

PC-PC 的 IP 电话实验

一、实验目的

在一定程度上了解网络通信中多媒体通信的编程方法和基本思路。在我们的样本程序中，对定义变量和函数调用有比较详细的注释，请同学们自己认真阅读。

二、预备知识

要求对 Winsock 编程和四年级多媒体课程中声卡实验有相当的了解。有关 Winsock 编程的详细知识，请参阅实验“TCP/IP 编程”的实验说明书；或参考文件“Winsock 编程指南”（该文件摘自是 BBS 上“Winsock”版）。有关声卡编程的知识，目前多媒体实验室没有电子版图书，请同学们自己到图书馆借阅相关的“多媒体编程”一类的书籍。

如果感觉到实验较困难，可与辅导老师协商做本科生“计算机网络实验”中“单向语音传输实验”。该实验为本实验的简化版本。

三、实验原理

1、VC 中用于描述波形文件格式的有关结构体定义

(1) 结构体 PCMWAVEFORMAT

PCMWAVEFORMAT 结构用于描述 PCM 结构波形数据的格式，其结构定义如下所示：

```
typedef struct {  
    WAVEFORMAT wf;  
    WORD        wBitsPerSample;  
} PCMWAVEFORMAT;
```

其中 wBitsPerSample 是采样率

其中 wf 为 WAVEFORMAT 结构，用于描述数据的一般结构信息，它包括了所有波形文件的通用格式信息。结构定义如下所示：

```
typedef struct {  
    WORD    wFormatTag;  
    WORD    nChannels;  
    DWORD  nSamplesPerSec;  
    DWORD  nAvgBytesPerSec;  
    WORD    nBlockAlign;
```

如果您在阅读过程中发现疏漏和错误，请您尽快和编者取得联系 network@ustc.edu.cn exh@ustc.edu.cn

```

        WORD  wBitsPerSample;
        WORD  cbSize;
    } WAVEFORMATEX;

```

具体参数解释如下：

wFormatTag

波形数据的格式，定义在 MMREG.H 文件中

nChannels

波形数据的通道数：单声道或立体声

nSamplesPerSec

采样率，对于 PCM 格式的波形数据，采样率有 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz 等几种

nAvgBytesPerSec

数据率，对于 PCM 格式的波形数据，数据率等于采样率乘以每样点字节数

nBlockAlign

每个样点的字节数

wBitsPerSample

采样精度，对于 PCM 格式的波形数据，采样精度为 8 或 16

cbSize

附加格式信息的数据块大小

(2) 设备头结构 WAVEHDR

结构 WAVEHDR 定义了指向波形数据缓冲区的设备头，具体定义如下：

```

typedef struct {
    LPSTR  lpData;
    DWORD  dwBufferLength;
    DWORD  dwBytesRecorded;
    DWORD  dwUser;
    DWORD  dwFlags;
    DWORD  dwLoops;
    struct wavehdr_tag * lpNext;
    DWORD  reserved;
} WAVEHDR;

```

lpData 波形数据的缓冲区地址

dwBufferLength 波形数据的缓冲区地址的长度

dwBytesRecorded 当设备用于录音时，标志已经录入的数据长度

dwUser 用户数据

dwFlags 波形数据的缓冲区的属性

dwLoops 播放循环的次数，仅用于播放控制中

lpNext 和 **reserved** 均为保留值

注意：参考 VC 中 `mmsystem.h` 文件中的有关定义可以知道，`PWAVEHDR`、`NPWAVEHDR`、`LPWAVEHDR`、`LPHWAVEIN` 均定义为指向结构 `WAVEHDR` 的指针。在实验的样本程序中，用到下列有关多媒体的结构：“`HWAVEIN`”、“`LPWAVEHDR`”、如果您在阅读过程中发现疏漏和错误，请您尽快和编者取得联系 network@ustc.edu.cn exh@ustc.edu.cn

“ MMRESULT”。

2、样本程序的设计思路

(1) 样本程序的基本思路

本实验样本程序的基本思路是：发送端将录入的声音分时打包发送；接收端将收到的 IP 包逐个播放。实验室给出的样本程序有 2 个文件：SendVoice 目录下的项目文件用于录入并发送声音、RecieveVoice 目录下的项目文件用于接收并播放声音。整个程序建立在 TCP/IP 编程的技术之上，思路非常简单：初始化 SOCKET、初始化声音设备、开始录音，然后由 WINDOWS 自动发出的缓冲区满的消息控制发送。在程序结束时，并没有将所有打开的设备关闭，希望同学们在实验过程中能够将程序结束的清理部分内容添入。

(2) 发送程序中有关声音的参数

程序中有关声音的参数定义如下：

```
char   pInBuffer1;           //录音缓冲区 1(char)
char   pInBuffer2;           //录音缓冲区 2(char)
HWAVEIN      hWaveIn;        //录音设备标志
LPWAVEHDR    pWaveHdr1;      //设备设置头 1
LPWAVEHDR    pWaveHdr2;      //设备设置头 2
MMRESULT     result;         //函数返回值
PCMWAVEFORMAT pcm;          //波形数据格式
BOOL m_RecFirst;            //缓冲区使用标志
```

以上用到的结构体在实验原理的第一部分中有较为简洁的说明。如果看完了实验原理第一部分以后仍然对 WINDOWS 中有关多媒体部分的结构定义不太清楚的话，请仔细阅读 VC 中的相关帮助或到查询其他有关的书籍。

在程序的初始化部分对声音采样的有关参数进行赋值

```
//Format of voice: 16 bits,11.025kHz,mono audio
pcm.wf.wFormatTag=WAVE_FORMAT_PCM; //格式为 PCM
pcm.wf.nChannels=1;                //单声道 mono
pcm.wf.nSamplesPerSec=8000;        //采样率
pcm.wf.nAvgBytesPerSec=8000*2;     //数据率
pcm.wf.nBlockAlign=2;              //每样点的字节数
pcm.wBitsPerSample=16;             //采样精度
```

并且分配缓冲区所需要的内存

```
//录音设备头的初始化
pWaveHdr1 =(LPWAVEHDR)GlobalAllocPtr(GHND|GMEM_SHARE,sizeof(WAVEHDR));
pWaveHdr2 =(LPWAVEHDR)GlobalAllocPtr(GHND|GMEM_SHARE,sizeof(WAVEHDR));
//录音缓冲区
pInBuffer1 =(char *)GlobalAllocPtr(GHND|GMEM_SHARE,PCMBUFFER_SIZE);
```

如果您在阅读过程中发现疏漏和错误，请您尽快和编者取得联系 network@ustc.edu.cn exh@ustc.edu.cn

```
pInBuffer2      =(char *)GlobalAllocPtr(GHND|GMEM_SHARE,PCMBUFFER_SIZE);
//初始化缓冲区控制变量, 首先使用第一个缓冲区
m_RecFirst=TRUE;
```

在录音过程中用了两个录音缓冲区 pInBuffer1、pInBuffer2, 大小都是 360*8。录音过程中轮流使用 pInBuffer1 和 pInBuffer2, 通过存取 BOOL 变量 m_RecFirst 来标志正在使用的是哪一个录音缓冲区。

(3) 接收程序中有关声音的参数

程序中有关声音的参数定义如下:

```
char    *    pOutBuffer1;    //放音缓冲区 1
char    *    pOutBuffer2;    //放音缓冲区 2
char    *    rBuffer;        //接收缓冲区 No USE
HWAVEOUT  hWaveOut;        //录音设备标志
LPWAVEHDR  pWaveHdr1;       //设备设置头 1
LPWAVEHDR  pWaveHdr2;       //设备设置头 2
MMRESULT  result;
PCM_WAVEFORMAT pcm;        //波形数据格式
BOOL m_PlayFirst;          //缓冲区使用标志
```

以上用到的结构体在实验原理的第一部分中有较为简洁的说明。如果看完了实验原理第一部分以后仍然对 WINDOWS 中有关多媒体部分的结构定义不太清楚的话, 请仔细阅读 VC 中的相关帮助或到查询其他有关的书籍。

在程序的初始化部分对声音采样的有关参数进行赋值

```
//Format of voice: 16 bits,11.025kHz,mono audio
pcm.wf.wFormatTag=WAVE_FORMAT_PCM; //格式为 PCM
pcm.wf.nChannels=1;                //单声道 mono
pcm.wf.nSamplesPerSec=8000;        //采样率
pcm.wf.nAvgBytesPerSec=8000*2;     //数据率
pcm.wf.nBlockAlign=2;              //每样点的字节数
pcm.wBitsPerSample=16;             //采样精度
并且分配缓冲区所需要的内存
pWaveHdr1  =(LPWAVEHDR)GlobalAllocPtr(GHND|GMEM_SHARE,sizeof(WAVEHDR));
pWaveHdr2  =(LPWAVEHDR)GlobalAllocPtr(GHND|GMEM_SHARE,sizeof(WAVEHDR));
//放音缓冲区
pOutBuffer1  =(char *)GlobalAllocPtr(GHND|GMEM_SHARE,PCMBUFFER_SIZE);
pOutBuffer2  =(char *)GlobalAllocPtr(GHND|GMEM_SHARE,PCMBUFFER_SIZE);
//接收缓冲区
rBuffer      =(char *)GlobalAllocPtr(GHND|GMEM_SHARE,PCMBUFFER_SIZE);
//初始化缓冲区控制变量, 首先使用第一个缓冲区
m_PlayFirst=TRUE;
```

在放音过程中用了两个放音缓冲区 pOutBuffer1、pOutBuffer2, 大小都是 360*8。录音如果您在阅读过程中发现疏漏和错误, 请您尽快和编者取得联系 network@ustc.edu.cn exh@ustc.edu.cn

过程中轮流使用 pOutBuffer1 和 pOutBuffer2，通过存取 BOOL 变量 m_PlayFirst 来标志正在使用的是哪一个录音缓冲区。

注意：在样本程序中，需要将 `stdlib.h`、`malloc.h`、`windows.h`、`mmsystem.h` 四个头文件包含进项目，并且需要在项目的连接设置中将库 `mmwin.lib` 连接入项目。

四、实验要求

1. 实验 2 人一组或一人在本机上自行操作演示。实验样本程序发送和接收是两个独立的程序，但均具备发送和接收两项功能，需要在实验过程中更改 IP 地址、端口号等基本信息，修改方法参见实验代码目录中的说明文件。编写自己的程序集成发送和接收功能。由辅导老师或实验室当值老师检查通过。

五、思考题

1. WINDOWS 下 SOCKET 编程中阻塞方式和非阻塞方式有什么区别？在你的程序中采用的是哪种方式？为什么采用这种方式？

2. 波形文件的格式定义包括哪些部分？在 VC 中有那些结构体定义与之相对应？这些结构体是在哪个头文件中声明的？

3. 为了保证发送实时的声音，在实验的样本程序中采用双缓冲区的结构，为什么？你采用的是何种办法？阅读有关的联机帮助或文献，是否有更好的办法？怎样实现？