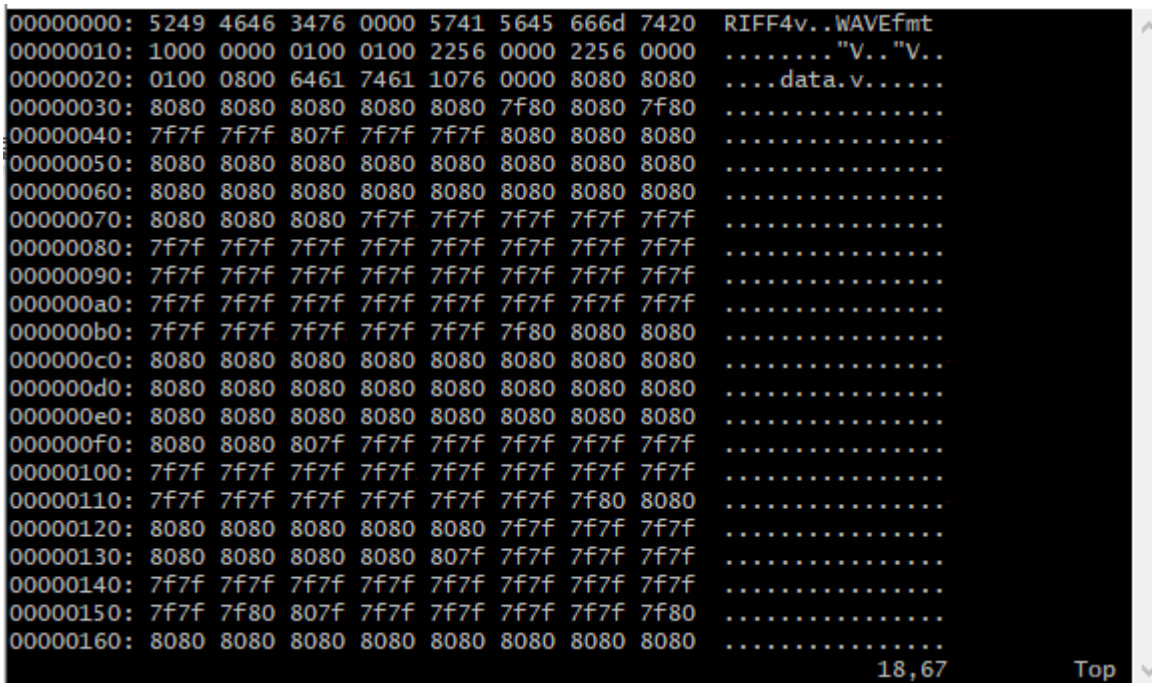
7. Homework

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
....



Xxd를 이용하여 wav 파일의 내용을 확인해보면 파일 크기, 형식, 내용 등 모든 데이터를 확인할 수 있다.

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);
.......
```

```c
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 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.

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  f14  f8       mycat.c    myxxd.exe  sw2-wav.txt    y.c
d1        ex1.exe  exdir    f12  f15  hw4.c    mycat.exe  newhw4     swvader03.wav
d2        ex2.c    f1       f13  f2   hw4.exe  myxxd.c    newhw4.c   x

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

2번 문제와 동일하게 진행하였고, 마지막 출력 부분만 수정을 해줬다. Fprintf를 사용했는데 이는 화면이 아닌 파일에 쓰라는 의미다. Fprintf를 사용하기 위해서는 파일 포인터 선언을 선언해줘야

한다. 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.

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".

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

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"
.....
```