Chapter 2  PROGRAMMING EXERCISE

Writing to an Audio Device in C++

This program introduces you to low level sound programming by having you write sound data directly to an audio device.

First, compile and run the program in Algorithm 1.1 on your own computer. Then try altering the program in some of these ways:

- change the program so that it uses the ALSA library rather than OSS
- change the data type of the sample so that it is only one byte long (It is short, which is two bytes, in the program given)
- play the notes one at a time instead of together as a chord
- write the audio data out to a binary file, read it into MATLAB or Octave, and graph it

```c++
//This program uses the OSS library.
#include <sys/ioctl.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <linux/soundcard.h>
#include <iostream>
using namespace std;
#define LENGTH 3   //number of seconds
#define RATE 44100  //sampling rate
#define SIZE 2   //size of sample, in bytes
#define CHANNELS 1  //number of stereo channels
#define PI 3.14159
#define BUFFSIZE (LENGTH*RATE*SIZE*CHANNELS)   // bytes sent to audio device
#define ARRAYSIZE (LENGTH*RATE*CHANNELS)     //total number of samples
#define HALF_STEP 1.05946
#define SAMPLE_MAX 32767

void writeToSoundDevice(short buf[], int deviceID)  {
    int status;
    status = write(deviceID, buf, BUFFSIZE);
    if (status != BUFFSIZE)
        perror("Wrote wrong number of bytes
    status = ioctl(deviceID, SOUND_PCM_SYNC, 0);
    if (status == -1)
        perror("SOUND_PCM_SYNC failed\n");
} 
```
```c
int main()
{
    int deviceID, arg, status, i, t;
    short buf[ARRAYSIZE];
    deviceID = open("/dev/dsp", O_WRONLY, 0);
    if (deviceID < 0) {
        perror("Opening /dev/dsp failed\n");
        exit(1);
    }
    arg = SIZE * 8;
    status = ioctl(deviceID, SOUND_PCM_WRITE_BITS, &arg);
    if (status == -1)
        perror("Unable to set sample size\n");
    arg = CHANNELS;
    status = ioctl(deviceID, SOUND_PCM_WRITE_CHANNELS, &arg);
    if (status == -1)
        perror("Unable to set number of channels\n");
    arg = RATE;
    status = ioctl(deviceID, SOUND_PCM_WRITE_RATE, &arg);
    if (status == -1)
        perror("Unable to set sampling rate\n");
    for (t = 0; t < ARRAYSIZE; ++t)
    {
        buf[t] = (floor(SAMPLE_MAX * sin(2*PI*262*t/RATE)) +
                  floor(SAMPLE_MAX * sin(2*PI*330*t/RATE)) +
                  floor(SAMPLE_MAX * sin(2*PI*392*t/RATE))
                     ) / 3;
        writeToSoundDevice(buf, deviceID);
    }
}
```

**Algorithm 1.1**