A Kind of Message Based Interprocess Communication

WU Zhen, CHEN Yao-wu
(*Institue of Advanced Digital Technology and Instrumentation, Zhejiang University, Hangzhou* 310027, *China*)

Abstract: In order to reduce the complexity brought by a great deal of data communication among multi-processes, we introduce a new kind of IPC based on message for Linux, propose the architecture and its work procedure. All kinds of IPC of Linux were listed and compared with the message based IPC. At last, we illuminate how to apply the message based IPC in DVR software architecture.

Key words: IPC; DVR; linux
EEACC: 6150

一种基于消息的进程间通信机制

吴 震, 陈耀武
(浙江大学 数字技术及仪器研究所, 杭州 310027)

摘 要: 为了简化多个进程间的大量数据通信给软件设计带来的复杂逻辑, 我们实现了一种 linux 下基于消息的进程间通信机制, 详细阐述了该进程通信机制的体系结构和工作流程, 并通过与传统的 linux 进程通信机制比较, 列举了各种进程通信机制的适用环境及限制。最后以嵌入式数字硬盘录像机(DVR)的软件设计为例说明了该进程通信机制的应用。

关键词: 进程间通信; DVR; linux;
中图分类号: **TN911** 文献标识码: **A** 文章编号: **1005 9490(2006)04 1218 05**

传统的 linux 进程间通信机制主要有管道、FIFO(命名管道)、消息队列、信号量、共享存储, 套接字, 信号等[1]。管道是进程之间的一个单向数据流, 一个进程写入管道的所有数据都由内核定向到另一个进程, 另一个进程由此就可以从管道中读取数据。命名管道在语义上与管道一样, 不同是一个命名管道是一个文件, 这个文件有目录项并可通过路径名来存取, 这样不相关的进程也可以通过这个文件相互通信。消息队列、信号量以及共享存储是 System V 的三种进程间通信机制[2]。消息队列允许进程发送格式化的数据流到任意的进程。信号量用于同步多进程对共享数据对象的存取。共享存储允许多个进程通过把公共数据放入一个共享内存段。此外, 将 linux 套接字的域设置为 AF_UNIX 也可以实现进程间通信。信号是一种软件中断, 用来向一个或多个进程发送异步事件信号。上述的进程间通信机制使用简单, 但并不适用于多个进程间的多参数通信机制[4]。如图 1 所示。

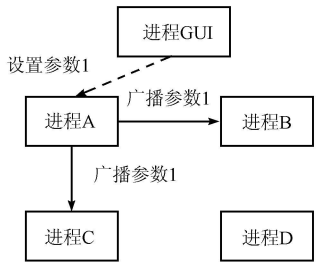

图 1 多 个进程间通信

通信的目标进程需要根据具体的通信内容(后文称之为参数)而定, 当目标进程处理完参数后, 还需要把参数更新通知给所有侦听此参数的其他进程。而且为了保证多个进程都能及时使用更新的参数, 还需要使用通知机制来保证实时性。比如在图 1 中, GUI 进程负责接收用户对参数的更改设置, 但

收稿日期: 2005 11 07
作者简介: 吴 震(1980), 男, 黑龙江哈尔滨人, 硕士研究生, 研究方向为嵌入式系统开发, zjwuzhen@hotmail.com;
陈耀武(1963), 男, 辽宁铁岭人, 教授, 博导, 从事嵌入式系统研究

GUI 进程只是负责与用户的交互, 实际设置参数是由其他参数维护进程来完成。假如用户更改了参数 1, GUI 进程把对参数 1 的设置通知给实际维护参数 1 的进程 A, 进程 A 根据用户设置完成了更改, 同时需要把参数 1 的更新通知给另外两个需要用到参数 1 的进程, 即进程 B 和进程 C, 在下文中, 这个过程称为广播。当有大量这样的参数需要在多个进程间通信时, 使用传统的 linux 进程通信机制将会使逻辑变得极为复杂。针对这种情况, 我们设计了一种基于消息的适用于多个进程间大量参数的通信机制。该机制除具有传递消息和更新通知功能之外, 还可在运行时动态的设置参数的维护进程和广播进程。

1　基于消息的进程间通信机制

1.1　设计目的

提供一种适用于复杂逻辑的多进程间信息传递的机制, 并保证多进程间通信的同步性; 提供通知机制, 方便进程及时处理更新; 由驱动模块实现, 减少用户进程相互依赖的关系, 增强系统的模块化和扩展性。

1.2　设计思路

1.2.1　用驱动程序实现通信

进程间通信通过一个由驱动模块实现的通信中心来完成, 进程发送消息至通信中心, 并由通信中心转发消息至目标进程, 目标进程可以是一个进程也可以是多个进程。通信中心为使用通信中心的进程创建一个如图 2 所示的结构, 其中 CMD_PARA_H 表示命令链表头, 链接了该进程维护的所有参数, 即该进程为这些参数的维护进程。BRD_PARA_H 表示广播参数链表头, 链接了该进程侦听的参数, 表示该进程希望当这些参数改变时得到通知。以上两个链表用来表示参数与进程的关系, 用来查找参数的维护进程和侦听进程, 真正的命令和广播是通过消息来发送的。CMD_MSG_H 是命令消息的链表头, 当一个进程需要设置/读取其他进程维护的参数时, 通信中心找到该参数的维护进程, 然后把设置/读取信息加入到命令消息的链表中。BRD_MSG_H 是广播消息的链表头, 当一个进程侦听的参数发生变化时, 由通信中心把参数更新消息添加到所有侦听此参数进程的广播消息链表中。RES_MSG_H 表示回应消息链表头。当一个进程设置了其他进程维护的参数时, 参数维护进程会给设置进程发来回应, 表示是否设置成功。回应消息链表头链接的元素就是回应消息。

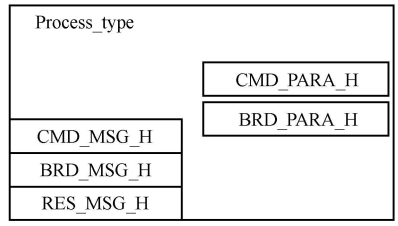

图 2　进程结构

1.2.2　以参数为核心进行构造命令路径

进程间通信内容被抽象成如图 3 所示的一个个 para_type 结构体, 每个结构体有一个参数名(Name), 进程链表头(Process_H), 左孩子(LeftChild)和右孩子(RightChild)。各个参数结构体以参数名为关键字构造成一棵排序二叉树或者是平衡二叉树如 AVL 树、红黑树等。

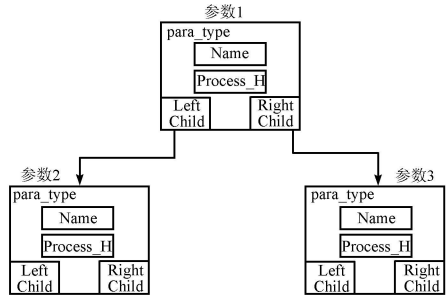

图 3　参数树

其中, Process_H 是表示链表头, 链接了该参数的维护进程, 而在图 2 中 CMD_PARA_H 链表头链接了该进程维护的所有参数, 考虑到一个参数可能有多个维护进程(一般情况下, 一个参数的维护进程只有一个, 即只可以通过一个进程设置其生效), 而一个进程又可能维护多个参数, 所以这里不能将参数节点与进程直接相连, 而是通过图 4 所示的连接器将参数结构与进程联系在一起。

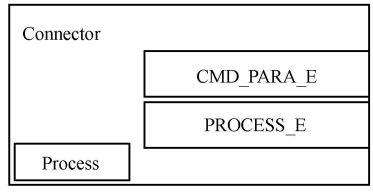

图 4　连接器

其中, CMD_PARA_E 链接在 Process_type 结构体中的 CMD_PARA_H 链表中, PROCESS_E 链接在 para_type 结构体中的 Process_H 链表中, Process 是一个指向 Process_type 结构体的指针, 指向参数的维护进程。通过 Connector 连接进程和参数树如图 5 所示。

图 5 所描述的进程和参数的维护关系如图 6 所示。参数 1 连接到 Connector1, 进程 A 连至 Con

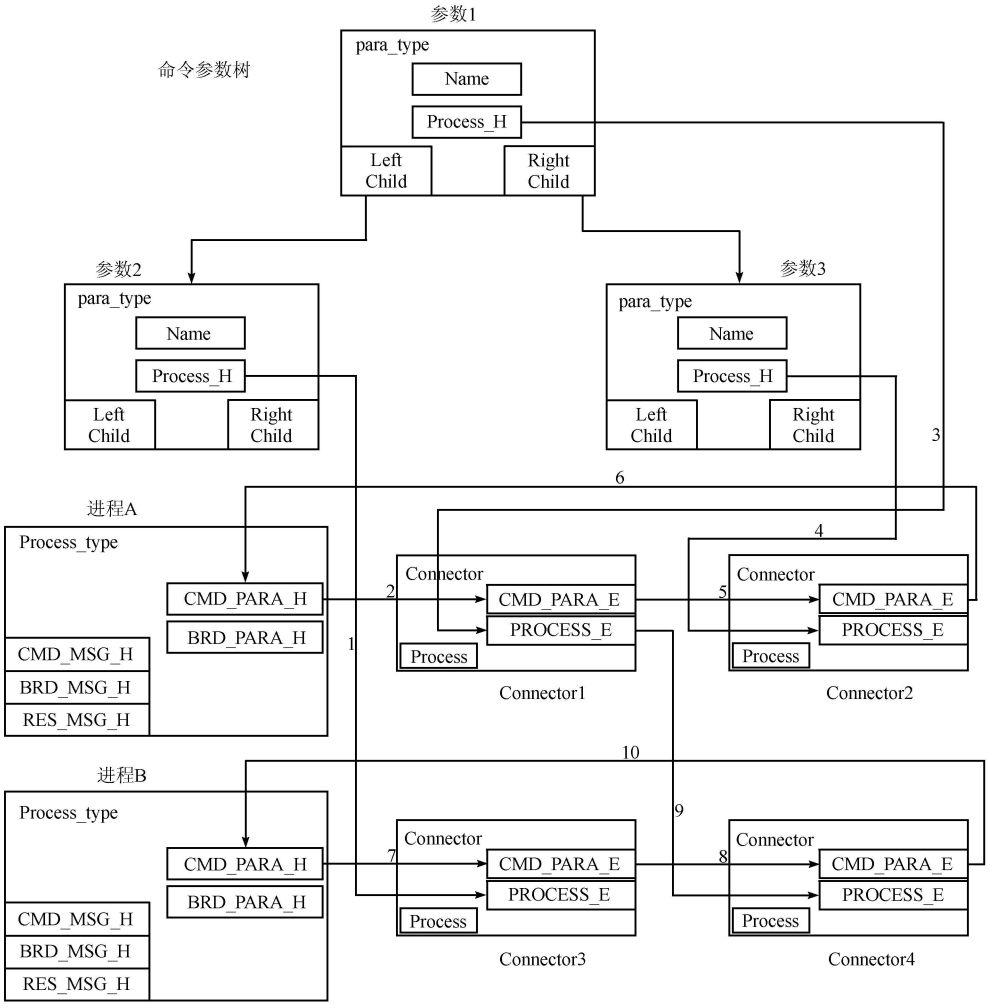

图 5 命令参数树

从图 5 可以看出，表示进程 A 维护参数 1，其余参数的维护进程也可以类推。在这里比较特殊的是，参数 1 有 2 个维护进程，所以 Connector1 又与 Connector4 相连，表示进程 B 也维护参数 1，即也可以通过进程 B 进行更改参数 1。在这样的情况下，当有其他进程发出命令请求设置参数 1 的值时，通信中心沿着参数 1 的 Process_H 域的链表找到第一个维护进程即进程 A，然后将设置请求填写至结构体 MSG 并添加到进程 A 的 CMD_MSG_H 链表中。

	参数1	参数2	参数3
进程A	维护		维护
进程B	维护	维护	

图 6 参数与进程的关系

```
struct MSG
{  //伪码
   bool set_or_read;
   unsigned long Parameter_name;
   unsigned char Parameter_value[ MAX];
};
```

进程 A 处理该请求，然后发送回应。当进程 A 退出的时候，参数 1 的维护进程只剩下进程 B，此时如果有其他进程再次发出设置命令请求的时候，沿着参数 1 的 Process_H 域的链表找到的第一个进程，即进程 B，将负责设置参数 1 的值，并发送回应。

1.2.3 以参数为核心进行构造广播路径

与命令通道类似，每个进程都有一个广播通道，广播通道用于接收参数维护进程发出的参数变化广播消息。该广播消息一定是参数维护才有权发送。每个进程都可以订阅感兴趣的广播消息。参数维护进程发送参数变化广播消息后，每个订阅了该参数广播消息的进程都将得到一份广播消息。进程在接收到该广播消息后，可以根据自己的需要进行处理，处理完毕后，不需进行回应。

图 7 所示的是广播参数树和进程之间的关系。与图 5 类似，不同的是这里的 Connector 是与进程

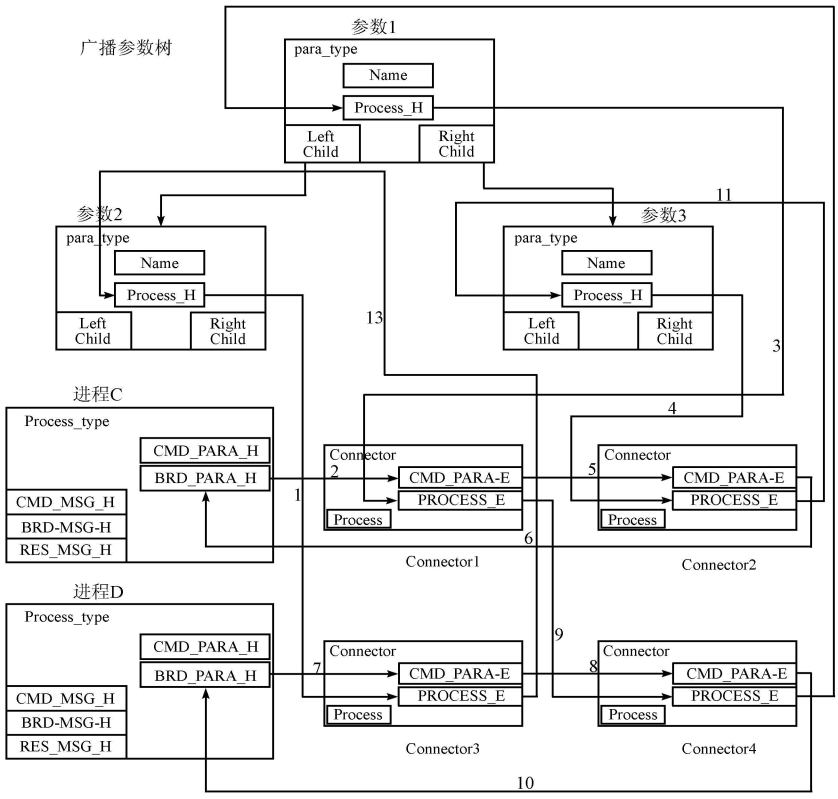

图 7　广播参数树

的 BRD_PARA_H 连接成链表的。图 7 表示的是进程 C 侦听参数 1 和参数 3, 进程 D 侦听参数 1 和参数 2。当参数 1 发生变化时, 通信中心沿着参数广播树查找参数 1 的 Process_H 域连接的所有 Connector, 即遍历整个 Process_H 链表, 找到所有侦听此参数的进程, 然后分别把参数更新通知添加到侦听进程的 BRD_MSG_H 链表中。最后所有侦听进程根据参数更新作出相应的调整, 见图 8。

	参数1	参数2	参数3
进程C	订阅		订阅
进程D	订阅	订阅	

图 8　进程订阅参数

1.2.4　工作流程

在图 5 的参数命令树中, 如果参数被维护进程更改, 则维护进程直接发送广播消息至参数中心。参数中心沿着图 7 的广播参数树查找所有侦听进程并发送广播通知。当非参数维护进程发出设置/读取命令, 通信中心记下发出命令的进程 id, 并在参数命令树中沿着该参数节点的 Process_H 链表找到该参数第一个维护进程, 把设置/读取命令添加到维护进程的 CMD_MSG_H 的链表中。维护进程处理完该命令请求后, 发送回应至通信中心, 最后通信中心把回应链接到发出命令的进程的 RES_MSG_H 的链表中。如果命令是设置参数并且维护进程设置成功时维护进程还需要发送广播通知至通信中心。通信中心沿着图 7 所示的参数广播树遍历参数节点的 Process_H 链表找到所有侦听进程, 并把更新信息链接到每个侦听进程的 BRD_MSG_H 链表中。

1.2.5　对应用程序设计的影响——事件驱动

使用通信中心的应用程序须以事件为核心进行设计。应用程序需要在进程中实现一个消息循环, 处理参数命令发送回应。需要使用通信中心读取或设置参数的应用程序调用通信中心的入口发送命令, 然后在回应通道等待回应。需要获得参数变化通知的应用程序, 同样需要实现一个消息循环, 等待广播消息。

2　在嵌入式数字硬盘录像机软件设计中的应用

2.1　嵌入式数字硬盘录像机需求及功能

嵌入式数字硬盘录像机(DVR)是广泛应用于安防领域的视频监控终端。如图 9 所示。每台 DVR 将采集到的音视频数据实时送至本地显示, 并将音视频数据压缩存储至本地硬盘, 并根据用户需

要发送音视频数据至网络客户端。

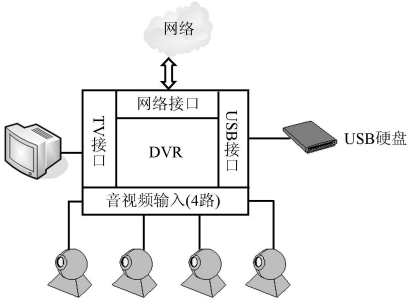

图 9 嵌入式数字硬盘录像机

2.2 软件模块

如图 10 所示,其中 GUI 和 OSD 提供了图形化操作用户界面。OSD(on screen display)提供的是快捷操作,遥控器上有快捷按键可以直接弹出 OSD 界面,设置色度、亮度、饱和度和对比度等,方便用户操作。存储管理进程负责音视频录像和报警事件等存储至硬盘。网络管理进程负责发送音视频录像至网络客户端,以及处理网络客户端对 DVR 系统参数的远程设置。报警检测进程侦听传感器的报警事件。日志管理进程负责记录用户操作日志以及其他进程的执行日志。AV 管理进程根据用户需要设置关于音视频的相关参数(比如视频码流,帧率等)和音视频 A /D 的相关参数(比如色度、亮度等)。备份管理进程根据用户选择将需要备份的音视频录像和日志备份至 CD 或者 USB 硬盘。参数通信中心是用驱动模块实现的进程通信机制。每个进程都维护一部分参数,比如 AV 管理进程维护所有的关于音视频设置以及音视频 A /D 的相关参数(比如色度、亮度等)。在这里使用参数通信中心时,每一个参数只有一个维护进程。

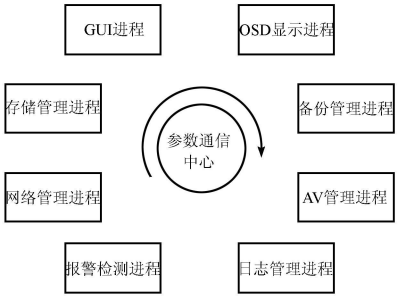

图 10 DVR 软件模块

2.3 进程通信机制

在 DVR 系统中,有大量的参数需要用户参数设置。用户可通过 GUI 进程、OSD 进程或者网络客户端远程设置参数,但是出于系统稳定性、模块化和维护性等方面的考虑,GUI 进程和 OSD 本身并没有加入设置参数的逻辑。他们记录下用户的设置后,需要与参数维护进程通信完成参数设置操作。但同时,有些其他进程也要针对这些参数的更新做相应调整。比如,用户通过 OSD 快捷方式修改了色度亮度等参数,实际的设置工作由 AV 管理进程完成,但与此同时 GUI 以及远程网络客户端所显示的参数值都要进行相应改变。这样,AV 管理进程成功设置新的参数后,需要通知网络管理进程和 GUI 进程参数更新。再比如,存储管理进程更换了当前活动硬盘,要通知 GUI 进程和日志管理进程(需要把日志记录在当前活动硬盘)。由于系统存在大量的参数,每个参数除了有维护进程还有侦听进程并且需要及时得到更新通知。在处理这样复杂的通信逻辑时,应用传统的进程间通信机制将使软件设计非常烦杂,容易出错且不易扩展。使用参数通信中心由驱动来实现设置和广播消息的转发,使得软件设计的模块化和维护性及扩展性都得到了提高。

除此之外,也有其他的可行方案:一是单独设立进程,与其他进程交互;另一个方案是使用共享库链接入各进程,直接完成进程间通讯。前者的缺点是每次参数操作会引起额外的进程切换,而且需要使用其他进程间交互机制,比如 FIFO、套接字等,这些机制需要为每对进程的交互通道指定名字,无法动态扩展。后者的缺点是,链接库会引起编译和链接时的依赖问题,增加了进程间的耦合性。

2.4 与其他 IPC 机制的比较

管道虽然灵活简单,但是其只能由相关进程使用,需要一个共同的祖先创建管道[4]。命名管道、消息队列以及套接字都允许不相关的进程间进行通信,这些机制在只有两个进程间通信时可以很好的工作,但是当多个进程需要相互通信,并且欲通信的目标进程取决于通信内容的时候,其逻辑将非常复杂,而且必须在使用前协商好名字,无法动态扩展。共享存储允许多个进程共享通信内容,但是这种通信方式缺乏通知机制。而信号量只是一个计数器,用于同步,并不能实现通信内容的传递。信号具有通知的功能,但是却没有给参数、消息等具体通信内容留有空间。我们设计的基于消息的进程间通信机制既具有传递消息和更新通知的功能,又可以简化多个进程间多参数传递的逻辑复杂性,在 DVR 系统中,采用这种进程间通信机制极大的简化了软件架构设计的复杂性。

3 结束语

我们介绍了一种基于消息的进程间通信机制。

(下转第 1226 页)

由图 8 可以看出，介质杆替代介质板使慢波电路的色散特性和耦合阻抗都产生了较大的变化，工作频率得到大幅提高，耦合阻抗也有了增加，大大增强了与电子束的作用。

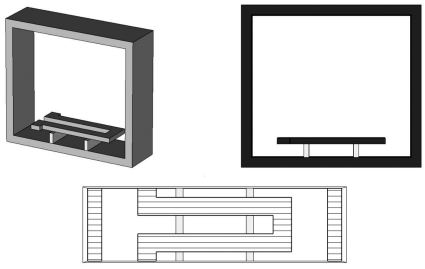

图 7 介质杆支撑的微带型曲折线慢波结构示意图

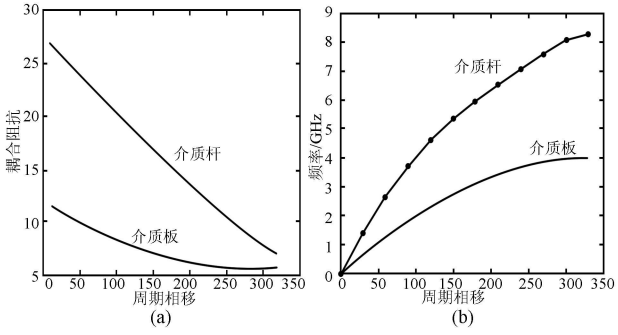

图 8 介质杆支撑的微带型曲折线慢波结构的色散特性和耦合阻抗

3 结论

本文对微带型曲折线慢波结构的计算机模拟结果与理论分析结果是基本符合的。这些处理方法普遍适用于对其他的周期性慢波结构的研究，可以为实际的工程设计提供比较精确的依据。此外，为了提高耦合阻抗，可以采用文献[1] 提到的双曲折线结构，同时可采用双面结构，类似文献[6] 的结构，器件的互作用效率也会得到相应的提高。

参考文献：

[1] 彭自安. 微带型曲折线慢波结构的研究[J] . 电子学报，1983：68 75.

[2] Roome Gerard T，hair Hugh A，Thin Ferrite Devices for Microwave Integrated Circuits[J] . IEEE Transaction on Electron Devices，July 1968，15：473 482.

[3] Rizwan Murji，M. Jamal Deen，A Scalable Meander Line Resistor Model for Silicon RFICs[J] . IEEE Transaction on Electron Devices，Jan. 2002，49(1) ：187 190.

[4] Rizwan Murji，Deen M Jamal，Accurate Modeling and Parameter Extraction for Meander Line N-Well Resistors[J] . IEEE Transaction on Electron Devices，July 2005，52(7) ：1364 1369.

[5] Carol Kory，Lawrence Ives，John Booske，Suede Bhattacharjee，et al.，Novel TWT Interaction Circuits for High Frequency Application[C] //IVEC2004.

[6] Dayton James A. 300GHz BWO with Biplanar Interdigital Circuit[C] //IVEC2005.

[7] Weiss J A. Dispersion and Field Analysis of a Microstrip Meander Line Slow Wave Structure[J] . IEEE Transaction on MTT，Dec. 1974，22：1194 1201.

[8] 周文等. 印刷电路返波管的特性计算与实验研究[J] . 浙江大学学报，1980，2：66 80.

[9] Kory Carol L，Dayton，Jr James A. Accurate Cold Test Model of Helical TWT Slow Wave Circuits[J] . IEEE Transaction on Electron Devices，April 1998，45(4) ：966 970.

[10] 李实等，螺旋线慢波结构的参数变化对其冷测特性的影响[J] . 电子学报，2004，32(9) ：1511～1514 .

[11] 刘盛纲，李宏福，王文祥等，微波电子学导论[M] . 国防工业出版社，1985.

[12] Mourier G. Small signal theory[M] . 1961.

(上接第 1222 页)

该机制适用于多个进程间的存在复杂维护和侦听逻辑的数据通信，方便了系统软件的模块化和松耦合度设计，并且易于系统维护和扩展。该进程通信机制采用驱动模块实现，减少了进程间切换的时间，从而提高了进程间通信的响应速度，特别适用于对实时性和稳定性要求较高的嵌入式系统软件设计。

参考文献：

[1] 王文义，武华北. Linux 中进程间信号通信机制的分析及其应用[J] . 计算机工程与应用，2005，41(3) ：108 110，115.

[2] 杜毅，杨金生，吴震华. Linux 消息队列分析及应用[J] . 计算机工程，2004，30(B12) ：175 177.

[3] 冯佳奇，林浒，吴文江. 基于 RT Linux 实时系统进程间通信的研究和在数控系统中的应用[J] . 小型微型计算机系统，2004，25(1) ：21 23.

[4] 吴舜歆，李洪海，洪玫，张冬，冯子亮. 基于 SCO UnixWare 的实时消息队列设计与实现[J] . 四川大学学报，自然科学版. 2004，41(1) ：92 96.

[5] 李小群，赵慧斌，孙玉芳. 进程间通信机制的分析与比较[J] . 计算机科学，2002，29(11) ：16 21.