Problem Set 6 –Wav Files and Reverb

Total points: 80
Use the structure in file reverb_wav.c and fill in code under the comment blocks Problem 1, 2 and 3.

Install libsndfile
On PC/Windows 7
Run Cygwin setup program, search for "libsndfile" and install

On Mac OS X
Open Terminal window and "cd" to the directory with ps6 files.
Install homebrew
    Go to brew.sh and follow instructions under "Install Homebrew"
Then execute
    brew install libsndfile

Wav file reference
http://soundfile.sapp.org/doc/WaveFormat/

libsndfile reference
http://www.mega-nerd.com/libsndfile/

C function reference
The following URL provides a good reference for C language library function usage:
http://www.cplusplus.com/reference/cstdlib/

Makefile tutorial
http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/

Problem 1: Parse command line and open all files (20 points)

Create a program that has the following command line usage:

```
    main ifile.wav reverb_file.wav [ofile.wav]
Where
    ifile.wav is input audio file
    reverb_file.wav is reverberation inpulse response file
    ofile.wav is output audio file
```

All files must have the same sampling rate and same number of channels and be no more than 2 channels.

Parse command line

Parse the command line. If parsing fails, print an error diagnostic and exit. If parsing succeeds, then print command line values after parsing.

Open WAV files
Use the libsndfile library to open WAV audio files and read the WAV header of each input file. Use error checking and error reporting in all operations.

Check that all files have the same sampling rate and same number of channels. If not, then print an error and exit.

Problem 2: Allocate buffers and read input audio signals into buffers (20 points)

Allocate sufficient storage to ibuf, rbuf and obuf (input buffer, reverb buffer and output buffer) using `malloc()` to read the entirety of each of input WAV files. In addition, allocate the same storage for output WAV file. The number of sample frames in obuf is the number in ibuf plus the number in rbuf.

Read the ifile.wav and rfile.wav audio data. In every case use error checking and error reporting. The data in the file will be interleaved by channel. Add code to de-interleave the sample frame data into separate channel buffers.

Problem 3: Reverberation function (40 points)

Write a function in a separate file to process the input and reverb signals to create the output signal. The function should operate on an arbitrary number of channels (up to MAX_CHN). Create three "versions," selected via a `#define MODE N` preprocessor directive:

#define MODE 1
Just copy the input file values to the output file, and copy an additional rframes-1 zeros. This can be used to test the fread() and fwrite() code in your main program.

#define MODE 2
Convolve the input signal with the reverberation impulse response. The convolution result has iframes+rframes-1 samples.

Compute the RMS value of the input and output signals for all samples and all channels.

Normalize the output signal such that it has the same RMS value as the input signal.

#define MODE 2
Use the supplied `convolve()` function to perform the convolution via FFTs.