编　　制	沈胜文
审　　核	
批　　准	
实施责任人	

Amendment history 修改历史记录

版本号	修改说明	修改批准人	修改人	日期	签收人
101	创建文档		沈胜文	2008-4-1	

编　　制	沈胜文
审　　核	
批　　准	
实施责任人	

Table of Content 目录

1. Overview 总述

1.1. Background 背景

下载实现是 MuFTAD 项目中的核心，是该项目中最重要的部分，同时，也是该项目所缔造的最大的扩展库，对它的设计和开发必须考虑到以后的扩展和完善，同时也应该做到最大的通用性，以方便于适用于其他项目；

在法电 MuFTAD 项目中，下载部分涉及的内容很多，包括：Http，Https 下载，HTTP 协议内容的解析；Cookies 的处理；多线程下载等等，这些都将是本扩展库需要完善，并且做到的。对 Https 的支持使用 Openssl 第三方库来完成，同时，也需要考虑到 SSL 的多线程支持；

在现阶段 Linux 环境下，并没有一个完善的工具具备上述功能：多线程，续传，支持 HTTPS 并无临时文件下载。

本项目开发过程中，参考了 Wget, Prozilla, Linuxdown 程序的实现和原理；

1.2. Illuminate 说明

下载程序的开发，大部分将会使用 Linux 系统库函数，而对于 SSL 的支持，将会采用 Openssl 库来实现；

为了便于使用，该程序将要求实现续传，多线程下载，并能完成所下载块的校验；

本扩展库的开发，将尽可能地按照独立的下载工具来实现；

1.3. Strcuture 结构

1.3.1. Rules Of Name 函数命名规则

该扩展库中函数一律使用 Mu_XXX 开头；

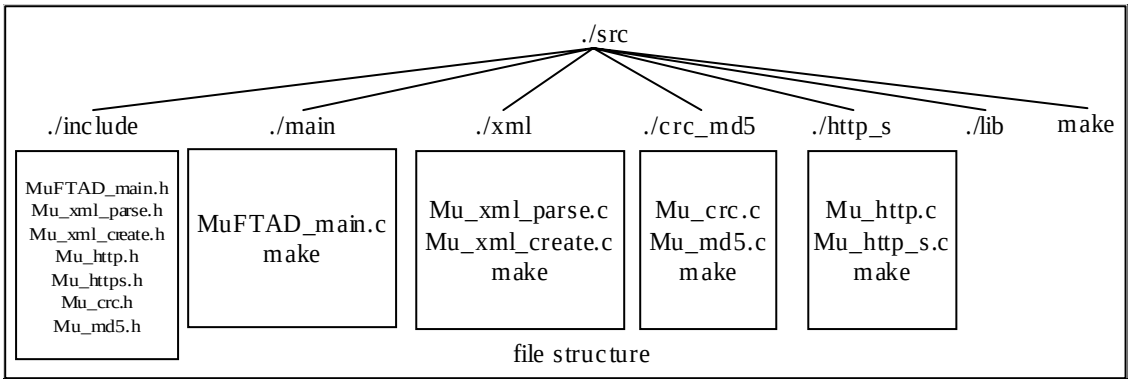
1.3.2. Library 标准库的使用

用 Openssl 库以完成 Https 安全连接；

1.3.3. File Organize 文件组织

[注意]:

1、此为整个项目的文件组织方式;



./src/include：文件夹，包含该项目中的所有头文件；

./src/main：文件夹，包含所有按法电《do3c00_SoftProtocol_0.1.0_RC1》流程所开发的程序；

./src/xml：文件夹，包含项目中所需要的 xml 处理库函数源代码；

./src/crc_md5：文件夹，包含项目中所需要的校验函数源代码，包括 CRC 和 MD5校验代码；

./src/http_s：文件夹，包含项目中所需要的与服务器交互的方式，包括 HTTP(s) GET、POST 方式；

./src/lib：文件夹，用于存储编译所生成的 xml、http 和 https、crc/md5库。软件编译连接时使用该文件夹下的库；

./src/make：文件，总的编译入口；

[注意]:

1、各对应文件夹下的源文件按需要添加，但是所作修改必须对 makefile 文件作相应的修改，以正确编译;

1.4. Reference 参考文献

O'Reilly.Network.Security.with.OpenSSL

An Introduction To Openssl Programming(Part I & II)
O'Reilly - HTTP Pocket Reference
openssl.org

Source Code of Wget
Source Code of Prozilla
Source Code of Linuxdown

rfc2246.pdf
rfc2617.pdf
rfc1738.pdf

2. Functions 功能描述

2.1. Resume 续传

续传部分的功能必须由下载库来保证，调用者只需要提供相应的 URL 地址，待写入文件句柄，还需要提供一个用于校验文件分段完整性的函数；

在本项目中，一个 URL 可以标记为一个文件也可标记为一个分段，在下载前，由调用者校验程序的完整性（CRC-32或 MD5校验）。调用本扩展库下载的 URL 内容存在二种可能：

1、重新下载，不使用续传功能；

2、以前下载过，现在需要续传；

这二种状态提供，以防止续传后，调用者校验内容不完整，从而放弃所有已下载内容，重新下载 URL 内容。

2.2. URL 内容大小获取

要实现多线程下载，我们必须首先获得我们需要下载内容的大小，才可以进行可靠，高效的多线程下载。

在本次实现过程中，我们设定最多使用线程数（MAX_THD），按线程来平均 URL 内容；同时，也考虑到 URL 内容过小，也限定线程最小下载内容（MAX_THD_LEN）。

在分线程的过程中，需要考虑到二种情况：

1、URL 内容整体小于 MAX_THD_LEN;

2、分多线程下载后，最后一个线程的下载块小于 MAX_THD_LEN；

对于第一种情况，只使用一个线程下载，对于指点二种情况，将最后一个线程要下载的内容合并到倒数第二个线程中去；

2.3. Multi-Threads 多线程下载

在多线程下载的过程中，需要考虑到写时的同步，记录写文件或是下载文件开始地址。对于续传，还必须考虑到续传开始地址（即定位）。

可以考虑对每个段使用一个 segment.seg 的记录文件，若未完成，则保留，若完成，则 unlink 该文件；

2.4. Thunder 线程异常

由于在整个过程中，没有人为的交互，通常用户不知道下载状态，也不可能进行相应的操作，所以，程序必须自己对多线程下载时的故障做出一些处理：

1、所有线程的网络不可达；

2、某个或是某几个线程下载不可达；（可以考虑使用平均速度来中断一个线程）；

2.5. Checksum 校验

在下载时,需要处理一些使用http://username:password等类型的，需要使用用户名和密码的HTTP方式，采用这种方式时，需要使用base64位加密用户名和密码username:password字段后，填充HTTP的报文头：Authorization字段。

2.6. Cookies

按照 HTTP 的标准，或是按照法电要求，set-cookies 字段是需要被处理的，在本项目中，不作任何处理，只需要将服务器返回的 set-cookie 字段内容存储并在下次请求服务器时，返还给服务器即可。

3. Requirement Of Routines 程序需求

3.1. 调用接口函数

函数在实现时，从调用者处获得相应的参数，包括处理类型，文件句柄，开始偏移量，是否续传，记录文件以及文件长度等参数。

处理类型包括：请求和下载；

设置此处理参数的目的是为了提高实现效率。针对普通的信令请求，服务器返回的信令不会太大，仅仅是 XML 文件；而下载文件一般会比较大；

若不分类型处理，在处理信令时，需要请求服务器，以获得实际长度，再分线程处理，这样将会降低信令请求的效率；而对于下载节目内容，如此处理将会提高整个下载效率；

文件句柄用于存储从服务器处获得的内容，包括信令和节目；

偏移量是从服务器取得内容时写入文件的开始位置，针对信令一般是文件的开始位置处，而对于节目下载，需要指定一个分段开始下载的位置；

是否续传用以续传或是在续传失败（校验结果出错）后，放弃全部分段内容，重新下载；

记录文件用于记录下载过程中是否已经下载过相应节目，重新开始可以从该记录文件中获取开始值，以续传节目；对于信令部分，不采用该参数；

[注意]:

1、对信令的请求访问，不采用续传，也不使用记录;

文件长度应用于下载分段时，用于分线程；

另外，还需要一个 URL 字串，由本扩展库负责解析出具体的 URL 信息；

3.2. Query For Length

在下载节目时，一律考虑以多线程的形式实现。在下载20X 类型的分段时，是可以预先知道分段大小的，但是对于40X 类型的节目，我们并不知道节目的大小。

为了便于分线程下载，我们在下载前需要以单线程请求服务器，以获得节目内容大小，以此对节目进行分线程下载；

请求节目长度类似于 GET 方式访问服务器文件，服务器返回该文件的内容。我们在此步骤中，需要处理的仅仅是服务器返回的 HTTP 报头,以获得 content_length 字段值；

3.3. Read For Record

为了方便续传，在下载前，我们必须考虑读取记录文件，以获得已写部分的开始位置，以及相关的线程信息；

3.4. Record

为了便于续传，在多线程下载过程中，需要一种记录文件，以便于下载被取消后，重新下载时可以接着文件未下载完成部分开始；

记录文件名在3.1.中说明；

3.5. Rates

在每一个线程开始下载时，都应该有取得一个下载时间，待写入一部分数据后，均需要计算下载速率，存入各个线程所维护的结构；

另外，平均下载速率以一个全局值存储，对该值的修改需要用到锁，以防止不同步；另外，在速率的计算上，若该值为0，则记入当前线程的速率值，若非0，则记入平均值；

3.6. Parse Header

项目中所有报文头，均为 HTTP 报文头，在处理服务器返回的内容时，必须首先处理报文头中的相关信息。包括：HTTP 状态码，Cookie 的存储，Location 值，正文段大小等信息；

上述信息均存入全局变量 HttpStatus 结构中；

3.7. Multi-Threads

在分线程处理时，需要考虑到二种情况：HTTP 和 HTTPS，二者在多线程实现时，差别很大，但是现阶段对于 HTTPS 的多线程，考虑预留接口；

在分线程时，考虑按线程数（MAX_THD）设置每一线程所需要下载的数据，并且，也必须限制最小可分线程数据大小 MAX_THD_LEN；

若所有线程的数据大小均小于 MAX_THD_LEN，线程数减半后设置线程，递归进行，若值为0则单线程下载；

若最后一个线程（分线程后的余量）的数据大小小于 MAX_THD_LEN，将该数据量合并入上一线程；

3.8. Build Query

构建请求报文，在整个项目中，请求报文分为二种：POST 和 GET，二者的请求报文格式是完全不一样的，所构建的报文以全局变量存储；

并且，报文的存储的区域考虑为动态存储区，以方便请求报文过大后，扩充存储区域；

3.9. Checksum

同2.5. 中说明

3.10. Cookies

同2.6. 中说明

3.11. Location

当向服务器发起请求后，服务器返303重定向状态码时，Device 必须能取得该值，并重新解析，重新连接服务器；

3.11. Parse Addr

程序从调用者处获得 URL 地址，在进行相应的处理前，必须解析该 URL 地址信息，存储到 url 结构中；

4. Environment 运行环境

4.1. Device 设备

达芬奇数字平台 TMS320DM6446 双核 ARM9+DM64X SRAM(8M)、FLASH(256M)、NAND flash (2G)存储。

4.2. Software 支持软件

在 Linux 内核主机上均可运行。

4.3. Interface 接口

本扩展库支持所有应用网络的项目；

4.4. Control 控制

略