一个分布式实时操作系统UECnet

电子科技大学计算机系

龚天富　张松梅　李广星　舒　敏　张　立　桑　楠

摘要　本文论述了分布式实时操作系统UECnet的概念和实现。UECnet是一个在三台PC/AT连成的分布式系统上开发的，基于进程模型的分布式实时操作系统原型。它旨在提供一个通用的分布式实时处理环境。在操作系统级，提供了满足实时限制的通信机制，尽量保证进程实时要求的调度算法。在应用程序级，提供构造实时系统的工具，减少大型分布式实时应用系统的开发周期。为适应系统可演化的需要，还提供系统配置（Configuration）能灵活变动的机制。

一、前言

分布式概念应用于实时处理，将产生新一代的实时应用系统。随着实时系统朝着大型、复杂和动态的方向发展，例如实验室控制，核电厂，空间站和飞船控制系统，以及机器人等，这些系统除了对响应时间要求非常严格外，还需要有较大的灵活性、可预测性和可靠性。因此，迫切需要研究并开发分布式实时操作系统及其实时系统的构造工具，为这类大型分布式实时软件提供一个舒适方便的开发环境。

在国外，分布式实时程序设计系统的研究非常活跃。有些是在面向专用任务开发的基础上构造出来的系统模型，如加拿大Queen大学研制的Rnet[1]模型；有的是从程序设计开发工具方面进行研究的，如英国帝国大学研制的Conic[2]系统（该系统不具实时性）；有些是较为实用的系统，还有一些面向对象的实时操作系统。目前商用的分布式实时操作系统还未见报道。

与国外相比，国内分布式实时操作系统的研究起步较晚，有较大差距。我们在承担预研项目“分布式实时操作系统”的模型研制任务中，参考了Rnet[1]系统，Conic[2]系统及DiCon[3]系统，在三台PC/AT连成的分布式系统中，实现了UECnet分布式实时操作系统原型。本文论述UECnet的概念和实现。

二、设计目标

UECnet系统的设计动机是提供一个通用的分布式实时处理环境。分布式实时系统的实现涉及从操作系统到应用程序设计的多级复杂范畴。一般讲，在操作系统层，为保证分布进程的实时同步与互斥，需要实时通信机制和同步机制；为保证应用进程对实时事件的有效处理，应采用实时调度算法。在应用程序层，应提供描述分布式实时应用软件的一般方法和工具。另外，大的嵌入式计算机系统，在其生存期内非常需要系统随应用环境的改变所提供的服务也改进，即系统是可演化的，因此应提供系统配置（Configuration）能灵活变动的机制。UECnet对这些需求提供有力的支持，可作为通用的分布式实时处理构造系统。

三、系统的物理构成

UECnet由三台IBM—PC/AT微机作为系统的结点机。每个主机配有二个RS-232—C标准的串行接口，通过双绞线电缆相连接。接口采用双工方式。各结点机平等自治，也可由其中任何一台作为主控结点。通信核心支持不同的网络拓扑结构。见图一所示。

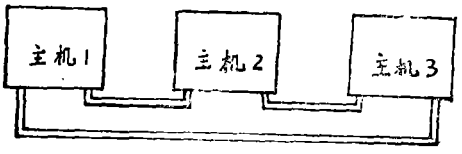

图1 UECnet系统物理结构

四、系统模型

UECnet系统的设计采用进程模型，消息通信机制，最早死限加剥夺的调度模型和动态重配置模型。试图从实时系统的基本特征出发，探讨描述分布式实时软件的基本方法，以及开发这类软件所需的支撑环境。

1. 进程模型

有三类进程存在于UECnet系统中。

(1) 实时进程　实时事件一般由周期事件和突发事件组成。同期事件由时间驱动，周而复始地执行。突发事件随机产生，但为了预测系统的状态，要求两事件发生的间隔大于某一时间值，称为突发事件的间隔周期。根据这一物理模型，UECnet中的实时进程由周期进程和突发进程组成。周期进程具有等待周期，突发进程具有间隔周期。

(2) 报警进程　实时系统中常有异常事件需要处理，报警进程试图提供用户处理异常事件的工具。但是，由于实时系统设计时难以对异常事件的类型、原因及解决办法进行无遗漏的估计，用户在此基础上还应采用其它冗余技术解决系统容错问题。

(3) 后台进程　无实时性，在没有实时进程和报警进程时执行这类进程。

2. 通信模型

在UECnet系统中，进程之间采用消息传递方式通信。本地进程间的通信与远程进程间的通信具有完全相同的语义。除某些特殊场合（如固定结点的进程）外，用户不必关心进程存放的位置。

消息通信采用带时间信息的端口(port)方式。进程P_1向portA（输出端口）发消息，进程P_2在portB（输入端口）接收消息，需要用静态连接（Link)方式将portA与portB连接，这二个进程就可以通过此通道进行通信。

port按类型分为三类：

(1) 接收消息port

inport，输入端口，仅出现在接收原语中。接收与inport相连接的port发来的消息。inport与一时间死限（deadline）相联系。当等待消息的时间超过该死限，系统自动报错。

(2) 发送消息port

outport，输出端口，仅出现在发送原语中。采用不等待方式，异步地发送消息。

(3) 进程开始port标志一进程的开始。

a. Sporadic Start port，突发进程开始port，仅出现在接收原语中。执行时，标志突发事件的间隔周期开始，等待消息。消息到来，且两突发事件发生的间隔大于间隔周期，激发该进程执行。

b. alarm start port 报警进程开始port，仅出现在接收原语中。执行时，报警进程等待消息，消息一到立即剥夺执行。

c. Period Start port，周期进程开始port，执行时，周期进程睡眠，等待周期结束，由时间自动激发执行。

消息的传送通过执行发送和接收原语实现。

3. 调度模型

在UECnet系统中，一个进程可由多个

任务（task）组成。任务的划分是自程序代码中出现的接收或发送原语（包括该原语）始，至下一个出现的接收或发送原语（不包括该原语）止。这样划分可以将port（出现在接收或发送原语中）的通信时间与任务代码的执行时间分开描述，使时间特性确定化，易于预测、分析和超时处理。组成进程的任务是顺序执行的。

任务的实时特性由下述时间量描述：

a．到达时间　任务处于就绪队列的时间；

b．计算时间　任务代码的执行时间；

c．死限　任务执行完成的截止时间；

d．周期　实时进程的周期。

其中a是以系统时钟为准的物理时间，b和c是相对于a的时间。

系统运行期间，根据以上实时特性，调度采用最早死限加剥夺的策略，保证具有最早死限的进程投入运行。每一物理结点上的资源由当前运行的进程独享。

系统将超时会导致灾难后果的进程称为硬实时进程，这类进程的超时处理是向用户报错，激发用户预先编制的报警进程运行。对于软超时进程，系统仅报错，允许该进程继续运行。

4．配置模型

（1）配置说明语言（UECspec）

一个分布式实时应用系统可由进程逻辑网描述。UECnet的程序由一组并行进程组成，各进程间的连接关系由配置说明语言进行说明，主要说明端口（port）间的连接关系。将一组必须分配在同一物理结点上的进程组合在一起，构成逻辑结点，它们必须分配在一个物理结点。在此意义下，程序被视为逻辑网，它由多个逻辑结点组成。另一方面，逻辑网到物理网存在一种映射关系，几个逻辑结点可以映射到同一物理结点。因此程序又可视为物理结点网。每个物理结点描述了单处理机上多任务实时处理环境。图2给出了逻辑网到物理网的映射关系。

UECSpec配置说明语言提供如下功能：

首先描述逻辑网络的完整结构，包括对每个逻辑结点的说明，逻辑结点所需的资源情况，用户将逻辑结点映射到物理结点的愿望等。

提供每个程序模块的说明手段，包括该程序模块用何种语言编写，放在哪个文件中，该程序模块的通信端口说明等。

给出每个模块实例（即进程）的说明，其中包括进程的实时特性说明，实时任务说明及进程到逻辑结点的约束等。

提供进程间端口静态连接的描述手段。

（2）UECnet静态配置方法

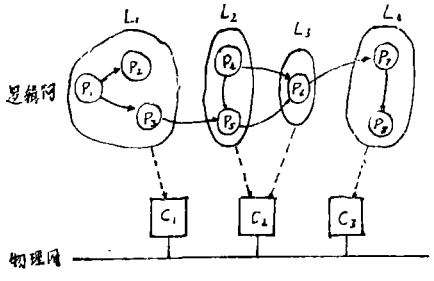

图2　UECnet逻辑网到物理映射关系

其中　Ci物理结点　Li逻辑结点　Pi进程

静态配置模型如图3所示。

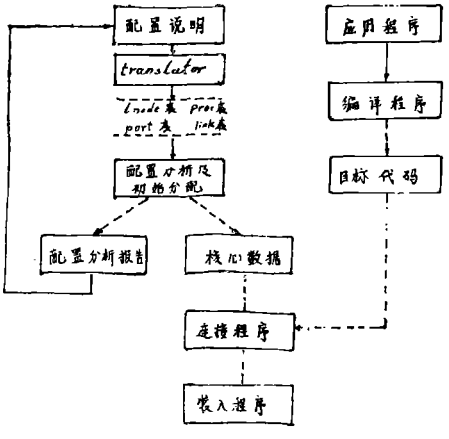

图3　静态配置结构

Translator是配置语言 UECpsec 的编译程序。它对配置说明进行语法和语义分

析，得到配置分析所需的数据：逻辑结点到物理结点的映射关系表Lnodemsg，进程表Proc，任务表Port及端口连接关系表Link。根据这些数据，静态配置分析程序进行时间和资源分析，并对进程进行初始分配。若不成功，将分析报告提交用户，供用户修改参考。这一过程是交互式的。

UECnet系统初启时采用静态配置方式。

(3) UECnet动态重配置模型

动态配置提供系统在运行时扩展和更新系统的方法。如图4所示。

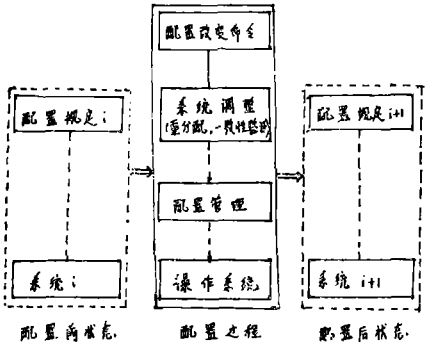

图4　动态配置模型

当系统接到配置改变命令时，系统立即进行调整。调整包括重分配，验证新的系统状态的正确性，修改配置说明等。调整完毕，由配置管理程序向操作系统发命令，修改系统的状态，系统正确继续执行。

五、系统实现

UECnet系统由三个物理结点组成，每个物理结点是一个多任务实时处理环境。物理结点之间由通信软件提供网络服务，系统提供带时间信息的通信原语，系统还具有动态重构，动态进程更新和扩展的功能。

UECnet系统实现由分布式实时操作系统核心和动态重配置处理程序两部分构成。分布式实时核心包括实时核心和通信核心。实时核心主要提供系统初始化，进程调度时钟管理和中断处理等功能。由于UECnet以Dos3.3为宿主操作系统，故没有实现存贮管理等功能。通信核心提供消息通信管理和网络服务程序。动态重配置处理包括配置分析、系统初始分配、动态重分配、动态重构处理和动态进程更新等。系统还提供了配置说明语言UECspec及其编译器。

1．实时核心

(1) 进程调度　系统核心维持一张全局进程表和一张局部进程表。进程调度采用最早死限加剥夺的策略。就绪队列的进程按死限大小排队，若两进程死限相同再考虑优先级。进程可通过Setpri原语动态改变优先级。非实时的后台进程处于后台队列。报警进程一旦激活，立即投入运行。

(2) 中断及设备处理　为每个设准设置一个预定义的核心port。在配置说明中，这个port与用户提供的设备处理进程的输入端口inport相连接，中断信号能通过核心port发送到用户进程，因而设备中断处理进程可作为用户进程。

2．通信核心

(1) 网络服务程序　主要实现物理结点间的通信。参照国际标准组织 (ISO) 提出的OSI参考模型，提供了物理层，数据链路层和通信管理层等低层服务。

(2) 消息通信机制　系统设置全局port表和局部port表，管理port的信息和时间信息。

消息发送原语Send为不等待发送。发送port不设缓冲区，将消息挂到（或由通信核心传到）接收port的缓冲队列，通过信号灯通知接收方。

接收原语receive分为等待接收和不等待接收。接收port具有缓冲区。

在这种异步通信方式下，用户必须关心发送速率和接收速率的匹配，不然会发生消息覆盖。

3．配置分析

以任务作为分析单位，时间可满足性为主要依据，考虑负载平衡和资源需求的情况，对配置说明进行分析，不成功则自动调整分析策略。若无论如何调整都难以满足，那么分析程序给用户提供分析报告，由用户修改配置说明。

4．应用系统分配

无论初始分配还是动态重分配，实质上都是在保证物理结点间通信量尽可能少，并满足上述分析条件的前提下，将进程逻辑网映射到物理网。这个映射过程是将进程逻辑网转化成在每个物理结点上可按时间顺序执行的任务序列。该算法采用启发式方法。

5．动态重构处理

UECnet系统的三个物理结点平等自治，其中一台为主控结点。当结点失效时，剩余结点立即选出新的主控结点，负责将失效结点的进程重分配到剩余结点。系统根据所保存的失效结点（由其它结点保存）的部分现场进行自动恢复，达到优美降级。

6．动态进程更新

UECnet系统提供配置改变命令，以更新或扩展进程。系统接到更新命令后，各物理结点同步，立即进行调整。调整包括重分配，验证系统状态，修改连接结构等。调整完毕后，由配置管理程序向操作系统核心发命令，系统进入新状态继续运行。

配置语言UECspec语言文本及其编译器将另文论述。

六、结束语

分布式实时操作系统是当前计算机应用非常重要的发展方向之一。分布式实时程序设计的困难在于应用系统非常复杂，各程序模块之间既是物理分布又是互相合作的，而且还具有实时性。操作系统设计者的任务是根据分布式实时应用软件的特点，找出描述这类应用软件的一般方法和工具，提供用户开发分布式实时软件的环境，减少用户编写这类软件的困难。本文以一个结构清晰，概念完整的分布式实时程序设计模型为基础，描述了UECnet分布式实时操作系统原型及其系统生成工具。我们的工作仅是初步的，今后将朝实用化方向努力。

七、参考文献

〔1〕C.Belzile et al, "Rnet: A HARD REAL TIME DISTRIBUTED PROGRAMMING SYSTEM", In IEEE Real Time System Symposium 1986, PP2—13.

〔2〕M.Sloman et al, "THE CONIC TOOKIT FOR BUILDING DISTRBUTED SYSTEM 1985, PP79—89.

〔3〕Insup Lee, "A PROGRAMMING SYSTEM FOR DISTRIBUTED REAL-TIME APPLICATIONS", in IEEE Real Time System Symposium 1984, PP18—27.

〔4〕V.P.Holmes and D.L.Harris, "A DESIGER's PERSPECTIVE OF THE HAWK MULTI-PROCESSOR OPERATING SYSTEM KERNEL", Operating System Review, 1989, 7.