Filip Mazur 226018 dr inż. Jacek Mazurkiewicz

Marcin Kozieł 225937 Data ćwiczenia: 10.10.2017

Urządzenia peryferyjne

**Obsługa karty dźwiękowej**

# Wstęp

Celem ćwiczenia było opanowanie wiedzy działania z zakresu obsługi kart dźwiękowych oraz poznanie właściwości formatu WAV

# Przebieg ćwiczenia

## PlaySound

Najprostszym sposobem odtworzenia dźwięku przy pomocy WinAPI jest użycie funkcji PlaySound. Jest bardzo prosty, w sumie jednolinijkowy i nie wymaga absolutnie żadnej wiedzy ani o formacie pliku, ani o przetwarzaniu plików binarnych, oto przykład:

PlaySound(TEXT("Input/JazzTrio.wav"), NULL, SND\_FILENAME);

## waveOut

O wiele ciekawszym sposobem jest użycie metod „waveOut\*”, wymagającej znalezienia w pliku WAV pewnych charakterystycznych danych, stanowiących ich nagłówek. Poniżej kluczowy dla tego celu fragment kodu:

std::string stringData(4, '0'); //four bytes to hold 'RIFF'

fread(&stringData[0], sizeof(stringData[0]), 4, file);

std::cout << "ChunkID: \t" << stringData << std::endl;

if (stringData == "RIFF")

{ //we had 'RIFF' let's continue

DWORD dIntegerData;

fread(&dIntegerData, sizeof(dIntegerData), 1, file); //4

std::cout << "ChunkSize : \t" << dIntegerData << std::endl;

fread(&stringData[0], sizeof(stringData[0]), 4, file); //8

std::cout << "Format : \t" << stringData << std::endl;

fread(&stringData[0], sizeof(stringData[0]), 4, file); //12

std::cout << "Subchunk1ID: \t" << stringData << std::endl;

DWORD Subchunk1Size;

fread(&Subchunk1Size, sizeof(Subchunk1Size), 1, file); //16

std::cout << "Subchunk1Size: \t" << Subchunk1Size << std::endl;

fread(&waveFormatex.wFormatTag, sizeof(WORD), 1, file); //20

std::cout << "AudioFormat: \t" << waveFormatex.wFormatTag << std::endl;

fread(&waveFormatex.nChannels, sizeof(WORD), 1, file); //22

std::cout << "NumChannels: \t" << waveFormatex.nChannels << std::endl;

fread(&waveFormatex.nSamplesPerSec, sizeof(DWORD), 1, file); //24

std::cout << "SampleRate: \t" << waveFormatex.nSamplesPerSec << std::endl;

fread(&waveFormatex.nAvgBytesPerSec, sizeof(DWORD), 1, file); //28

std::cout << "ByteRate: \t" << waveFormatex.nAvgBytesPerSec << std::endl;

fread(&waveFormatex.nBlockAlign, sizeof(WORD), 1, file); //32

std::cout << "BlockAlign: \t" << waveFormatex.nBlockAlign << std::endl;

fread(&waveFormatex.wBitsPerSample, sizeof(WORD), 1, file); //34

std::cout << "BitsPerSample: \t" << waveFormatex.wBitsPerSample << std::endl;

fseek(file, 20 + Subchunk1Size, SEEK\_SET); //jump to begining of data sub-chunk

fread(&stringData[0], sizeof(stringData[0]), 4, file);  
 std::cout << "Subchank2ID: \t" << stringData << std::endl;

fread(&(waveHeader.dwBufferLength), sizeof(DWORD), 1, file);

std::cout << "Subchunk2Size: \t" << waveHeader.dwBufferLength << std::endl;

g\_buffer = std::malloc(waveHeader.dwBufferLength);

waveHeader.lpData = (LPSTR)g\_buffer;

fread(waveHeader.lpData, waveHeader.dwBufferLength, 1, file);

waveHeader.dwFlags = 0;

waveHeader.dwLoops = 0;

waveOutOpen(&hwaveout, WAVE\_MAPPER, &waveFormatex, NULL, NULL,

CALLBACK\_NULL;

waveOutPrepareHeader(hwaveout, &waveHeader, sizeof(waveHeader));

waveOutWrite(hwaveout, &waveHeader, sizeof(waveHeader));

}

A poniżej wynik wczytania pliku przykładowego pliku WAV razem z krótkim objaśnienie

ChunkID: RIFF - identyfikator Resource Interchange File Format  
ChunkSize : 1516122 - Rozmiar pliku w bajtach pomniejszony o 8(bez ChunkID i Size)  
Format : WAV - identyfikator Waveform Audio File Format  
Subchunk1ID: fmt - identyfikator bloku „format”  
Subchunk1Size: 18 - rozmiar bloku „format” bez pól ID i Size  
AudioFormat: 1 - typ kodowania. Wartość „1” oznacza Pulse-code modulation(PCM)   
NumChannels: 2 - liczba kanałów. Wartość „2” oznacz dźwięk stereo  
SampleRate: 44100 - częstotliwość próbkowania, w Hertzach.  
ByteRate: 176400 - SampleRate \* BitsPerSample/8 \* NumChannels,  
 ile bajtów pamięci zajmuje sekunda nagrania  
BlockAlign: 4 - wyrównanie bloku danych, zwykle BitsPerSample/8 \* NumChannels  
BitsPerSample: 16 - rozmiar próbki  
Subchank2ID: data - identyfikator bloku danych  
Subchunk2Size: 1516036 - rozmiar bloku danych

## DirectSound

Trzecią metodą jaką próbowaliśmy użyć, jest DirectSound, mający większe możliwości, np. zbuforowanie i odtworzenie części pliku, zamiast zbuforowanie go w całości, niestety zawiła dokumentacja Microsoftu oraz brak przykładów użycia na których można by się wzorować sprawiły że nie udało nam się uzyskać żadnych efektów.

# Konkluzje

Zapis plików multimedialnych to bardzo ciekawy temat, te laboratoria pozwoliły nam zyskać trochę wiedzy na temat formatu WAV oraz przekonać się jak wygląda jego niskopoziomowa obsługa.