

文章编号:1001-9081(2006)03-0692-03

## 支持网络传感器的嵌入式操作系统设计

王鸚鵡, 郑扣根, 瞿心杭

(浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

(ywwangzju@yahoo.com.cn)

**摘要:** 简要阐述了普适计算环境下网络传感器应用的一些特点, 针对这些特点设计了一个以网络传感器应用为目标的微型嵌入式操作系统  $\gamma$ OS。阐述了  $\gamma$ OS 的几个特点: 支持低能耗的微内核、微线程的系统架构、组件化的功能设计、支持传感通信的接口等。应用  $\gamma$ OS 机制, 建立了一个典型应用平台, 在该平台上简单分析了  $\gamma$ OS 的性能。

**关键词:** 普适计算、嵌入式操作系统、组件化、 $\gamma$ OS

**中图分类号:** TP316.89 **文献标识码:** A

## An embedded operating system designed for network sensors

WANG Ying-wu, ZHENG Kou-gen, QU Xin-hang

(College of Computer Science and Technology, Zhejiang University, Hangzhou Zhejiang 310027, China)

**Abstract:** The characteristics of network sensor application in ubiquitous computing environment were analyzed. Based on the analysis, an embedded operating system,  $\gamma$ OS, was designed aiming at network sensor application.  $\gamma$ OS had following features: low power consuming micro kernel, micro-threading architecture, component-based design and sensor-interaction interfaces. A typical application platform was set up applying  $\gamma$ OS, and the performance of  $\gamma$ OS on the platform was also analyzed.

**Key words:** ubiquitous computing; embedded operating system; component-based;  $\gamma$ OS

### 0 引言

网络传感器是集传感器技术、嵌入式计算技术、现代网络及通信技术、分布式信息处理技术于一身的资源受限的嵌入式设备, 是“普适计算”<sup>[1]</sup>在微型嵌入式领域的一种重要应用模式。

网络传感器的研究过去一直受限于硬件平台而发展缓慢。随着半导体技术、通信技术、微电子技术和微机械技术的不断进步, 低功耗、低价格、多功能的传感器网络系统得到了快速发展, 使得制作微小、有弹性、低功耗的传感器节点成为现实。

### 1 背景

#### 1.1 应用特点

网络传感器应用有其自身的特点, 主要有以下几个方面: 小尺寸和低功耗、并发密集操作、有限的物理并行性和控制层次、多样化的设计和使用。一方面, 传感器资源极其有限, 给底层嵌入式程序设计带来较大的限制; 另一方面, 传感器上运行的应用程序和系统内核通常是紧密结合在一起的, 且运行时需要的任务数量、执行时间、执行结果以及内存消耗等是可以较好预计的。

此外, 传感器种类繁多, 针对不同应用场合需要不同种类的传感器; 在军事应用、空间探索等特定应用场合下, 更是需

要大量的不同种类的传感器协同合作来完成特定应用事件。因而传感器上运行的软件系统如果能够具备相对较好的灵活性、可配置性和可重用性, 将能更好地满足应用需求。

#### 1.2 现有嵌入式 OS 比较

当前存在众多的嵌入式操作系统, 其中具有代表性的如 Vxwork, Windows CE, pSOS 和 Neculeus 等, 它们的优点是: 功能强大; 具有丰富的 API 和嵌入式应用软件; 具备良好的实时性能, 尤以 Vxwork 为代表; 具备良好的稳定性。缺点是: 价格昂贵; 源代码不公开, 以及由此导致的诸如对设备的支持、应用软件的移植等一系列的问题; 另外对于传感器器件来说, 这些嵌入式 OS 都显得过于“庞大”了一些。

uc/os 和嵌入式 Linux 当前正获得越来越广泛的应用。它们的优点是: 执行效率高、占用空间小、可扩展性能好, 同时是免费且源代码公开的。uc/os 具备良好的实时性能, 嵌入式 Linux 的实时性能有待进一步提高。缺点是: 它们都是相对通用的嵌入式操作系统, 不能完全适应传感器应用领域的需求, 如嵌入式 Linux 最小仍然需要上百 K 的 ROM 和 RAM 空间才能工作, 而 uc/os 的内核尽管可缩减至几 K, 但是对于某些传感器应用来说, 仍然显得不够精简。

UC Berkeley 设计开发了无线传感器网络应用的嵌入式操作系统 TinyOS<sup>[2]</sup>以及系统编程语言 nesC<sup>[3]</sup>。我们在剖析现有嵌入式 OS 特别是 TinyOS 的基础之上, 设计实现了支持网络传感器的微型嵌入式操作系统  $\gamma$ OS, 并开发了系统编程

收稿日期: 2005-09-12 修订日期: 2005-12-29 基金项目: 国家 863 计划项目(2003AA1Z2080)

作者简介: 王鸚鵡(1980-), 男, 浙江台州人, 硕士研究生, 主要研究方向: 嵌入式系统、编译器、Linux; 郑扣根(1964-), 男, 江苏镇江人, 教授, 主要研究方向: 操作系统、嵌入式 Linux、计算机网络、地理信息系统、人工智能; 瞿心杭(1975-), 男, 浙江嘉兴人, 硕士研究生, 主要研究方向: 嵌入式系统、计算机网络。

语言 AntC<sup>[4]</sup>。

### 2 $\gamma$ OS 设计

$\gamma$ OS 是以网络传感器应用为目标的微型嵌入式操作系统,针对网络传感器的前述应用特点, $\gamma$ OS 的设计具备几个特性:支持足够微小的硬件系统,便于传感器设备在检测环境中的任意撒布;支持足够低的系统功耗,保证传感器设备具备足够长的生命期;支持集成可与物理世界交互的传感设备,实现数据的采集和传输;同时兼顾适度灵活的可重用性、可配置性。 $\gamma$ OS 还必须解决传感器网络的两个突出问题:

- 1) 由于网络传感器操作的并发密集性,因而必须保证众多不同数据流的并发即时传输;
- 2) 系统必须提供高效的模块化管理策略,具体硬件设备和具体应用组件必须紧密地结合在一起,减小处理和存储开销。

为此, $\gamma$ OS 的设计重点主要集中在以下几个方面:低能耗的微型内核;微线程的系统架构;组件化的功能设计;支持传感通信的接口。

#### 2.1 支持低能耗的微型内核

为了降低能耗, $\gamma$ OS 设计了一种相对简单的内核机制,它由两部分组成:系统初始化代码以及一个微小的核心调度组件。系统初始化代码具有平台相关性;核心调度组件实现基于优先级的两级调度机制,它分别由两个调度队列组成:事件队列和任务队列。事件队列优先级高于任务队列的优先级,每个队列内部基于 FIFO 调度机制。

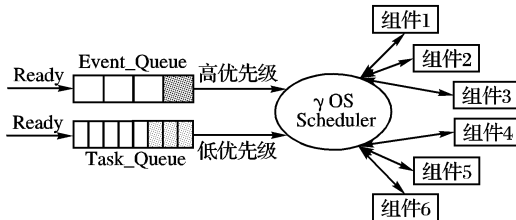


图1  $\gamma$ OS 的内核调度示意图

此外,为了降低能耗,在借鉴 TinyOS 的能耗管理算法<sup>[5]</sup>的基础之上,设计实现了能耗控制组件:动态电源管理 DPM 组件和动态电压调整 DVS 组件。

#### 2.2 微线程的系统架构

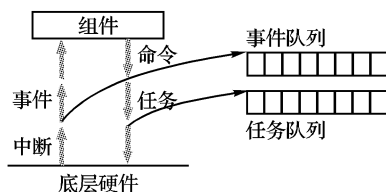


图2 微线程的系统架构

如图2所示, $\gamma$ OS 的通过组件来实现基于事件驱动模式的微线程系统架构,采用事件触发去唤醒相应的功能组件工作。每个功能组件可以由以下几个部分组成:

事件处理函数 用以实现对底层硬件中断的处理,如 MCU 外部中断、定时器中断等,它可以向核心调度组件提交任务,但并不等待任务的执行。事件优先级高,可抢占任务执行,可以传递。它提供了一个简明的方法用于抽象软硬件之间的边界,使得支持硬件中断变得非常简单。

命令 用以执行对底层组件的操作,是非阻塞的,且必须向调用者返回命令执行的结果(成功或失败)。

任务 用于表示组件中计算相对集中的一组操作。任务不具有抢占性,任务与任务之间是原子化的,以先进先出的方式执行,即一个任务必须执行完之后才能执行下一个任务。但任务可以被事件处理函数抢占。

组件状态 用以表示组件当前的工作状态,可以被自己的功能函数或其他组件所参考。

利用微线程的系统架构, $\gamma$ OS 可有效降低上下文切换代价;同时,通过引入原子语句来处理任务和事件,甚至事件和事件之间的并发操作,实现微线程异步通讯机制,有效地避免阻塞、轮询和数据资源竞争。

#### 2.3 组件化的功能设计

在特定应用场合下,需要大量的不同种类的传感器协同合作来完成特定应用事件,因而传感器上运行的软件系统具备相对较好的灵活性和可配置性。

为此, $\gamma$ OS 提供了对组件化的功能设计方式的支持。 $\gamma$ OS 可分解为一个核心调度组件和若干功能组件。现有的功能组件主要包括:能耗控制组件如动态电源管理 DPM 组件和动态电压调整 DVS 组件, AntIP 协议(支持微型嵌入式 TCP/IP 协议)组件, USB 驱动组件,网卡驱动组件和 XML 分析器组件等。

根据不同应用配置不同的功能组件,以实现特定的目标。 $\gamma$ OS 支持静态配置和动态加载两种方式。静态配置组件最少可只包括一个核心调度组件,而其他的功能组件可根据相应的应用需求选择预先静态配置方式或者动态加载方式。

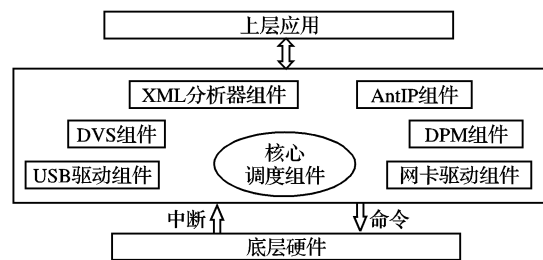


图3  $\gamma$ OS 组件示意图

#### 2.4 支持传感通信的接口

$\gamma$ OS 通过 AntIP 组件实现对传感器通信的接口支持,主要有支持传感器节点间的对等 (Peer-To-Peer) 通信和组播通信模式,支持传感器节点与 PC 间的对等通信模式和支持基于事件的异步通信处理模式。

AntIP 是一个适用于 8/16 位机的微型嵌入式 TCP/IP 协议栈,它尽管去掉了许多全功能协议栈中不常用的功能,但仍然保留了网络通信所必要的协议机制,支持 ARP, IP, ICMP, TCP, UDP 等协议,并且提供了简易的应用层接口和设备驱动层接口。AntIP 的设计借鉴了 uip<sup>[6]</sup> 的设计思想。

### 3 典型应用

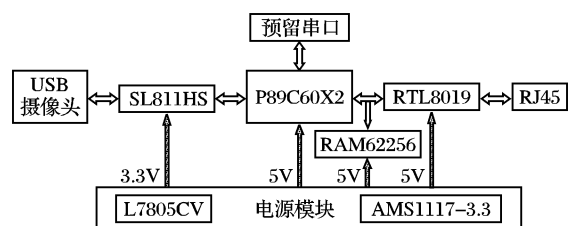


图4 硬件平台示意图

$\gamma$ OS 是以网络传感器应用为目标的,它可以运行在多种

目标传感器上。我们采用  $\gamma$ OS 机制,针对图像数据采集方面的应用,设计了一套较典型的网络图像传感器系统。

该系统主要由主控模块、存储模块、USB host 模块、以太网模块、摄像头模块和串口模块(预留接口)组成,采用的芯片分别为 Philips 公司的 P89C60X2(80C51 芯片)、USB 控制芯片 SL811HS、RAM 芯片 62256 和 NIC 芯片 RTL8019AS。

平台部分初始化代码(AntC 语言)如下:

```
use SL811HS
use Camera
use RTL8019
class Platform
{
public static int Init()
{
    RTL8019. Init();
    if (SL811HS. UsbInit() == FALSE)
        return -1;
    if (Camera. CameraInit() == FALSE)
        return -1;
    if (Camera. CameraStart() == FALSE)
        return -1;
    AntIP. Init();
    return 1;
}
public static void Start()
{
    post AntIP. Run;
}
public static void main()
{
    Init();
    Start();
}
}
```

该部分代码做了网卡模块、USB 模块以及摄像头模块的初始化工作。

(上接第 687 页)

当迁移实例提出资源服务请求后,MIRequestDispatcher 捕获请求对象,根据访问协议和访问策略,按顺序回调 InterceptorChain 中的各截取器方法,各截取器调用相应服务组件进行相应处理,如安全检测,日志记录等。之后请求对象将被送至 ServiceChooser,ServiceChooser 根据本地规则将它送至合适的目标实体 Service Agent。ServiceChooser 的设计采用了策略模式,可根据具体需要,设计多种不同的分配策略。

## 5 结语

本文基于迁移 workflow 服务的概念,提出了一种停靠站服务器的结构模型,重点介绍了该模型的体系结构和内核结构,分析了其服务集成框架和组织模式。该模型支持运行期间服务的定制、动态扩展和重置,具有良好的柔性特性,克服了停靠站服务器缺少灵活性、扩展性,难以适应部署环境和应用需求的动态变化的不足。同时,通过分离迁移、通信、安全等功能服务,减少这些服务与停靠站服务器的耦合程度,可以裁减不必要的服务,节省系统资源;也使得停靠站服务器能够很

## 4 结语

采用  $\gamma$ OS 的网络图像传感器的各模块的代码量及所需数据空间大小如表 1 所示。从该表中可看出  $\gamma$ OS 的核心代码量基本接近 TinyOS 的核心代码量。

表 1 模块代码量与所需数据空间

Component name	Code Size/b	Xdata Size/b
Camera	2551	355
RTL8019	1329	280
SL811HS	4516	751
AntIP	5825	544
Platform_init	349	267
$\gamma$ OS scheduler	236	35
Total	13726	2232

在这篇文章中,我们简要阐述了网络传感器在普适计算环境下的应用特点,介绍了一个以网络传感器为应用目标的嵌入式操作系统  $\gamma$ OS 的设计和几个特点,并建立了一个典型应用平台,最后简要给出了  $\gamma$ OS 在该平台上的性能。

### 参考文献:

- [1] WEISER M. Hot Topics: Ubiquitous Computing [J]. IEEE Computer, 1993, 26(10): 71-72.
- [2] [http://www.tinyos.net/\[EB/OL\]](http://www.tinyos.net/[EB/OL]), 2004-03.
- [3] GAY D, LEVIS P, von BEHREN R, et al. The nesC language: A holistic approach to net-worked embedded systems[A]. ACM SIGPLAN Conference on Programming Language Design and Implementation [C], 2003.
- [4] LUO X, ZHENG K, WU Z, et al. A systems programming language for wireless networked embedded systems[A]. The 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT05) [C]. Dalian, China, 2005.
- [5] HOHLT B, DOHERTY L, BREWER E. UCB//CSD-03-1293, Flexible Power Scheduling for Sensor Networks[R], 2003.
- [6] [http://www.sics.se/~adam/uip/\[EB/OL\]](http://www.sics.se/~adam/uip/[EB/OL]), 2004-11.

容易替换这些服务组件,集成第三方产品。

目前,该停靠站服务器结构模型已成功应用于 Migflow 2.0 的开发。Migflow 2.0 是我们在 Migflow 1.0<sup>[3]</sup> 的基础上自行开发的迁移 workflow 管理系统,主要应用于迁移 workflow 系统模型理论和系统方法验证。

### 参考文献:

- [1] 曾广周,党妍. 基于移动计算范型的迁移 workflow 研究[J]. 计算机学报, 2003, 26(10): 1343-1349.
- [2] 杨公平,曾广周,卢朝霞. 迁移 workflow 系统中停靠站服务器的设计与实现[J]. 计算机工程与应用, 2004, 40(19): 111-112.
- [3] BUSCHMANN F, MEUNIER R, ROHNERT H, et al. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns [M]. England: John Wiley & Sons Ltd, 1996.
- [4] SCHMIDT D, STAL M, ROHNERT H, et al. Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects [M]. England: John Wiley & Sons, Ltd, 2000.
- [5] 张鹏,曾广周. 基于 Java 的迁移 workflow 管理系统原型实现研究[J]. 计算机应用研究, 2004, 21(11): 51-53.