成功返回文件指针，失败返回 NULL
void getAllCourseInfo()	调用使用 Python 从网站获取所有用户所有课程的课程信息

2) 调用 Python 函数

函数名	功能
int getCourseList(struct User * user)	获得 user 的所有课程的课程号和课程名，存入 user->courseList 链表
int getCourseNote(struct User * user, struct Course * course)	将 user 下 course 的所有课程公告下载到缓存文件夹，并将文件描述结构 cFile 连入 course->noteList
int getCourseInfo(struct User * user, struct Course * course)	获得 user 下 course 的课程公告，文件内容存入缓存 course->info
int getHomeworks(struct User * user, struct Course * course)	获得 user 下 course 的所有课程作业的信息，文件存入缓存文件夹，将文件描述结构加入 course->homeworkList
int getDownloadInfo(struct User * user, struct Course * course)	获取 user 下 course 的所有课程文件名、文件大小和下载链接但不进行下载
int downloadFile(struct User * user, struct cFile *file)	根据 file 中存储的下载链接下载 user 的 file，存储至缓存文件夹
int uploadHomework(struct User * user, struct Course * course, struct cFile*uphwkName)	若未上传，则上传 uphwkName 所附带的提交文件 uphwkName->homework 至 user 的 course

3) 系统调用操作定义函数

函数名	功能
static int hello_readdir(const char *path, void *buf, fuse_fill_dir_t filler, off_t offset, struct fuse_file_info *fi)	获取 path 下所有的文件和目录，按四级结构进行查找，若首次访问课程公告、课程文件或课程作业文件夹，则调用 getCourseNote、getDownloadInfo 或 getHomeworks 获取信息并显示，若不是则直接遍历相应链表显示

static int hello_getattr(const char *path, struct stat *stbuf)	根据 path 寻找该文件的信息，读取相应链表并显示，没有任何 Python 交互
static int hello_open(const char *path, struct fuse_file_info *fi)	打开一个文件，遍历四级目录，若不存在则返回-ENOENT，存在则返回 0，在第一次打开已经提交作业时，会调用 uploadHomework 上传作业
static int hello_read(const char *path, char *buf, size_t size, off_t offset, struct fuse_file_info *fi)	读取文件，若是某个课程信息，则从 course->info 的内存中直接读取，若读取缓存的文件，则使用 C 库打开并读取
static int hello_write(const char *path, const char *buf, size_t size, off_t offset, struct fuse_file_info *fi)	仅在提交作业时使用，将内容写至缓存文件夹 file->bufPath 的对应文件中
static int hello_create(const char *path, mode_t mode, struct fuse_file_info *fi)	为 file->homework 分配内存创建一个要提交的文件，建立缓存文件夹中的文件

四、困难与思考

1. 此次工程遇到的第一个困难就是 Python-C 库的使用，在开始时没有运行 `sys.path.append('./')`，没有将自己模块所在的路径加入系统搜索路径，不能使用自己定义的模块，导入的 module 为空，会产生段错误，最终通过网上的搜索才意识到这个问题。
2. 其次就是对系统函数参数的理解，明白 `getattr`、`readdir` 等函数参数的使用方法，通过在函数中添加相似的语句，然后打开路径观察，才看出来各种参数的含义，最终能够开始自定义的函数编写。
3. 对文件浏览器会调用的函数次序不明白，不了解各嘎哈是农户在发生何种操作的时候会调用。通过在函数中添加 `printf` 语句，了解了调用关系。同时发现了一个问题，在进入一个文件夹之后，由于系统要读取文件的概要，会直接打开文件，对于文件较大的课程文件不能直接下载，于是创建了打开计数，只有第二次打开的时候才下载文件，这样就不会一进入文件夹就下载所有文件了。
4. 在爬取网页的过程中也遇到了麻烦，其中对于文字有很多 ` ` 之类的转义符号，但是 Python 的 `HTMLParser` 会调用 `handle_entityref(self,name)` 函数处理转义而非 `handle_data`，故而文件名或公告中一旦有空格就不能抓取完整的名字，我还以为是 C 函数的问题，最终通过百度才发现问题。
5. 在课堂中学习到的有关存储的前沿知识对此次编程有很大的帮助，能够在编程之前了解文件系统的概念和架构，对理解 FUSE 工作方式有很大帮助。
6. 此次作业是我写过的最大的 C 程序之一，代码量达到了 1800 行，日夜连续写了五天左右，代码量大，BUG 出现的方式也千奇百怪，调试的过程遇到了很多麻烦但是逐个解决了，对我对 linux 系统有了更深的了解。