c语言 图片隐写术,C/C++信息隐写术（二）之字符串藏入BMP文件

转载

食色也 于 2021-05-27 10:42:34 发布 135 收藏
文章标签： c语言 图片隐写术

我们这一节用代码实现，把字符串藏入BMP文件，并且能正常读取出来。

看这篇的同学请先阅读第一篇了解理论：

下面开始进入此节：

从上一节，我们知道了Bmp文件的结构，如下图所示：

其中最关键的两个结构体BITMAPFILEHEADER和BITMAPINFOHEADER，这里面保存了这个Bmp文件的很多信息。

恰好，Windows给我们提供了这个两个结构体，如下图所示：

```
typedef struct tagBITMAPFILEHEADER {
WORD bfType;
DWORD bfSize;
WORD bfReserved1;
WORD bfReserved2;
DWORD bfOffBits;
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER, *PBITMAPFILEHEADER;
typedef struct tagBITMAPINFOHEADER{
DWORD biSize;
LONG biWidth;
LONG biHeight;
WORD biPlanes;
WORD biBitCount;
DWORD biCompression;
DWORD biSizeImage;
LONG biXPelsPerMeter;
LONG biYPelsPerMeter;
DWORD biClrUsed;
```

DWORD biClrImportant;

} BITMAPINFOHEADER, FAR *LPBITMAPINFOHEADER, *PBITMAPINFOHEADER;

我们在到010Editor看看(图如下)：

关于高度为负的问题，在第一节已经说明，不知道的同学请从文章最上面的链接进入第一节。在此不再说明。

那么问题就简单了，现在这个程序的思路就是：

1.用C/C++代码读取图片文件里面的这两个结构体。

2.读取这个文件到内存中。

3.获取bfOffBIts，再获取alpha通道(+4)。

4.把数据拆分，插入到alpha通道。

5.保存文件。

6.读取被修改文件的alpha通道，组合成字符串。

理论就是这么简单：

下面是程序源码打包下载地址：

下面是程序源码：

dwBmpSize.h

```
#pragma once
#include
#include
using namespace std;
class CBMPHide
{
public:
CBMPHide();
~CBMPHide();
bool setBmpFileName(char* szFileName);//设置Bmp文件名
int getBmpWidth();//获取宽度
int getBmpHeight();//获取高度
int getBmpBitCount();//获取Bit总数
bool save();
bool hideString2BMP(char* szStr2Hide);//隐藏String到BMP文件中
```

```
void showStringInBmp(char* szBmpFIleName=NULL);//展示
private:
DWORD dwBmpSize;//图片文件大小
string sBmpFileName;
LPBYTE pBuf;//用于存放图片信息的内存
BITMAPFILEHEADER* m_fileHdr;
BITMAPINFOHEADER* m_infoHdr;
};
```

dwBmpSIze.cpp

```
#include "dwBmpSize.h"
CBMPHide::CBMPHide()
{
sBmpFileName = "";
pBuf = 0;
dwBmpSize = 0;
}
CBMPHide::~CBMPHide()
{
}
bool CBMPHide::setBmpFileName(char* szFileName)
{
this->sBmpFileName = szFileName;
if (pBuf)//如果已经生成就释放掉
{
delete[]pBuf;
}
HANDLE hfile = CreateFileA(szFileName, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ |
FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0, 0);
if (hfile == INVALID_HANDLE_VALUE)
{
return false;
```

```
}
//和struct BITMAPFILEHEADER bmfh里面的 bfSize的大小应该是一样的。
dwBmpSize = GetFileSize(hfile, 0);//获取文件的大小
pBuf = new byte[dwBmpSize];
DWORD dwRead = 0;
ReadFile(hfile, pBuf, dwBmpSize, &dwRead, 0);
if (dwRead != dwBmpSize)
{
delete[]pBuf;
pBuf = 0;
return false;
}
CloseHandle(hfile);
m_fileHdr = (BITMAPFILEHEADER*)pBuf;
m_infoHdr = (BITMAPINFOHEADER*)(pBuf + sizeof(BITMAPFILEHEADER));
return true;//成功话就是文件的内容读取到pBuf里面
}
int CBMPHide::getBmpWidth()
{
return m_infoHdr->biWidth;
}
int CBMPHide::getBmpHeight()
{
return m_infoHdr->biHeight;
}
int CBMPHide::getBmpBitCount()
{
return m_infoHdr->biBitCount;
}
bool CBMPHide::save()
{
```

```
string sDstFileName = sBmpFileName + ".hide.bmp";
HANDLE hfile = CreateFileA(sDstFileName.c_str(),
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
CREATE_ALWAYS, 0, 0);
if (hfile == INVALID_HANDLE_VALUE)
{
return false;
}
DWORD dwWritten = 0;
WriteFile(hfile, pBuf, dwBmpSize, &dwWritten, 0);
if (dwBmpSize != dwWritten)
{
return false;
}
CloseHandle(hfile);
return true;
}
//隐藏一个字符串到图片中，把字符串拆成字节，写入每个像素的alpha通道中
bool CBMPHide::hideString2BMP(char* szStr2Hide)
{
LPBYTE pAlpha = pBuf + m_fileHdr->bfOffBits + 3;//第一个像素的通道位置
int nHide;//成功隐藏的字节数
//每次循环写入一个字节，吸入alpha通道
///(pAlpha - pBuf) < m_fileHdr->bfSize这个是判断字符串是太大，图片不能隐藏
for (nHide = 0; (pAlpha - pBuf) < m_fileHdr->bfSize && szStr2Hide[nHide] != 0; nHide++, pAlpha += 4)
{
*pAlpha = szStr2Hide[nHide];//写入一个字节
}
return true;
```

```
}
void CBMPHide::showStringInBmp(char* szBmpFIleName/*=NULL*/)
{
string sDstFileName="";
if (szBmpFIleName == 0)
{
sDstFileName = sBmpFileName + ".hide.bmp";
}
else
sDstFileName = szBmpFIleName;
HANDLE hfile = CreateFileA(sDstFileName.c_str(),
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
OPEN_EXISTING, 0, 0);
if (hfile == INVALID_HANDLE_VALUE)
{
return;
}
DWORD dwSize = GetFileSize(hfile, 0);
LPBYTE pBuf1 = new byte[dwSize];
DWORD dwRead = 0;
ReadFile(hfile, pBuf1, dwSize, &dwRead, 0);
CloseHandle(hfile);
//文件内容读取到pBuf1中
BITMAPFILEHEADER *pHdr = (BITMAPFILEHEADER *)pBuf1;
LPBYTE pStr = pBuf1 + pHdr->bfOffBits + 3;
char szTmp[1280];
RtlZeroMemory(szTmp, 1280);
for (int i = 0; i < 1280; i++)
{
```

```
if (*pStr == 0 || *pStr == 0xFF)
{
break;
}
szTmp[i] = *pStr;
pStr += 4;
}
printf_s(szTmp);
delete[]pBuf1;
}
```

main.h

```
#include
#include "dwBmpSize.h"
int main()
{
CBMPHide hide;
hide.setBmpFileName("test.bmp");
printf_s("test.bmp width:%d,height:%d,bitCount%d\n",
hide.getBmpWidth(),
hide.getBmpHeight(),
hide.getBmpBitCount());
hide.hideString2BMP("Hello Word");
hide.save();
hide.showStringInBmp("test.bmp.hide.bmp");
getchar();
return 0;
}
```

程序运行结果如下：

在此不一一举出。

在这里：可能出现特殊情况，比如写入了0或oxFF(判断自有数据是否结束标志)

在下面一节中，我们解决这个问题，并且，把一个不大的txt文本插入到图片里面去。