

7. Homework

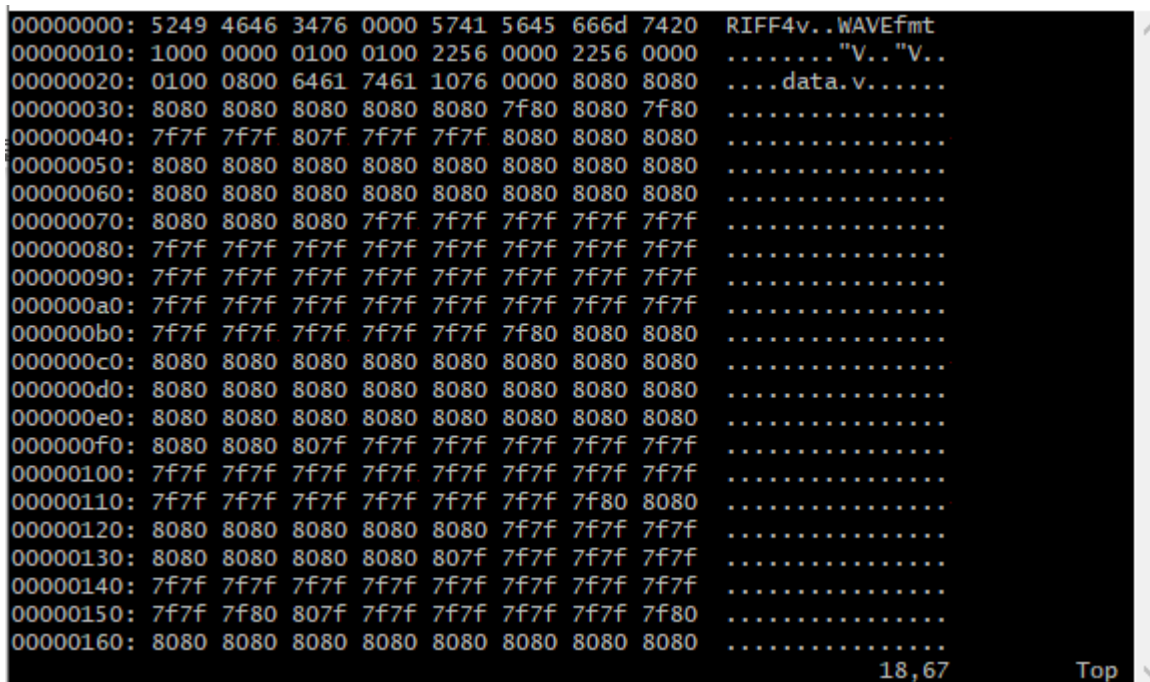
12201922 이규민

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 64 74 20 : fmt
Subchunk1Size: 10 00 00 00 : 16
.....
```



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);
.....
```

```
int main(){
    int x,y;
    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", O_RDONLY, 00777);

    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;

    for(int i=0;i<4;i++) y=read(x, &AudioFormat, 2);
    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);

    return 0;
}
```

```
kyumin@DESKTOP-NUDFAPK ~
$ g++ -g -o ex1 ex1.c
kyumin@DESKTOP-NUDFAPK ~
$ ./ex1
ChunkID:RIFF
ChunkSize:30260
Format:WAVE
AudioFormat:1
```

Wav 파일의 내용을 읽어 각 변수에 저장하는 코드를 만들었다.

wav 파일의 데이터에서 첫 4글자는 ChunkID를 나타낸다. 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);
.....
```

```
int main(){
    int x,y;
    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", O_RDONLY, 00777);
    FILE *fout=fopen("sw2-wav.txt", "w"); // output file

    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;

    for(int i=0;i<4;i++) y=read(x, &AudioFormat, 2);
    y=read(x, &AudioFormat, 2); // read next 2 bytes and store at address &AudioFormat

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

    return 0;
}
"ex1.c" 33L, 1040B 30,51-55 Bot
```

```
kyumin@DESKTOP-NUDFAPK ~
$ g++ -g -o ex1 ex1.c

kyumin@DESKTOP-NUDFAPK ~
$ ./ex1

kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup  ex1.c  ex2.exe  f11
d1        ex1.exe exdir    f12
d2        ex2.c  f1       f13

kyumin@DESKTOP-NUDFAPK ~
$ cat sw2-wav.txt
ChunkID:RIFF
ChunkSize:30260
Format:WAVE
AudioFormat:1
```

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(W0)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.

```
int main(){
    int sw, cpsw, y, ChunkSize;
    char buf[20];
    sw=open("swvader03.wav", O_RDONLY, 00777);
    cpsw=open("sw2.wav", O_RDWR | O_CREAT | O_TRUNC, 00777); // output file

    //원본 파일을 복사 하는 과정
    for(;;){
        y=read(sw,buf,20);
        if(y==0) break;
        write(cpsw,buf,y);
    }

    //묵을 처리
    buf[0]=0;

    //파일 크기 구하기
    lseek(sw,4,SEEK_SET);
    y=read(sw,&ChunkSize,4);

    //중간 부분까지 묵을 처리하기
    lseek(cpsw,44,SEEK_SET);
    for(int i=0;i<(ChunkSize-36)/2;i++) write(cpsw, buf,1);

    return 0;
}
```

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 $(\text{ChunkSize} - 36)/2$ 만큼 반복해줬다. (40번째 데이터를 저장했다면 데이터 부분의 크기만을 변수에 저장할 수 있다.)

```

kyumin@DESKTOP-NUDFAPK ~
$ g++ -o ex1 ex1.c

kyumin@DESKTOP-NUDFAPK ~
$ ./ex1

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 549
drwx-----+ 1 kyumin ㄹ  0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin ㄹ  0 Apr 1 09:03 d1
drwxr-xr-x+ 1 kyumin ㄹ  0 Mar 6 12:47 d2
-rw-r--r--+ 1 kyumin ㄹ 661 Apr 2 18:37 ex1.c
-rwxr-xr-x+ 1 kyumin ㄹ 66901 Apr 2 18:37 ex1.exe
-rw-r--r--+ 1 kyumin ㄹ 976 Mar 31 00:42 ex2.c
-rwxr-xr-x+ 1 kyumin ㄹ 71461 Mar 31 00:42 ex2.exe
drwxr-xr-x+ 1 kyumin ㄹ  0 Mar 14 13:02 exdir
-rw-r--r--+ 1 kyumin ㄹ  6 Apr 2 16:32 f1
-rwxr-xr-x+ 1 kyumin ㄹ  0 Mar 31 01:43 f2
-rwxr-xr-x+ 1 kyumin ㄹ 20 Mar 30 00:37 f8
-rw-r--r--+ 1 kyumin ㄹ 430 Mar 25 10:45 hw4.c
-rwxr-xr-x+ 1 kyumin ㄹ 66926 Mar 30 17:02 hw4.exe
-rw-r--r--+ 1 kyumin ㄹ 487 Mar 29 20:05 mycat.c
-rwxr-xr-x+ 1 kyumin ㄹ 67093 Mar 29 20:05 mycat.exe
-rw-r--r--+ 1 kyumin ㄹ 537 Mar 29 20:12 myxxd.c
-rwxr-xr-x+ 1 kyumin ㄹ 67093 Mar 29 20:07 myxxd.exe
-rwxr-xr-x+ 1 kyumin ㄹ 430 Mar 28 16:12 newhw4
-rwxr-xr-x+ 1 kyumin ㄹ 430 Mar 28 16:13 newhw4.c
-rwxr-xr-x+ 1 kyumin ㄹ 30268 Apr 2 18:37 sw2.wav
-rwx-----+ 1 kyumin ㄹ 30268 Mar 31 18:54 swvader03.wav
-rw-r--r--+ 1 kyumin ㄹ  0 Apr 2 16:16 test
-rw-r--r--+ 1 kyumin ㄹ 128652 Apr 1 09:05 x
-rw-r--r--+ 1 kyumin ㄹ  54 Mar 14 12:57 y.c

```

```

kyumin@DESKTOP-NUDFAPK ~
$ xxd sw2.wav
00000000: 5249 4646 3476 0000 5741 5645 666d 7420  RIFF4v..WAVEfmt
00000010: 1000 0000 0100 0100 2256 0000 2256 0000  ..... "V.. "V..
00000020: 0100 0800 6461 7461 1076 0000 0000 0000  ....data.v.....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000  .....

```

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".

```
int main(){
    int sw, cpsw, y, ChunkSize;
    char buf[20];
    sw=open("swvader03.wav", O_RDONLY, 00777);
    cpsw=open("sw2.wav", O_RDWR | O_CREAT | O_TRUNC, 00777); // output file

    //원본 파일을 복사하는 과정
    for(;;){
        y=read(sw,buf,20);
        if(y==0) break;
        write(cpsw,buf,y);
    }

    //파일 크기 구하기
    lseek(sw,4,SEEK_SET);
    y=read(sw,&ChunkSize,4);

    //reading / write 포인트 구해서 cpsw에 MASTER 음성 입력하기
    lseek(sw, ChunkSize/2+26, SEEK_SET);
    lseek(cpsw,44,SEEK_SET);

    for(int i=0;;){
        y=read(sw,buf, 20);
        i+=y;
        if(i>=(ChunkSize-36)/2){
            write(cpsw,buf,(ChunkSize-36)/2 % 20);
            break;
        }
        write(cpsw, buf, 20);
    }

    return 0;
}
```

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.

```

kyumin@DESKTOP-NUDFAPK ~
$ g++ -o ex1 ex1.c

kyumin@DESKTOP-NUDFAPK ~
$ ./ex1

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 552
drwx-----+ 1 kyumin ㄹ 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin ㄹ 0 Apr 1 09:03 d1
drwxr-xr-x+ 1 kyumin ㄹ 0 Mar 6 12:47 d2
-rw-r--r--+ 1 kyumin ㄹ 803 Apr 2 19:18 ex1.c
-rwxr-xr-x+ 1 kyumin ㄹ 66901 Apr 2 19:18 ex1.exe
-rw-r--r--+ 1 kyumin ㄹ 976 Mar 31 00:42 ex2.c
-rwxr-xr-x+ 1 kyumin ㄹ 71461 Mar 31 00:42 ex2.exe
drwxr-xr-x+ 1 kyumin ㄹ 0 Mar 14 13:02 exdir
-rw-r--r--+ 1 kyumin ㄹ 6 Apr 2 16:32 f1
-rwxr-xr-x+ 1 kyumin ㄹ 0 Mar 31 01:43 f2
-rwxr-xr-x+ 1 kyumin ㄹ 20 Mar 30 00:37 f8
-rw-r--r--+ 1 kyumin ㄹ 430 Mar 25 10:45 hw4.c
-rwxr-xr-x+ 1 kyumin ㄹ 66926 Mar 30 17:02 hw4.exe
-rw-r--r--+ 1 kyumin ㄹ 487 Mar 29 20:05 mycat.c
-rwxr-xr-x+ 1 kyumin ㄹ 67093 Mar 29 20:05 mycat.exe
-rw-r--r--+ 1 kyumin ㄹ 537 Mar 29 20:12 myxxd.c
-rwxr-xr-x+ 1 kyumin ㄹ 67093 Mar 29 20:07 myxxd.exe
-rwxr-xr-x+ 1 kyumin ㄹ 430 Mar 28 16:12 newhw4
-rwxr-xr-x+ 1 kyumin ㄹ 430 Mar 28 16:13 newhw4.c
-rwxr-xr-x+ 1 kyumin ㄹ 30268 Apr 2 19:18 sw2.wav
-rwx-----+ 1 kyumin ㄹ 30268 Mar 31 18:54 swvader03.wav
-rw-r--r--+ 1 kyumin ㄹ 0 Apr 2 16:16 test
-rw-r--r--+ 1 kyumin ㄹ 128652 Apr 1 09:05 x
-rw-r--r--+ 1 kyumin ㄹ 54 Mar 14 12:57 y.c

```

```

kyumin@DESKTOP-NUDFAPK ~
$ xxd sw2.wav
00000000: 5249 4646 3476 0000 5741 5645 666d 7420 RIFF4v..WAVEfmt
00000010: 1000 0000 0100 0100 2256 0000 2256 0000 ..... "V..
00000020: 0100 0800 6461 7461 1076 0000 7477 7572 ....data.v..twur
00000030: 6f6d 6e72 767b 7e7d 7975 7271 757b 838b omnrv{~}yurqu{..
00000040: 9295 9593 9190 9297 9da3 a6a4 9f99 9595 .....
00000050: 989f a5a9 aaa7 a29e 9a98 989a 9b9a 9690 .....
00000060: 8a84 8182 858a 8d8d 8882 7a74 706f 7071 .....ztpopq
00000070: 716e 6a65 615f 6164 676a 6b69 6663 6060 qnjea_adgjkifc`
00000080: 6163 676b 6e6f 6f6e 6e6e 7175 7a7d 7e7e acgknoonnquz}~
00000090: 7c7a 7876 7474 7576 7779 7a7a 7b7e 8184 |zxvtuvwyzz{~..
000000a0: 888b 8c8c 8c8c 8d8d 8d8d 8b89 8787 898b .....
000000b0: 8d8f 8f8e 8c8b 8c8e 9295 9695 918c 8886 .....
000000c0: 878a 8d8f 8e8b 8886 8586 888b 8d8d 8b86 .....
000000d0: 827d 7978 797a 7b7a 7978 7675 7678 7b7f .}xyz{zyxvuvx{.
000000e0: 8284 8480 7a74 706f 6f71 7373 7271 7276 ....ztpooqssrgrv
000000f0: 7c83 8788 857f 7871 6c69 6a6e 747a 7d7c |.....xqlijntz}|
00000100: 7976 7476 7b81 8586 8480 7c7a 7a7c 8086 yvtv{.....|zz|..
00000110: 8c8f 8e8b 8682 8283 888c 8e8e 8b86 807d .....}
00000120: 7b7c 7f82 8585 8482 7f7c 7a7a 7b7d 7c79 {|.....|zz{|y
00000130: 746e 6b6a 6c72 777b 7c7c 7976 7371 7173 tnkjlrw{|lyvsqq
00000140: 7678 7877 7676 797c 8286 8c90 9495 9798 vxxwvvy|.....
00000150: 9b9d a0a3 a4a4 a29e 9b98 9696 9797 9797 .....
00000160: 9593 918e 8c8a 8786 8482 7f7d 7a77 7472 .....}zwtr
00000170: 7070 6e6c 6a67 6564 6362 6261 605f 5f5e ppnljgedcbba`__^
00000180: 5e5d 5b59 5757 5a61 6a72 7575 726e 6b6a ^][YwwZajruurnkj
00000190: 6c6f 7378 7c7f 8182 8283 8688 8c90 9294 losx|.....
000001a0: 9492 9090 9294 9696 9490 8c89 898b 8f93 .....
000001b0: 9596 9490 8d8b 8a8a 8c8f 908e 8b87 827e .....~

```

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.

```
int main(){
    int sw, cpsw, y, ChunkSize, SubChunk2;
    char buf[20];
    sw=open("swvader03.wav", O_RDONLY, 00777);
    cpsw=open("sw2.wav", O_RDWR | O_CREAT | O_TRUNC, 00777); // output file

    //원본 파일을 복사하는 과정
    for(;;){
        y=read(sw,buf,20);
        if(y==0) break;
        write(cpsw,buf,y);
    }

    //ChunkSize와 Subchunk2Size 구하기
    lseek(sw,4,SEEK_SET);
    y=read(sw,&ChunkSize,4);
    lseek(sw, 40, SEEK_SET);
    y=read(sw, &SubChunk2,4);

    //ChunkSize와 Subchunk2Size 수정 후 입력하기
    lseek(cpsw, 4, SEEK_SET);
    ChunkSize = 2 * ChunkSize - 36;
    write(cpsw, &ChunkSize, 4);

    ChunkSize = (ChunkSize +36)/2;
    lseek(cpsw, 40, SEEK_SET);
    SubChunk2 *= 2;
    write(cpsw, &SubChunk2, 4);

    //음성 데이터를 한 번 더 쓰기
    lseek(sw, 44, SEEK_SET);
    lseek(cpsw, ChunkSize + 8, SEEK_SET);

    for(;;){
        y=read(sw,buf, 20);
        if(y==0) break;
        write(cpsw, buf, 20);
    }

    return 0;
}
```

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 subchunk2size 데이터를 수정해줬다.

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

```
kyumin@DESKTOP-NUDFAPK ~
$ g++ -g -o ex1 ex1.c

kyumin@DESKTOP-NUDFAPK ~
$ ./ex1

kyumin@DESKTOP-NUDFAPK ~
$ xxd sw2.wav | head -5
00000000: 5249 4646 44ec 0000 5741 5645 666d 7420  RIFFD...WAVEfmt
00000010: 1000 0000 0100 0100 2256 0000 2256 0000  ..... "V.."V..
00000020: 0100 0800 6461 7461 20ec 0000 8080 8080  ....data .....
00000030: 8080 8080 8080 8080 8080 7f80 8080 7f80  .....
00000040: 7f7f 7f7f 807f 7f7f 7f7f 8080 8080 8080  .....

kyumin@DESKTOP-NUDFAPK ~
$ xxd swvader03.wav | head -5
00000000: 5249 4646 3476 0000 5741 5645 666d 7420  RIFF4v..WAVEfmt
00000010: 1000 0000 0100 0100 2256 0000 2256 0000  ..... "V.."V..
00000020: 0100 0800 6461 7461 1076 0000 8080 8080  ....data.v.....
00000030: 8080 8080 8080 8080 8080 7f80 8080 7f80  .....
00000040: 7f7f 7f7f 807f 7f7f 7f7f 8080 8080 8080  .....

kyumin@DESKTOP-NUDFAPK ~
$ |
```

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="RIFFW000..."             ==> 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"
.....

```

```

kyumin@DESKTOP-NUDFAPK ~
$ gdb ex1
GNU gdb (GDB) (Cygwin 13.2-1) 13.2
Copyright (C) 2023 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-pc-cygwin".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ex1...
(gdb) b main
Breakpoint 1 at 0x10040108d: file ex1.c, line 20.
(gdb) r
Starting program: /cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/ex1
[New Thread 25060.0x7c80]
[New Thread 25060.0x1720]
[New Thread 25060.0x2cc8]

Thread 1 "ex1" hit Breakpoint 1, main () at ex1.c:20
20         x = open("./swvader03.wav", O_RDONLY, 00777);
(gdb) n
21         x = read(x, chunkID, 4);
(gdb) p x
$1 = 3
(gdb) n
22         chunkID[y] = 0;
(gdb) p chunkID
$2 = "RIFF\377\377\a\000\000"
(gdb) n
[New Thread 25060.0x2ebc]
23         y = read(x, &chunkSize, 4);
(gdb) p chunkID
$3 = "RIFF\377\377\a\000\000"
(gdb)

kyumin@DESKTOP-NUDFAPK ~
$ g++ -g -o ex1 ex1.c

kyumin@DESKTOP-NUDFAPK ~
$ ./ex1
chunkID : RIFF chunkSize : 30260 format : WAVE
chunkID : fmt chunkSize : 16 AudioFormat : 1 NumChannel : 1 ByteRate : 22050 B
ockAlign : 1 BitsPerSample : 8
chunkID : data chunkSize : 30224

```

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.