7. Homework

12201922 이규민

1) Download swvader03.wav from iclass and read it with "xxd". For Cygwin, you can access C drive with /cygdrive/c. For virtual box, use email to download the wav file. If you cannot access the wav file from your virtual machine, use Cygwin. Interpret all fields in the header. Look at the file with xxd.

$ xxd swvader03.wav > x

$ vi x

......

The answer should show the byte sequence and the value of each field as follows:

ChunkID: 52 49 46 46 : RIFF

ChunkSize: 34 76 00 00 : 30260

Format : 57 41 56 45 : WAVE

Subchunk1ID : 66 6d 74 20 : fmt

Subchunk1Size: 10 00 00 00 : 16

.............

텍스트, 스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

Xxd를 이용하여 wav 파일의 내용을 확인이 가능하다. 내용에는 파일 크기, 형식, 내용 등 모든 데이터가 담겨있다.

2) Write a program that reads swvader03.wav and displays the content as above.

..............

char ChunkID[10]; // use char array for text data

int ChunkSize; // use "int" for 4 byte data

char Format[10];

........

short AudioFormat; // use "short" for 2 byte data

........

x=open("./swvader03.wav", ...........);

y=read(x, ChunkID, 4); // read first 4 bytes into ChunkID[]

ChunkID[y]=0; // to print as a string

y=read(x, &ChunkSize, 4); // read next 4 bytes and store at address &ChunkSize

y=read(x, Format, 4); // read "WAVE"

Format[y]=0;

.......

y=read(x, &AudioFormat, 2); // read next 2 bytes and store at address &AudioFormat

..........

printf("ChunkID:%s\n", ChunkID);

printf("ChunkSize:%d\n",ChunkSize);

printf("Format:%s\n",Format);

.......

printf("AudioFormat:%d\n", AudioFormat);

.......

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Wav 파일의 내용을 읽어 각 변수에 저장하는 코드를 만들었다.

wav 파일의 데이터에서 첫 4글자는 ChunkID를나타낸다. 첫 4글자를 ChunkID 캐릭터 배열에 저장한 후 출력해보면 RIFF가 출력되는 것을 볼 수 있다.

그 다음 4글자는 ChunkSize를 가지고 있는데 xxd 명령어로 보면 3476 0000 인데 이는 0x00007634 = 30260을 뜻한다.

동일한 방식으로 Format과 AudioFormat을 출력했다. 여기서 Format과 AudioFormat 사이에는 다른 데이터들이 있다. 그래서 중간에서 lseek(x, 20, SEEK\_SET) 명령어를 사용하여 리딩 포인트를 옮기는 방법도 있다. 그러나 위 코드에서는 2글자씩 리딩을 4번한 후 다시 AudioFormat을 리딩하는 방식을 썼다.

이렇게 코드를 작성한 후 프로그램을 실행하면 wav파일의 정보를 볼 수 있다.

3) Same as 2), but display the content in file sw2-wav.txt. Using "write()" to write into a text file is very hard. Use fopen() and fprintf() for formatted output.

..........

x=open("./swvader03.wav", ...........); // input file

FILE \*fout=fopen("sw2-wav.txt", "w"); // output file

y=read(x, ChunkID, 4); // read "RIFF"

ChunkID[y]=0; // to print as a string

y=read(x, &ChunkSize, 4); // read chunk size

y=read(x, Format, 4); // read "WAVE"

Format[y]=0;

.......

fprintf(fout,"ChunkID:%s\n", ChunkID); // write to sw2-wav.txt

fprintf(fout, "ChunkSize:%d\n",ChunkSize);

fprintf(fout, "Format:%s\n",Format);

.......

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

2번 문제는 파일의 정보를 화면에 출력하였지만 이번에는 다른 파일에 내용을 입력해보았다.

우선 새로운 파일을 만들 때 우선 포인터를 선언한 후 fopen을 사용하였다. 그리고 원본 파일에서 데이터 정보가 담긴 부분을 읽고 배열과 정수 변수에 저장해줬다. 이 때 배열에 저장하고나서 ChunkID[y]=0과 같이 마지막 글씨를 0으로 수정해줬다. 이는 원본 파일을 읽고 그대로 배열에 저장하기 때문에 입력종료(\0)가 없으므로 따로 만들어준 것이다.

그리고 출력하는 함수를 fprintf를 사용했는데 이는 화면이 아닌 파일에 쓰라는 의미다. Fprintf를 사용하기 위해서는 파일 포인터 선언을 선언해줘야한다. 마지막 결과 화면을 보면 정상적으로 내용 입력이 된 것을 알 수 있다.

4) swvader03.wav contains a sentence, “Yes, my master”. Write a program that modifies the file such that it contains only “master”. Move the file read pointer to the start of the actual sound data with lseek() and write 0 for half of the sound data, since "Yes, my" and "master" take about half of the sound data each. It will be better that you copy swvader03.wav to sw2.wav and modify sw2.wav.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

원본 파일을 sw, 복사된 파일을 cpsw라고 하였다. 우선 sw의 내용을 그대로 cpsw에 복사해줬다.

그리고 cpsw의 SubChunk2의 data 항목의 절반을 0으로 바꾸어 “master” 소리만 들리도록 만들었다. 그러기 위해서 우선 파일의 크기 확인이 필요하므로 sw의 4번째 글자부터 7번째 글자까지 읽어서 정수 변수인 ChunkSize에 넣어주었다.

데이터 크기 확인이 완료되었다면 data 내용의 절반만 0으로 바꾸어주어야한다. Buf를 0으로 저장해줬다. 그리고 cpsw의 리딩 포인트를 44번째로 옮겨주었다. 음성 데이터가 저장 되어있는 곳이기 때문이다. 데이터의 내용을 0으로 바꾸는 과정을 한글자씩 진행하였고, 데이터의 절반 크기인 (ChunkSize – 36)/2만큼 반복해줬다. (40번 째 데이터를 저장했다면 데이터 부분의 크기만을 변수에 저장할 수 있다.)

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

원본 파일과 동일한 크기의 sw2.wav가 생성되었고, xxd를 이용하여 Data 항목의 내용을 보면 모두 0인 것을 알 수 있다. 중간 부분부터는 기존 내용이 저장되어있고, wav 파일을 재생해보면 “Master”소리만 들리는 것을 확인할 수 있다.

5) Write a program that modifies the wav file such that it contains "master" twice. That is, when you play this file you should here "master master".

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Master 음성을 두 번 출력하기 위한 코드이다.

우선 원본 파일 sw를 복사한 cpsw 파일을 만들었다.

그리고 원본 파일 데이터의 중간부분부터 마지막까지의 내용을 cpsw 데이터의 처음부분에 입력해줬다. 그렇게 하여 Master 음성 데이터가 cpsw에 두 번 입력되었다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Sw2.wav 파일을 재생해보면 Master 소리가 두 번 들린다. Xxd로 내용을 확인해보면 Master의 음성 데이터 내용이 입력된 것을 확인할 수 있다.

6) Write a program that modifies the wav file such that it contains “Yes my master” twice.

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Yes my master 음성이 두 번 들리게 만들기 위하여 우선 ChunkSize와 Subchunk2Size를 늘려주었다.(Subchunk2Size만 두 배로 늘려줘도 문제 없으나 ChunkSize도 늘려주었다.)

Lseek를 이용하여 subchunk2size 정보가 저장된 부분을 읽어 변수 Subchunk2Size에 저장했다. 이후 두 배로 그리고 변수에 두 배를 해주었고, 다시 subchun2size 데이터를 수정해줬다.

그리고 lseek로 sw의 음성 데이터가 저장된 부분에 포인트를 잡고 복사된 파일 마지막 부분에 데이터를 다시 복사해줬다.

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

ChunkSize와 Subchunk2Size를 모두 수정해줬기 때문에 4번째 데이터와 40번째 데이터가 수정된 것을 볼 수 있다. Wav 파일을 재생해보면 “yes my master” 소리가 두 번 들린다.

7) Use gdb to debug the error in following code.

#include<fcntl.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

int main(){

char chunkID[10];

int chunkSize;

char format[10];

short AudioFormat;

short NumChannel;

int SampleRate;

int ByteRate;

short BlockAlign;

short BitsPerSample;

char data[20];

int x,y;

x = open("./swvader03.wav", O\_RDONLY, 00777);

x = read(x, chunkID, 4);

chunkID[y] = 0;

y = read(x, &chunkSize, 4);

y = read(x, format, 4);

format[y] = 0;

printf("chunkID : %s ", chunkID);

printf("chunkSize : %d ", chunkSize);

printf("format : %s ", format);

printf("\n");

y = read(x, chunkID, 4);

chunkID[y] = 0;

y = read(x, &chunkSize, 4);

y = read(x, &AudioFormat, 2);

y = read(x, &NumChannel, 2);

y = read(x, &SampleRate, 4);

y = read(x, &ByteRate, 4);

y = read(x, &BlockAlign, 2);

y = read(x, &BitsPerSample, 2);

printf("chunkID : %s ", chunkID);

printf("chunkSize : %d ", chunkSize);

printf("AudioFormat : %d ", AudioFormat);

printf("NumChannel : %d ", NumChannel);

printf("ByteRate : %d ", ByteRate);

printf("BlockAlign : %d ", BlockAlign);

printf("BitsPerSample : %d", BitsPerSample);

printf("\n");

y = read(x, chunkID, 4);

chunkID[y] = 0;

y = read(x, &chunkSize, 4);

printf("chunkID : %s ",chunkID);

printf("chunkSize : %d", chunkSize);

printf("\n");

return 0;

}

$ gcc -g -o ex2 ex2.c ==> compile with -g to use gdb

$ gdb ex2

b main

r

     x=open("swvader03.wav",...);

n                             ==> run "x=open(...)"

     x=read(x, chunkID, 4);     ==> next statement to debug

p x                            ==> print x to see the result of "x=open(...)"

$1=7                          ==> swvader03.wav file is now file no 7

n                              ==> run "x=read(x, chunkID, 4)"

   chunkID[y]=0               ==> next statement to debug

p chunkID       ==> print chunkID to see the result of "x=read(x, chunkID, 4)"

$5="RIFF\000..."              ==> we have RIFF in chunkID

n                             ==> run "chunkID[y]=0"

   y=read(x, ...);              ==> next statement to debug

p chunkID                    ==> check chunkID again after "chunkID[y]=0"

......................

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

디버깅 진행하였고, 음성 파일 오픈 후 x의 값은 3이 나오므로 정상적으로 열렸다는 것을 알 수 있다. 그러나 x=read(~,~,4)라고 하였으니 변수 x에는 리딩한 길이인 4가 저장되고, 변수 y에는 아무것도 저장되어있지 않다. 그 상태에서 chunkID[y]=0이라고 저장해줘도 y가 정확한 위치에 저장되지 않았으므로 chunkID를 확인해보면 RIFF 뒤에 \0이 있지 않으니 올바른 메시지가 출력되지 않을 수 있다. 여기서 21 행을 y = read(x, ~,~)으로 수정해주면 올바른 결과를 얻을 수 있다.

두 번째 사진은 21을 수정한 후 정상적으로 출력되는 화면이다.