

BEEPを鳴らすせ!

著: 11 e8l

0. 事件概要

C/C++言語を用いて Windows API に存在するビーブ音を鳴らす関数を利用し、音楽を演奏させようとした事件。容疑者は「なんとなくやった。今は反省している。」と述べています。

1. 基礎事項

1.1 Beep()関数

問題の関数です。ビーブ音を鳴らします。利用には windows.h ヘッダーファイルをインクルードする必要があります。

```
BOOL Beep(  
    DWORD dwFreq,  
    DWORD dwDuration  
); [MSDN ライブラリより引用]
```

dwFreq は鳴らすビーブ音の音の高さをヘルツ[Hz]単位で指定します。

(有効値は 37~32767[Hz])

dwDuration はビーブ音を鳴らす時間をミリ秒[ms]単位で指定します。

戻り値は成功すると TRUE、失敗すると FALSE が返ってきます。

見てもらえれば分かりますが、この関数には音の高さと発声時間しか指定できませんので、それ以外のは弄れません(パンとかボリュームとか)。

また、この関数は同期関数といって、処理が終わるまでは呼び出し元に制御が戻りません。ということで同時発音数はなんと1つなのです。(CH 数もね。)

1.2 Sleep()関数

面倒なことに Beep()関数では休符を表現できないのです。ということでこの関数で休符の代わりにさせます。

```
VOID Sleep(  
    DWORD dwMilliseconds  
); [MSDN ライブラリより引用]
```

これは dwMilliseconds に指定した時間

[ms] だけスレッドの実行を中断させる

関数です。こんな関数に休符の代わりにさせるのはどうかと思いますが、これしかないのご了承ください。なお、C言語にもともとある sleep()関数とは別物です。こちらは Windows API の関数です。

もちろん、利用には windows.h ヘッダーファイルが必要です。

1.3 発声時間

Beep()関数、Sleep()関数にわたすために、音を出す時間を求めなければいけません。しかし、普通は*分音符で表します。どうしたらいいでしょうか？

曲のテンポを表すものとして BPM (Beat Par Minute) というものがあります。これは「一分間における拍数」(Wikipedia テンポより引用) のことなのですが、ここでは「一拍 = 四分音符」で進めます(適宜、四分音符の部分は読み替えてください)。

というわけで、四分音符の発声時間[ms]は次のようになります。

$\text{四分音符 [ms]} = 60 \times 1000 \div \text{BPM}$

これさえ分かれば後は全ての音符の長さが出ます。全音符の 1/2 の長さが二分音符、二分音符の 1/2 の長さが四分音符、四分音符の 1/2 の長さが八分音符、……のように求めることが可能です。

1.4 音階と周波数

今度は「じゃあドの周波数はいくつだってばよ!? (怒)」という話です。

1オクターブ間には12個の音階があり、1オクターブ上がると周波数が2倍になるように等比数列的に周波数が変化していきなるといいますが、データを載せたいと思います。(次のページにあります。)



C	261.6255653005986	F#/G♭	369.9944227116344
C#/D♭	277.1826309768721	G	391.99543598174927
D	293.6647679174076	G#/A♭	415.3046975799451
D#/E♭	311.1269837220809	A	440
E	329.6275569128699	A#/B♭	466.1637615180899
F	349.2282314330039	B	493.8833012561241

単位 Hz

(音階と周波数の対応表
<http://www.yk.rim.or.jp/~kamide/music/notes.html> より引用)

なお上のデータは MIDI 作曲においてオクターブ数が 3 にあたるものの音階の周波数です。さっきもちらっと述べましたが、1つ上のオクターブの音はこれの 2 倍の周波数になります。

なお Beep()関数に渡せる引数は正の整数値ですので、小数点以下はうまく処理して整数にしてください。

1.5 とりあえず鳴らしてみよう

下のコードは上で示した A の音を 100 ミリ秒鳴らすプログラムです。一度コンパイルして聞いてみてください。

```
#include<windows.h>

int main(void){

    Beep( 440 , 100 );/*ラの音*/

    return 0;
}
```

2.実際に音楽を書いてみよう

2.1 下準備

いくらなんでも Beep()関数とかに数値データをそのまま渡していたのでは、そのうちワケが分からなくなります。

そこで、次のような準備をしましょう。まず、BPM を表す定数を定義します。それをもとに各音符の発声時間を表す定数を定義しましょう。

その後、各音階の周波数を表す定数をいっぱい作るか、オクターブ数と発生時間を指定すると、その音を指定時間分だけ再生するような自作関数 C()、A()、などを作りましょう。そうすれば後で見たとき分かりやすくなりますし、テンポも BPM の値を

変えるだけで済みます。C++ならクラスにしてみるといいかも。

```
/*Sample*/
const DWORD BPM = 120;

/*各音符の発声時間*/
const sibuonpu = 60 * 1000 / BPM;
const hatibuonpu = sibuonpu / 2;
const nibuonpu = sibuonpu * 2;
const zenonpu = nibuonpu * 2;
/*以下略*/

/*音程の定義*/
/*定数の場合*/
const double c = 261.6255653;
/*同様に基準となる周波数を
入力していく*/

const DWORD C1 = (DWORD)c/4;
const DWORD C2 = (DWORD)c/2;
const DWORD C3 = (DWORD)c;
const DWORD C4 = (DWORD)c*2;
const DWORD C5 = (DWORD)c*4;
/*以下略*/

/*関数の場合*/
BOOL A(DWORD octave,
        DWORD dwNoteLength){
    const double A = 440;
    DWORD note;
    /*
    note に octave や A の値から
    適切な値を代入する処理を
    書く
    */
    return Beep(note, dwNoteLength);
}
/*以下略*/
```

2.2 じゃあ書いてみて

一応これで準備は整いました。あとは main()関数内に書いていくだけです。

コメントを入れたり、ある程度の塊を自作関数でまとめたりしながら、自分が作業していくのに十分な程度に見やすく書いていきましょう。

3. 発展的内容

3.1 無理やり和音とか鳴らせないの？

下のように細かい分散和音（アルペジオ）でいくらかそれっぽくなるかもしれませんが、ただ、一つの音を少なくとも 10ms くらい鳴らさないと綺麗に音はでないようです。誤差が少し大きい気がするけどまあ大丈夫だよな？

```
/*Sample*/
/*多分 bpm が 200 以下なら OK*/
VOID Chord(DWORD note1,
           DWORD note2,
           DWORD note3,
           DWORD dwNoteLength){
    DWORD Note[3];
    Note[0] = note1;
    Note[1] = note2;
    Note[2] = note3;

    for(DWORD dwCount = 0;
        (dwCount * 10) < dwNoteLength;
        dwCount++){
        Beep(Note[dwCount%3], 10);
    }
    return;
}
```

上みたいな方法でちょっとずつ周波数を変えた音を鳴らせばグリッサンド（ピッチベンド？）みたいなことも可能です。

でもあえて言おう。**単音だけの方が大分綺麗に鳴らせると。**

3.2 C#とか VB とかやりたいのだけど…。

Windows API を利用する方法と .NET Framework 2.0 の Console.Beep を用いる方法があります。詳しくはご自身でお調べ下さい。なお、Console.Beep の引数等の指定は上の説明と同様です。C#にはこれ以外にも方法がありますが、お手数ですがご自身でお調べ下さい。

4. 最後に

すごく説明がざっくりばらんでしたし、ビープ音自体ものすごくチープですが、是非気分がのったりしたら試してみてください。

5. 参考文献？というか参考にした所。

- ・ Wikipedia
(<http://ja.wikipedia.org/wiki/>)
- ・ MSDN ライブラリ
(<http://msdn.microsoft.com/ja-jp/library/default.aspx>)
- ・ 音階と周波数の対応表
(<http://www.yk.rim.or.jp/~kamide/music/notes.html>)
- ・ @it 2006/04/21 の記事 『ビープ音などのシステム・サウンドを鳴らすには？』
(http://www.atmarkit.co.jp/fdotnet/dotnet_tips/440systemsounds/systemsounds.html)

