7-zip 压缩算法及 C SDK 使用

Auth: dwh0403@163.com

1. 介绍

官方网址：中文：http://sparanoid.com/lab/7z/ 英文：http://www.7-zip.org/
SDK 下载网址：中文：http://sparanoid.com/lab/7z/ 英文：http://www.7-zip.org/sdk.html
SDK 开发支持语言：Java C/C++ C#

缺点：LZMA SDK 相关文档不完整.

7-zip 当前最新稳定版本为：7-Zip 9.20 稳定版，最后更新时间为：2010-11-18
7-zip 当前最新版本为：7-Zip 9.32 alpha，最后更新时间为：2013-12-01

7z 是一种全新的压缩格式，它拥有极高的压缩比。7z 格式的主要特征：

- 开放的结构
- 高压缩比
- 强大的 AES-256 加密
- 能够兼容任意压缩、转换、加密算法
- 最高支持 16000000000 GB 的文件压缩
- 以 Unicode 为标准的文件名
- 支持固实压缩
- 支持文件头压缩

7z 已公开了结构编辑功能，所以它可以支持任何一种新的压缩算法。到目前为止，下列压缩算法已被整合到了 **7z** 中：

压缩算法	备注
LZMA	改良与优化后的 LZ77 算法
LZMA2	改良的 LZMA 算法
PPMD	基于 Dmitry Shkarin 的 PPMdH 算法
BCJ	32 位 x86 可执行文件转换程序
BCJ2	32 位 x86 可执行文件转换程序
BZip2	标准 BWT 算法
Deflate	标准 LZ77-based 算法

LZMA 算法是 7z 格式的默认算法。LZMA 算法具有以下主要特征：

- 高压缩比
- 可变字典大小（最大 4 GB）
- 压缩速度：运行于 2 GHz 的处理器可达到 1 MB/秒

- 解压缩速度：运行于 2 GHz 的处理器可达到 10-20 MB/秒
- 较小的解压缩内存需求（取决于字典大小）
- 较小的解压缩代码：约 5 KB
- 支持 Pentium 4 的超线程（Hyper-Threading）技术及多处理器

LZMA 压缩算法非常适于应用程序的内嵌。LZMA 发布于 GNU LGPL 许可协议之下，如果您想使用 LZMA 的代码，您可以通过 发送信息到 LZMA 开发部 来咨询和自定义设计代码及制定开发者的使用许可。您也可以点击此处来查看有关 LZMA SDK 的信息： LZMA SDK.
7z 是 7-Zip 发布于 GNU LGPL 许可下的子程序。您可从 下载页面 下载 7-Zip 的源代码。
支持 7z 压缩格式的应用程序：WinRAR、PowerArchiver、TUGZip、IZArc。

2 LZMA SDK 介绍

SDK 下载网址：中文：http://sparanoid.com/lab/7z/ 英文：http://www.7-zip.org/sdk.html
SDK 开发支持语言：Java C/C++ C#
9.20 版本下载地址：http://downloads.sourceforge.net/sevenzip/lzma920.tar.bz2，新增用于安装包的精简版 SFX 自释放模块。

3. LZMA SDK 代码分布

下载 lzma920.tar.bz2 后，解压目录如下：

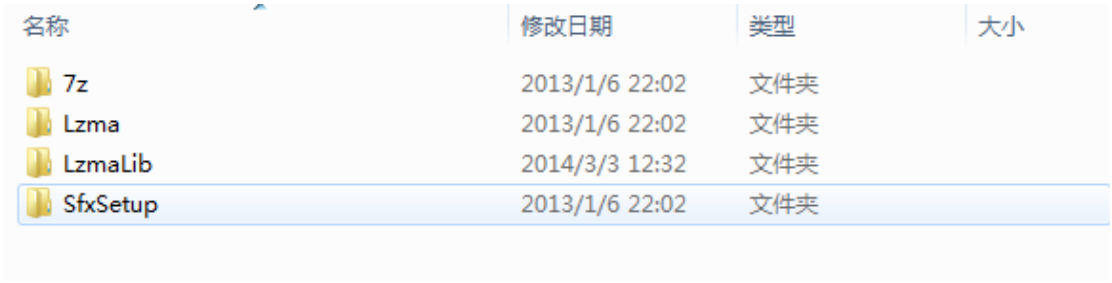
LZMA SDK 包含以下内容:

- **C++** source code of LZMA Encoder and Decoder
- **C++** source code for .7z compression and decompression (reduced version)
- **ANSI-C** compatible source code for LZMA / LZMA2 / XZ compression and decompression
- **ANSI-C** compatible source code for 7z decompression with example
- **C#** source code for LZMA compression and decompression
- **Java** source code for LZMA compression and decompression
- **lzma.exe** for .lzma compression and decompression

- **7zr.exe** to work with 7z archives (reduced version of 7z.exe from 7-Zip)
- **ANSI-C** and **C++** source code in LZMA SDK is subset of source code of 7-Zip.

ANSI-C LZMA 解压缩代码是从原始的 **C++ 源代码**转换到 **C**。并简化和优化了代码的大小。但它依然和 7-Zip 的 LZMA 完全兼容。

C 目录：
Util 和相对应的文件。
Util 目录内容如下：



目录名	说明	支持平台
7z	生成可执行程序 7z	Linux/Windows
Lzma	生成可执行程序 lzma	Linux/Windows
LzmaLib	生成 LZMA.dll 动态库	Windows
SfxSetup	生成可执行程序 7zS2.sfx	Windows

CPP 目录内容如下：

目录名	说明	支持平台
7z	生成可执行程序 7z	Linux/Windows

Windows: CPP\7zip\UI\Client7z -> client7z.exe
CPP\7zip\Bundles\Alone7z –> 7zr.exe
CPP\7zip\Bundles\LzmaCon-> lzma.exe
Linux: CPP\7zip\Bundles\LzmaCon -> lzma

Common	公共包含的文件	Linux/Windows
Windows	Windows 平台下包含的文件	Windows

Java 目录
主要包含 7zip.jar 和使用的 Java 源代码

结论：对于 Linux 下程序集成开发采用 C 语言 SDK 更加方便。

4. 使用 LZMA C SDK

C 版本 SDK 已经实现了针对输入文件压缩和解压缩的功能，具体功能在：
C/Util/Lzma/LzmaUtil.c 中的 main2 函数中实现，可以从 main 函数中直接调用。
int main2(int numArgs, const char *args[], char *rs)

实现 lzma 程序的 main 函数如下：

```
int MY_CDECL main(int numArgs, const char *args[])
{
  char rs[10*1024*1024] = { 0 };          // 用于中间过程的内存，原始大小为 80K
  int res = main2(numArgs, args, rs);
  fputs(rs, stdout);
  return res;
}
```

对于 lzma 程序来讲，使用帮助如下：

```
lzma <e|d> inputFile outputFile
                    e: encode file
                    d: decode file
```

因此如果使用文件解压缩的话，只需要将 LzmaUtil.c 中的 main 函数使用宏定义控制，将相关文件编译成动态库使用即可。

例如解压缩函数可定义如下：

```
int decode_file(const char *in_file_name, const char* out_file_name)
{
    char buf[10*1024*1024];
    char *argvs[4];
    argvs[0] = NULL;
    argvs[1] = "d";
    argvs[2] = in_file_name;
    argvs[3] = out_file_name;
    return main2(4, argvs, buf);
}
```

压缩函数只需要将 argv[1]=”d”替换成，argv[1]=”e”即可