# 屏幕文本的语音合成

杨鸿武[1]，蔡莲红[2]，陶建华[2]
(1.西北师范大学 物理与电子工程学院,甘肃 兰州 730070;
2.清华大学 计算机系媒体所,北京 100084)

摘要:本文介绍了 TTS 系统的原理和 Win32 API 截获技术的实现方法，并利用 Win32 API 截获技术和清华大学 SinoSonic 系统实现了一个桌面文本的语音输出系统，可以合成桌面上鼠标指针下的任意文本。
关键词： TTS；API 截获技术；

## 1. 引 言

计算机语音合成系统又称文语转换系统（TTS 系统），它的主要功能是将计算机中任意出现的文字，转换成自然流畅的语音输出。目前，语音合成系统已经较为成熟并已大量应用在不同场合。一般认为，TTS 系统采用的技术有基于参数合成和基于波形拼接合成两种。基于波形拼接的语音合成系统包括三个主要组成部份：文本分析模块、韵律生成模块和声学模块。文本分析模块主要处理输入文本，获得文本的语境参数，产生文本的语音学表示。韵律生成模块决定最终系统能够用来进行声信号合成的具体韵律参数。声学模块则根据文本分析的结果从语音数据库中选出相应的语音基元，由韵律生成模块修正后，产生合成语音的输出。其系统结构如图 1 所示。

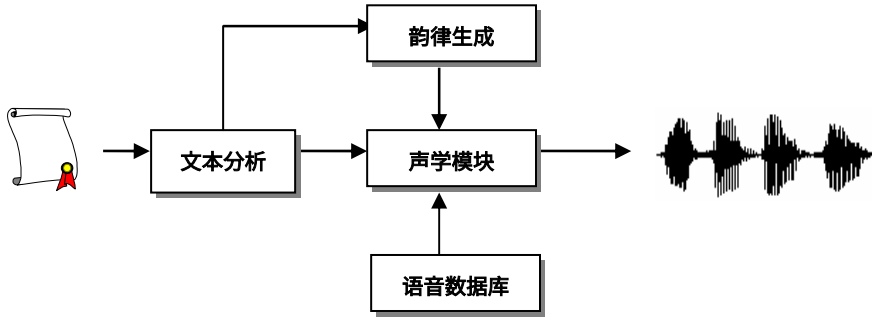



图 1 TTS 系统框图

API 截获技术是指在程序运行期间，用一条无条件分支语句来替换 Windows 系统要调用的 Win32 API 函数的前几条指令，使得 Windows 系统对原 API 函数的调用转向到用户自己的函数，从而改变原 API 函数的功能。也就是说，利用 API 截获技术，我们可以用一个新的函数部分或全部替换 Windows API 函数，从而在不改动 Windows 操作系统的情况下，增强或改变 Windows 操作系统的部分功能。由于对 API 函数的修改是程序运行时在内存中动态完成的，所以我们可以在某个特定的应用程序中截获 API 而不影响其它应用程序对该 API 函数的调用。在 Windows 系统中，屏幕文本的输出是由 GUI 中的几个文本输出 API 函数来完成的，所以，我们通过截获这几个 API 函数，就可以获得 Windows 系统向屏幕输出的文本内容。将这些文本内容取出来后，通过语音合成系统，就可以将这些文本转换成自然流畅的语音输出。这样，我们就可以让计算机读出屏幕上的任意文本，使人机交互变得简单。

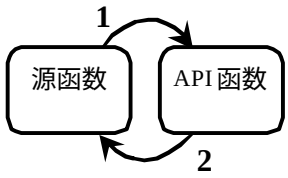
## 2. API 函数的截获方法

要截获 Win32 API 函数，首先取得 API 函数的地址，然后将 API 函数的前 5 个字节用一条 JMP 指令替换（JMP 指令需 5 个字节），JMP 指令跳转到用户定义的截获函数。API 函数的前 5 个字节指令则保存在一个跳转表中。跳转表由 API 函数中移出的前 5 个字节指令和一条

JMP 指令组成，该 JMP 指令跳转到 API 函数的剩余部分。

在没有截获时，Windows 系统直接调用 API 函数。当截获了 API 函数后，由于 API 函数的前 5 个字节已经被 JMP 指令替换，所以当 Windows 系统调用 API 函数时，JMP 指令直接将控制转移到用户定义的截获函数。用户定义的截获函数完成操作后跳转到跳转表，然后由跳转表中的 JMP 指令完成对 API 函数的调用。这时候，API 函数相当于截获函数的一个子函数。当 API 函数完成后，它将控制交还给用户定义的截获函数，用户定义的截获函数接着进行其它操作，最后将控制交还给系统源函数。图 2 显示了在截获和不截获情况下的控制转移流程。

*不截获调用：*




*截获调用：*

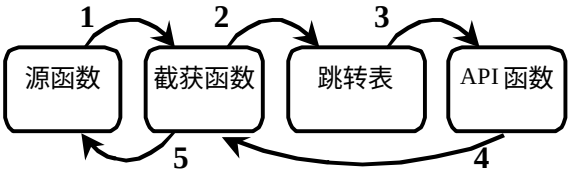



图 2. 在截获和不截获情况下的函数调用流程。

整个截获过程是通过重写 API 函数的进程间二进制映像来实现的。对每一个 API 函数，截获时重写了 API 函数的前 5 个字节和相应的跳转表。图 3 显示了截获前和截获后的 API 函数和跳转表的内存映象。

截获前：

```
;; API 函数
APIFunction:
  push  ebp
  mov   ebp,esp
  push  ebx
  push  esi
  push  edi
;; 跳转表
Trampoline:
  jmp  APIFunction
          …
```

截获后：

```
;; API 函数
APIFunction:
  jmp   截获函数

APIFunction+5:
  push  edi
           …
;; 跳转表
Trampoline:
  push  ebp
  mov   ebp,esp
  push  ebx
  push  esi
  jmp   APIFunction+5
           …
```

图 3. 截获前和截获后的跳转表和 API 函数

为了让截获函数能够注入到应用程序的地址空间，只需将截获函数写到一个动态链接库中，然后将应用程序和动态链接接库一起编译即可。截获时，首先获得 API 函数的地址，这一步可通过调用 GetProcessAddress()来完成。接着为跳转表分配内存，用 GetCurrentProcess()取得当前进程的地址，并且改变进程的虚拟地址空间的页保护属性，容许 Windows 重写 API 函数和跳转表。接下来，将 API 函数的前 5 个字节保存到跳转表中，并将这些字节改写成一条 JMP 指令，其分支地址为截获函数的入口地址。跳转表的前 5 个字节存 API 函数的前 5 个字节的内容，并再跟一条 JMP 指令，其分支地址为 API 函数的第 6 个字节。JMP 指令的生成、插入 API 函数以及指令的拷贝通过查一个反汇编表来完成。当指令的插入和拷贝完成后，调用 FlushInstructionCache()来实现指令缓存的更新。

### 3. 屏幕文本的获取

Win32 系统在屏幕上输出文字时，用到 GDI32.dll 中的以下几个函数：TextOutA、TextOutW、ExtTextOutA、ExtTextOutW、BitBlt。所以，利用 API 截获技术截获了以上几个函数，就能够得到 Windows 系统发给这几个函数的参数，从而获取屏幕上的文本输出。

用“鼠标钩子”或“定时器”得到鼠标的位置，如果鼠标移动了，那么在鼠标位置下放置一个很小的窗口，Windows 系统会发出 WM_PAINT 消息，指示桌面应用程序重绘屏幕，在应用程序响应 WM_PAINT 时，会调用 TextOut(),ExtTextOut()等 API 函数来绘制文本，如果我们在应用程序的堆栈中截获 TextOut(),ExtTextOut()的参数，就能获取屏幕上的文本。

具体来说，首先，我们仿照 TextOutA、TextOutW、ExtTextOutA、ExtTextOutW、BitBlt 写出相应的 5 个截获函数，分别截获以上 5 个 API 函数。截获函数的参数和 API 函数完全一样。截获函数执行的操作是，先根据系统传进来的参数取出屏幕输出文本，然后向应用程序发出文本获取完成消息,最后利用跳转表调用 API 函数,使 Windows 能够在屏幕上绘出文本。其次，用 SetWindowsHookex（）函数安装鼠标钩子，或用 SetTimer()安装计时器，在鼠标钩子过程或计时器过程中，得到鼠标的当前位置，判断鼠标是否移动，如果鼠标位置改变了，则在鼠标指针下放置一个小窗口，以便让 Windows 系统发出 WM_PAINT 消息，这时，截获函数会被调用，我们就可以取出鼠标指针底下的屏幕文本。第三，在应用程序中，响应文本获取完成消息，从内部缓存中取出获取的文本，并交给合成语音系统进行合成。

### 4. 截获文本的语音合成

由于中英文读法不同，所以在语音合成时，对中文文本和英文文本要用不同的合成系统来合成，对于符号，还要确定符号的特殊读法（如数字、日期的读法）。应用程序截获的文本，有可能即有中文，又有英文，还可能包含一些符号。在合成时，首先将截获文本中不发音的符号去掉，然后将中文、英文和符号区分开。对于中文，由于发音的基本单元为音节，所以以音节为单位进行分析，获取音节的位置、属性等语境信息。对于英文，其基本发音单元为双音子，所以以双音子为单位进行分析，获取其语境信息。对于符号，则要根据符号在句子中的位置以及符号前后字、词的属性决定其特殊读法。发音基本单元的语境信息获得后，利用语境参数在语音数据库中选取基本发音单元的波形数据，用 PSOLA 技术拼接后，再进行韵律修正，最后将结果以 WAV 格式缓存，并用 Windows 的 PlaySound（）函数播放。

### 5. 结论

利用 Win32 API 函数截获技术和屏幕文本的获取技术，我们在清华大学 SinoSonic 语音合成系统的基础上，开发了一个屏幕文本的语音合成程序，该程序能将桌面上的文本以合成语音的方式读出来。该程序可在 Windows 2000/NT、Windows 98/95 下运行。

参考文献

[1]蔡莲红.语音合成系统综述及其应用[J].计算机世界,2000,(3):20

[2]陶建华，蔡莲红.计算机语音合成的关键技术及展望[J].计算机世界,2000,(3):20

[3] Detours: Binary Interception of Win32 Functions, Microsoft Company ［EB/OL］. http://research.microsoft.com/Detours, 1999, 12

[4]白瑜.鼠标屏幕取词技术的原理和实现［EB/OL］.

http://www.eschool.com.cn/document/20010102/2001010211331501.shtml, 2001,1

**A Synthesis System of Screen Text to Speech**

YANG Hong-wu[1] TAO Jian-hua[2] CAI Lian-hong[2]

(1.School of Physics and Electronics Engineering, Northwest Normal University, Lanzhou,730070, china;2.Department of Computer Science & Technology, Tsinghua University,Beijing,100084,china)

**Abstract:** The theory of TTS system and technology of intercepting Win32 functions by re-writing target function images are introduced in this paper. A Synthesis System of Screen Text to Speech also realized by using technology of intercepting Win 32 functions under SinoSonic System in the paper.

Key words: TTS; API intercepting technology

作者简介：杨鸿武（1969-）、男、甘肃合作人、讲师、硕士、从事计算机语音合成方面的研究。