lect7 HW

시스템프로그래밍 1분반 12180626 성시열

1) Read swvader03.wav with "xxd". Interpret all fields in the header.

First copy swvader03.wav file from ../../linuxer1 directory into current directory. "." means current directory.

$ cp ../../linuxer1/swvader03.wav .
(or cp ../../linuxer2/swvader03.wav . in 165.246.38.152)

```
-bash-4.2$ cp ../../linuxer1/swvader03.wav  .
-bash-4.2$ ls
cphw4     ex1     ex2     ex5     ex.8    f2  f7   f93?     myxxd          y.c
cphw4.c   ex13    ex2.c   ex5.c   ex8.c   f3  f8   hw4      myxxd.c
d1        ex13.   ex3     ex6     ex9     f4  f9   hw4.c    newhw4.c
d2        ex13.c  ex3.c   ex6.c   ex9.c   f5  f91  mycat    swvader03.wav
d3        ex1.c   ex4.c   ex8     f1      f6  f92  mycat.c  x
```

cp 명령어와 . 명령어를 통해 로그인 디렉토리에 swvader03.wav 파일을 복사하였다.
ls를 통해 로그인 디렉토리에 모든 파일들을 확인하면 swvader03.wav가 생긴 것을 알 수 있다.

Look at the file with xxd.

```
-bash-4.2$ xxd swvader03.wav > x
-bash-4.2$ vi x
```

xxd 명령어를 통해 swvader03.wav에 저장된 값을 16진수로 표현하고 그 결과를 파일 x에 저장한다. 이후 vi 편집기를 통해 x에 저장된 값을 확인한다.

```
0000000: 5249 4646 3476 0000 5741 5645 666d 7420  RIFF4v..WAVEfmt
0000010: 1000 0000 0100 0100 2256 0000 2256 0000  ........"V.."V..
0000020: 0100 0800 6461 7461 1076 0000 8080 8080  ....data.v......
0000030: 8080 8080 8080 8080 8080 7f80 8080 7f80  ................
0000040: 7f7f 7f7f 807f 7f7f 7f7f 8080 8080 8080  ................
0000050: 8080 8080 8080 8080 8080 8080 8080 8080  ................
0000060: 8080 8080 8080 8080 8080 8080 8080 8080  ................
0000070: 8080 8080 8080 7f7f 7f7f 7f7f 7f7f 7f7f  ................
0000080: 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f  ................
0000090: 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f  ................
00000a0: 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f 7f7f  ................
00000b0: 7f7f 7f7f 7f7f 7f7f 7f7f 7f80 8080 8080  ................
00000c0: 8080 8080 8080 8080 8080 8080 8080 8080  ................
```

(출력 결과가 길어 일부만 첨부하였습니다.)

wav 파일은 음성 데이터를 포함하고 있고, 이러한 16진수로 표현된 값을 분석한다.

목록	16진수	설명
ChunkID	52 49 46 46	ChunkID를 나타내는 아스키코드이고 문자열로 보면 RIFF이다.
ChunkSize	34 76 00 00	ChunkSize는 Chunk 이후의 크기를 나타내는데 36 + SubChunk2Size이다. 8byte를 제외하여 계산한다. 16진수를 10진수로 변환 후 계산하면 30260이다.
Format	57 41 56 45	Format(형태)가 WAVE임을 의미한다.
Subchunk1ID	66 6d 74 20	Subchunk1ID가 fmt임을 의미한다.
Subchunk1Size	10 00 00 00	0x00000010을 little endian으로 뒤에서부터 거꾸로 나타내기 때문에 16을 의미한다.
AudioFormat	01 00	0x0001을 little exndian으로 뒤에서 거꾸로 나타낸 것이다.
NumChannels	01 00	채널의 개수를 의미하고 1개(Mono)이다.
SampleRate	22 56 00 00	초당 샘플링 속도를 의미한다.
ByteRate	22 56 00 00	초당 몇 byte를 나타내는지 알려준다.
BlockAlign	01 00	채널의 개수까지 고려한 샘플의 개수이다.
BitsPerSample	08 00	샘플 당 비트 수를 나타낸다. 8bit이므로 샘플 당 1byte이다.
Subchunk2ID	64 61 74 61	문자열이므로 big-endian으로 표현한다. Subchunk2부터 실질적인 음파의 데이터가 저장된다.
Subchunk2Size	10 76 00 00	샘플 수 * 채널 수 * 샘플당 비트수/8 로 계산하며 거꾸로 읽어서 00 00 76 10으로 볼 때 크기는 30224이다. 위에서 구한 30260과 비교했을 때 정확히 36byte 차이가 난다.
Data	80 80 80 80 ...	이후로는 실제 소리데이터를 의미한다.

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

```c
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>

int main(){
        int x, y;
        char ChunkID[10];
        int ChunkSize;
        char Format[10];
        char subChunk1ID[10];
        int subChunk1Size;
        short AudioFormat;
        short numchannels;
        int samplerate;
        int byterate;
        short blockalign;
        short BPS;
        char subChunk2ID[10];
        int subChunk2Size;

        x = open("./swvader03.wav", O_RDONLY, 00777);

        y = read(x, ChunkID, 4);
        ChunkID[y] = 0;

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

        y = read(x,Format, 4);
        Format[y] = 0;

        y = read(x, subChunk1ID, 4);
        subChunk1ID[y] = 0;

        y = read(x, &subChunk1Size, 4);
        y = read(x, &AudioFormat, 2);
        y = read(x, &numchannels, 2);
        y = read(x, &samplerate, 4);
        y = read(x, &byterate, 4);
        y = read(x, &blockalign, 2);
        y = read(x, &BPS, 2);

        y = read(x, subChunk2ID, 4);
        subChunk2ID[y] = 0;

        y = read(x, &subChunk2Size, 4);
```

```c
        printf("ChunkID : %s\n", ChunkID);
        printf("ChunkSize : %d\n", ChunkSize);
        printf("Format : %s\n", Format);
        printf("subChunk1ID : %s\n", subChunk1ID);
        printf("subChunk1Size : %d\n", subChunk1Size);
        printf("AudioFormat : %d\n", AudioFormat);
        printf("numchannels : %d\n", numchannels);
        printf("samplerate : %d\n", samplerate);
        printf("byterate : %d\n", byterate);
        printf("blockalign : %d\n", blockalign);
        printf("BPS : %d\n", BPS);
        printf("subChunk2ID : %s\n", subChunk2ID);
        printf("subChunk2Size : %d\n", subChunk2Size);

        return 0;
}
```

코드를 보면 위에서

$ xxd swvader03.wav > x

$ vi x

를 통해 파악했던 wave 파일의 정보들을 출력하는 형태이다.

swvcader03.wav 파일을 읽기 모드로 open하고 파일 번호를 x에 저장한다.

이후 파일을 4byte까지만 ChunkID에 저장하고, 읽은 bytes 수만큼 4를 y에 저장한다.

ChunkID[] 배열은 문자열이므로 y번째 인덱스에 0을 넣어 문자열을 완성시킨다.

이후 문자열은 0으로 마무리 짓는 과정을, 정수는 read함수를 통해 각 변수의 주소값에 값을 넣어 각 부분에 해당하는 정보들을 뽑아낸다.

```
-bash-4.2$ vi ex2.c
-bash-4.2$ g++ -o ex2 ex2.c
-bash-4.2$ ./ex2
ChunkID : RIFF
ChunkSize : 30260
Format : WAVE
subChunk1ID : fmt
subChunk1Size : 16
AudioFormat : 1
numchannels : 1
samplerate : 22050
byterate : 22050
blockalign : 1
BPS : 8
subChunk2ID : data
subChunk2Size : 30224
```

컴파일 후 실행하여 결과를 확인하였다.

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.

```c
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>

int main(){
        int x, y;
        char ChunkID[10];
        int ChunkSize;
        char Format[10];
        char subChunk1ID[10];
        int subChunk1Size;
        short AudioFormat;
        short numchannels;
        int samplerate;
        int byterate;
        short blockalign;
        short BPS;
        char subChunk2ID[10];
        int subChunk2Size;
```

```c
        x = open("./swvader03.wav", O_RDONLY, 00777);
        FILE *fout = open("sw2-wav.txt", "w");

        y = read(x, ChunkID, 4);
        ChunkID[y] = 0;

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

        y = read(x,Format, 4);
        Format[y] = 0;

        y = read(x, subChunk1ID, 4);
        subChunk1ID[y] = 0;

        y = read(x, &subChunk1Size, 4);
        y = read(x, &AudioFormat, 2);
        y = read(x, &numchannels, 2);
        y = read(x, &samplerate, 4);
        y = read(x, &byterate, 4);
        y = read(x, &blockalign, 2);
        y = read(x, &BPS, 2);

        y = read(x, subChunk2ID, 4);
        subChunk2ID[y] = 0;

        y = read(x, &subChunk2Size, 4);
```

```
        fprintf(fout, "ChunkID : %s\n", ChunkID);
        fprintf(fout, "ChunkSize : %d\n", ChunkSize);
        fprintf(fout, "Format : %s\n", Format);
        fprintf(fout, "subChunk1ID : %s\n", subChunk1ID);
        fprintf(fout, "subChunk1Size : %d\n", subChunk1Size);
        fprintf(fout, "AudioFormat : %d\n", AudioFormat);
        fprintf(fout, "numchannels : %d\n", numchannels);
        fprintf(fout, "samplerate : %d\n", samplerate);
        fprintf(fout, "byterate : %d\n", byterate);
```

2번과 동일하지만 화면에 출력하는 것이 아닌 파일에 저장하는 코드이다. 이를 위해 FILE *fout = fopen("sw2-wav.txt", "w")를 사용하여 txt 파일을 만들어주고 fprintf를 사용하여 화면이 아닌 파일에 정보를 저장한다.
컴파일 과정에서 warning이 발생하였지만 진행하였다.
./ex2를 통해 실행하였으나 Segmentation fault (core dumped)가 발생하였다. 이는 포인터 관련 오류인데 메모리 보호 기법에 의해 발생한 오류로 허가되지 않은 메모리의 접근을 방지하기 위해서이다.

```
-bash-4.2$ cat sw2-wav.txt
cat: sw2-wav.txt: No such file or directory
```

cat을 통해 파일의 내용을 확인해보려 했으나 파일이 없는 것으로 보아 파일이 저장되지 않았음을 확인하였다.

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. When you modified the file, you need to download it to your PC using psftp (look at Section 7 for the explanation for psftp).

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <string.h>

int main(){
        int x;
        int y = 0;
        x = open("sw2.wav", O_RDWR, 00777);
        lseek(x, 44, SEEK_SET);

        int i = 0;
        for (i; i<15108;  i++){
                write(x, &y, 1);
        }
        return 0;
}
~
~
```

```
-bash-4.2$ vi ex44.c
-bash-4.2$ gcc -o ex44 ex44.c
-bash-4.2$ ./ex44
```

psftp를 다운받고 수정한 음성파일(sw2.wav)을 다운받았다.
수정을 하기전의 음성파일(swvader3.wav)은 "Yes my master"를 출력하고

수정을 한 후의 음성파일(sw2.wav)은 "master"를 출력한다.

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".

```c
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <string.h>

int main(){
        int x;
        int y = 0;

        int z;
        int copyMaster;

        x = open("sw2.wav", O_RDWR, 00777);
        copyMaster = open("sw3.wav", O_RDWR, 00777);

        lseek(x, 44, SEEK_SET);
        lseek(copyMaster, 44, SEEK_SET);

        int i = 0;
        for (i; i<15108;  i++){
                z = read(x, &y, 1);
                write(copyMaster,  &y, 1);
        }
        return 0;
}
~
~
```

```
-bash-4.2$ vi ex44.c
-bash-4.2$ gcc -o ex44 ex44.c
-bash-4.2$ ./ex44
-bash-4.2$
```

먼저 cp를 사용하여 swvader03.wav파일을 sw3.wav라는 새로운 파일에 복사하여 주었다.
c프로그램은 4번과 거의 동일하나 새로운 사운드 데이터를 저장할 copyMaster변수를 만들어주었고 sw3.wav파일을 오픈해주었다.
lseek를 통해 4번에서 사용한 사운드 파일인 sw2.wav(x)의 r/w pointer가 "Yes my"를 담고있는 데이터 이후인 15152번째 인덱스를 가르키게 하였다.
sw3.wav(copyMaster)는 lseek를 이용해 r/w pointer의 위치를 actual sound data로 옮겨주었다.
for 반목문을 만들고 반복을 하는 횟수는 sw3.wav 의"Yes my"부분을 "master"로 변경하는 byte의 크기인 15108로 설정해주었다.
반복문안에서 sw2.wav파일의 r/w pointer가 가르키는 인덱스 이후의 데이터 ("master")를 읽고 이 데이터를 write를 사용하여 copyMaster 즉,sw3.wav에 저장하여 주었다.
psftp를 다운받고 수정한 음성파일(sw3.wav)을 다운받았다.
수정을 한 후의 음성파일(sw3.wav)은 "master master"를 출력한다.

6) 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"
.....
```

```c
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <string.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;
}
```

```
-bash-4.2$ vi ex6.c
-bash-4.2$ gcc -o ex6 ex6.c
-bash-4.2$ ./ex6
chunkID :  chunkSize : 0 format :
chunkID :  chunkSize : 0 AudioFormat : 0 NumChannel : 0 ByteRate : 0 BlockAlign :
 BitsPerSample : 1
chunkID :  chunkSize : 0
-bash-4.2$
```

주어진 코드를 작성하고 컴파일 하였더니 문제없이 실행이 되었으나 결과창을 보면 각 변수에 헤더 정보들이 정확히 저장되지 않은 것을 확인할 수 있었다.
이를 해결하기 위해 gdb로 디버깅을 해보았다.

```
-bash-4.2$ gdb ex6
GNU gdb (GDB) Fedora (7.5.1-38.fc18)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/sp32/12181784/ex6...done.
(gdb) b main
Breakpoint 1 at 0x400614: file ex6.c, line 21.
(gdb) r
Starting program: /home/sp32/12181784/ex6

Breakpoint 1, main () at ex6.c:21
21              x=open("./swvader03.wav", O_RDONLY,00777);
Missing separate debuginfos, use: debuginfo-install glibc-2.16-31.fc18.x86_64
(gdb) n
22              x=read(x,chunkID,4);
(gdb) p chunkID
$1 = "\320\b@\000\000\000\000\000 \005"
(gdb) n
23              chunkID[y]=0;
(gdb) p chunkID
$2 = "RIFF\000\000\000\000 \005"
(gdb) n
24              y=read(x,&chunkSize,4);
(gdb) p chunkID
$3 = "\000IFF\000\000\000\000 \005"
(gdb) p chunkSize
$4 = 0
```

`디버깅을 하고 코드를 한 줄씩 실행시키며 변수에 올바른 값이 저장되나 확인하였다.
p를 사용하여 chunkID 찍어보니 계속 값이 바뀌는 것을 확인할 수 있었다.
심지어 n을 입력했을 때 다음 코드가 실행되지 않고 무한대기 현상까지 발생하였다.
gdb디버깅을 통해 chunkID에 누적해서 변수가 저장되는 것을 확인하였으며 이를 통해 문제를 해결을 위한 정보를 얻을 수 있었다.

```
x=open("./swvader03.wav", O_RDONLY,00777);
y=read(x,chunkID,4);
```

x를 y로 수정하였다.

```
-bash-4.2$ vi ex6.c
-bash-4.2$ gcc -o ex6 ex6.c
-bash-4.2$ ./ex6
chunkID : RIFF chunkSize : 30260 format : WAVE
chunkID : fmt  chunkSize : 16 AudioFormat : 1 NumChannel : 1 ByteRate : 22050 Bloc
kAlign : 1 BitsPerSample : 8
chunkID : data chunkSize : 30224
```

수정한 코드를 다시 컴파일하여 돌려보니 올바른 값을 출력해주었다.