

计算机电话界面设计和实现

曾晓勤 (河海大学)

摘要:本文讨论了利用电话来构造计算机系统语音界面的技术。

一、引言

长期以来,人们对计算机系统的访问都要在计算机或计算机终端上才能进行。近年来,随着计算机与通信技术的迅速发展,有许多计算机通信产品问世,电话语音卡就是一例。该卡能把电话一端的语音信号数字化后送给计算机,还能将计算机中数字化的语音数据还原成语音信号送电话播放。电话语音卡在电话与计算机系统之间架起了一座桥梁,使人通过电话与计算机系统通信成为可能。

二、电话语音卡的功能

目前,商品化的电话语音卡种类已有许多。卡按可接电话线的多少而有线数之分,线又有内线(接电话机)或外线(接电信局交换机)之别。各种卡的功能大体相近,提供给用户的接口通常是 C 语言函数接口和汇编语言函数接口。下面结合 C 语言的接口函数给出电话语音卡常见的功能:

1. 铃流检测 int RingDetect(int LineNum)

检测某一线有电话打入时的振铃信号。

2. 摘机 Void PickUp (int LineNum)

模拟电话机的摘机动作,接通某一线电话。

3. 放音 int PlayData (char * DataPtr, int DataLen, int LineNum)

将指定内存区域中数字化语音数据播放到某一线电话上。

int PlayFile (char * FileName, int LineNum)

将指定的数字化语音文件播放到某一线电话上。

int CheckPlayEnd (int LineNum)

检测某一线放音是否结束。

int StopPlay (int LineNum)

强行停止某一线的放音操作。

4. 录音 int RecordData (char * DataPtr, int DataLen, int LineNum)

将某一线电话语音录制成数字化语音数据存放在指定的内存区域中。

int RecordFile (char * FileName, int LineNum)

将某一线电话语音录制成指定名的数字化语音数据文件。

int CheckRecordEnd (int LineNum)

检测某一线电话录音是否结束。

int StopRecord (int LineNum)

强行停止某一线的录音操作。

5. 接收双音频(DTMF)电话键码

int GetDTMFCode (char * DTMFPtr, int DTMFLen, int LineNum)

将某一线电话接通后由话机键拨入的 DTMF 码数据存放在指定的内存区域中。

int CheckGetDTMFEnd (int LineNum)

检测某一线接收 DTMF 码是否结束。

int StopGetDTMF (int LineNum)

强行停止某一线接收 DTMF 码。

6. 发送双音频(DTMF)电话键码

int DialPhone (char * DTMFPtr, int DTMFLen, int LineNum)

将指定内存区域的 DTMF 码从某一线发出,即自动拨号向外打电话。

7. 信令检测 int CheckSignal (int LineNum)

检测某一线电话线路返回的信号状态,如拨号音、忙音等。

8. 挂机 Void HangUp (int LineNum)

模拟电话的挂机动作,挂断某一线电话。

除此以外,还有些其他功能,有的卡甚至还具有传真的功能,这里就不一一罗列了。

三、计算机电话界面的特点

电话是用语音交流信息的工具。电话语音卡的上述功能,使得计算机能够“说话”,能够“听”到电话里的讲话。由于现有的计算机语音产品已经能将汉字编码或字符编码转换成相应汉字或字符的语音数据,所以把数据库中的数值、字符、和文本数据用语音播放到电话上已是可行的,可以做到无限词汇的语音输出。但是由于计算机语音识别技术还没有很好解决,计算机虽然能“听”到语音,却“听”不懂语义,人用语音还难以随意地操纵计算机。好在电话语音卡还能够使计算机接收到按电话键送来的双音频码,并且计算机能够识别双音频码,人可以通过按电话键来向计算机传递语义信息。因此现阶段人机用电话交互还不能完全语音化,只能半语音化,它的特点是:人按电话键,计算机“讲话”。显然,人靠按电话键来向计算机传递信息是很不方便的。尽管如此,能远离计算机用电话查询到计算机数据库中的信息,是十分有用并且是非常需要的。适合这样特点的应用领域有:

公共信息查询, 银行、证券、邮电等部门的信息服务, 政府机关和事业单位的办公自动化等。

四、计算机电话界面

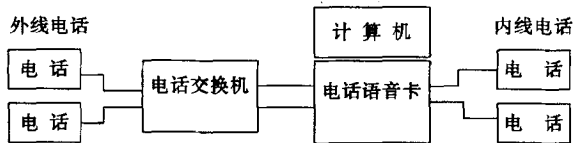


图1 硬件组成结构(4线卡, 2内线, 2外线)

1. 电话界面的硬件构成

电话界面指的是实现人用电话与计算机系统通信规程的软件, 但是它必须要有相应硬件的支持。将电话语音卡插入计算机任一扩展槽中, 接上电话线就可使电话与计算机在硬件上互连。该计算机可以是单机系统, 也可以是多机系统中一台机器。

硬件组成结构如图1所示:

2. 电话界面的功能

(1) 语音交互功能。电话界面要利用电话语音卡提供的接口函数, 模拟打电话的过程, 用语音与人交互, 完成电话端来的服务请求。

(2) 实时处理功能。电话界面要能实时地响应电话用户的请求, 当多个用户同时打电话访问计算机数据库时, 要能及时为用户服务。

(3) 安全保障功能。电话是公共的通信工具, 电话界面使得计算机中的数据库面向公众。电话界面必须确保数据库的安全性和保密性。

(4) 主动打电话功能。电话界面可利用电话语音卡的自动拨号功能来建立触发器机制, 即当数据库中的数据使得某个条件成立时, 主动打电话与相关的用户联系。

3. 电话界面的设计

电话界面的半语音交互过程是按一定的应答规程进行的, 也就是电话打通后, 计算机用语音提示人如何按键, 人按电话键回答计算机, 经过反复的问与答, 最后计算机完成人所需的操作。设计时可根据应用领域的需要为电话语音卡的每一线定义一组状态, 规定每一状态应做的动作。同时, 整个系统还要维护一个状态转换图, 该图是个有向无回路图, 它指出各状态之间的次序关系以及转换的条件。任一时刻某一线只可能处于一种状态, 系统根据该线当前的状态。从初始状态演变到结束状态表示一次交互过程的完毕。值得一提的是, 在交互过程中要允许用户在提示语音还未播放完之前就按键回答, 对使用熟练的用户来说, 是有可能的, 这时要立即中断播音并转到交互的下一步去。再就是要防止用户长时间不按键回答或者突然挂机, 这时要结束此次交互过程。

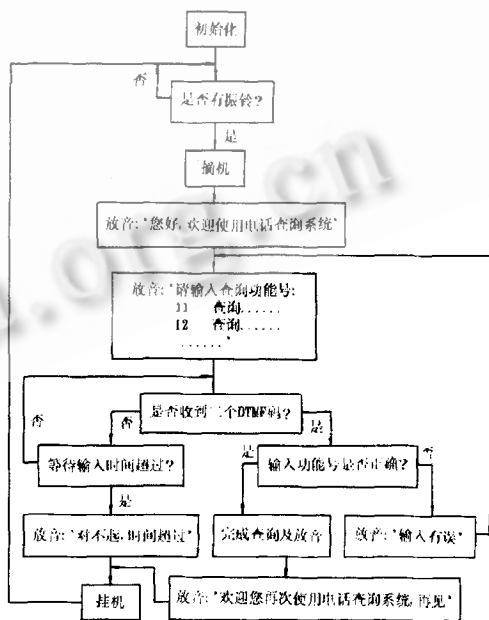
在多任务操作系统环境下, 利用系统的并发控制机制可以实现电话界面的实时处理功能。在单任务操作系统环境下, 就要在设计上做文章。电话语音卡的各线有并行处理功能, 并且卡与 CPU 也可并行执行。卡接口函数中所有函数都不是过程型的, 而是指令型的, 即函数执行后, (将任务交给卡硬件去做), 马上返回。因此在设计时, 只要控制 CPU 轮流为各线服务, 就可以在单任务系统环境下实时为多用户服务。况且人对电话实时性的要求不是很严, 能在几秒钟内响应都可以接受。安全控制方面可用常见的加锁技术。

触发器机制是主动数据库所研究的课题, 目前的数据库管理系统都只能被动地存储数据。但是对于少量简单条件的触发器, 电话界面是可以实现的, 维护这些触发条件, 每当有数据更新时就去判这些条件的真假, 以决定是否主动拨号打电话。

五、一个实例框架的实现

以下的例子是从一个已实现的电话银行查询系统中剪裁出的一个框架, 裁去了权限检查及业务管理等方面的操作。主要是给出电话界面实现的方法, 不同应用领域的用户可根据业务需要在此基础上构造出相应系统的电话界面。

1. 程序流程图(对每一线)



2. 程序清单 (因篇幅较长, 程序清单略, 有兴趣的读者请与作者联系。联系地址: 南京北祖师庵 60 号 701 室, 210015)