

文章编号: 1002-0446(2001)03-0261-05

基于 Linux 的远程机器人控制系统研究

薛广涛 陈一民 张 涛

(上海大学 计算机工程与科学学院 200072)

摘 要: 本文对远程机器人控制系统的体系结构和最新进展进行了论述, 提出了一种新的远程机器人控制系统的软件体系结构. 文中就远程机器人的基本原理及远程机器人的状态监控技术进行了详细的叙述, 并详细说明了与这种系统相关的重要技术问题, 最后给出了一个基于 Linux 操作系统的远程机器人控制程序的具体实现例子.

关键词: Linux; 机器人控制器; Socket; 遥操作; 远程机器人技术

中图分类号: TP24 **文献标识码:** B

RESEARCH ON TELE-ROBOTICS CONTROL SYSTEM BASED ON LINUX

XUE Guang-tao CHENG Yi-min ZHANG Tao

(School of Computer Engineering and Science, Shanghai University 201800)

Abstract: This paper discusses the architecture and the up-to-date development of the tele-robotics control system, and introduces a new type of software architecture concerning the tele-robotics control system. In this paper, the basic principles of tele-robotics and the monitoring status technology of tele-robotics are presented in great detail, and some related important techniques are also presented. At the end, an application about the tele-robotics control program based on Linux is given.

Keywords: linux, robot controller, socket, remote control, tele-robotics technology

1 引言 (Introduction)

具有远距离控制功能的基于网络的远程机器人在海洋探测、军事及人类自身不能到达的各种危险和特殊场所具有独到的作用. 在网络环境中运行的远距离控制功能的机器人, 不仅要满足实时性和可靠性的要求, 还要能够在传输实时控制数据的同时进行实时图象的传输, 从而依据三维机器人运行状态图控制机器人的运行. 基于 PC 及 Linux 平台的远程机器人控制系统更是值得研究开发, 但目前尚未见这方面的有关报道.

由于 Linux 本身所具有的开放性、稳定性、高效性和强大的网络功能并且是公开源码的自由软件, 采用运行 Linux 操作系统的工业控制个人计算机作为机器人控制器开发平台是目前工业机器人技术和产业发展的方向和趋势. 该方案对于降低系统开发

生产成本、提高产品竞争力均有重要意义. 上海大学机器人实验室原有二台具有五个自由度的 PT600 机器人, 原控制系统为多 CPU 结构, 在仔细分析了原控制系统后, 我们研究开发了以工控 PC 机为核心的机器人控制系统, 并在 Linux 平台上进行了远程机器人控制的研究.

2 系统结构 (System architecture)

机器人的性能很大程度上决定于控制系统, 传统机器人控制系统以实现内部控制为主要目的, 对外通信能力弱, 缺乏与外界进行信息的高速交换能力, 缺乏与外界设备间进行协同工作的能力, 也缺乏对外界环境的适应能力, 很难实现对机器人的远程控制. 传统机器人控制系统的种种不足, 很在程度上是由于网络通信能力的不足而造成的. 我们所研制的机器人控制系统正是通过引入了网络通信技术,

使整个系统的功能得到了加强. 整个系统由机器人控制器、离线编程系统、图形仿真系统三个部分组成, 并通过网络连接起来.

系统中数据交换主要有以下几种情形:

1) 离线编程系统与机器人控制器的通信: 经过离线编程生成的机器人指令序列, 以文件的形式传递给机器人控制器.

2) 离线编程系统和图形仿真与监控子系统的通信: 在离线编程时需要模拟机器人实际运动过程, 这一功能是通过离线编程系统中内置的解释器对机器人指令进行解释, 实时生成机器人关节运动信息, 并通过网络输送到图形仿真与监控系统, 由该子系统

来完成图形仿真的功能. 由于通过网络传送的信息不是以文件的形式而是以实时数据流, 因而必须通过网络编程来实现.

3) 机器人控制器和通信仿真与监控子系统的通信: 在机器人控制器控制机器人运行时, 控制器将当前的状态数据传送到图形仿真系统, 以便使用户能够通过图形仿真与监控系统实时监视机器人的实际运动状态, 在必要时用户还可通过图形仿真与监控系统对控制器的实际操作进行干预. 这里传送的数据也是实时的数据流, 因而也必须通过网络编程的方法来实现. 远程机器人控制系统按层次和功能分为以下模块, 它们之间的关系如图 1 所示.

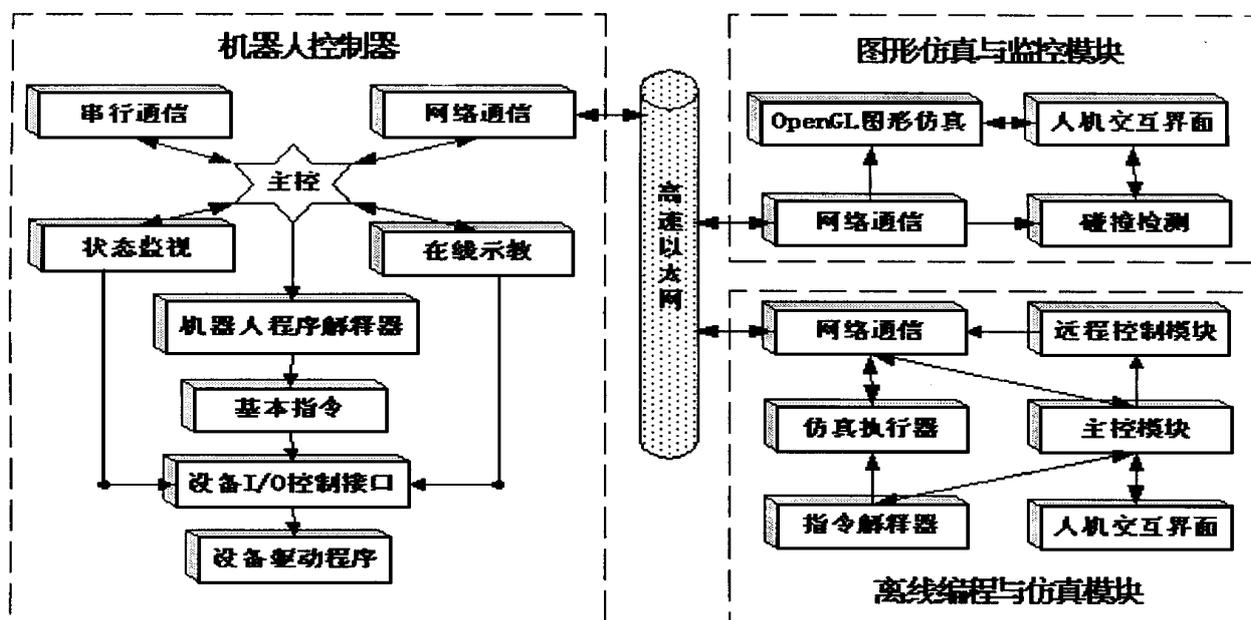


图 1 远程机器人控制系统

Fig. 1 Tele-robotics control system

要实现远程控制必须具有良好的可靠性. 可利用软件和硬件的 Watchdog 进行监控. 远程的客户发出机器人指令通过高速以太网发送给本地的遥操作系统, 该指令由本地的控制系统解释执行, 并将执行情况(机器人运行状态)返回给客户. 在服务器与客户机程序的相互通讯中双方都遵循传输协议, 在接收对方的数据时都要进行相应的指令解析.

多用户同时操作的问题也需在软件设计中予以考虑. 根据设计要求, 在同一时刻只能有一个远程客户具有对机器人的控制权, 当机器人处于空闲状态(无远程客户控制状态)时, 它等待客户机的连接; 这时若有客户机进行连接, 并对机器人控制的发出申

请, 它就可以获得对机器人的控制, 通过发出机器人指令, 由服务器(主控计算机)接收指令通过主控程序的运行, 控制机器人的运动, 并且发回当前的各关节位置, 通过客户机上的图形仿真模块进行三维的机器人仿真. 当机器人处于运行状态(有远程客户控制状态), 不具有控制权的客户需要对其控制的可以发出申请, 按其申请的先后进入等待队列, 等待对机器人的控制权, 这时, 不具有控制权的客户只能接收到机器人运动时发回的当前的各关节位置的指令.

由于在软件设计中利用了多线程支持, 不仅能使各模块并发运行并将模块间的耦合度控制在较低水平, 以后添加新功能也比较方便. 比如可以在状态

监视模块中加入模糊识别故障的功能.

3 Linux 下实现网络通信 (Implementation of internet communication with linux)

Linux 的网络实现支持 BSD 套接口, 支持完整的 TCP/IP 协议. BSD 套接口是最早的网络通信的实现, 也是 UNIX 系统中通用的网络接口. 它不仅支持各种不同的网络结构类型, 而且也是一种内部进程间的通信机制. 开发套接口的目的就是为隐藏网络底层复杂的结构与协议, 使编程人员能够抽象而简单地对网络进入操作, 随着 Internet 在全球范围内的广泛应用, 套接口已渐渐成为网络编程的通用接口.

在网络上大部分的通信都是客户机/服务器模式下进行的, 比起基于主机的运行环境, 主/从运行环境和共享设备的运行环境, 客户机/服务器的运行环境具有更高的分布程度, 更优良的性能. 针对传输

的数据需要连续数据流, 且需保证数据可靠的特点, 我们选择了流式 Socket 方式, 编程直接采用通用 Socket 的方案.

Socket 的编程基于套接字的系统调用. 应用程序首先必须通过调用 Socket() 创建套接字, 然后调用 Bind() 将套接字地址与所创建的套接字联系起来; 通过 Connect() 和 Accept() 建立套接字的连接, 其中 Connect() 用于建立连接, Accept() 用于使服务器等待来自客户的实际连接; Listen() 用于面向连接的服务器, 表示它愿意接受连接, Listen() 需在 Accept() 之前调用. 当建立一个连接后, 可以传输数据了, 常用的系统调用有 Send() 和 Recv(). Send() 用于在制定的已经连接的套接字上发送输出数据, Recv() 调用用于在指定的已建立的套接字上接受输入数据. 最后调用 CloseSocket() 关闭套接字, 并释放所分配给套接字的资源. 如图 2 所示.

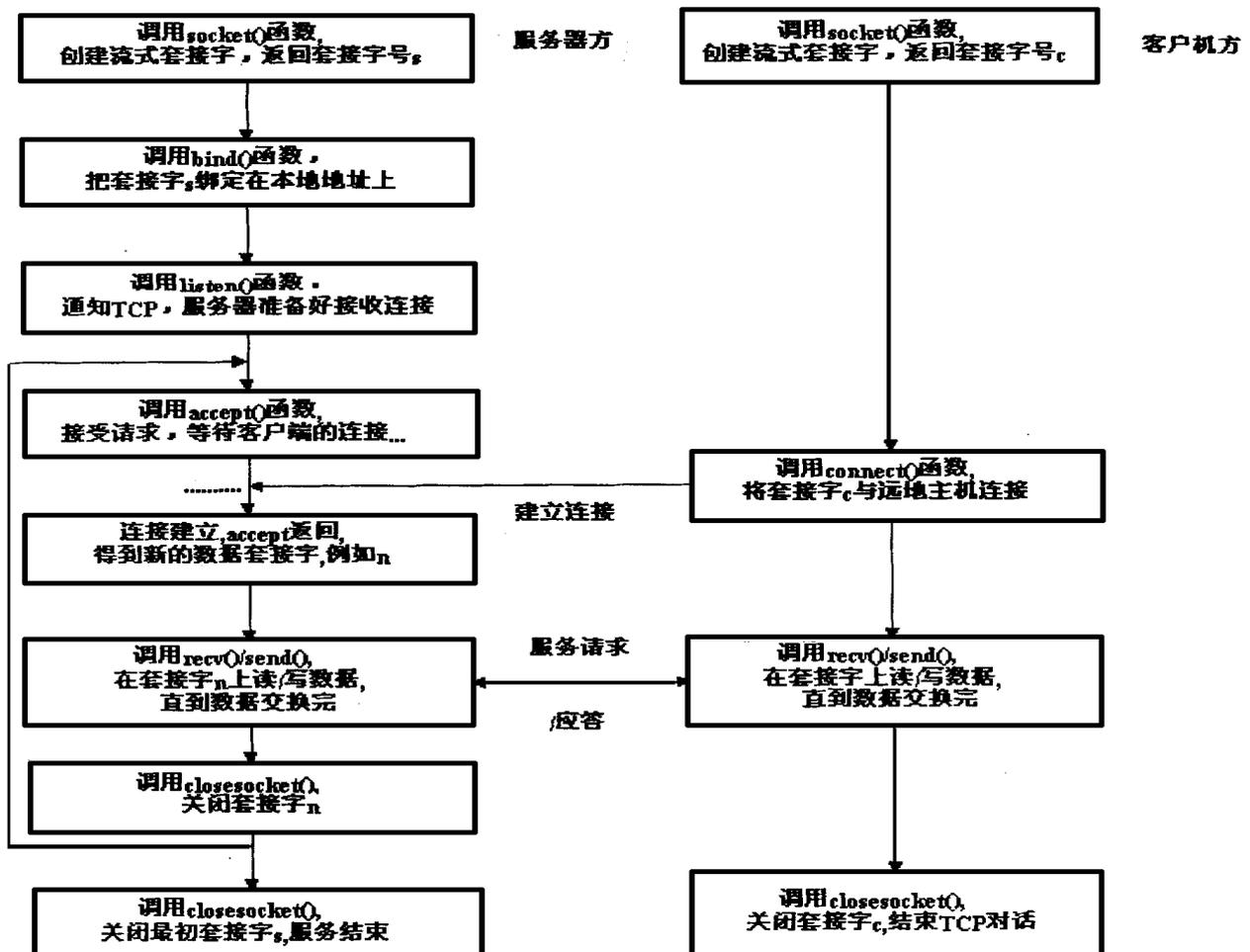


图 2 Socket 的编程流程

Fig. 2 The procedure base on socket programming

由于 socket 进行传输的是字节流 (byte stream), 必须在其上定义自己的高层协议, 才能实际用于状态数据的传送. 我们按照如下格式来定义传输的数据结构, 如图 3 所示.

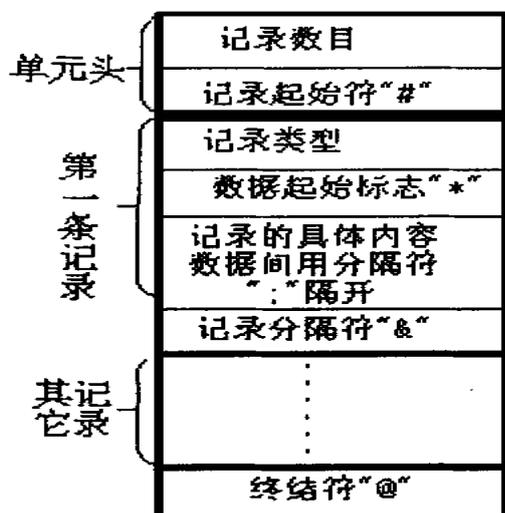


图 3 高层协议单元格式

Fig. 3 High level agreement format

高层协议单元由一个单元头和若干个记录, 以及终结符组成. 记录的个数是可变的, 以适应不同场合的需要. 单元头和记录与记录之间由记录分隔符来进行分隔. 每条记录通常分为两个域, 域之间用域分隔符隔开. 第一个域表示该记录的具体类型, 如是数据还是命令, 具体为什么命令等. 为简便起见, 直接采用一个 32 位整数来表示. 第二个域为该记录的具体数据, 数据间用域内分隔符隔开. 下面为一个协议单元的具体实例:

```
2# COMMAND* trace. home&ABSOLUTE_
DEG* 100.00; 30.45; 92.66; -67.10; -25.85@
```

"2" 表示该单元中有两条记录, "#" 为分隔符, 标志着第一条记录的起始. "COMMAND" 表示该记录为命令; "trace. home" 为命令的具体内容, 打开捕捉功能, 并执行回位操作. "*" 为记录分隔符, 表示第一个记录已结束, 下面将开始第二个记录. "ABSOLUTE_DEG" 表示该记录为绝对角度坐标数据. 分隔符后面为具体的绝对坐标数据, 各数据间用内部分隔符为 ";" 隔开. 整个单元由终结符 "@" 结束.

由于系统随时可能需要进行数据的发送和接收, 以得到需要的信息; 应用程序之间的数据传输很频繁且不定时的, 而系统又要求数据的传输不能消耗过多的系统资源, 不能妨碍用户界面上的操作和显示. 因此使用多线程实现多任务并行, 以达到系统

的要求.

4 程序实现 (Programming implementation)

服务器端程序必须具备多机连接的功能, 既要处理监听套接口, 又要处理已连接套接口的通讯, 即具有这样的功能: 如果一个或多个 I/O 条件满足 (例如, 输入已准备好被读, 或者套接字可以承接更多的输出) 时, 我们就被通知, 因而需要采用 I/O 复用, I/O 复用是由函数 select 和 poll 支持的; 使用函数 select 可以同时检查多个套接字是否就绪, 当有套接字就绪时, 函数 select 成功返回. 服务器方程序具体实现如下:

1. 首先调用 socket()、bind()、listen() 函数, 建立套接字, 做好接收准备, 然后需要调用 ioctl (serversock, FIONBIO, &flag) 设置主套接字为非阻塞式套接字 FD_SET (serversock, &readfds) 打开主套接字的监听位.

2. 接着调用 pthread_create (&tid, NULL, (void *) &SrvFunc, NULL) 创建后台工作线程 SrvFunc(). 这时, 服务器方通讯的初始化完成返回避免由于通讯造成主程序阻塞不能对机器人进行控制, 以及妨碍用户界面上的操作和显示的结果. 具体的通讯在后台工作线程 SrvFunc() 中进行, 它完成下述循环过程:

```
while(1)
{
select (MAXCONNECTION, &rfd, &wfd,
&efds, &TmeOut);
```

//调用 select, 等待某个事件发生: 或是新客户连接的建立; 或是数据、FIN 或 RST 的到达. 其中 MAXCONNECTION 表示最多可以连接的客户端机的数量, 参数 TmeOut 定时器的值为 0, 表示立即返回

```
if (FD_ISSET (serversock, &rfd))
//有新客户连接, 建立连接
{
connfd= accept(serversock, &cliaddr, &client);
//调用 accept, 建立与客户机的连接
for (i = 0; i < MAXCONNECTION; i++)
//检查所有的连接
if (client[i] < 0) {
client[i] = connfd;
//保存与客户机通讯的套接字
break;
```

```

}
if ( i == MAXCONNECTION )
err_quit(" too many clients");
//太多的连接客户
FD_SET( connfd, &allset);
//设置新的监听标志位
if ( connfd > maxfd)
maxfd = connfd;
if ( i > maxi)
maxi = i;
//client[ ]用户队列中最大的客户标号
}
for ( i = 0; i <= maxi; i++ )
//检查所有连接的客户
{
if ( ( sockfd = client[ i] ) < 0)
continue;
//连接的套接字已经关闭, 返回
if ( FD_ISSET( sockfd, &rset))
//套接字可读
{
SrvReceive( sockfd, buf, 32);
//调用函数 SrvReceive 进行读取处理
}
}
}

```

客户机端程序与服务器程序相类似, 这里就不赘述了。

5 结论(Conclusion)

由于我们研制的远程机器人监控系统采用了发展最为迅速的 PC 硬件平台和开放稳定的 Linux 软件平台, 因此可利用软硬件的最新技术成果来推动远程机器人监控系统技术的发展。所以, 我们研制的远程机器人监控系统能够满足我们对于远程机器人控制的实时性的需要, 并且具有极其优异的性能价格比、通用性、开放性和扩展性。基于 PC 和 Linux 远程机器人监控系统在价格、性能和发展潜力上都具有竞争力, 代表了未来的发展方向, 有很好的应用前景。

参考文献 (References)

- 1 任昊星, 翁海华, 杨杨, 陈坚. 基于 Web 的机器人遥操作的研究与实现, 计算机工程[J], 1999, 25(5): 85- 88
- 2 ALESSANDRO RUBINI. Linux 设备驱动程序[M], 中国电力出版社, 2000
- 3 Di Wang, Yin in Chen, Minglun Fang, Yongyi He. A Study on a New Robot Simulation and Monitoring System Based on PC. Proceedings of the 3rd World Congress on Intelligent Control and Automation: 28- 31
- 4 Richard Stevens W. UNIX 网络编程[M]. 清华大学出版社, 2000
- 5 鲍天麟, 陈一民, 何永义. Windows 95 平台上机器人控制软件的研究. 机器人[J], 1998, 20: 87- 91

作者简介:

薛广涛 (1976-), 男, 硕士生. 研究领域: 多媒体技术.

陈一民 (1961-), 男, 教授. 研究领域: 计算机控制, 网络与多媒体技术的研究.

张涛 (1976-), 男, 硕士生. 研究领域: 多媒体技术.