

SAMS
PUBLISHING

DOS和Windows 环境下的声音编程技术

[美] Nathan Gurewich & Ori Gurewich 著

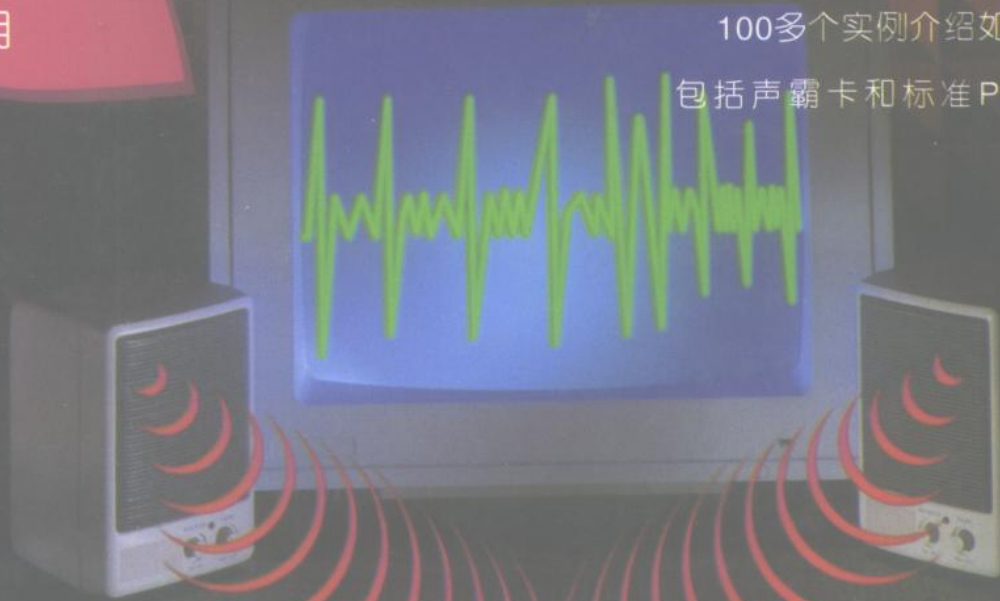
冯文全 马维丽 李良珍 译

黄志瑜 审校

无需扩充任何
硬件就可拓展
PC应用

创建DOS和Windows新应用
—— 逼真的话音和音乐

100多个实例介绍如何生成声音
包括声霸卡和标准PC扬声器等



配套磁盘包含
本书程序的源代码
C函数库
实用的声音程序



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

DOS 和 Windows 环境下的 声音编程技术

[美] Nathan Gurewich 和 Ori Gurewich 著

冯文全 马维丽 李良珍 译

陈 熺 审校



电子工业出版社

内 容 简 介

本书完整地介绍了在 Windows 和 DOS 下的声音编程技术。全书共分十三章:第一章是声音编程技术的概述;第二章到第九章,详细介绍了如何通过 PC 机内置扬声器播放声音的编程技术;第十章介绍如何编写 Windows 的声音独立程序(包含声音文件在内的 EXE 文件);第十一章介绍通过声霸卡(Sound Blaster)播放声音;第十二章和第十三章介绍了 DOS 下的声音编程技术;最后一部分为附录,介绍了声音动态链接库(DLL)并给出本书中常见术语索引表。

本书附赠磁盘中包含了可调用的 C 语言函数库。

本书既是声音编程技术的入门指导,又是一本内容全面的使用手册。书中既有技术内容的剖析,又包含许多程序实例。全书内容论述准确深入,图文并茂。本书是软件开发人员、操作人员、大中专院校学生以及各类计算机爱好者学习、研究和必备的常用参考书。

"Authorized translation from the English language edition published by Sams"

Copyright © 1995

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission in writing from the Publisher.

Chinese language edition published by
Publishing House of Electronic Industry, China

本书英文版为美国 Macmillan Computer Publishing(之分公司 Sams)所拥有,电子工业出版社从该公司购得该书的中文专有出版权。版权所有,侵权必究。

DOS 和 Windows 环境下的声音编程技术

[美] Nathan Gurewicz 和 Ori Gurewicz 著

冯文全 马维丽 李良珍 译

陈 焯 审校

责任编辑 玉庆育(特约) 史明生

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

电子工业出版社计算机排版室排版

中国科学院印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:23 字数:600 千字

1996 年 5 月第一版 1996 年 5 月北京第一次印刷

印数:3500 册 定价:48.00 元

ISBN 7-5053-3381-X/TP·1301

著作权合同登记号 图字:01-1995-568

鸣 谢

我们非常感谢 Stacy Hiquet 女士,她是 Sams 出版社的资深编辑。衷心感谢她决定出版本书,尤其要感谢的是她在本书的编写与制作过程中的各种推荐和建议。

我们也衷心感谢本书责任编辑 Judy Brunetti 女士,感谢她在编辑手稿过程中杰出的工作和她的许多建议、调查和协商。

我们也衷心感谢技术编辑 Burce Graves,他亲自编译、连接了本书中所有程序并验证了编译连接后程序的执行情况。

作者简介

Nathan Gurewich 在纽约州哥伦比亚大学取得电子工程硕士学位,在纽约长岛的哈佛大学取得了电子工程学士学位。随着 PC 的推广,作者一直致力于 PC 机的商业软件包的设计与创作。他是 PC 程序设计领域的专家,并在局域网络,广域网络,数据库管理与设计等方面提供咨询服务。

Ori Gurewich 在纽约州 Stony Brook 大学获得工程硕士学位。他是软件开发专业多媒体和 Windows 应用程序的公司的高级软件工程师和软件顾问。他是 PC 编程技术和网络通讯领域的专家,开发出多种用于 PC 的多媒体和声音编程软件。

目 录

第一章 声音程序设计	(1)
1.1 PC 机说话与播放音乐的概念	(1)
1.1.1 适合人视听感的程序	(1)
1.2 PC 机有无附加硬件时发音的概念	(2)
1.3 安装本书附带的磁盘	(2)
1.4 TSENGINE	(3)
1.5 本书附带的磁盘	(3)
1.5.1 安装本书的磁盘	(3)
第二章 声音编程技术和声音库	(5)
2.1 如何使 PC 机发声	(5)
2.2 如何在程序中运用声音	(5)
2.3 Windows 操作系统	(5)
2.3.1 在程序管理器中生成新的图标组	(5)
2.4 执行应用程序	(11)
2.5 本书内容的安排	(11)
2.6 DOS 下的声音文件	(11)
第三章 Generic1 Windows 程序	(13)
3.1 Generic1.C 和 Generic2.C	(13)
3.2 Generic1.C 程序的文件	(13)
3.3 快速浏览 Generic1	(20)
3.4 Generic1.C 的 #include 部分	(20)
3.5 Generic1.C 中的 WinMain() 函数	(20)
3.5.1 WinMain() 函数的局部变量	(21)
3.5.2 共用 Generic1 实例	(21)
3.5.3 更新和注册 Window 类	(21)
3.5.4 建立、更新和显示 Program Window 程序窗口	(22)
3.5.5 Generic1.C 的消息循环	(22)
3.6 Wndproc() 函数	(23)
3.6.1 Wndproc() 函数的消息开关(Switch)	(23)
3.6.2 处理 WM_CREATE 消息	(24)
3.6.3 处理 WM_PAINT 消息	(24)
3.6.4 处理 WM_COMMAND 消息	(25)
3.6.5 处理 WM_DESTROY 消息	(26)
3.7 源文件 Generic1.rc	(27)

3.8 模块定义文件 Generic1.def	(28)
3.9 用于 Microsoft 编译器的 MAKE 文件:Generic1.mak	(28)
3.9.1 - C 开关	(29)
3.9.2 - G2sw 开关	(29)
3.9.3 - Ow 开关	(29)
3.9.4 - W3 开关	(29)
3.9.5 - Zp 开关	(30)
3.10 Generic1.mak 的连接部分	(30)
3.11 TegoWin.lib 库	(31)
3.12 NMAKE 程序	(31)
3.12.1 用 Microsoft C 编译器编译并连接	(31)
3.12.2 用 Microsoft 程序员工作平台(PWB)编译并连接	(32)
3.12.3 Borland 编译器的 Make 文件 Generic1.mak	(32)
3.13 执行 Generic1.exe 程序	(32)
3.13.1 在 Windows 下执行 Generic1	(32)
3.13.2 为程序生成一个图标	(32)
3.13.3 在 DOS 命令行中执行的 Generic1	(33)
3.14 编写基于 Generic1.C 格式的程序	(33)

第四章 Hello.c 程序	(35)
4.1 用 Microsoft 编译器编译、连接 Hello 程序	(35)
4.2 用 Borland 编译器编译、连接 Hello 程序	(35)
4.3 执行 Hello 程序	(35)
4.4 Hello 程序的文件	(36)
4.4.1 Make 文件 Hello.mak	(43)
4.4.2 源文件 Hello.rc	(43)
4.4.3 模块定义文件 Hello.def	(43)
4.4.4 #include 文件 Hello.h	(43)
4.4.5 Hello.c 文件	(43)
4.5 DLL 函数对静态 sp_ 函数	(44)
4.6 其它 sp_ 函数的用法	(44)
4.7 sp_OpenSession()函数的参数	(45)
4.7.1 声音文件 Hello.ts	(45)
4.7.2 独立参数	(45)
4.7.3 sp_OpenSession()函数的第三个参数	(45)
4.7.4 文件类型参数	(46)
4.8 sp_PlayF()函数	(46)
4.9 sp_START_OF_FILE 和 sp_END_OF_FILE 标识符	(47)
4.10 Hello.c 中的其它 sp_PlayF()语句	(48)
4.11 TS Sound Editor 程序	(48)
4.11.1 在退出程序时向用户说 Good-Bye	(49)
4.11.2 sp_PlayLabelF()函数和 sp_PlayTimeF()函数	(50)

4.12 Sections 程序	(51)
4.12.1 Sections 程序的编译和连接	(52)
4.12.2 Sections 程序的运行	(53)
4.12.3 Sections 程序的声音段	(53)
4.12.4 Sections 程序的文件	(53)
4.12.5 WM_CREATE case	(61)
4.12.6 播放整个声音文件	(61)
4.12.7 播放声音文件的声音段	(62)
4.12.8 WM_ABOUT case	(62)
4.12.9 WM_DESTROY case	(62)
4.13 Push2Say 程序	(63)
4.13.1 编译、连接 Push2Say 程序	(64)
4.13.2 Push2Say 程序的文件	(64)
4.13.3 Push2Say 程序的按钮	(72)
第五章 声音文件类型	(75)
5.1 FileType 程序	(75)
5.1.1 编译连接 FileType 程序	(76)
5.1.2 执行 FileType 程序	(76)
5.1.3 FileType 程序的文件	(76)
5.1.4 位图文件	(88)
5.2 FileType 程序概述	(88)
5.2.1 作为主窗口的对话框	(88)
5.2.2 处理 WM_COMMAND 消息	(89)
5.2.3 sp_OpenSession 的其他返回值	(91)
5.2.4 处理键盘信息	(91)
5.2.5 About 对话框	(92)
5.2.6 SetFocus() 函数	(92)
5.2.7 显示位图	(92)
5.2.8 FileType 的背景	(94)
第六章 动画	(95)
6.1 Dog 程序	(95)
6.1.1 编译、连接和运行 Dog 程序	(95)
6.1.2 什么是动画制作	(96)
6.1.3 建立位图	(97)
6.1.4 Dog 程序文件	(97)
6.1.5 SP_PlayF() 函数返回值的作用	(110)
6.1.6 Dog.c 的 WinMain() 函数	(110)
6.1.7 Dog.c 的 WndProc()	(110)
6.1.7.1 WM_CREATE case	(111)
6.1.7.2 WM_PAINT case	(111)
6.1.7.3 WM_DESTROY case	(112)

6.1.7.4 WM_CHAR case	(113)
6.1.7.5 WM_LBUTTONDOWN case	(113)
6.1.8 BarkingShow()函数	(114)
6.1.9 PlayIt()函数	(114)
6.1.10 DisplayDogWithOpen()和 DisplayDogWithClose()函数	(116)
6.2 HearMe 程序	(116)
6.2.1 HearMe 程序的文件	(116)
6.2.2 编译、连接和运行 HearMe 程序	(116)
6.2.3 播放 Good-Bye 和退出程序	(129)
6.2.4 HearMe 程序的面部表情	(130)
6.2.5 放音与动画制作	(130)
6.2.6 DisplayFaceWithOpen()、DisplayFaceWithClose() 和 DisplayFaceWithMid()函数	(132)
6.2.7 改进 HearMe 程序	(133)
6.3 Dance 程序	(133)
6.3.1 编译、连接和运行 Dance 程序	(133)
6.3.2 Dance 程序的文件	(133)
6.3.3 Dance 程序的 WndProc()	(149)
6.3.4 舞姿显示	(150)
6.3.5 DanceShow()函数	(150)
第七章 与语音同步移动文本	(155)
7.1 Press 程序	(155)
7.1.1 编译连接并运行 Press 程序	(155)
7.1.2 Press 程序的文件	(156)
7.1.3 Press.c 的 Wndproc()函数	(157)
7.1.4 PlayInstruction()函数	(158)
7.1.5 与 Press 相似的程序	(160)
7.2 PlzWait 程序	(160)
7.2.1 编译连接 PlzWait 程序	(160)
7.2.2 执行 PlzWait 程序	(161)
7.2.3 PlzWait 程序的文件	(162)
7.2.4 PlzWait 的 WndProc()函数	(162)
7.2.5 SpeakShow()函数	(165)
7.2.6 DisplayPlease()函数	(167)
7.2.7 完成显示的另一个方法	(169)
7.2.8 应用 Paintbrush	(169)
7.2.9 创建显示的建议	(169)
第八章 控制	(171)
8.1 Rotate 程序	(171)
8.1.1 编译、连接、运行 Rotate 程序	(171)
8.1.2 Rotate 程序文件	(172)
8.1.3 打开声音话路	(186)

8.1.4	运行 Rotate 对话框	(186)
8.1.5	初始化 Rotate 对话框	(186)
8.1.6	确定 Speed 滚动条的最小值与最大值	(187)
8.1.7	确定 Speed 滚动条的自然位置	(187)
8.1.8	保存速度滚动条的当前位置	(188)
8.1.9	装入 Rotate 对话框中用到的位图文件	(188)
8.1.10	Rotate 对话框的 WM_PAINT Case	(188)
8.1.11	左移或右移滚动条	(189)
8.1.12	拖动 Speed 滚动条	(189)
8.1.13	ChangeSpeedwasRequested()函数	(189)
8.1.14	sp_SetNewSpeed()函数	(190)
8.1.15	Rotate 对话框的 PLAY_PB 按钮	(190)
8.1.16	Rotate 对话框的 PLAY_BACKWARD_PB Case	(191)
8.1.17	sp_PlayB()函数	(191)
8.1.18	ROTATERIGHT()与 ROTATELEFT()函数	(192)
8.1.19	关于 sp_PlayB()函数的进一步解释	(192)
8.1.20	退出 Rotate 对话框	(193)
8.2	Controls 程序	(193)
8.2.1	编译、连接和运行 Controls 程序	(193)
8.2.2	Controls 程序的文件	(195)
8.2.3	Controls 程序中的新 sp_ 函数	(211)
8.3	关于 Controls 程序	(212)
8.3.1	在播放中显示声音文件的位置	(212)
8.3.2	使用户能够停止播放	(213)
8.3.3	使用户能在播放过程中改变播放速度	(214)
8.3.4	选择播放组的大小	(214)
第九章	基于 Generic2 的程序——多任务	(215)
9.1	Windows 操作系统	(215)
9.1.1	Generic1 的消息循环	(215)
9.1.2	Generic2 的消息循环	(216)
9.2	Generic2 程序	(217)
9.2.1	编译和连接 Generic2 程序	(217)
9.2.2	执行 Generic2 程序	(218)
9.2.3	Generic2 文件	(218)
9.2.4	Generic2 的 WinMain()函数	(226)
9.2.5	允许使用鼠标	(227)
9.2.6	消息循环前的初始化	(227)
9.2.7	在消息循环里的播放声音	(228)
9.2.8	Generic2 的 WndProc()函数	(229)
9.2.9	禁止鼠标	(229)
9.2.10	多例子	(230)
9.3	SayName 程序	(230)

9.3.1 编译和连接 SayName 程序	(230)
9.3.2 执行 SayName 程序	(230)
9.3.3 SayName 文件	(231)
9.3.4 SayName 的 WinMain()函数	(240)
9.3.5 SayName 的消息循环	(240)
9.3.6 监视 Windows 对话	(241)
9.4 Organ 程序	(242)
9.4.1 编译连接和执行 Organ 程序。	(242)
9.4.2 Organ 的文件	(244)
9.4.3 Organ 的 WinMain()函数	(244)
9.4.4 Organ 的消息循环	(245)
9.4.5 初始化 Scroll Bar	(246)
9.4.6 滚动条变化的结果	(246)
9.4.7 处理键盘键	(247)
9.4.8 演奏风琴	(248)
9.4.9 处理按钮	(248)
9.4.10 指令对话框	(248)
9.4.11 Organ 程序升级	(249)
9.5 Control2 程序	(249)
9.5.1 编译、连接和执行 Control2 程序	(249)
9.5.2 Control2 文件	(250)
9.5.3 程序主窗口对话框	(251)
9.5.4 打开声音话路	(251)
9.5.5 Control2 的消息循环	(252)
9.5.6 Control2 的 WM-CREATE case	(253)
9.5.7 Control2 的 WM-PAINT case	(253)
9.5.8 Control2 的 WM-CHAR case	(254)
9.5.9 Control2 的 WM-HSCROLL case	(255)
9.5.10 处理按钮	(255)
9.5.11 允许和禁止使用鼠标	(256)
第十章 独立程序	(257)
10.1 改进 Dog 程序为独立程序	(257)
10.1.1 TSlink 实用程序	(259)
10.1.2 运行独立 Dog 程序	(260)
10.2 使用 TSLabels	(260)
10.3 参数 _argv[]	(260)
10.4 sp_OpenSession()函数的第三个参数	(261)
10.5 转换其它程序	(261)
10.6 WhoAmI 程序	(261)
10.6.1 编译和连接 WhoAmI 程序	(268)
10.6.2 将一个声音文件和 WhoAmI 程序连接起来	(268)
10.7 LpszCmdline 变量	(268)

第十一章 在 Windows 应用程序中使用声卡	(271)
11.1 Windows 操作系统与声卡	(271)
11.2 安装声卡	(271)
11.2.1 安装声卡的硬件和软件	(271)
11.2.2 声卡的 Windows 驱动程序	(272)
11.3 Hello2 程序	(274)
11.3.1 为 Hello2 程序录制 .WAV 文件	(274)
11.3.2 编译、连接和运行 Hello2 程序	(275)
11.3.3 Hello 程序的文件	(275)
11.3.4 Hello2 的 #include 语句	(285)
11.3.5 Hello2 程序的 Make 文件	(285)
11.3.6 Hello2 程序的 WinMain() 函数	(286)
11.3.7 Hello2 程序的 WndProc() 函数	(287)
11.3.8 打开 Wave 话路	(289)
11.3.9 独立程序	(290)
11.3.10 Hello2 程序的 WM_CHAR case	(290)
11.3.11 Hello2 程序的 WM_COMMAND case	(291)
11.3.12 ts_PlayWave() 函数	(291)
11.4 Record 程序	(293)
11.4.1 编译、连接和运行 Record 程序	(293)
11.4.2 Record 程序的文件	(294)
11.4.3 Record 程序的 WM_CREATE case	(295)
11.4.4 在 WM_COMMAND case 下处理播放请求	(296)
11.4.5 在 WM_COMMAND case 下处理 Record 程序的请求	(297)
11.4.6 在已存在的 WAV 文件中插入新的录制	(299)
11.4.7 停止录制和播放	(299)
11.4.8 录制的取样率和其它 ts_ 函数	(301)
11.5 Record2 程序	(301)
11.5.1 编译、连接和运行 Record2 程序	(301)
11.5.2 Record2 程序的文件	(303)
11.5.3 Record2 程序的 WM_COMMAND case	(303)
11.6 Dog2 程序	(307)
11.6.1 编译、连接和 Dog2 运行程序	(307)
11.6.2 Dog2 程序的文件	(307)
11.6.3 Dog2 程序的动画制作	(309)
11.7 建立一个异步显示的过程	(309)
11.7.1 预置计时器	(309)
11.7.2 预置画面帧	(309)
11.7.3 显示时播放声音	(312)
11.7.4 动画制作	(313)
11.7.5 结束显示	(313)
11.8 PressAny 程序	(314)

11.8.1 编译、连接和运行 PressAny 程序	(314)
11.8.2 PressAny 程序的文件	(316)
11.8.3 PressAny 程序的显示	(316)
11.9 播放乐器数字化接口(MIDI)文件	(319)
11.9.1 PlayMIDI 程序的文件	(319)
11.9.2 编译、连接和运行 PlayMIDI 程序.....	(320)
11.9.3 PlayMIDI 程序的代码	(320)
11.9.4 其它 ts_ MIDI 函数	(321)
11.9.5 合成 WAV 文件和 MIDI 文件	(321)

第十二章 DOS 下的声音 C 语言编程..... (323)

12.1 编写 DOS 下通过 PC 机内置扬声器播放声音的 C 语言程序	(323)
12.2 MAKEexe.bat 文件	(323)
12.3 PlayTs.c 程序	(324)
12.3.1 生成 TS 声音文件.....	(324)
12.3.2 编译、连接 PlayTS 程序	(324)
12.3.3 执行 PlayTS.exe 程序	(325)
12.3.4 声音文件命名.....	(325)
12.4 PlayTS 程序的 C 代码	(325)
12.4.1 # include 文件	(328)
12.4.2 从 DOS 命令行拾取声音文件名	(328)
12.4.3 打开一个声音话路(session)	(328)
12.4.4 清键盘缓冲区	(329)
12.4.5 播放声音文件	(330)
12.4.6 循环播放	(330)
12.4.7 中止 while 循环.....	(331)
12.4.8 标准 C 函数 _disable()与 _enable()	(331)
12.4.9 其它 sp_play()函数.....	(331)
12.5 内存管理与虚拟内存	(332)
12.6 PlayS 程序	(332)
12.6.1 编译、连接和执行 PlayS 程序	(332)
12.6.2 PlayS.C 的代码	(333)
12.6.3 使用 S 文件	(333)
12.7 播放组的大小	(333)
12.8 独立程序.....	(334)
12.8.1 把 PlayTS 程序转换为独立的声音程序	(334)
12.8.2 执行独立程序 NiceDay	(336)
12.9 转换其它程序	(336)
12.10 Tegosoft 声音库的附加 sp_ 函数	(336)
12.11 Info4TS 程序.....	(336)
12.11.1 编译,连接和执行 Info4TS 程序	(337)
12.12 Info4TS.c 程序的 C 代码	(337)

12.12.1	禁止 Ctrl-C	(337)
12.12.2	从 TS 文件中提取信息	(338)
12.12.3	显示播放的当前位置	(338)
12.13	Info4S 程序	(339)
12.13.1	编译、连接和执行 Info4S 程序	(339)
12.13.2	sp_get_ 函数族用于其它类型的声音文件(.WAV、.VOL、.SND)	(339)
12.14	改变播放速度	(340)
12.15	与声音同步移动文本	(340)
12.15.1	编译、连接和执行 SayPress 程序	(340)
12.15.2	SayPress 程序的代码	(341)
12.15.3	文字到语言	(341)
12.16	动画、图形和播放	(341)
第十三章	DOS 下的声卡(sound Blaster Card)	(343)
13.1	执行 PlayVOC 程序	(343)
13.2	用 Borland C 编译器编译和连接 PlayVOC	(343)
13.3	用 Microsoft C 编译器编译和连接 PlayVOC	(343)
13.4	PlayVOC 代码	(344)
13.4.1	装载和初始化 Sound Blaster Driver 程序	(344)
13.4.2	将 VOC 文件装入内存	(346)
13.4.3	播放声音文件	(346)
13.4.4	停止播放	(347)
13.4.5	释放 VOC 缓冲区和声卡	(347)
13.5	其它声霸卡函数	(348)
13.6	用声霸卡在 DOS 下建立多媒体程序	(350)
附录	声音动态链接库 DLL	(351)
F1.1	使用 DLL 声音函数的优点	(351)
F1.1.1	Visual Basic 和其他 Windows 程序设计语言使用 DLL 声音库	(351)
F1.2	使用 DLL 声音库的缺点	(351)
F1.2.1	使用 DLL 声音库	(351)
F1.2.2	使用 DLL4snd.DLL 动态连接库(DLL)	(351)

第一章 声音程序设计

你想你的 PC 机会说话和播放音乐吗？如果是这样的话，本书正适合你。

1.1 PC 机说话与播放音乐的概念

现在，PC 机的价格越来越便宜、功能越来越强，应用更广泛。它们已进入各行各业和许多家庭。对 PC 机来说软件是必不可少的。PC 机软件已变得功能更强，更有竞争力且价格更便宜。例如，几年前优秀的字处理软件包为数很少，但是今天能够以远低于几年前的价格买到许多优秀的文字处理软件包。

因为 PC 机速度的提高与功能的增强，用户希望其软件能够完成更多的任务并且充分利用 PC 机的功能，MS-Windows 操作系统就是软硬件发展的产物。在拥有高速的 80386/486 处理器，足够的 RAM，足够的硬驱，一个 VGA 显示器或其它的高分辨率显示器的 PC 机上，这种操作系统工作性能最好。

程序员现在面临着一个新的挑战，即编出能够满足用户的需求且吸引更多用户的程序。几年前，PC 机用户期望具备操作功能强大的软件包的能力，而程序员也常常责备终端用户不知如何应用这些软件。

随着 Windows 的引入，现在界面已经确定了。有经验的用户都知道如何使用 Windows 操作系统。用户知道如何通过双击程序图标启动程序，如何运行程序，如何把文本和图形对象从一个程序拷贝、粘贴、剪切到另一个程序，如何使用程序菜单，如何执行其它的普通 Windows 任务。

作为一个程序员，你面临着一个挑战，即写出一种懂得这些基本操作的用户能够毫不费力地运行的程序。

1.1.1 适合人视听感的程序

除了易于操作外，你的程序对用户来说必须具有吸引力并且使用起来有乐趣。为此，你的程序应该能够具有吸引人的感觉，即视觉与听觉。

为了让程序吸引人的视觉，应用程序应该工作于图形方式并具有色彩。在可能的情况下，程序应当包括动画。（例如，当用户双击一个 Windows 应用程序中的按钮时应显示该按钮被按动。）

为了让程序吸引人的听觉，应使应用程序说话并播放动听的音乐。几年前，几乎所有的程序使用内置扬声器，它会发出蜂鸣声，以提醒用户程序运行过程中的某些情况或事件。现在，一个好的程序应当对用户说话，用人的语音告诉用户报警的原因，而不是发出令人讨厌的蜂鸣声。

使 PC 机发出人的语音有两种办法，通过使用声卡或使用每台 PC 机中都有的内置扬声器。

1.2 PC 机有无附加硬件时发音的概念

个人计算机(PC机)是一种设计来用以完成任何想象得到的任务的运算机器。然而,有一项功能是设计PC机时不曾考虑的,即产生真正的人声讲话和播放动听的音乐。为了使PC机具备这种功能,人们发明了声卡。它插在PC机母板上的一个扩展槽中,包括用以连接外部麦克风和扬声器的插孔。你的程序对声卡发出指令,“告诉”它去播放声音文件。

声卡是一种使程序功能更加完善的外部设备。它需要麦克风、外部扬声器等装置。然而,事实上并非所有的PC机都配有声卡。

通过这本书,你将学会如何编写能使PC机通过声卡或者是内置扬声器发声的C语言程序,如何编写使PC机发声的DOS应用程序和Windows应用程序。PC机播放的声音文件发出的是真正的人的语音(而非合成的声音),PC机播放的音乐是动听的音乐。你还将学习在程序中如何检测PC机中是否配备声卡,如何把声音导向声卡或内置扬声器。当程序发现PC机没有配备声卡时,它将通过内置扬声器播放。这个小扬声器(所有PC机都配有)可以发出真正的人的语音与动听的音乐。以上概念可用图1.1来表示。

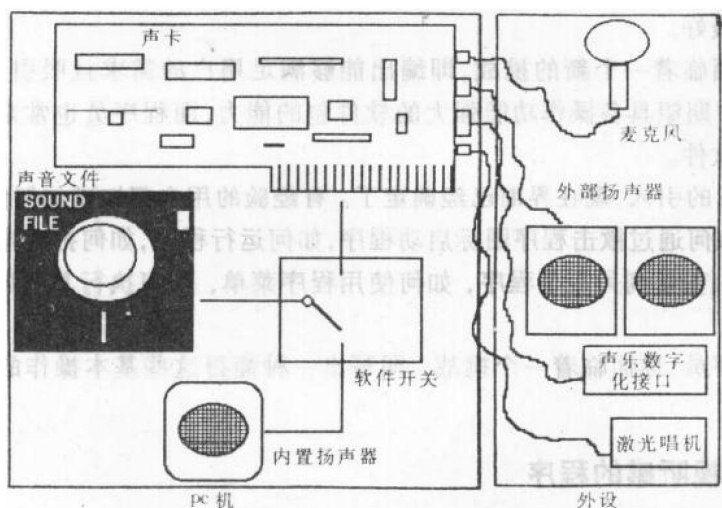


图 1.1 用 PC 机内置扬声器取代声卡播放的概念

一旦你学会了如何在你的程序中加入声音,你将再也不会编写没有声音的程序了。作为程序员,你可能早已会应用PC机的扬声器在程序运行中发出一些简单的蜂鸣声。从现在起,你将用语音代替那讨厌的蜂鸣声。

1.3 安装本书附带的磁盘

为了编写可以录制和播放声音文件的C语言程序,需要多种软件。本书附带的磁盘包含了应用程序可以调用的C语言函数库。为了调用C语言库函数,必须用C语言编程,然后使用Microsoft C编译器或Borland C编译器进行编译连接。

1.4 TSENGINE

本书附带的磁盘中的 C 语言库函数是标准的 C 语言函数, DOS 或 Windows 下的 C 语言程序均可调用,这样,将声音加入程序就很容易了。这个功能强大的 C 语言函数库被称为 TSEngine。

1.5 本书附带的磁盘

本书附带的磁盘中包含以下各项的 C 源码:

- 程序
- 应用软件
- 声音文件
- 用于 Microsoft C 编译器创建 Windows 应用程序的 C 函数库
- 用于 Borland C 编译器创建 Windows 应用程序的 C 函数库
- 用于 Microsoft C 编译器创建 DOS 应用程序的 C 函数库
- 用于 Borland C 编译器创建 DOS 应用程序的 C 函数库
- 其它应用软件

这个软件是 Tegosoftware 公司的 TS Sound Plus 库的简版。虽然是简版它提供的软件能够编译连接和执行本书中所有的程序和自己编写类似的程序。

1.5.1 安装本书的磁盘

这本书的软件是以压缩格式存储的。必须首先安装在硬盘上。请按下面的步骤将该软件安装在硬盘上:

1. 在 DOS 提示符下,设置缺省驱动器为插入安装盘的驱动器。例如,如果软盘插在驱动器 A 中,键入 A:后,按回车键。
2. 键入 INSTALL,然后回车。

这样将创建一个名为 C:\SPSDK 的目录,并在该目录下装入所有的文件。硬盘至少要有 4.6M 的自由空间。

注:文件必须安装在硬盘 C 中,这个驱动器号在某些程序中是固定的。如果你想改变,只需更改硬盘盘符即可。

文件装入硬盘后,自动建立了几个子目录。请务必阅读子目录 LICENSE 中的软件许可协议。

这本书的正文和磁盘的内容是一致的。但是,也可能会发现书中出现的代码与磁盘中的

代码有一些细微差异,可能是最后一刻作了修改的缘故,但差别很细微。例如,磁盘可能会含有不同的程序图标。无论如何,所有书中的程序都已经过编译连接与测试。因此,应该把磁盘中的代码作为最佳代码。

正如前面所说,你应该能够自己编译、连接书中的所有程序。然而,磁盘也包括已经编译连接过的 Windows 程序,这样你可以立即执行书中的程序并且去“听”它们。这样,你会更好地了解你可从书中学到什么。下一章将讲述如何运行这些程序。

本书中采用易学、循序渐进的方式解释程序。请放松一下,准备去一趟轻松的“旅行”。