普及计算终端的自适应性技术研究

李 允,熊光泽,罗 蕾,胡卫东
(电子科技大学计算机科学与工程学院,四川成都 610054)

摘 要: 在普及计算领域中,终端设备所能够使用的资源是动态变化的,要求终端具有自适应处理能力,能够根据资源的变化作出相应的响应. 文中以嵌入式操作系统为基础,提出了终端自适应处理的体系结构,详细分析了采用控制理论的自适应算法,并对系统的平衡性、稳定性和灵敏性进行了分析. 文中以多任务系统为例,对系统的自适应性进行了实验说明.

关键词: 普及计算; 自适应系统; PID 控制论; 嵌入式操作系统
中图分类号: TP316; TP273 **文献标识码**: A **文章编号**: 0372-2112 (2002) 08-1121-05

The Adaptive Technology of Pervasive Computing Terminal

LI Yun, XIONG Guang-ze, LUO Lei, HU Wei-dong
(Computer Science and Engineering College, University of Electronic Science and Technology, Chengdu, Sichuan 610054, China)

Abstract: In pervasive computing, the terminal's available resources will change dynamically and unpredictably. The terminal should be adaptive to changes in the supply of resources and respond to these changes. The paper proposes the adaptive architecture based on an embedded operating system, describes the adaptive algorithm using control theory, and analyzes the equilibrium, stability and agility of system. The paper demonstrates the adaptivity of the system according to a multi-task system.

Key words: pervasive computing; adaptive system; PID control theory; embedded operating system

1 引言

随着计算机及相关技术的深入发展,计算将不再是局限于桌面系统的分离行为,网络计算和移动计算将很快成为人们日常生活的一部分[1],并逐渐呈现出普及计算(Pervasive Computing)的模式. 普及计算是一种新型的计算模式,在该模式下,计算以人为中心,人机交互类似于人与人之间的自然交流方式(如语言,姿势,书写等),用于计算的设备无处不在,弥漫在人们生活的环境中,并能够随时随地为人们提供所需要的服务,而使用计算设备的人却感知不到计算机的存在. 普及计算是一种非常广泛的计算概念,包括了无线计算、网络计算、移动计算、日常计算,以及嵌入式计算和实时计算等侧重点不同的内容,其目的在于突破基于桌面系统的计算模式,使人们能够在随时随地的情况下获得计算服务. 在普及计算模式下,PDA 和移动电话、家用电器、可穿戴式计算机等通用设备,以及用于医疗、军事、娱乐等方面的专用设备将协同工作,以快速、高效和便捷的模式为人们提供服务. 目前,已经出现了很多不同类型的终端设备,但这些设备只具有普及计算的某些简单特征,还远未达到普及计算的需求. 为满足服务使用的便捷性,普及计算设备应该能够工作于移动环境. 因此,同固定设备相比,普及计算设备主要具有资源有限和安全性比较低等局限性. 资源有限性主要是由于终端在设计方面存在以下需求:功耗低,重量轻和物理尺寸比较小. 根据这些需求设计出来的设备所拥有的资源在以下方面存在不足:磁盘容量,物理内存大小,处理能力,缓存大小和屏幕大小等. 尽管随着技术的发展,普及计算终端的资源会得到不断增强,但固定设备同样也能从技术发展中获益,且固定设备不用考虑大小,功耗和重量等约束[2]. 因此,资源有限性是普及计算终端的固有特性,技术的发展过程也不能弥补它同固定设备之间的差距. 终端的安全性比较低,主要是由于终端的移动性造成的. 由此,普及计算终端不能作为数据的可靠存放地方. 一方面,有限的资源很难用来存储所有需要的内容,另外,从安全性考虑,也不宜把数据长期存放在终端上. 因此,数据应该保存在拥有丰富资源,并具有高安全性的服务器上,移动设备可以访问或是维护服务器上的数据,以客户端设备的形式存在.

除终端本身的局限性外,终端还受到所处网络环境的限制,网络的连接性和远程资源对终端来说都是动态变化的. 根据环境的变化,终端可能使用有线网络,更多的情况则是使用无线网络. 同有线网络相比,无线网络通常具有低带宽和高延迟的特性. 并且,由于物理实现和经济方面的考虑,无线网络由于基站信号强弱的限制,基站覆盖区域内的移动终端不可能获得统一的高质量网络服务[3,4]. 对终端来说,在移动环境中所得到的网络连接特性可能是强连接,也可能是弱连接,甚

收稿日期:2001-07-10;修回日期:2002-03-10
基金项目:国家"九五"预研项目(No. 15.3.1.2)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

至失去同网络的连接. 终端所能得到的远程资源也是变化的, 当工作于学校或办公环境时, 所得到的打印服务可能是高分辨率、彩色的, 若工作于公共场所, 得到的可能是低分辨率的黑白服务. 因此, 终端既不能作为数据的可靠存放场所, 也不能指望得到持续的高质量的网络服务.

因此, 普及计算终端的环境是变化的, 可得到的资源也是变化的, 并且这种变化是动态的, 不可预期的. 这就要求终端具有自适应的环境处理能力[5,6], 根据资源的变化做出相应的响应, 当资源丰富时, 为应用提供高质量的服务, 当资源缺乏时, 则提供降级服务.

关于自适应系统, Coda[7] 采用了系统自适应模型, S McCanne[8] 的 RLM 和 J Inouye[9] 的视频播放器则采用了应用自适应模型. 若单纯由系统进行自适应处理, 不能反映应用的特殊性; 单纯由应用进行自适应处理则失去了并行应用共享资源的统一控制点. Brian D Noble[2] 提出了图 1 所示用于移动数据访问的系统和应用相协作的自适应模型 ——Odyssey, 其中, Viceroy 用来监控资源的可获得性, 并通过 Upcall 把资源情况传送给应用进行自适应决策, Interceptor 用来处理 UNIX 文件相关的操作, Warden 用来在系统级提供与特定应用类型相关的功能, 但该结构由 4.4BSD UNIX 扩展而来, 过分依赖于 UNIX 的特有特性, 并且每类应用还需要在系统中添加相应的 Warden 处理模块, 不利于处理应用的多样性. 另外, Odyssey 还缺乏对系统在资源分配上的平衡性、稳定性和灵敏性分析. 本文分析了基于嵌入式操作系统的、平台无关的终端自适应系统的体系结构, 提出了采用 PID 控制理论的自适应算法, 分析了系统的平衡性、稳定性和灵敏性, 并通过多任务系统对系统的自适应性进行了实验分析.

2 自适应系统体系结构

2.1 具有自适应特性的嵌入式操作系统

嵌入式实时操作系统 Delta OS 是电子科技大学微机所经过"八五"和"九五"项目研究的结果, 其内核主要包括任务管理、内存管理、任务间通信、中断管理和时钟管理等模块.

为了使普及计算终端能对环境变化作出自适应处理, 需要在 Delta OS 中添加自适应处理功能, 满足终端的自适应需要. 具有自适应特性的嵌入式操作系统 Delta αOS 的体系结构如图 2 所示, 在 Delta OS 的基础上, 主要添加了三个模块: 资源监控模块, 名字管理模块和自适应处理模块. 资源监控模块监测到普及计算终端的资源发生变化时, 从名字管理模块获取关于自适应处理模块的引用, 并通过获取到的引用, 以事件对象的方式把资源事件发送给自适应处理模块, 由自适应处理模块根据系统当前的情况对资源进行处理, 处理后的资源再交给应用程序, 由应用程序根据分配到的资源数量, 作出应用特定的自适应处理决策. 体系结构的灵活性在于通过名字管理模块把资源的处理过程同资源的监控模块相分离, 使系统易于维护和扩展, 并使自适应处理模块能够方便地移植到其它系统.

Delta αOS 采用系统—应用协作自适应模型[10], 系统可以对资源的可用性进行监控, 对资源的分配进行决策, 并对终端使用的资源进行优化. 应用则可以通过系统提供的资源信息, 根据应用自身的特性进行决策.

2.2 多任务自适应应用程序

在普及计算终端中, 应用程序通常需要根据资源状况进行自适应处理, 典型的自适应策略包括: 数据压缩, 数据预取, 数据缓存, 负载分布等内容. 通过自适应策略, 为应用提供与当前系统资源情况相一致的稳定服务. 在嵌入式环境中, 应用通常被组织为任务的形式, 即系统中的应用是由一系列任务构成的, 这些任务可通过任务流模型来表示. 在任务流模型中, 任务根据输入数据, 使用一定的资源, 产生相应的输出数据. 因此, 可以把对自适应应用的考虑转化为对具有自适应特性的任务的研究, 并把这种具有自适应特性的任务称为目标任务, 对应为自适应处理模块的控制对象.

目标任务通过名字管理模块同自适应处理模块相联系, 从自适应处理模块获取所需要的资源, 并根据可获得的资源数量进行应用特定的自适应决策.

2.3 自适应处理模块

每类资源对应一个自适应处理模块, 以对系统中的某类资源进行自适应处理, 如网络自适应处理模块, CPU 自适应处理模块等. 自适应处理模块由自适应处理任务和观察任务构成, 其结构如图 3 所示.

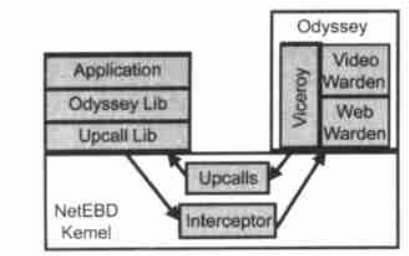

图 1 Odyssey 的体系结构

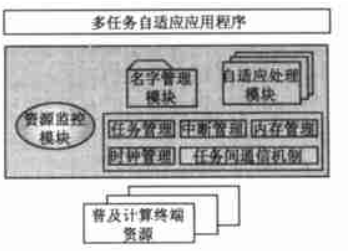

图 2 Delta αOS 系统的体系结构

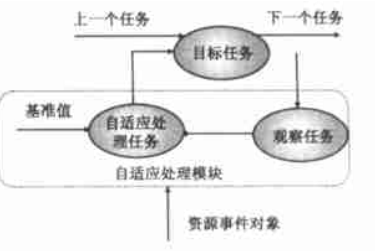

图 3 自适应控制模型

在图 3 所示的自适应控制模型中, 自适应处理模块采用了控制论的方法, 以保证资源分配的公平性和系统运行的稳定性. 自适应处理任务作为目标任务的控制器, 根据观察任务观察到的目标任务的状态, 通过控制算法确定分配给目标任务的资源数量. 目标任务的状态表示能够反映目标任务中资源消耗的动态特征, 属于目标任务的内部参数. 如, 浏览器任务中的数据传输率即为与网络带宽相对应的任务状态.

2.4 名字管理模块

每个自适应处理模块对应一个名字, 该名字用于资源监控模块和应用程序对自适应处理模块的引用. 在创建自适应

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

处理模块的时候,需要向名字管理模块注册,由名字管理模块保证自适应处理模块名字的唯一性,并为其他模块检索提供对需要的自适应处理模块的引用. 当资源监控模块需要把事件对象传送给自适应处理模块时,就通过名字管理模块获取相应自适应处理模块的引用.

2.5　资源监控模块

资源监控模块用来监控普及计算终端的移动环境,并根据从名字管理模块获取到的关于自适应处理模块的引用,把获取到的事件对象提供给相应的自适应处理模块. 当环境状态的变化情况被监测到时,监控模块创建一个新的事件对象,并把该对象传送给自适应处理模块.

2.6　自适应系统的工作过程

图 4 以普及计算终端的网络带宽资源为例,通过 UML 序列图的方式对系统工作过程进行了说明. 图中包含 Network Bandwidth、Bandwidth Resource Monitor、Name Manager Module、Bandwidth Adaptive Control 和 Browser 等五个对象,分别表示网络带宽、带宽资源监控模块、名字管理模块、带宽自适应控制任务和浏览器. 这些对象分别对应着 Resource、Resource Monitor、Name Manager、Adaptive Control 和 Target Task,分别表示普及计算终端的资源、资源监控模块、名字管理模块、自适应处理模块和表示应用的目标任务. 图 4 中的箭头表示对象之间为完成所需功能而传递的消息,箭头自上而下的排列,意味着消息发生在时间关系上的先后关系.

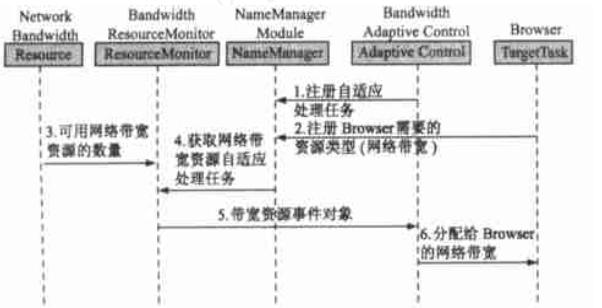

图 4　处理网络带宽资源事件对象的序列图

3　自适应资源分配的控制算法

根据输入,目标任务请求一定的资源并执行一系列操作,以产生输出信息. 如果请求的资源能够得到满足,称该请求为被接受的请求,任务成功获得该资源;如果请求暂时不能得到满足,则称该请求为未被接受的请求,任务或者在不需要该资源的情况下继续执行,或是处于等待状态,直到资源可用.

定义 1　在资源分配控制模型中,系统用 S 表示,对应一个四元组 $< R, T, D_{\text{input}}, D_{\text{output}} >$. 其中,$R$ 表示系统中可用资源的集合;T 表示系统所有任务的集合;D_{input} 表示系统的输入数据集合;D_{output} 表示系统输出数据的集合.

定义 2　在资源分配控制模型中,用 $r(r \in R)$ 表示某类特定的资源,对应一个六元组 $< c, k, x(k), M(k), A(k), N(k) >$.

其中,c 表示资源 r 的能力,用资源的最大数量表示;k 表示时间;$x(k)$ 表示在时间 k,系统中所有任务对资源 r 的请求未被接受的资源 r 的数量;$M(k)$ 表示在时间 k,系统中需要竞争资源 r 的活动任务的集合;$A(k)$ 表示在时间 k,被自适应处理的任务的集合;$N(k)$ 表示在时间 k,未被自适应处理的任务的集合. 这里,$M(k), A(k), N(k) \subseteq T$.

定义 3　在资源分配控制模型中,目标任务用 $\tau(\tau \in T)$ 表示,且每个任务 τ 对应一个四元组 $< d_{\text{input}}, d_{\text{output}}, k, r(k), u(k), w >$. 其中,$d_{\text{input}}$ 表示任务的输入数据的集合;d_{output} 表示任务的输出数据的集合;k 表示时间;$u(k)$ 表示在时间 $[k, k+1]$ 内,自适应任务的控制算法决定分配给任务 τ 的资源 r 的数量;$r(k)$ 表示任务 τ 在时间 $[k, k+1]$ 内期望得到的资源 r 的数量;w 表示任务的静态权值,用来表示任务在资源分配方面的重要性.

对于任务 τ,如果 $r(k) < u(k)$,表示任务 τ 请求的资源数量少于自适应任务对任务 τ 的资源请求数量的限制,则使 $u(k) = r(k)$,该任务属于 $N(k)$. 若 $r(k) > u(k)$,则表示需要对任务 τ 请求的资源数量进行自适应处理,称该任务为需要被自适应处理的任务,属于 $A(k)$. $r(k)$ 表示任务的内部属性,$u(k)$ 则表示自适应任务的自适应处理结果.

由此,在时间 k,系统中所有任务对于资源 r 的请求未被接受的资源 r 的数量可用式(1)来近似表示.

$$\frac{x(k) - x(k-1)}{k - (k-1)} = x(k) - x(k-1) = \sum_{i=0}^{M(k-1)} u_i(k-1) - c \tag{1}$$

式(1)表示了自适应系统内部的动态特性. 自适应控制的目的在于使 $x(k)$ 趋近一个基准值 $x^c(k)$. 根据式(1),对目标任务 τ_i,可在自适应任务中采用式(2)所示的控制算法来计算 $u_i(k)$,该算法根据 PID(Proportional-Integral-Derivative)控制[11]得出.

$$u_i(k) = u_i(k-1) + \alpha[x^c(k) - x(k)] + \beta\{[x^c(k) - x(k)] - [x^c(k-1) - x(k-1)]\} \tag{2}$$

式(2)中,α 和 β 是可配置的参数,分别对应着 PID 控制算法的积分和微分设置. 选择 PID 控制算法的原因在于:(1)根据控制论,该算法为简单线性算法,并在排除干扰和自适应的稳定性方面表现出了良好的性能. (2)该算法易于实现,计算复杂性比较低. 由于 $u(k) \leq r(k)$,式(2)中的 PID 控制算法可用式(3)来表示.

$$u_i(k) = \Psi_{r_i^{(k)}}\{u_i(k-1) + \alpha[x^c(k) - x(k)] + \beta\{[x^c(k) - x(k)] - [x^c(k-1) - x(k-1)]\}\} \tag{3}$$

其中,

$$\Psi_b(a) = \begin{cases} 0, & a < 0 \\ b, & a > b \\ a, & \text{其他} \end{cases}$$

由于 $N(k)$ 中任务的 $u(k) = r(k)$,式(1)可以表示为式(4)的形式.

$$x(k) = x(k-1) + \sum_{\tau_i \in A(k-1)} u_i(k-1) + \sum_{\tau_i \in N(k-1)} r_i(k-1) - c = x(k-1) + l(k-1)\,\bar{u}(k-1) + R(k-1) - c \tag{4}$$

其中,$l(k)$ 表示 $A(k)$ 的数量,即 $l(k) = |A(k)|$;$R(k) =$

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

$\sum_{\tau_i \in N(k)} r_i(k)$；$u_i(k) = \omega_i(k)\, l(k)\, \bar{u}(k)$.

$\bar{u}(k)$ 表示在时间 k，所有满足 $u_i(k) < r_i(k)$ 的任务 $u_i(k)$ 的的平均值. $\tilde{\omega}_i(k)$ 表示任务 τ_i 的动态权值，用来确定资源分配的优先级，且满足 $\sum_{\tau_i \in A(k)} \tilde{\omega}_i(k) = 1$. 动态权值通过任务的静态权值计算而来，计算过程如式(5)所示.

$$\tilde{\omega}_i(k) = \frac{w_i}{\sum_{\tau_j \in A(k)} w_j} \tag{5}$$

平衡性分析. 比较理想的情况为，$x(k)$ 趋近于 $x^c(k)$，而 $u_i(k)$ 则在对应资源的竞争任务间保持资源分配的公平性. 根据文[12]，如果在特定的时间周期 $[k^1, k^2]$ 内，$x^c(k)$、$l(k)$、$M(k)$ 和 $R(k)$ 是稳定的，分别表示为 x_s^c、l_s、M_s 和 R_s，则有：

定理 1 在时间 $[k_1, k_2]$ 内，由式(3)和式(4)确定的系统中，未被接受请求的资源数量 x 趋近于一个平衡值 x_s^c. 并且，系统根据任务的静态权值，能够在各竞争资源的任务间公平地共享资源. 稳定性分析. 自适应系统的稳定性主要体现在两个方面：(1)在多任务环境中，多个目标任务同时共享有限可获得的资源，在一个特定的时间周期内，如果活动任务的数量保持不变，则分配给每个任务的资源数量应该趋近于一个稳定值. 这也意味着，如果有新的任务变为活动任务，已存在的活动任务应该调整相应的资源使用情况，以确保系统能在比较短的时间范围内达到新的平衡. (2)稳定性也意味着即使资源的可获得性出现了不可预测的变化，如，无线通讯网络在发生短暂信号衰减时，会导致网络暂时性地失去连接，而自适应行为不会出现振荡现象. 在这种情况下，如果根据信号的短暂强弱来作为判断网络连接的依据，其结果将不会满足用户的需要，因为信号很快又会恢复到能够进行正常通信的水平. 并且，如果频繁地对瞬时变化进行处理，也会导致系统处于一种不稳定的状态. 振荡是不期望出现的现象，因为可能导致过量的自适应行为，从而导致占用过多资源而使系统超载. 为了使系统达到平衡，需要证明系统是稳定的. 根据文[12]，有：

定理 2 由式(3)和式(4)确定的系统中，对于任务 τ_i 来说，在 $\alpha > 0$，$\beta > 0$，且 $\alpha + 2\beta < 4\tilde{\omega}_i$ 的情况下，在一个局部的时间区间内是趋于稳定的.

推论 对于 α 和 β，如果 $\alpha > 0$，$\beta > 0$，且对于 $\forall_i, k$，有

$$\alpha + 2\beta < 4\frac{w_{\min}}{\sum_{\forall i} w_i} < 4\frac{w_i}{\sum_{\tau_i \in A(k)} w_j} \tag{6}$$

则系统中的所有任务是稳定的. 式(6)中，$w_{\min}$ 表示所有的最小值. 自适应系统的灵敏性分析. 除稳定性需求外，也期望系统能够快速地对资源可获得性的变化作出响应. 自适应的灵敏性定义为自适应系统对资源可获得性进行检测和响应的速度和准确性，即自适应系统对环境动态变化的敏感度. 在任务控制模型中，自适应的敏感性由控制算法 PID 的配置参数确定. 在满足定理 2 及其推论的情况下，可以对式(3)所描述的控制算法中的参数 α 和 β 进行配置，具体数据由系统对敏感性的需求来确定. PID 控制算法的可配参数 α 和 β 的选择策略为：(1) α 和 β 的选择应该满足定理 2 和相应的推论；(2) α 控制瞬时响应速度，α 越大，系统越灵敏，但将导致系统稳定性的降低；β 控制系统的稳定性，β 越大，系统越稳定.

4 实验分析

在单处理器环境下的多任务系统中，多个任务竞争共享的 CPU 资源. 为此，可以使用自适应系统的控制算法对 CPU 的分配进行控制，确保应用任务能够公平地获取 CPU 资源，同时保证系统在资源分配上的稳定性和灵敏性.

实验中，系统中存在四个应用任务：三个模拟任务和一个目标任务. 模拟任务为非周期任务，用来模拟除目标任务以外的多个任务竞争 CPU 资源的情况；目标任务则为自适应系统的控制对象，称为 τ_1. τ_1 为具有自适应处理能力的周期任务，周期为 400ms，最长计算时间为 200ms，其自适应处理策略为：

```
if r_cpu is equal to 0.5, then ComputationPath is p1
else if r_cpu is equal to or greater than 0.375, then ComputationPath is p2
else if r_cpu is equal to or greater than 0.25, then ComputationPath is p3
else if r_cpu is equal to or greater than 0.125, then ComputationPath is p4
```

其中，r_{cpu} 为 τ_1 实际获得的 CPU 资源的数量；p_1，p_2，p_3 和 p_4 为中四条不同的执行路径，对应为 τ_1 需要完成的操作的四个重要性等级，执行时间分别为 200ms，150ms，100ms 和 50ms. 在上述自适应策略中，τ_1 根据获得的 CPU 资源数量选取不同的执行路径. 根据文[13]，任务得到的 CPU 资源的数量可以定义为：在一段时间内，任务获得的处理时间与实际间隔时间的比值. 正常情况下，τ_1 的执行情况如图 5(a)所示，每 400ms 执行一次. 但在有新任务加入或其它不确定因素影响的情况下，τ_1 可能出现图 5(b)所示的情况，τ_1 在新周期开始 10s 时才得到调度，其截止时间也不能得到满足.

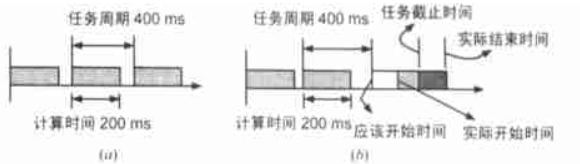

图 5 目标任务执行情况

为使任务能够得到完整运行，τ_1 需要的 CPU 资源的数量为：$R_{cpu} = \frac{T}{P} = \frac{20}{40} = 0.5$，其中，$R_{cpu}$ 为 τ_1 在正常情况下所需要的 CPU 资源的数量；T 表示任务的计算时间；P 表示任务的周期. τ_1 在运行过程中，每个运行周期能够实际获得的 CPU 资源数量为：$r_{cpu} = \frac{(T - B)}{P}$，其中，$r_{cpu}$ 为 τ_1 实际获得的 CPU 资源数量；T 为任务计算时间；B 为相对于任务周期开始时间的任务执行开始时间；P 为任务周期. 对于图 5(b)的情况，$r_{cpu} = 0.25$，此时，τ_1 请求未得到满足的 CPU 资源的数量为：$x_{\tau_1} = R_{cpu} - r_{cpu} = 0.5 - 0.25 = 0.25$. 因此，在时间 k，所有应用任务请求未得到满足的 CPU 资源的数量可表示为：$x(k) = \sum_{\tau_i \in \text{应用任务}} x_{\tau_1}(k)$. 设定应用任务在资源分配上的优先级相同，为 0.25，且对任务 τ_1，设控制算法的 $\alpha = 0.04$，$\beta = 0.05$，满足

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

定理 2 中的系统稳定性要求. 在 0～4000ms,自适应控制算法分配给任务 τ_1 的 CPU 资源数量 u_{τ_1} 如表 1 所示,图 6 为其图形表示. 图 7 为根据自适应控制算法分配的 CPU 资源的数量,任务 τ_1 进行自适应处理后所选择的执行路径.

表 1　自适应控制算法分配给任务 τ_1 的 CPU 资源的数量

参数 \ k	0-400	400-800	800-1200	1200-1600	1600-2000	2000-2400	2400-2800	2800-3200	3200-3600	3600-4000
x^c	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
x	0.2	0.4	0.8	0.8	0.6	0.5	0.7	0.7	0.4	0.4
u_{τ_1}	0.5	0.46	0.41	0.38	0.37	0.36	0.33	0.31	0.32	0.31

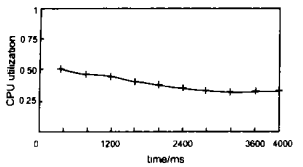

图 6　分配给任务 τ_1 的 CPU 资源的数量

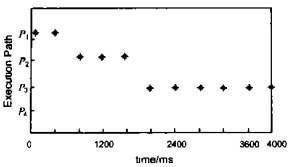

图 7　任务 τ_1 的执行路径

对于图 6 所示的资源情况,如果任务不采取自适应措施,其执行情况将不能反映系统中各个应用任务对 CPU 资源的使用情况,并将在多个周期内导致任务 τ_1 大大超过截止时间的要求. 由图 7 可以看出,在采用自适应处理策略后,任务 τ_1 能够根据当前 CPU 资源的可获得情况选择不同的执行路径,从而较好地满足任务 τ_1 对截止时间的要求. 采用自适应处理后,任务 τ_1 既能保证在资源比较丰富的情况下能够执行任务的完整操作,也能在资源不足时改善任务的截止时间满足情况,从而获得比较稳定的系统性能.

5　结束语

普及计算终端的环境是不断变化的,可得到的资源也是变化的,并且这种变化是动态的,不可预期的. 这就要求终端具有自适应的环境处理能力,能够根据资源的变化做出相应的响应. 本文根据系统 - 应用协作自适应模型提出了自适应系统的体系结构,详细描述了结构中各个组成部分的功能原理. 该结构主要包含资源监控、名字管理和自适应处理等模块,其目的在于为普及计算终端提供一个平台无关的、可伸缩的资源自适应处理结构,从而使得设计出的应用能够工作于各种普及计算终端. 文中还提出了采用控制论的 PID 控制模型进行系统自适应处理的方法,从理论上保证了系统在资源分配上的平衡性、稳定性和灵敏性. 文中以多任务系统为例,对体系结构进行了实验分析. 通过实验说明,在资源分配的自适应控制下,具有自适应处理策略的目标任务在所需要的资源发生变化时,能够获得比较稳定的系统性能.

通过本文的研究工作,对设计具有自适应特性的嵌入式操作系统和自适应应用,以及开发普及计算终端设备都具有重要的指导意义.

参考文献:

[1] Randy H Katz. Adaptation and mobility in wireless information systems [J]. IEEE Personal Communications, 1995, 1: 6 - 17.

[2] Brian D Noble. Mobile data access [D]. Carnegie Mellon University, 1998. http://www.cs.cmu.edu/ats/cs/project/coda/Web/docdir/bnoble-thesis.pdf.

[3] J Mello, P Wayner. Wireless mobile communications [J]. Byte Magazine, 1993: 147 - 154.

[4] 袁东风,王承祥,姚旗. 移动衰落信道中多级编码的多级译码与并行译码的性能比较 [J]. 电子学报, 2001, 29(4): 560 - 562.

[5] Vaduvur Bharghavan, Vijay Gupta. A framework for application adaptation in mobile computing environments [A]. Computer Software and Applications Conference '97 [C]. Washington: IEEE, 1997. http://citeseer.nj.nec.com/bharghavan97framework.htm1.

[6] Dushyanth Narayanan, Jason Flinn, M Satyanarayanan. Using history to improve mobile application adaptation [A]. In Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems and Applications [C]. Monterey, CA: 2000. 61 - 66.

[7] M Satyanarayanan, J J Kistler, P Kumar, M E Okasaki, E H Siegel, D C Steere. Coda: A highly available file system for a distributed workstation environment [J]. IEEE Transactions on Computers, 1990, 39(4): 447 - 459.

[8] J Inouye, S Cen, C Pu, J Walpole. System support for mobile multimedia applications [A]. In Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video [C]. St. Louis, MO: 1997. 143 - 154.

[9] S McCanne, V Jacobson, M Vetterli. Receiver-driven layered multicast [A]. In Proceedings of the ACM SIGCOMM '96 Conference [C]. Stanford CA: 1996. 117 - 130.

[10] M Satyanarayanan. Fundamental challenges in mobile computing [A]. Fifteenth ACM Symposium on Principles of Distributed Computing [C]. Philadelphia, PA: 1996. 1 - 7.

[11] 李人厚. 智能控制理论和方法 [M]. 西安:西安电子科技大学出版社, 1999.

[12] Baochun Li, Klara Nahrstedt. A control theoretical model for quality of service adaptations [A]. In Proceedings of IEEE International Workshop on Quality of Service [C]. Napa, CA: 1998. 127 - 136.

[13] Clifford W. Mercer, Stefan Savage, Hideyuki Tokuda. Processor capacity reserves: operating system support for multimedia applications [A]. In Proceedings of the IEEE International Conference on Multimedia Computing and Systems [C], Boston, MA: 1994. 90 - 99.

作者简介:

李　允　男,1971 年 9 月出生于重庆开县,1993 年 7 月于西南石油学院获得计算机专业学士学位,1999 年 4 月于电子科技大学获得计算机软件与理论专业硕士学位,1999 年 4 月于电子科技大学攻读计算机应用博士学位至今,长期从事嵌入式操作系统和嵌入式应用方面的研究工作,先后从事过国防科技"九五"预研项目实时并发软件可靠性测试评价系统和超微内核嵌入式实时操作系统的研究工作,并结合项目成果转化,进行了移动终端软件平台的实际产品研制过程. Email: liy@mail.coretek.com.cn.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net