DOI:10.19414/j.cnki.1005-1228.2011.02.023

文章编号 :1005-1228 (2011) 02-0064-03

两种开源嵌入式操作系统的比较

王 超,孟祥娟,黄宇博
(新疆医科大学高等职业技术学院,乌鲁木齐市 830054)

摘 要 :由于嵌入式系统处理能力大幅度提升,需求的功能也越来越复杂,特别是在网络以及多任务方面。单片机式的开发方式渐显落后,迫切需要用嵌入式操作系统来提高效率。文章主要介绍了目前较流行的两种较成熟的嵌入式操作系统:uClinux 和 eCos,方便相关的开发人员参考和选型。

关键词 :嵌入式 ;操作系统 ;uClinux ;eCos
中图分类号 :TP316 文献标识码 :A

The Comparison of Two Open- source Embedded Operating System

WANG Chao,MENG Xiang- juan,HUANG Yu- bo
(Vocational and Technical college of Xinjiang Medical University, Urumqi 830054,China)

Abstract :As embedded systems greatly enhance the capacity, demand functions are more complex, especially in terms of network and multi- task. MCU- style way of starting to prove behind the development of the urgent need for an embedded operating system to improve efficiency. This paper describes the current more mature than the two popular embedded operating system: uClinux and eCos. Related development to facilitate reference and selection of staff.

Key words: embedded; operating system; UClinux; ECos

1 eCos 和 uClinux 简介

eCos 和 uClinux 都是当前被广泛应用且比较成熟的嵌入式操作系统。相对于 uC/OS,它们是真正的免费且公开源码,因为 uC/OS 的网络协议栈不是开源的,商用是需要收取费用的。eCos 适合小型控制系统,微内核结构优秀,eCos 最小版本只有几百个字节,占用空间小、运行效率高、而且实时性能优良,可扩展性强,支持多种 API。一般一个完整的网络应用,其二进制代码也就 100KB 左右。uClinux 是 Linux 在嵌入式领域的第一代产品,针对嵌入式处理器的特点对原始 linux 进行很大的裁剪,主要裁去了任务管理、内存管理等组件,但也保留了 TCP/IP 协议栈,支持多种文件系统。编译后的目标文件可控制在几百 K 的量级[1]。

1.1 两种嵌入式操作系统主要性能比较

和通用操作系统一样,嵌入式操作系统也是一个作业管理系统,所以拥有系统软硬件资源的最高调度权,核心任务是以高效的方法管理多个应用软件,同时使用 CPU 等共享资源,而且不能发生逻辑错误。为了可以方便和 PC 机共享数据,也越来越需要嵌入式系统支持常用的文件系统。

嵌入式操作系统实际应用中最常出问题的地方有 :系统移植、进程调度和文件系统支持。这些问题如果处理不好,就发挥不了嵌入式操作系统的优势,甚至会让使用者觉得还不如裸机开发容易,因此我们就这几个方面,对 eCos 和 uClinux 进行分析比较。

1.2 进程调度

进程调度也被称为任务调度或 CPU 调度,因为操作系统中的任务都是断断续续执行的,为了描述这种状态,我们将任务称为进程。通常启动一个应用软件就是操作系统新建一个进程,我们开发时至少要启动一个进程。进程调度模块主要是协调各个任务对系统内资源 (如内存、I/O 设备、CPU) 的争夺使用,因为这些资源都是紧缺的,每个时间点只能服务于一个任务,来提高计算机效率,是多道程序操作系统的基础。

进程调度有“抢占式”和“非抢占式”两种基本方式。前者,操作系统可以按优先级或其他规则“抢占”或不“抢占”某个运行的进程 ;后者,每个进程都有自己的

收稿日期 :2011- 01- 06
作者简介 :王 超 (1980-),男,新疆乌鲁木齐人,讲师,硕士研究生,研究方向 :计算机软件开发 ;通讯作者 :孟祥娟 (1970-),女,新疆乌鲁木齐人,副教授,研究方向 :计算机应用。

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

时间片，时间一到，操作系统就将资源分给下一个时间片的进程[2]。

很明显，实时操作系统必须使用抢占机制，且要保持最小的中断延迟，就是最好的响应速度，而且不能造成逻辑错误。为了实现这个目的，需要使用必要的同步原语，最好支持常用的多种调度机制。

eCos 具有两个调度器，即两个调度管理模块，位图调度器和多级队列调度器，前者设置若干个不同的线程优先级。线程个数有严格的限制。这种调度器非常适合小型管理系统，效率会很高。多级队列调度器具有优先级的 FIFO 调度策略，时间片轮转的策略，时间片大小可通过配置工具设置，这种策略较复杂，不过能保证进程的公平，不会使优先级低的进程饿死，也不会使实时性能降低。

uClinux 的进程调度沿用了 Linux2.0 内核，采用“非抢占式”时间片轮转的调度策略，时间片的轮转由时钟计时中断控制，系统每隔一定时间会产生一个时钟中断，来挂起正在运行的进程，并决定进程什么时候拥有它的时间片。用户需要编程完成进程切换，父进程调用 fork 函数生成子进程，这时 CPU 由子进程占据，父进程被挂起。如果父进程还要生成子进程，fork 调用完成后，还可以使用 exec 系统调用生成一个新的进程，子进程代替父进程执行，直到子进程 exit 退出。但是父进程不能自动唤醒，需要子进程使用 wakeup 原语把父进程唤醒[3]。

uClinux 没有 MMU 模块，需要所有程序直接访问实际的物理地址。各个进程实际上共享一个内存空间，需要每个进程自己保护自己的数据，甚至用户程序会误操作使用到系统内核空间，造成整个 uClinux 的崩溃。这样确实增加了开发人员的工作难度。这种开发方式退回了 Dos 系统时代。开发人员必须参与系统的内存管理。从编译内核代码开始，开发人员必须在源码中配置系统开发平台上拥有多少的内存，这样系统在启动的初始化阶段才能对内存进行正确的分页动作，标记好系统已使用以及未使用的内存地址，应用程序启动时就能正确的使用这些分页内存了[3]。

由上述分析可以得知，eCos 内核是针对实时系统的要求设计实现的，使用相对简单，可以满足较高的实时性要求。而 uClinux 则在结构上继承了标准 Linux 原始的多任务实现方式，实时性能也较差。

1.3　操作系统的移植性

嵌入式操作系统不是通用系统，不同的硬件环境需要对嵌入式操作系统进行必要的配置和修改才能正常运行，这个配置和修改的过程就是“移植”。eCos 和 uClinux 都是完全公开源码的嵌入式操作系统，且它们都是模块化的设计，而且主体代码都是用 C 语言编写的，完全可以把与 CPU 相关的部分分离出来，所以被移植到新的处理器上是完全可行的。以下对两种系统的移植分别予以说明。

(1) eCos 的移植

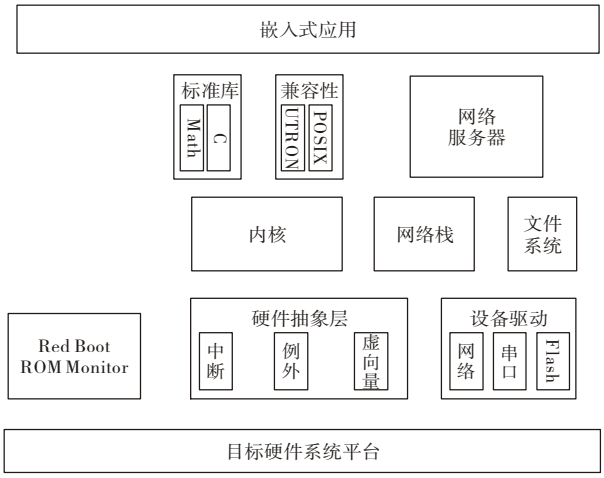

图 1　eCos 结构图

图 1 为 eCos 结构图，eCos 其核心组件包括：内核（kernel），硬件抽象层（Hardware Abstract Layer，HAL），标准 C 和数学库，GNU 调试器（GDB），设备驱动程序。这些组件分别位于系统的不同层次，具有可重用性。用户可以根据需要，创建一个最精简的 eCos 镜像，只要用专用的配置工具去掉相应的功能组件即可。

硬件抽象层（HAL）处于 eCos 层次结构的最底层，如其英文缩写所述，抽象不同的处理器，对上层模块提供统一的接口，它是 eCos 系统移植的关键。根据微内核系统的设计原则，HAL 层又被抽象为平台抽象层、变体抽象层和体系结构抽象层，这些层次来描述不同处理器的差别和联系，处理器制造时就有不同的架构和不同的系列来满足不同市场的需求，例如 ARM 系列、PowerPC 系列等，不同的公司在相同架构下又有不同的产品。

平台抽象层（Platform HAL）移植，该子层次主要抽象某种架构处理器，当这种架构的处理器出现变体产品，结构变化不大，主要是性能提升较多时，就可以认为还是在使用原来已经使用过的处理器，一般修改初始化代码中的内存布局等就能完成。

变体抽象层（Variant HAL）移植，该子层抽象了同类体系结构的普通 CPU 与当前 CPU 之间的明显差异，当前处理器如果变化不是很大，就不用这方面的抽象。该层次的移植通过对系统中断、高速缓存及其他特性的重定义来覆盖平台抽象层中的默认实现[4]。

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

体系结构抽象层(Architecture HAL)移植。该层次的移植实际上就不是移植，是对 eCos 代码的一次扩充,因为 eCos 还没有支持我们要使用的处理器。我们的工作是添加代码而不是修改配置已有的代码。但同样可以选择相近的 HAL 作为参照模板,但是每种体系需要专门的 eCos GNU C/C++ 编译器支持,若已有的编译器不支持新体系结构，则需要先进行新体系编译器的移植。

当抽象的角度不同时，硬件抽象层的这 3 个子模块之间没有很明显的界限。对于不同的目标平台,这种区分具有一定的模糊性，相同的代码可以被定义到不同的抽象层,但都是封装成不同的包来加以实现。

⑵ uClinux 的移植

由于 uClinux 其实是 Linux 针对嵌入式系统的一个精简版本,虽然也是模块化的设计,但结构还是比较复杂,毕竟脱胎于通用操作系统,其移植也复杂得多。主要体现在 uClinux 支持的处理器有限，主要是通用处理器,而且通用处理器要比嵌入式处理器复杂得多,需要处理较多的代码冗余，还需要具有较大容量的存储空间,包括外部 ROM 和易失性的 RAM。

uClinux 的移植大致可以分为由易到难的 3 个层次：

• 板级移植　移植的工作量最小，因为所用处理器已被支持,只要修改相应板的目录下的链接描述文档 rom.ld 或 ram.ld 和启动代码 crt0_rom.s 或 crt0_ram.s。然后再修改或增加驱动程序,变更环境变量设置等内容。

• 平台层次的移植　这种移植工作量要大些，uClinux 没有支持目标处理器,但已经支持相同架构的其他处理器,所以借鉴的地方较多。需要在相关体系结构目录下建立相应目录并编写相应代码，甚至会涉及到添加修改部分汇编代码。

• 结构层次的移植　如果待移植处理器没有参考结构,则需要修改 linux/arch 目录下与处理器结构相关的文件。这种情况的移植工作量最大一般由厂家提供,需要移植者非常熟悉处理器架构和 LINUX 架构,完成初始化过程内存映射、中断处理上下文、任务上下文等的代码,然后才能再做其他层次的移植[3]。

2 应用效果

我们开发了一个监控衡器，开始使用 uClinux,编程比较复杂,因为涉及到多任务,调试也麻烦。又使用 linux 的开发库,参考的例程倒是比较多,占用的资源还可以,不到 256kB,但是客户要增加功能的时候就很麻烦,新的功能往往要增加新的进程来实现,加新进程的时候不光要编写新的功能代码，还要调试已有的其他任务代码，因为新的任务代码很容易和原有的任务代码冲突。后来使用了 eCos,移植工作很顺利,兼容 linux 的库。当用户增加其他功能的时候,不用再次调试大部分其他代码,维护工作轻松了很多。

3 结语

通过以上对 eCos 和 uClinux 的分析比较以及自己的应用,发现这两种操作系统各有优劣。eCos 不是像 uClinux 从通用系统精简继承而来。而是就嵌入式市场的特点量身定制的。所以其运行效率较高,实时性能优良,支持更多的处理器,且针对新处理器的移植也相对简单，并且率先支持了多处理器甚至多核级的系统架构,而且其开发工具有 Windows 版和其他平台版的,且都提供了图形化的开发工具,适合不同开发者的习惯。uClinux 脱胎于通用 linux,占用空间相对较大,支持的嵌入式处理器较少,移植相对复杂,尤其是对于新架构的 CPU，因为嵌入式 CPU 架构和通用 CPU 架构还是有很大区别。不过,uCLinux 也同样继承了 Linux 的丰富应用,对于传统的 Linux 开发人员更加适宜,也有成功的产品应用。例如思科公司的部分低端交换机就是基于 uClinux 操作系统开发的。

总之，操作系统的选择是由嵌入式系统的需求和实际情况决定的,应用的效果和综合成本都要兼顾。一般情况下 eCos 小巧,更适合实时性要求较高的小型控制系统,而且如果开发者习惯 C++,那采用 eCos 会更合适。当然选 uClinux 是不错的选择,可以综合降低系统的成本,也是一个教学研究的良好样本。

参考文献：

[1] 邹思轶.嵌入式 Linux 设计与应用[M]. 清华大学出版社.2002 .

[2] Alessandro Rubini 和 Jonathan Corbet .魏永明,耿岳,钟书毅,等译. LINUX 设备驱动程序 (第 3 版) [M].中国电力出版社,2002.

[3] Rick Grehan ,Robert Moote ,Ingo Cyliax,许汝峰译.32 位嵌入式系统编程[M].中国电力出版社,2001.

[4] 蒋句平.嵌入式可配置实时操作系统 eCos 开发与应用[M].机械工业出版社,2004.

[5] 王学龙.嵌入式 Linux 系统设计与应用[M].清华大学出版社,2001.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net