

whyask37's blog

Project SMD

5. SFmpq (ShadowFlare's MPQ Library) and examples



why do you ask
2013. 9. 11. 9:43

add neighbor

**This document is a course-style log of developing the StarCraft editor .
Since I am also in the position of learning, what is the point of calling this a 'course';
This is a project to make a map editor, realizing that making a map editor involves understanding the overall structure of the game.**

<http://blog.naver.com/whyask37>

You don't allow right-mouse; I just did it.

From this time on, I will have to work hard to print the unit. Yes.

Unit output is more than 10^{10} times the terrain output, so there's a lot to do. Terrain output is easy, but the difficulty of unit output is...

In order to print the unit, a lot of lectures are required. It requires quite a bit of basic knowledge.

So let's get started.

Let's get started. First of all, you need to know what MPQ is, right?

Simply put, MPQ can be thought of as a folder with no findfirst and only fopen.

The file is not listed, but the file can be opened. (*)

(Because the detailed explanation of MPQ is needed for why this is, below)

The function to read the MPQ is in storm.dll.

Any MPQ protect will be stormed. In fact, of course? Star reads mpq with storm.

But the function that writes MPQ is not in storm.dll. (Based on storm.dll of Star1. From Star2)

If you are a map editor, you should basically be able to read and write scx files.

Storm alone can only read scx files (which is also MPQ), not write them. It's really difficult.

whyask37's blog

(staredit.exe can obviously read and write this map. The function to write the map is in staredit.exe.)

(It's not even staredit.DLL, but it's amazing how you can call and use internal application functions)

stormlib: Reversing storm.dll so you can read and write MPQ files without storm.dll.

mpqctl: I do not use VB, so I do not know. This is the MPQ library used in previous versions of WinMPQ. I think it's for VB.

sfmpq : **[which we will use]** Implemented after the MPQ format was released.

Other than that, I tried making mpqrw (my own work) for practice, but it's not stable enough to use here.

We will use sfmpq, which has been used for WinMPQ and numerous protectors and unprotectors, and has proven stability in the future.

(cf) Ladik's MPQ Editor uses stormlib.)

(*) You need to know the MPQ format itself to know if the file listing is not working. If the file listing is not already in (listfile), findfirst is not possible. Did you load StarDat.mpq using ListFile.txt in the previous lecture? Open ListFile.txt and you'll understand what I mean...

like. Let's download sfmpq.

<http://sfsrealm.hopto.org/downloads/SFmpqapi.html>

Let's download sfmpqapi.zip from

sfmpq.h, sfmpq_static.lib, and sfmpq_static.h are required in the compressed file. Put this in your project folder appropriately.

(If you open a zip file, the file name pops up right? This may not be the case in MPQ.)

(When did you create the project? Make it a console application now.)

And... Let's put the contents of chkparse.zip (attachment from last lecture) into the project and make it the same as the attachment start zip

One

whyask37's blog

(I named the project mapuecompiler. You can do anything)

So let's get started. MPQ related code.

Main source: http://sfsrealm.hopto.org/inside_mopaq/chapter3.htm#opening_a_file_inside_an_mpq

Add the following code to main.cpp.

Colored By [Color Scripter](#)™

```

One // omit the first part
2
3 //Open the MPQ file and extract staledit\scenario.chk as scenario.chk.
4 {
5     DWORD fsize;
6     FILE* fout;
7     MPQHANDLE hMPQ;
8     MPQHANDLE hFile;
9
10    //One. Open map.scx and open staledit\scenario.chk inside it.
11    if (!SFileOpenArchive( "map.scx" , 0, 0, &hMPQ)) {
12        // SFileOpenArchive([file name], 0, 0, &[MPQ handle to be returned]);
13        printf( "SFileOpenArchive failed. [%d]\n" , GetLastError());
14        return -1;
15    }
16
17    SFileOpenFileEx(hMPQ, "staledit\scenario.chk" , 0, &hFile);
18    // SFileOpenFileEx([MPQ handle], [file name], 0, &[return file handle]);
19
20    fsize = SFileGetFileSize(hFile, NULL); //파일의 크기를 얻는다.
21
22    //2. scenario.chk를 연다.
23    fout = fopen("scenario.chk", "wb");
24    if(fout == NULL) {
25        SFileCloseFile(hFile); //fclose에 해당
26        SFileCloseArchive(hMPQ); // MPQ 닫기
27        printf("Cannot open scenario.chk!\n");
28        return -1;
29    }
30
31    //3. 4096byte씩 map.scx\staledit\scenario.chk에서 scenario.chk로 복사.
32    char buffer[4096];
33    while(fsize > 0) {
34        DWORD transfersize = min(4096, fsize);
35        DWORD readbyte;

```

One

whyask37's blog

```

41     SFileCloseArchive(hMPQ);
42     return -1;
43 }
44     fwrite(buffer, 1, readbyte, fout);
45     fsize -= transfersize;
46 }
47
48 //4. 핸들을 다 닫는다.
49     fclose(fout);
50     SFileCloseFile(hFile);
51     SFileCloseArchive(hMPQ);
52 }
53
54 //뒤에부분 생략

```

주석을 보고 이해해주시길 바랍니다. 간단히 말해서

1. map.scx를 엽니다.
2. map.scx\staredit\scenario.chk를 scenario.chk로 복사합니다.
3. map.scx를 닫습니다.

입니다. 이렇게 하면 main.cpp가 첨부파일의 main.cpp와 같이 됩니다.

컴파일하면 map.scx 안의 scenario.chk가 자동으로 읽혀서 chkparse되는것을 볼 수 있습니다.

sfmpq 함수 설명

sfmpq에서 MPQHANDLE는 읽거나 쓸 MPQ / 파일의 핸들을 의미합니다. (WinAPI에서 CreateFile에서 리턴되는 것과 비슷)

1. 읽기용 (storm에서 쓰는 인터페이스를 그대로 따릅니다. storm은 MPQ 읽기만 가능한거 잊지 않으셨죠?)

SFileOpenArchive(fname, 0, 0, &hMPQ) : MPQ 파일을 읽기용으로 엽니다.

SFileCloseArchive(hMPQ) : SFileOpenArchive로 연 MPQ 파일을 닫습니다. 쓰기용으로 연 MPQ 파일은 이걸로 닫으면 큰일납니다;

SFileOpenFile(hMPQ, fname, 0, &hFile) : MPQ 안에 들어있는 파일을 엽니다.

SFileCloseFile(&hFile) : SFileOpenFile로 열었던 파일 핸들을 닫습니다.

SFileReadFile(hFile, buffer, wantedbyte, &readbyte, NULL) : wantedbyte만큼 hFile에서 읽어가지고 buffer에 저장합니다. 만약 wantedbyte만큼 데이터를 읽는데 실패했을 경우에는 wantedbyte보다 조금 적게 읽을수도 있습니다. 언제나 실제로 읽은 바이트수를 readbyte에 리턴합니다. (readbyte는 fread의 리턴값과 비슷한 역할)

2. 쓰기용 (Impqapi에서 쓰는 인터페이스를 그대로 따릅니다. Impqapi는 stardit.exe를 해킹해서 MPQ 쓰기도 가능하게 한 라이브러리)

hMPQ = MpqOpenArchiveForUpdate(fname, flag, maxfilen) : MPQ 파일을 쓰기용으로 엽니다. maxfilen은 1024면 됩니다.

MpqCloseArchive(hMPQ) : MpqOpenArchiveForUpdate로 열린 MPQ 파일을 닫습니다. 읽기용으로 연건 이걸

whyask37's blog

MAFA_ENCRYPT : 암호화. MPQ 파일을 암호화한다고 합니다만 암호 뚫기가 장난 아니게 쉽습니다;

MAFA_MODCRYPTKEY : MAFA_ENCRYPT에서 쓰는 암호화 키를 변조를 한번 더 해줍니다.

MpqAddWaveFromBuffer(hMPQ, data, datasize, fname, flag, complevel) : 이걸 WAV 파일을 넣어줍니다. complevel은 다음이 될 수 있습니다.

MAWA_QUALITY_HIGH : 고품질이라고 하더군요. 막귀라서 진짤진 잘 모르겠습니다; 용량이 조금 압축되요.

MAWA_QUALITY_LOW : 저품질이라고 하더군요. 막귀라서 진짤진 잘 모르겠습니다; 용량이 약간 더 압축되요

MAWA_QUALITY_MEDIUM : 중간정도 품질이라고 하더군요. 막귀라서 진짤진 잘 모르겠습니다; 용량이 좀 더 압축되요.

MpqAddFileFromBufferEx 등등 다른 함수의 설명은 sfmpqapi.h와 다음 링크를 참조하세요.

http://web.archive.org/web/20120222093346/http://wiki.devklog.net/index.php?title=The_MoPaQ_Archive_Format

그런데 여기서 끝내면 뭔가 서운한것같아요.

제 생각에 sfmpq 인터페이스 자체가 뭔가 길고 **복잡하고 짜증난것같네요.**

sfmpq를 쓰기 쉽도록 만들어주는 래퍼를 만들어보도록 합시다. 클래스화시키는거죠.

대충 이런 형식으로요.

Colored By **Color Scripter™**

```

1  #ifndef MPQFCNT_HEADER
2  #define MPQFCNT_HEADER
3
4  #include "SFmpq_static.h"
5  #include "chunk.h"
6
7  class MpqRead {
8  public:
9      MpqRead();
10     ~MpqRead();
11
12     bool open(const char* fname);
13     void close();
14
15     bool extract(const char* fname, Chunk* chk);
16
17 private:
18     MPQHANDLE _mpqh;
19 };
20

```

whyask37's blog

```

26 ~MpqWrite();
27
28 bool open(const char* fname);
29 void close();
30
31 bool appendfile(const char* fname, const Chunk* chk);
32 bool appendwave(const char* fname, const Chunk* chk, int complevel = MAWA_QUALITY_HIGH);
33 private:
34     MPQHANDLE _mpqh;
35 };
36
37 #endif
38

```

이러이러한 함수가 있습니다.

함수명	설명
MpqRead::open(const char* fname);	MPQ 파일을 읽기 목적으로 엽니다.
MpqRead::close();	MpqRead로 열린 MPQ 파일을 닫습니다.
MpqRead::extract(const char *fname, Chunk *dst);	MPQ 안의 fname 파일을 dst에 복사합니다.
MpqWrite::open(const char* fname);	MPQ 파일을 쓰기 목적으로 엽니다.
MpqWrite::appendfile(const char* fname, const Chunk *src);	MpqWrite로 열린 MPQ 파일에 파일을 집어넣습니다.
MpqWrite::appendwave(const char* fname, const Chunk* src, int complevel);	MpqWrite로 열린 MPQ 파일에 웨이브(.wav) 파일을 집어넣습니다.

총 7개의 함수밖에 없어요. 참 간단하죠? 이 인터페이스로 mpqfcnt.cpp, mpqfcnt.h를 구현했습니다.

Colored By [Color Scripter™](#)

```

1 MpqRead mr;
2 Chunk chk;
3
4 mr.open("test.scx");
5 mr.extract("staredit\\scenario.chk", &chk);
6 mr.close();

```

하면 test.scx\staredit\scenario.chk 가 chk에 저장되는거죠. chk는 메모리에 있고요. scenario.chk 등등 파일을 만들지 않아도 됩니다.

그러면 scenario.chk를 fopen하듯이 Chunk 파일을 fread fwrite 등등 할 수 있으면 더 좋겠습니다. 해

whyask37's blog

```

4   Chunk chk;
5   uint32 fpos;
6 }ChkFile;
7
8 ChkFile* chkopen(Chunk &chk);
9 void chkclosel(ChkFile* f);
10 int chktell(ChkFile* f);
11 int chkseek(ChkFile* f, int offset, int origin);
12 int chkread(void* dst, size_t size, ChkFile *f);
13 int chkwrite(const void* src, size_t size, ChkFile *f);

```

이렇게 chunk.h에 추가해보면 될듯합니다. fopen -> chkopen 등등...
(chkscanf나 chkprintf같은건 필요없을것같아서 만들지 않았습니다)

이제 이 함수들을 구현하면 됩니다. 한가지 문제가 있는것만 제외하면요.

크기 0인 파일에 fwrite를 하면 파일 크기가 자동으로 늘어납니다.

chkwrite도 마찬가지로 해야겠죠? 그런데 Chunk는 크기가 고정되어있습니다. (malloc으로 메모리를 고정시켜서 할당)

아무래도 ChunkData 구조체를 data[1] 트릭 대신에 std::vector<uint8>를 쓰면 더 좋을것같아요.

이런 생각을 기반으로 수정된 chunk.cpp, 그리고 mpqfcnt.cpp입니다. std::vector의 생성자를 불러주기 위해 malloc ~ free를 쓰지 않고 new~delete를 쓰고 있으며, 그 외에는 그냥 uint8[1]을 std::vector<uint8>로 바꾸기 위한 작업입니다.

chunk.cpp

Colored By Color Scripter™

```

1  #include "chunk.h"
2  #include <stdlib.h>
3  #include <vector>
4
5  struct ChunkData {
6      uint8 _type[4];
7      uint32 _size;
8      std::vector<uint8> _arr;
9  };
10
11 Chunk::Chunk() : _data(nullptr) { init("", "", 0); }
12 Chunk::~Chunk() {}
13
14

```

whyask37's blog

```
20
21 int Chunk::init(const void* type, int size) {
22     ChunkData *chkdata = new ChunkData; //std::vector의 생성자를 불러주기 위해서
23     if(chkdata == nullptr) throw std::bad_exception("Memory allocation failed.");
24     memcpy(chkdata->_type, type, 4);
25     chkdata->_size = size;
26     memset(chkdata->_arr.data(), 0, size);
27
28     _data.reset(chkdata);
29
30     return 0;
31 }
32
33
34
35 int Chunk::resize(int newsize) {
36     _data->_arr.resize(newsize);
37     _data->_size = newsize;
38     return 0;
39 }
40
41
42
43 uint8& Chunk::at(uint32 index) {
44     if(index > _data->_size) throw std::bad_exception("Index violation");
45     return _data->_arr[index];
46 }
47
48
49 uint8 Chunk::at(uint32 index) const{
50     if(index > _data->_size) throw std::bad_exception("Index violation");
51     return _data->_arr[index];
52 }
53
54
55 uint8* Chunk::data() { return _data->_arr.data(); }
56 const uint8* Chunk::data() const { return _data->_arr.data(); }
57
58
59 uint32 Chunk::size() const {
60     return _data->_size;
61 }
62
63
64 const uint8* Chunk::type() const {
65     return _data->_type;
66 }
```


whyask37's blog

```
72 ChkFile* chkopen(Chunk &chk) {
73     ChkFile* f = new ChkFile;
74     f->chk = chk;
75     f->fpos = 0;
76     return f;
77 }
78
79
80 void chkclosel(ChkFile* f) {
81     delete f;
82 }
83
84
85
86 int chktell(ChkFile* f) {
87     return f->fpos;
88 }
89
90
91 int chkseek(ChkFile* f, int offset, int origin) {
92     switch(origin) {
93     case SEEK_SET:
94         f->fpos = offset;
95         break;
96
97     case SEEK_CUR:
98         f->fpos += offset;
99         break;
100
101     case SEEK_END:
102         f->fpos = f->chk.size() - offset;
103         break;
104
105     default:
106         break;
107     }
108
109     if(f->fpos > f->chk.size()) f->fpos = f->chk.size();
110     return 0;
111 }
112
113
114
115 int chkread(void* dst, size_t size, ChkFile *f) {
116     uint32 copylen = f->chk.size() - f->fpos;
117     if(copylen > size) copylen = size;
118     memcpy(dst, f->chk.data() + f->fpos, copylen);
```

whyask37's blog

```

124 int chkwrite(const void* src, size_t size, ChkFile *f) {
125     uint32 copylen = f->chk.size() - f->fpos;
126     if(copylen > size) {
127         copylen = size;
128     }
129     else if(copylen < size) {
130         f->chk.resize(size + f->fpos); //공간을 조금 늘려줍니다.
131     }
132     memcpy(f->chk.data() + f->fpos, src, copylen);
133     f->fpos += copylen;
134     return copylen;
135 }
136

```

mpqfcnt.cpp

Colored By [Color Scripter™](#)

```

1  #include "mpqfcnt.h"
2
3  // MpqRead
4
5  MpqRead::MpqRead() : _mpqh(nullptr) {}
6  MpqRead::~MpqRead() { close(); }
7
8
9  bool MpqRead::extract(const char* fname, Chunk* dst) {
10     if(!_mpqh) return false;
11
12     MPQHANDLE _mpqf;
13     if(!SFileOpenFile(fname, &_ampqf)) return false;
14     DWORD fsize = SFileGetFileSize(_mpqf, NULL);
15     DWORD rsize;
16
17     Chunk chk;
18     chk.init(fsize);
19     SFileReadFile(_mpqf, chk.data(), fsize, &rsize, NULL);
20     SFileCloseFile(_mpqf);
21
22     if(rsize != fsize) return false;
23     *dst = chk;
24     return true;
25 }
26
27
28 bool MpqRead::open(const char* fname) {
29     if(!_mpqh) SFileCloseArchive(_mpqh);
30     if(!SFileOpenArchive(fname, 0, 0, &_mpqh)) {

```

whyask37's blog

```
35 }
36
37
38 void MpqRead::close() {
39     if(_mpqh) {
40         SFileCloseArchive(_mpqh);
41         _mpqh = nullptr;
42     }
43 }
44
45
46
47 // MpqWrite
48
49 MpqWrite::MpqWrite() : _mpqh(nullptr) {}
50 MpqWrite::~MpqWrite() { close(); }
51
52
53 bool MpqWrite::appendfile(const char* fname, const Chunk *chk) {
54     //MpqAddFileFromBuffer is mad. They don't use LPCVOID;
55     Chunk chk2;
56     chk2.init(chk->size());
57     memcpy(chk2.data(), chk->data(), chk->size());
58     BOOL res = MpqAddFileFromBuffer(_mpqh, chk2.data(), chk2.size(), fname, MAFA_COMPRESS | MAFA_ENCRY
59     return res != 0;
60 }
61
62
63 bool MpqWrite::appendwave(const char* fname, const Chunk *chk, int complevel) {
64     //MpqAddFileFromBuffer is mad. They don't use LPCVOID;
65     Chunk chk2;
66     chk2.init(chk->size());
67     memcpy(chk2.data(), chk->data(), chk->size());
68     BOOL res = MpqAddWaveFromBuffer(_mpqh, chk2.data(), chk2.size(), fname, MAFA_COMPRESS | MAFA_ENCI
69     return res != 0;
70 }
71
72
73 bool MpqWrite::open(const char* fname) {
74     _mpqh = MpqOpenArchiveForUpdate(fname, MOAU_CREATE_ALWAYS, 1024); // 1024개 이상 파일을 넣을일이 없
75     return _mpqh != nullptr;
76 }
77
78
79 void MpqWrite::close() {
80     if(_mpqh) {
81         MpqCloseArchive(_mpqh);
```

whyask37's blog

대충 이런 함수를 가지고 main 함수를 새로 짜보죠.

Colored By [Color Scripter™](#)

```
1 int main() {
2     std::vector<Chunk> chktb;
3     std::vector<Chunk>::iterator it;
4
5
6     // chkparse 파트
7     MpqRead mr;
8     Chunk chk;
9
10    //1. map.scx를 열어서 chk 추출
11    mr.open("map.scx");
12    mr.extract("staredit\\scenario.chk", &chk);
13
14    //2. chk를 접근하는 핸들을 하나 만들자.
15    ChkFile *f;
16    f = chkopen(chk);
17    const uint32 fsize = chk.size();
18
19    //3. chkparse 로직을 돌린다
20    while(1) {
21        if(chktell(f) + 8 > fsize) break;
22
23        uint8 type[4];
24        int len;
25
26        chkread(type, 4, f);
27        chkread(&len, 4, f);
28
29        if(len > 0) {
30            if(chktell(f) + len > fsize) break;
31            Chunk chk;
32            chk.init(type, len);
33            chkread(chk.data(), len, f);
34            chktb.push_back(chk);
35        }
36
37        else {
38            if(chktell(f) + len < 0) break;
39            chkseek(f, len, SEEK_CUR);
40        }
41    }
42}
```

One

whyask37's blog

```

48  for(it = chktb.begin() ; it != chktb.end() ; it++) {
49      uint8 type[5] = {0};
50      uint32 len;
51      uint32 crc;
52      //후략

```

이런 식으로 바꾸면 됩니다. 전체 소스는 end.zip으로 첨부했습니다.

MPQ 포맷에 대해 간략하게 (간략하게) 설명하겠습니다.

MPQ 파일은 파일들을 해시 테이블에 놓고 관리합니다.

파일 찾는 순서 : [찾고자 하는 파일 이름] -> [Hash Table] -> [Block Table] -> array[Block]

Hash Table은 각 파일들의 entry가 어느 Block에 속해있는지를 알려줍니다. 구조는

[hashA, hashB, locale, BlockTableEntry] * (MPQ 파일의 최대 파일 갯수)

이런 식으로 되어있습니다. MPQ 파일은 MPQ 안에 들어있는 파일명을 저장하지 않습니다. 파일명의 해쉬값만 저장합니다.

Hash Table은 이름에서도 알 수 있듯이 **해시 테이블**입니다. 파일을 찾는 속도가 엄청 빠르죠. strcmp를 가지고 파일이름을 일일이 비교해가면서 '내 파일이 어디있나' 찾을 필요가 없습니다. 선형 연쇄로 충돌을 처리합니다. 이중해싱이었으면 더 좋겠습니다만

Block Table은 각 파일들이 어느 Block에 저장되어있는지를 알려줍니다. 원본크기(압축전), 압축된크기, 파일상에서 블록의 위치 등등을 가지고 있는 엔트리들의 배열로 되어있습니다.

Block은 각 파일들을 압축하고 암호화한 것들입니다.

아 막판에 조금 살렸다.

fix 1 : 웬지 말투가 sfmpq를 제가 만든것같은 느낌이 들어서 수정했습니다.

sfmpq는 ShadowFlare님이 만드셨습니다 : <http://sfsrealm.hopto.org/>

#IT·컴퓨터

One

whyask37's blog

main.cpp
end.zip

1



왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 Project SMD 카테고리 글

7. 지형을 다시 한번 출력해보자.

2013. 9. 19.
6

6. 기초 작업

2013. 9. 12.
0

5. SFmpq (ShadowFlare's MPQ Library) 와 예제

2013. 9. 11.
1

4. scenario.chk

2013. 9. 10.
0

whyask37's blog



이 블로그 인기글

MPQ 가지고 놀기 (1) - 간단한 MPQ 파일 분석

2013. 10. 19.

11

[별강의] 13. 트리거 프로그래밍 - TRIG-MRGN 루프

2014. 2. 24.

0

4. scenario.chk

2013. 9. 10.

0

[별강의] 2. 데스 사이의 대입, 더하기

2014. 1. 19.

1

[별강의] 5. 포인터 예제 - 유닛 제한 줄이기

2014. 1. 23.

0



[back to top](#)

blog market

특수목으로 만든 무전력 스피커



One

whyask37's blog
