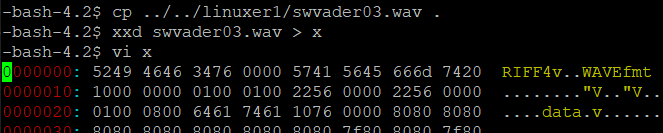
**System Programming Lect6 HW**

**001분반 / 12161756 / 윤성호**

8. Homework

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



**: 현재 directory로 swvader03.wav 파일을 복사한 후 xxd로 swvader03.wav를 연 결과를 파일 x에 저장하였다. WAV파일은 RIFF header와 Subchunk1, Subchunk2 총 세 부분으로 나뉘어진다. 세 부분을 캡쳐 이미지에서 순서대로 빨강, 노랑, 초록색 테두리로 표현하였다.**

**< 1. RIFF header >**

**[var] [hexadecimal] [decimal & ascii] [note]**

**ChunkID 52 49 46 46 RIFF**

**ChunkSize 34 76 00 00 30260 -bytes**

**Format 57 41 56 45 WAVE**

**< 2. Subchunk1 >**

**[var] [hexadecimal] [decimal & ascii] [note]**

**Subchunk1ID 66 6d 74 20 fmt**

**Subchunk1Size 10 00 00 00 16 -bytes**

**AudioFormat 01 00 1 PCM**

**NumChannels 01 00 1 Mono**

**SampleRate 22 56 00 00 22050**

**ByteRate 22 56 00 00 22050**

**BlockAlign 01 00 1**

**BitsPerSample 08 00 8 -bits**

**< 3. Subchunk2 >**

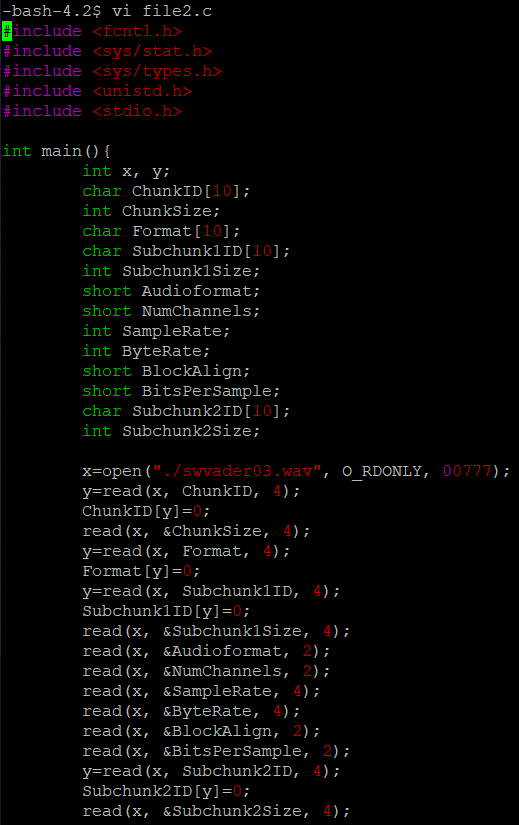
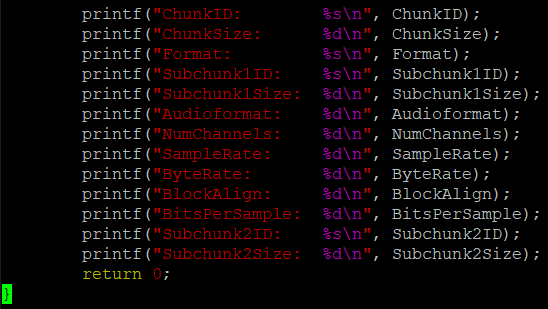
**[var] [hexadecimal] [decimal & ascii] [note]**

**Subchunk2ID 64 61 74 61 data**

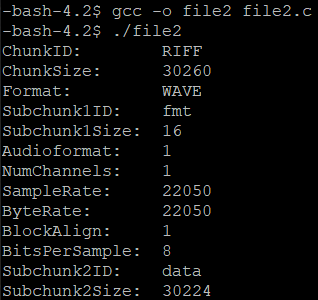
**Subchunk2Size 10 76 00 00 30224 -bytes**

**Data 80 80 80 80 ……**

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

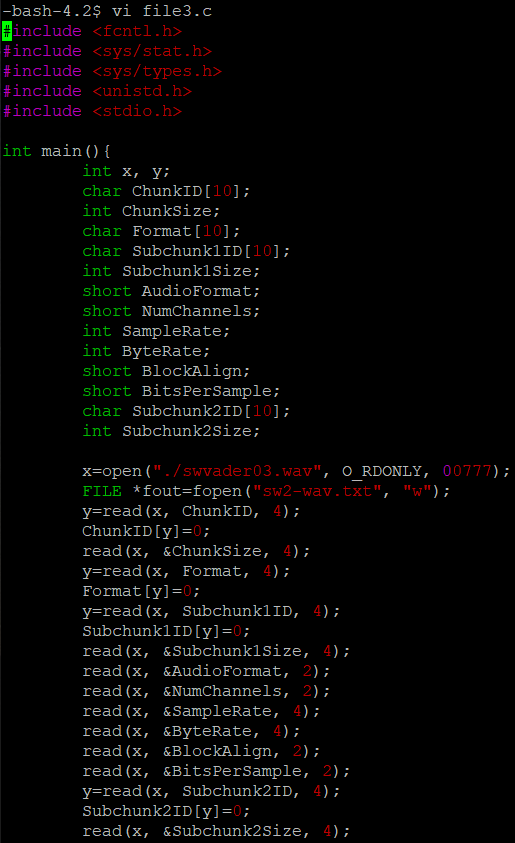
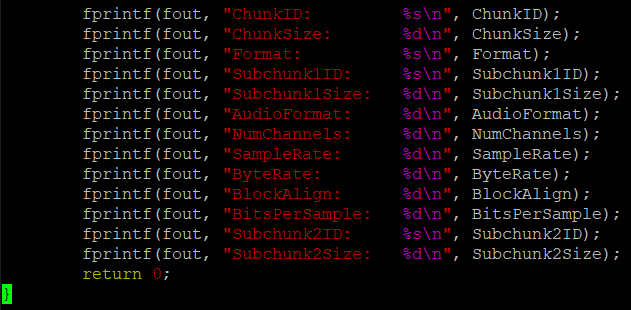
 

**: 숫자 데이터의 경우 4bytes는 int, 2bytes는 short를 이용하고 문자열은 char로 선언한다. 사운드 파일을 읽기 전용으로 open한 다음, 변수의 길이만큼 read하여 값을 각 변수에 저장한다. 문자열 변수의 경우 read의 결과(변수에 저장된 값의 길이)를 y에 저장하는 과정이 추가된다. y를 이용해 마지막 문자 뒤에 0을 입력하여 string으로 변환한다. 결과는 printf를 이용하여 출력한다.**

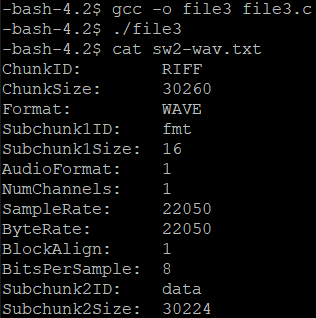


**: 변수 값이 1)에서 해석한 값과 일치하는 것을 확인할 수 있다.**

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.

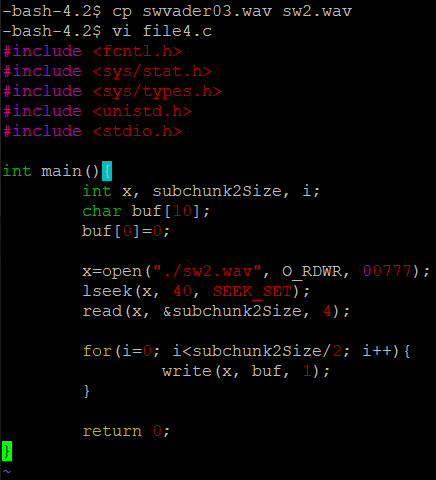
 

**: 2)의 코드와 변수 선언부와 read함수 부분은 동일하다. 다만, [FILE \*]로 파일 포인터를 선언한 후 쓰기모드로 fopen한 output파일을 가리키게하는 코드가 추가되었다. 그리고 기존에 printf로 결과를 화면에 출력하던 것을 fprintf를 이용해 output파일에 기록하도록 변경하였다.**



**: 코드를 실행한 후 output파일을 cat로 열자 2)와 동일한 결과가 출력되었다.**

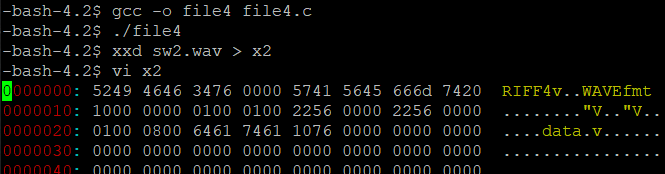
4) swvader03.wav contains a sentence, “Yes, my master”. Write a program that modifies the file such that it contains only “master”. Move the file read pointer to the start of the actual sound data with lseek() and write 0 for half of the sound data, since "Yes, my" and "master" take about half of the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

**(1) cp를 이용해 swvader03.wav를 복사하여 sw2.wav를 생성한다.**

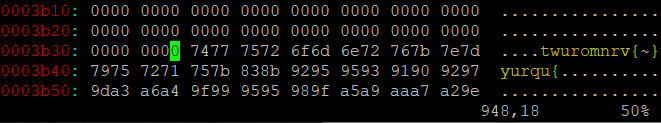
**(2) sw2.wav파일을 read & write 모드로 연다. 40bytes~43bytes 부분은 sound data size 정보를 담는 Subchunk2Size가 위치한다.**

**(3) 이 점을 이용하여 lseek로 40bytes 부분으로 read/write 포인터를 이동한 후 여기서부터 4bytes까지의 데이터를 subchunk2Size 변수에 담는다.**

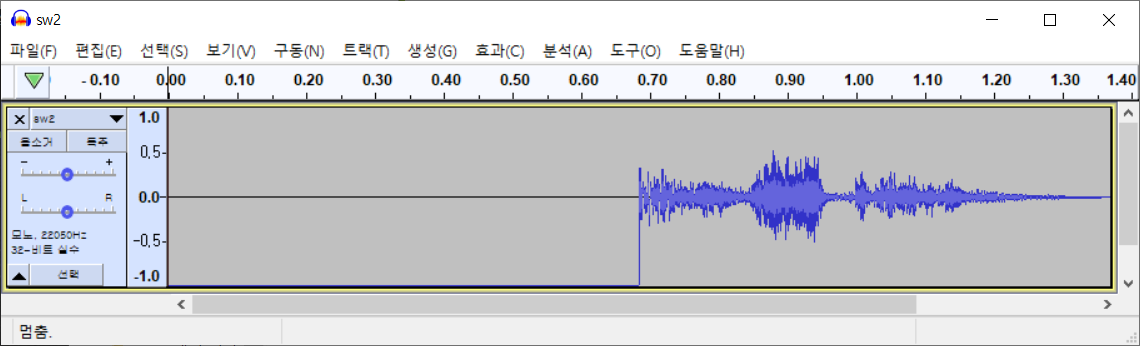
**(4) 위의 [read(x, &subchunk2Size, 4)] 이후 read/write 포인터의 위치는 44bytes에 위치한다. 여기서부터 sound data가 시작되며 for문과 0이 들어있는 buf를 이용해 data길이의 절반까지 0으로 write한다.**



**(중략)**

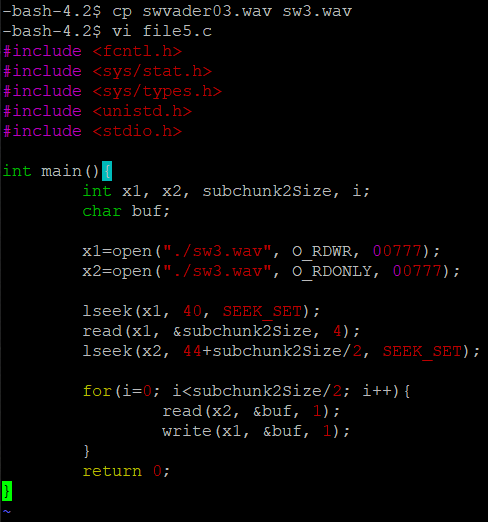


**: 프로그램을 실행 후 sw2.wav의 xxd 결과를 통해 sound data 절반이 0으로 변경된 것을 확인하였다.**



**: psftp를 이용해 내 PC로 sw2.wav를 다운받아 Audacity 응용프로그램으로 실행해보았다. 사운드의 절반이 묵음처리된 것을 확인할 수 있다. 실제 재생시에도 ‘master’만 들렸다.**

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".

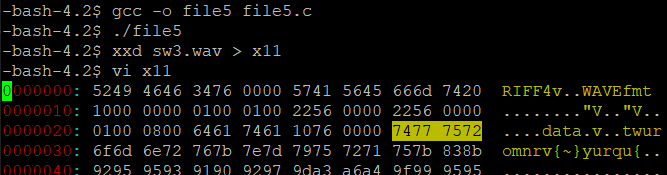
**(1) swvader03.wav를 복사하여 sw3.wav를 생성한다.**

**(2) sw3.wav파일을 하나는 read & write 모드로 열어 x1에 나머지 하나는 읽기전용으로 열어 x2로 파일 번호를 넘겨준다.**

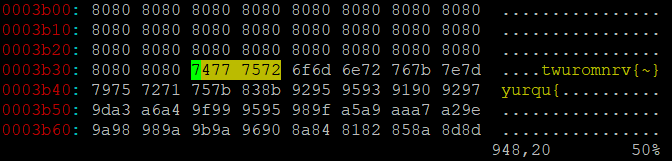
**(3) 4)에서와 동일한 방법으로 lseek와 read를 이용해 subchunk2Size를 추출한다. 이렇게되면 x1은 44bytes부터 시작하게 된다.**

**(4) x2는 lseek를 통해 ‘44+subchunk2Size/2’로 이동한다. 즉, x2는 sound data 절반부터 시작하게 된다.**

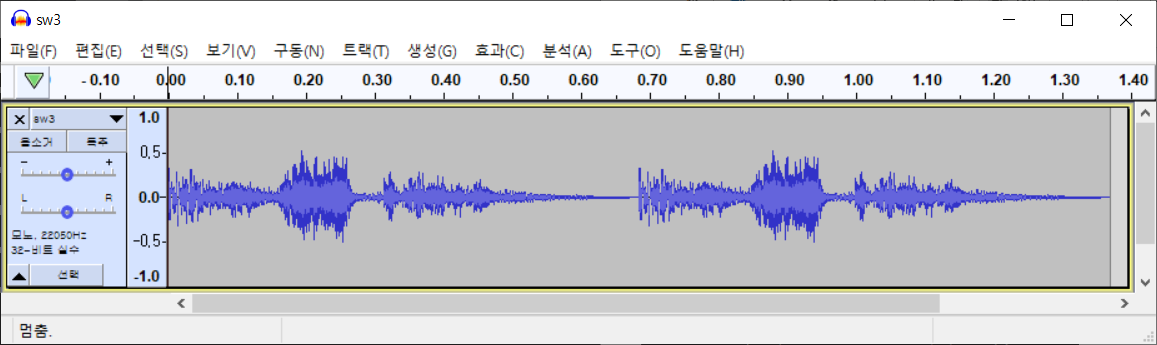
**(5) for문을 이용해 sound data size 반절 동안 x2를 1byte씩 읽어 buf에 저장하고 이 값을 x1에 write하는 것을 반복하도록 하였다.**



**(중략)**



**: [xxd sw3.wav] 결과를 vi로 열어 sound data의 첫 4bytes(7477 7572)를 pattern 검색했다. 그 결과 ‘50%’ 시점에서 같은 패턴이 시작되었다.**



**: 반복되는 파형을 확인할 수 있다. 재생 시에도 ‘master’이 두 번 들렸다.**