http://blog.chinaunix.net/uid-27106528-id-3328766.html

```
#include<stdio.h>
#include<stdlib.h>
#include <string.h>
#include <alsa/asoundlib.h>

struct WAV_HEADER
{
  char rld[4]; //riff 标志符号
  int rLen;
  char wld[4]; //格式类型（wave）
  char fld[4]; //"fmt"

  int fLen; //sizeof(wave format matex)

  short wFormatTag; //编码格式
  short wChannels; //声道数
  int nSamplesPersec ; //采样频率
  int nAvgBitsPerSample;//WAVE文件采样大小
  short wBlockAlign; //块对齐
  short wBitsPerSample; //WAVE文件采样大小

  char dld[4]; //”data“
  int wSampleLength; //音频数据的大小

} wav_header;

int set_pcm_play(FILE *fp);

int main(int argc,char *argv[])
{

  if(argc!=2)
  {
    printf("Usage:wav-player+wav file name\n");
    exit(1);
  }

  int nread;
  FILE *fp;
  fp=fopen(argv[1],"rb");
  if(fp==NULL)
  {
    perror("open file failed:\n");
    exit(1);
  }

  nread=fread(&wav_header,1,sizeof(wav_header),fp);
  printf("nread=%d\n",nread);

  //printf("RIFF 标志%s\n",wav_header.rld);
```

```
    printf("文件大小rLen：%d\n",wav_header.rLen);
    //printf("wId=%s\n",wav_header.wId);
    //printf("fId=%s\n",wav_header.fId);

   // printf("fLen=%d\n",wav_header.fLen);

    //printf("wFormatTag=%d\n",wav_header.wFormatTag);
    printf("声道数：%d\n",wav_header.wChannels);
    printf("采样频率：%d\n",wav_header.nSamplesPersec);
    //printf("nAvgBitsPerSample=%d\n",wav_header.nAvgBitsPerSample);
    //printf("wBlockAlign=%d\n",wav_header.wBlockAlign);
    printf("采样的位数：%d\n",wav_header.wBitsPerSample);

   // printf("data=%s\n",wav_header.dId);
    printf("wSampleLength=%d\n",wav_header.wSampleLength);





    set_pcm_play(fp);
    return 0;
}

int set_pcm_play(FILE *fp)
{
        int rc;
        int ret;
        int size;
        snd_pcm_t* handle; //PCI设备句柄
        snd_pcm_hw_params_t* params;//硬件信息和PCM流配置
        unsigned int val;
        int dir=0;
        snd_pcm_uframes_t frames;
        char *buffer;
        int channels=wav_header.wChannels;
        int frequency=wav_header.nSamplesPersec;
        int bit=wav_header.wBitsPerSample;
        int datablock=wav_header.wBlockAlign;
        unsigned char ch[100]; //用来存储wav文件的头信息



        rc=snd_pcm_open(&handle, "default", SND_PCM_STREAM_PLAYBACK, 0);
        if(rc<0)
        {
                perror("\nopen PCM device failed:");
                exit(1);
        }


        snd_pcm_hw_params_alloca(&params); //分配params结构体
```

```
if(rc<0)
{
        perror("\nsnd_pcm_hw_params_alloca:");
        exit(1);
}
 rc=snd_pcm_hw_params_any(handle, params);//初始化params
if(rc<0)
{
        perror("\nsnd_pcm_hw_params_any:");
        exit(1);
}
rc=snd_pcm_hw_params_set_access(handle, params, SND_PCM_ACCESS_RW_INTERLEAVED); //初始化访问权限
if(rc<0)
{
        perror("\nsed_pcm_hw_set_access:");
        exit(1);

}

//采样位数
switch(bit/8)
{
case 1:snd_pcm_hw_params_set_format(handle, params, SND_PCM_FORMAT_U8);
        break ;
case 2:snd_pcm_hw_params_set_format(handle, params, SND_PCM_FORMAT_S16_LE);
        break ;
case 3:snd_pcm_hw_params_set_format(handle, params, SND_PCM_FORMAT_S24_LE);
        break ;

}
rc=snd_pcm_hw_params_set_channels(handle, params, channels); //设置声道,1表示单声>道，2表示立体声
if(rc<0)
{
        perror("\nsnd_pcm_hw_params_set_channels:");
        exit(1);
}
val = frequency;
rc=snd_pcm_hw_params_set_rate_near(handle, params, &val, &dir); //设置>频率
if(rc<0)
{
        perror("\nsnd_pcm_hw_params_set_rate_near:");
        exit(1);
}

rc = snd_pcm_hw_params(handle, params);
if(rc<0)
{
perror("\nsnd_pcm_hw_params: ");
exit(1);
}

rc=snd_pcm_hw_params_get_period_size(params, &frames, &dir); /*获取周期
```

```
长度*/
        if(rc<0)
        {
                perror("\nsnd_pcm_hw_params_get_period_size:");
                exit(1);
        }

        size = frames * datablock; /*4 代表数据快长度*/

        buffer =(char*)malloc(size);
    fseek(fp,58,SEEK_SET); //定位歌曲到数据区

    while (1)
        {
                memset(buffer,0,sizeof(buffer));
                ret = fread(buffer, 1, size, fp);
                if(ret == 0)
                {
                        printf("歌曲写入结束\n");
                        break;
                }
                 else if (ret != size)
                {
                 }
                // 写音频数据到PCM设备
        while(ret = snd_pcm_writei(handle, buffer, frames)<0)
           {
                 usleep(2000);
                 if (ret == -EPIPE)
                {
                  /* EPIPE means underrun */
                  fprintf(stderr, "underrun occurred\n");
                  //完成硬件参数设置，使设备准备好
                  snd_pcm_prepare(handle);
                 }
                 else if (ret < 0)
                 {
                          fprintf(stderr,
                      "error from writei: %s\n",
                      snd_strerror(ret));
                 }
            }

    }

        snd_pcm_drain(handle);
        snd_pcm_close(handle);
        free(buffer);
        return 0;
}
```