

Linux操作系统的声音设备编程实例Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_Linux\\_E6\\_93\\_8D\\_E4\\_BD\\_c103\\_644785.htm](https://www.100test.com/kao_ti2020/644/2021_2022_Linux_E6_93_8D_E4_BD_c103_644785.htm)

Linux下的声音设备编程比大多数人想象的要简单得多。一般说来，我们常用的声音设备是内部扬声器和声卡，它们都对应/dev目录下的一个或多个设备文件，我们象打开普通文件一样打开它们，用ioctl（）函数设置一些参数，然后对这些打开的特殊文件进写操作。由于这些文件不是普通的文件，所以我们不能用ANSI C（标准C）的fopen、fclose等来操作文件，而应该使用系统文件I/O处理函数（open、read、write、lseek和close）来处理这些设备文件。ioctl（）或许是Linux下最庞杂的函数，它可以控制各种文件的属性，在Linux声音设备编程中，最重要的就是使用此函数正确设置必要的参数。下面我们举两个实际的例子来说明如何实现Linux下的声音编程。由于此类编程涉及到系统设备的读写，所以，很多时候需要你具有root权限，如果你将下面的例子编译后不能正确执行，那么，首先请你检查是否是因为没有操纵某个设备的权限。对内部扬声器编程内部扬声器是控制台的一部分，所以它对应的设备文件为/dev/console。变量KIOCSOUND在头文件/usr/include/linux/kd.h中声明，ioctl函数使用它可以来控制扬声器的发声，使用规则为：ioctl（fd, KIOCSOUND, (int) tone). fd为文件设备号，tone是音频值。当tone为0时，终止发声。必须一提的是它所理解的音频和我们平常以为的音频是不同的，由于计算机主板定时器的时钟频率为1.19MHZ，所以要进行正确的发声，必须进行如下的转换：扬声器音频值=1190000/我们

期望的音频值。 扬声器发声时间的长短我们通过函数usleep ( unsigned long usec ) 来控制。 它是在头文件/usr /include /unistd.h中定义的 , 让程序睡眠usec微秒。 下面即是让扬声器按指定的长度和音频发声的程序的完整清单 :

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <getopt.h>
/* 设定默认值 */ #define
DEFAULT_FREQ 440 /* 设定一个合适的频率 */ #define
DEFAULT_LENGTH 200 /* 200 微秒 , 发声的长度是以微秒为
单位的*/ #define DEFAULT_REPS 1 /* 默认不重复发声 */
#define DEFAULT_DELAY 100 /* 同样以微秒为单位*/ /* 定义
一个结构 , 存储所需的数据*/ typedef struct { int freq. /* 我们期
望输出的频率 , 单位为Hz */ int length. /* 发声长度 , 以微秒为
单位*/ int reps. /* 重复的次数*/ int delay. /* 两次发声间隔 , 以
微秒为单位*/ } beep_parms_t. /* 打印帮助信息并退出*/ void
usage_bail ( const char *executable_name ) { printf ( "Usage: \n \t%s
[-f frequency] [-l length] [-r reps] [-d delay] \n ", executable_name
). exit(1). } /* 分析运行参数 , 各项意义如下 : * "-f gt." * "-l gt."
* "-r gt." * "-d gt." */ void parse_command_line(char **argv,
beep_parms_t *result) { char *arg0 = *(argv). while ( *argv ) { if (
!strcmp( *argv,"-f" )) { /*频率*/ int freq = atoi ( *( argv ) ). if ( ( freq
gt. 10000 ) ) { fprintf ( stderr, "Bad parameter: frequency must be
from 1..10000\n" ). exit ( 1 ) . } else { result->freq = freq. argv . }
100Test 下载频道开通 , 各类考试题目直接下载。 详细请访问
www.100test.com
```