电话查询系统软件设计

方路平 （浙江大学生医系 310027）

目前利用电话进行查询的业务已广泛涉及金融业，比如在一些商业银行中出现的“电话银行查询系统”，在证券交易所中的“股票电话委托系统”，信用卡公司的“信用卡查询及挂失系统”，国外有些地区的“电话购物服务”等等，所有的这些服务的一个特点是方便、快捷。我们在利用电话提供服务这一方面，进行了深入的研究，在这一领域积累了一些经验，在此愿与同行交流。下面先简单介绍一下电话服务系统的结构如图 1 所示。

1.**电话服务系统的结构**

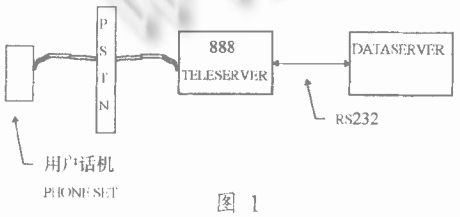

图 1

如框图所示，整个系统由 TELESERVER（电话服务器）和 DATASERVER（数据库服务器）组成，其中 DATASERVER 可以同时是一个计算机网络的组成部分。用户拨通服务中心号码，如 888，电话服务器（TELESERVER）会自动应答用户的服务。用户根据 TELESERVER 的语音提示，在自己的话机（PHONE SET）上选择相应的按键，电话服务器或从数据库服务器得到信息，然后把信息转换成相应的语音播放，用户可在听筒中听到查询结果；或把用户键入的信息对数据库服务器中的数据进行更新或增加。本文主要介绍软件系统的设计，故不准备对硬件设计的细节作讨论。在上述框图中，TELESERVER 和 DATASERAVER 是通过 RS32 串口连接的，这是硬件设计的一种模式，也有设计成插卡方式的情况。而我们认为，这种模式有其两个明显的优点。第一，有益于 DATASERVER 设备的保护。因为电话服务设备与公共电话网相连，来自于电话线中的电气冲击可能会进入电话服务设备。如果是采用插卡形式，很可能危及 DATASERVER 机器的安全，而 DATASERVER 有可能是其他计算机系统的核心，比如是银行数据库的主机，这样就可能造成重大损失。而采用 TELESERVER 和 DATASERVER 分离的模式，则安全得多。第二，DATASERVER 的主机类型是不固定的，若采用插卡式设计，可能会有相当多的局限，而 RS32 标准是一种公共标准，不会出现上述的局限。

针对上面介绍的模式，我们进行软件的整体设计。由于许多重要的商业应用的计算机是基于 UNIX 或 XENIX 操作系统的，故着重介绍基于这种操作系统的应用软件的开发。

2.**软件总体设计**

整个软件分为两个部分。一部分运行于 TELESERVER 上，一般用 98 或 51 编程（取决于 TELESERVER 是 98 或是 51 系统）。另一部分运行于 DATASERAVER 中，用 C 语言编程。根据 DATASERVER 的 DBMS 类型，有时需用 ESQL－C 等语言开发，如 DBMS 系统采用 INFORMIX 或 ORACLE 数据库。

(1)命令/任务驱动模式

这两部分软件均设计成“命令/任务驱动”模式。与 WINDOWS 程序的“消息驱动”的结构有类似之处。即程序运行时一直处于等待状态。一旦收到命令，则执行相应的任务。与 TELESERVER 相关的命令集列于表 1 中。

表 1 命令集

命令	描述
DEVICEINIT	初始化 TELESERVER
HOKOFF	摘机
HOOKON	挂机
RINGDETECT	振铃检测
DTMF	接收双音多频信号
SOUND	模拟语音发声

在 TELESERVER 中，由这六条命令驱动，可以完成任何形式的查询、操作等功能。可以看出这六条命令与任何查询内容无关，故可以不加任何修正而运用于“电话银行”，“信用卡挂失”等应用中。这种灵活性来源于“命

令/任务驱动"的软件结构设计。在与国内同类应用的比较中可以看出这种设计的优秀性能。TELESERVER 中的软件有两点需着重解释:①任务(或称命令)的组合形式与具体的应用是有关的,那么如何做到 TELESERVER 的设计与具体的应用无关呢?答案很简单。任务表存放于 DATASERVER 中,在具体动作中,由 RS232 通信线向 TELESERVER 逐条发出任务。②语音内容是具体的,如何做到 TELESERVER 的软件编写与应用无关?答案同①。语音表由 DATASERVER 在启动时通过 RS232 向 TELESERVER 动态灌注语音对照表及语音数据。解决了①和②这两个问题,TELESERVER 的软硬件不需任何改动,即可应用于"电话银行查询"、股票电话委托"等系统。充分体现了这种软件设计思路的灵活性。

(2)多进程设计

现在开始 DATASERVER 软件设计。这一部分采用的是"命令/任务驱动"模式。任务表由用户自己设定。在运行中,依据任务表驱动 DATASERVER 软件系统。其中,有些命令通过 RS232 下载到前端机驱动 TELESERVER。任务表的生成与处理在下面进行设计,现在先介绍多进程结构的设计。

整个查询系统的设计着眼于多路并发的情况处理。所谓并发,即多个用户同时打电话请求服务,一般分为 2 路/4 路/…64 路等。此项功能在类似"股票电话委托系统"的应用中显得尤为突出。这一需求,是我们最终采用多进程设计的初衷。后来,我们发现多进程的结构给系统带来了更大的灵活性(见多任务驱动)。

利用 FORK()功能,我们共生成了若干进程(PROCESS)及二个管道(PIPELINE)。一为通信进程 COMM-PROCESS,负责从串口读取数据并传入管道 TO-DAD。二为主进程 MAIN-PROCESS,依照任务表执行任务,从 TO-DAD 管道获取通信进程及子进程传递给主进程的数据,同时通过 TO-SON 管道将数据传递给子进程。三为数据库子进程(SON-PROCESS),负责对数据库的操作。因为数据库操作是整个操作的瓶颈部分,为了支持多路并发,故数据库子进程根据需要可复制二个、三个…相同进程。这样对于单个用户而言,几乎感觉不到多用户同时请求服务时系统负担变重的变化。同时,在通信数据包(PACKET)中引入通道号 CHANNEL NO 以区分各通道的数据。

(3)多任务驱动设计

在 DATASERVER 程序启动后与 TELESERVER 联通,接着将语音对照表及数据下载(DOWNLOAD)到 TELESERVER 中,利用生成的进程协调一致地工作。为了支持在一个系统中支持多种查询应用,比如,一个通道处理信用卡方面的业务,一个通道查询银行的帐户资金情况,又有四个通道接受股票委托业务。要完成这样复杂的需求,市场中一般的技术需三套系统来应付不同的服务请求。而我们所设计的一套系统就能完成所有的需求。所作的操作只是在程序启动时调入三套任务表,

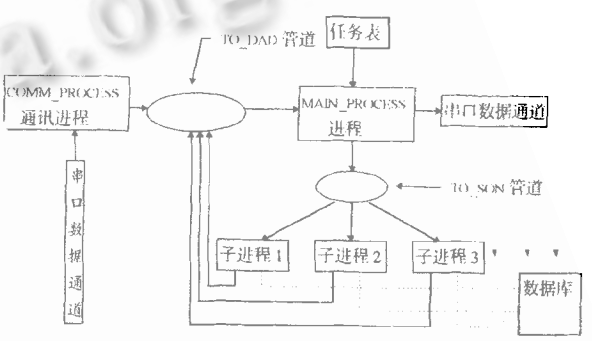

图 2

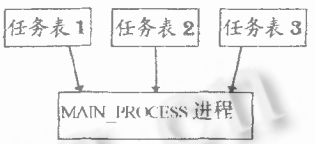

图 3

依次分配给相应的数据库查询子进程。比如说任务表 1 所涉及的数据库操作由 SON-PROCESS1 负责,任务表 2 所涉及的数据库操作由 SON-PROCESS2 负责。

3.任务表语言的设计

为了对任务表有一初步认识,现举例:

DEVINIT
RINGDETECT
HOOKOFF
SAYHELLO
HOOKON

上述例子为一典型的振铃检测后,向用户问候的例子。为了支持"任务驱动"的结构设计,我们故设计了一种类 BASIC 语言的解释器,对任务表进行分析和执行。任务表语言支持的内容包括:

(1)标识符、变量(字符型和数值型)、常量、运算符(+ - ×/括号)及表示式(由数值型变量、常量、运算符组成)

(2)通信命令:HOOKON(挂机)HOOKOFF(摘机)RINGDETECT(振铃检测)DTMF(接收 DTMF 输入)PABX(交换机功能)

(3)字符串操作:STRCAT(字符串连接)STRCPY(字符串拷贝)STRCMP(字符串比较)STRCUT(字符串分割)STRSTR(字符串搜寻)

(4)顺序控制:GOTO 标号 IF 条件 THEN 命令

(5)赋值: =

(6)函数调用:CALL 函数名 WITH 参数表

(7)杂项:DEFINE(宏定义)MACRO(扩展功能)PRINT(屏幕打印) END(流程结束)REM(注释) SAY(发语音)

该语言的设计成功,使得任务表的生成非常容易

(收稿时间:1996 年 8 月)