Windows 下的 MCI 编程实验报告

姚苏航 PB22061220

2025 年 12 月 13 日

1 实验目的

1. 熟悉 Windows 下声卡编程的基本过程与原理，掌握 Media Control Interface（MCI）编程的基本方法；
2. 了解 sndPlaySound 和 PlaySound 等播放波形文件的 API，并能在程序中调用这些函数；
3. 学会使用高级 MCI 函数播放 MIDI、录制声音并保存到文件；
4. 理解音频播放与录制流程，能够分析并解答实验指导书中提出的思考题。

2 实验原理

2.1 MCI 接口简介

媒体控制接口（MCI）是 Microsoft 在早期 Windows 平台上提供的高层多媒体设备控制机制。MCI 与具体设备驱动配合工作，为程序提供了设备无关的控制功能 [1]. 通过统一的 `open`、`play`、`record`、`pause` 与 `stop` 命令，MCI 可以控制多种设备，如 CD Audio、MIDI sequencer、wave audio 等。由于采用统一命令集，开发者无需关心底层设备差异，即可实现播放或录制音频/视频资源的功能 [1]. MCI 提供的设备类型包括波形音频（`waveaudio`）、MIDI 序列器（`sequencer`）、CD Audio（`cdaudio`）等，可扩展性较好。

2.2 播放波形文件的几种方法

Windows 平台提供了多种播放波形（WAV）文件的方法：

- **`sndPlaySound`：**此函数可以播放指定文件或系统注册表中的声音条目，功能较简单，是较早提供的 API。它只能播放外部文件或系统声音，不支持从资源或内存播放，主要为了向后兼容 [2].
- **`PlaySound`：**功能比 sndPlaySound 更丰富。除了支持文件名，还可以通过传入模块句柄和资源标识符播放嵌入在资源中的声音，能够设置同步/异步播放、循环播放等标志，因此是推荐使用的 API[2].
- **MCI 播放：**通过组合 mciSendCommand 或 mciSendString，可以打开 wave 设备、设置播放参数并开始播放。这种方法在控制上更灵活，可以查询状态、控制定位等，但代码相对复杂。

- **底层 waveOut 接口**：利用 waveOutOpen、waveOutPrepareHeader 等低级函数，可完全控制缓冲区管理和数据流，适合编写专业音频应用。然而编程复杂，通常用于需要实时控制或特殊音频处理的场合。

2.3 指针类型问题：DWORD 与 DWORD_PTR

`mciSendCommand` 函数的第四个参数用于传递指向 MCI 参数结构的指针。早期示例代码在 32 位 Windows 中直接将指针强制转换为 32 位 DWORD，在 64 位 Windows 下会导致访问冲突。`DWORD_PTR` 是指针精度的无符号长整型，用于执行指针算术或在 32 位参数扩展为 64 位时保持精度 [3]. 因此，在 64 位环境下应使用 DWORD_PTR 来保存指针的整数值。例如：

```
MCI_OPEN_PARMS mciopen;
// 错误：在 64 位下直接转换为 DWORD 会截断指针
// mciSendCommand(0, MCI_OPEN, MCI_OPEN_TYPE|MCI_OPEN_ELEMENT,
//                (DWORD)(LPVOID)&mciopen);
// 正确：使用 DWORD_PTR 类型
mciSendCommand(0, MCI_OPEN, MCI_OPEN_TYPE|MCI_OPEN_ELEMENT,
               (DWORD_PTR)&mciopen);
```

上述修改可以避免在 Windows 11 等 64 位系统中出现访问冲突。

2.4 MediaPlayer 与 MCI 的区别

Microsoft 在 Windows 10/11 引入了新的 *MediaPlayer* API，用于替代旧的 MCI。与 MCI 相比，MediaPlayer 封装更现代，支持播放本地文件、网络流媒体和 DRM 受保护内容，同时提供对音频焦点管理、媒体会话、播放列表、元数据和后台播放的统一支持。MediaPlayer 提供基于事件的异步模型，简化了多媒体开发，也更好地支持 UWP 和 WinUI 应用；而 MCI 主要针对早期 Win32 应用，功能有限，已被微软标记为遗留接口。

3 实验内容及结果

3.1 环境配置

本实验在 Windows 11 64 位系统上进行，使用 Visual Studio 2022 Community（版本 17.x）作为开发环境。项目基于 MFC 对话框模板创建，需勾选“使用 Unicode 字符集”，并在 `MciTestDlg.cpp` 中包含头文件 MMSYSTEM.H 以使用多媒体函数。声卡正常工作，实验中使用了系统自带的 `C:\Windows\Media\chimes.wav` 和 `onestop.mid` 作为示例音频，录音文件保存到 `e:\tmp\eeis.wav`。

3.2 补全示例程序

根据实验指导书，需要在示例代码中完成以下修改：

1. **修改注册表项字符串**。在 CMciTestApp::InitInstance 中调用 SetRegistryKey 时使用有意义的公司或组织名，避免默认字符串。本文将其修改为 ”USTC Multimedia Lab”。
2. **使用 `DWORD_PTR` 传递参数结构指针**。所有调用 mciSendCommand 的地方均改用 DWORD_PTR 传递参数结构体指针，以解决 64 位系统上的访问冲突。

3. **释放资源。**在主对话框关闭（按下 OK 或 Cancel）时，显式调用 mciSendCommand(MCI_DEVTYPE_WAVEF 和 mciSendCommand(MCI_DEVTYPE_SEQUENCER, MCI_CLOSE, 0, NULL) 关闭设备，保证程序退出前释放音频资源。

4. **路径调整。**播放 WAV 和 MIDI 时，修改路径为系统存在的文件；录音前先复制一个占位 WAV 文件到目标路径。

图 1 给出了完成后的程序界面示意图

图 1: 完成后的 MCI 测试程序界面示意图

3.3 调试与运行

在 Visual Studio 中编译并运行修改后的程序，按下“PlaySound”按钮可以同步播放系统自带的 `Ring01.wav`(原设置无法使用，调整为本机的 wav 资源)，按下“sndPlaySound”按钮同样可以播放同一文件但调用的是旧版 API。按下“Play”按钮将通过 MCI 打开 MIDI 设备并播放 `onestop.mid`；按下“Stop Play”可发送 `MCI_STOP` 命令终止播放。点击“Record”后程序通过 MCI 打开波形设备并开始录音，随后点击“Stop Record”保存录音到项目路径下的 `.\tmp\Classic.wav`。

程序测试过程中，首先确认音频文件路径正确，否则会收到 MCI 错误提示。利用 `DWORD_PTR` 传递参数结构指针后，在 Windows 11 64 位环境下不再出现访问冲突。录制的 `Classic.wav` 可通过系统播放器正常回放。

4 思考题

1. **什么是 MCI 接口？MCI 所囊括的内容有哪些？**MCI（Media Control Interface）是 Windows 的高层多媒体控制接口，为应用程序提供设备无关的播放和录制命令 [1]. 它通过调用 MCI 设备驱动程序解释并执行命令，支持播放波形音频、MIDI 文件、CD 音频、视频光盘等多种设备。MCI 的命令集分为系统命令（如 `sysinfo`）、必需命令（`open`、`close`、`info`、`status` 等）、基本命令（`play`、`record`、`pause`、`stop`、`save` 等）以及针对具体设备的扩展命令。设备类型包括 `waveaudio`、`sequencer`、`cdaudio`、`digitalvideo` 等 [1]。

2. **Windows 下播放波形文件可以采用几种方法，各有什么优缺点？**一是调用 sndPlaySound，语法简单但仅能播放外部文件或系统声音条目，是早期接口；二是调用 PlaySound，支持从文件、资源或内存播放，并可设置同步/异步播放及循环标志，因此应用更广 [2]. 三是使用 MCI 高级命令（ mciSendCommand/ mciSendString），可以查询播放状态、定位播放进度及控制设备，但编程较为复杂。四是直接调用底层 waveOut 系列函数，实现最灵活的缓冲区控制，适合专业应用，但开发工作量最大。

3. **在打开设备时为什么要使用 `DWORD_PTR` 类型指针？**在 32 位系统中，指针大小与 `DWORD` 相同，因此早期示例代码通过 (DWORD)(LPVOID)&mciopen 将指针强制转换为 32 位整数。但在 64 位系统中，指针宽度为 64 位，强制转换为 `DWORD` 会截断高位，导致 `mciSendCommand` 读取错误地址而引发访问冲突。`DWORD_PTR` 是为指针精度定义的无符号整型，用于在 32/64 位环境下安全地存储指针值 [3]. 因此将指针转换为 `DWORD_PTR` 可以在不同平台上正确传递参数结构体指针。

4. **MediaPlayer 具备哪些功能？**MediaPlayer 是微软在 Windows 10/11 中推荐的多媒体播放组件，取代了 MCI。它支持播放本地和网络媒体文件，能够处理音频与视频流，并提供播放列表、媒体会话与后台播放控制等高级功能。MediaPlayer API 支持异步调用和事件回调，集成了对系统媒体控制、音频焦点管理、字幕和元数据的处理，适合现代应用开发。相比之下，MCI 功能单一、接口老旧，不支持流媒体或复杂的会话管理。

参考文献

[1] Techs Helps. *Media Control Interface*. 可从 `https://techshelps.github.io/MSDN/DNWIN95/HTML/S714A.HTM` 获得。该文章指出 MCI 提供设备无关的命令集并列出支持的设备类型。

[2] Stefan Trost. “Delphi: Difference between PlaySound and SNDPlaySound.” `https://www.askingbox.com/question/delphi-difference-between-playsound-and-sndplaysound` 文中说明 `sndPlaySound` 是旧版本函数，而 `PlaySound` 提供了播放资源的扩展功能。

[3] Microsoft. “DWORD_PTR - Windows Data Types.” `https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-dtyp/66d61dd8-2191-4a37-b963-e49bf0dc2579`。指出 `DWORD_PTR` 用于指针精度，适用于 32/64 位之间扩展的参数。