

在 51 单片机上移植 uC/OS—II 关键问题的解决

Solution of key problems in the transplant of uC/OS- II to MCS- 51 MCU

(1.中国科学院沈阳自动化研究所; 2.中国科学院研究生院)田志鑫^{1,2} 张 雷^{1,2} 赵明扬¹

TIAN ZHIXIN ZHANG LEI ZHAO MINGYANG

摘要:由于 MCS- 51 系列微控制器存在着硬件堆栈小的结构缺陷,无法满足多任务环境下进行任务切换的需求,因此很难将 uC/OS—II 移植到 MCS- 51 系列微控制器上。本文给出的“堆栈映射”方式很好的解决了这个问题。同时还对 uC/OS—II 移植过程中一些关键性问题给予了详细的论述。包括:uC/OS—II 可移植的条件、内核配置和裁剪以及内核调试。

关键字:堆栈映射; uC/OS—II; 移植; 嵌入式实时操作系统

中图分类号:TP316.2, TP368.1

文件标识码:B

Abstract: Because of the intrinsic hardware stack limitation which results in the incapacity of task switching under multi-task environment, it's difficult that transplant uC/OS- II to MCS- 51 architecture MCU. The paper designs and implements a method called "stack mapping" that can solve this problem effectively. The paper also illustrates particularly other key problems including transplant conditions, real-time kernel configuration and debugging.

Key word: stack mapping, uC/OS—II, transplant, RTOS

1 概述

目前市场上的商用嵌入式操作系统,如 Vxwork、Linux 和 Windows CE 等已经十分成熟,开发商们提供功能强大的开发和调试工具,但其开发成本昂贵,而且只适用于 32/64 位带有 MMU 的高端嵌入式微处理器上。uC/OS—II 是一种多任务实时操作系统。内核源代码公开,短小精干,移植性强,可裁减性好,非常适用于 8/16 位低端嵌入式微处理器。本文详细描述了将 uC/OS—II 移植到 MCS- 51 系列单片机上的关键问题,包括操作系统的配置、裁减、堆栈结构设计以及内核的调试。

1.1 uC/OS—II 简介

uC/OS—II 实时嵌入式操作系统是基于优先级的抢占式实时多任务操作系统,包含了实时内核、任务管理、时间管理、任务间通信同步和内存管理等功能。uC/OS—II 具有如下特点:

1) 可移植性好:大部分代码用移植性很强的 ANSI C 语言实现,因此 uC/OS—II 可以在绝大多数 8、16、32 位微处理器、微控制器及数字信号处理器上运行。

2) 可裁剪:可以根据实际应用对其功能进行配置和裁剪,以达到减少存储器使用空间的目的。

3) 可确定性:uC/OS—II 中几乎所有系统调用的执行时间具有确定性。除了函数 OSTimeTick()和某些事件标志服务,uC/OS—II 系统调用的执行时间都不依赖于用户层软件任务的数量。

4) 任务栈:每个任务都有单独的栈空间,以便更高效的利用 RAM 空间。

5) 稳定性与可靠性:uC/OS—II 的每行代码都经过了严格的测试,并通过了美国联邦航空管理局的安全性认证,可应用于工业控制、航空航天等对于稳定性、可靠性要求很高的场合。

1.2 使用实时嵌入式操作系统的优、缺点

使用嵌入式实时操作系统同传统的前/后台系统相比,可以

将应用分解成多个任务,简化应用层软件的设计,并可通过内核提供的任务管理、时间管理、任务同步机制构建各种结构复杂的软件;良好的多任务设计,有助于提高软件系统的稳定性与可靠性;同时使得软件系统的功能扩展更加方便。但是,使用嵌入式实时内核也会带来一些额外负担。

传统的前/后台系统对程序存储器的需求仅取决于应用程序代码;而使用嵌入式操作系统内核的话,则内核本身需要额外的程序存储空间。同样,前/后台系统对数据存储器的需求仅取决于应用程序;而使用嵌入式操作系统内核,则需要额外数据存储空间进行任务切换。

实时嵌入式操作系统比前/后台系统需要更多的程序存储空间和数据存储空间。同时,由于嵌入式内核的存在,微控制器要承受 15%左右的额外负荷。这两方面的因素阻碍了实时嵌入式操作系统在实际中的应用。但随着存储器价格的不断下降,特别是嵌入式微处理器性能的飞速提高,使得嵌入式操作系统的应用逐渐成为一种低成本、高效、可靠的解决方案。

2 uC/OS—II 的移植

操作系统的移植,就是使一个通用操作系统内核在某一体系统结构的微控制器运行。uC/OS—II 大部分代码用 C 语言编写,而且实现的功能与具体硬件结构无关,因此保证了 uC/OS—II 的可移植性。但依赖于具体硬件结构的操作必须要用汇编语言来完成,如寄存器存取、硬件堆栈操作等等。操作系统的移植非常依赖于微控制器具体的结构和所使用的编译器。本文以 MCS- 51 结构微控制器为例,编译器使用 Keil uVision3。但其方法适用于任何体系结构的微控制器。

2.1 uC/OS—II 的结构与可移植条件

uC/OS—II 的内核采用模块化结构,使得 uC/OS—II 结构清晰,原代码可读性好,给操作系统的移植也带来方便。uC/OS—II 的文件结构如图 1 所示。uC/OS—II 的配置集中在 OS_CFG.H 和 INCLUDES.H 两个文件;uC/OS—II 的移植集中在 OS_CPU.H,

田志鑫:硕士研究生

OS_CPU_A.ASM 和 OS_CPU_C.C 三个文件。

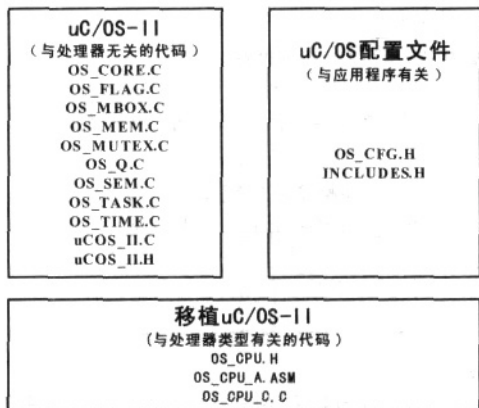


图1 uC/OS-II 内核结构框图

uC/OS-II 支持几乎所有的 8、16 位、32 位微控制器,但并非所有结构的微控制器都能进行 uC/OS-II 的移植,能够移植 uC/OS-II 的微控制器必须满足如下条件:

- 1) 微控制器的 C 编译器能够产生可重入代码;
- 2) 微控制器支持中断,并且能产生定时中断;
- 3) 编译器支持 C 语言进行开/关中断操作;
- 4) 微控制器具有硬件堆栈;
- 5) 微控制器有将堆栈指针以及其他寄存器的内容读出、并存储到堆栈或内存中去的指令。

uC/OS-II 支持 MCS-51 系列微控制器的移植。但像 Motorola 6805 系列单片机不具有硬件堆栈和堆栈操作指令,故无法进行 uC/OS-II 的移植。

2.2 uC/OS-II 的配置与裁剪

uC/OS-II 具有很好的可裁剪性。它能根据具体应用对其各个功能模块进行组合与定制,包括应用层软件要使用的最大事件控制块数目、最大任务数、最小优先级、内存分区大小以及功能选择开关,如信号量、邮箱、事件标志组、任务管理服务等,以达到减少存储器使用空间的目的。因此 uC/OS-II 适合于从简单到复杂各种程度的应用。包含全部功能模块的内核大约为 10K,如果经过裁减只保留核心代码,则可压缩到 2K 左右。uC/OS-II 的可裁减性是通过条件编译实现的。uC/OS-II 根据宏定义来完成对其功能的配置与裁剪。

在各项配置宏中,OS_TICKS_PER_SEC 是最重要的。该宏定义了操作系统时钟节拍的频率,uC/OS-II 在每一个节拍都要检查有没有更高优先级的任务在等待执行,若有,就要进行任务切换。OS_TICKS_PER_SEC 决定了应用程序的时间精度,该值越大,应用程序的时间精度越高,但微控制器的额外负担也越大。因此,在配置的时候,该值不宜设置过大,以满足应用要求为宜。

另一个重要的宏是 OS_STK_GROWTH,它定义了微控制器硬件堆栈的增长方向。MCS-51 系列微控制器硬件堆栈向下递减,应将该值设置为 1。对于 X86 系列等硬件堆栈向上递增的微控制器,将它的值设置为 0。

2.3 uC/OS-II 的堆栈结构设计

uC/OS-II 工作核心原理是:近似地让具有最高优先级的就绪任务处于运行状态。操作系统将在下面两种情况下进行任务调度:用户调用影响任务优先级和改变任务就绪状态的 API 函

数,调用中断服务程序。在进行任务调度时,uC/OS-II 会模拟一次中断,从用户程序角度看,好像程序被中断程序打断一样。

uC/OS-II 的移植最关键的是堆栈设计。这里有两个最重要的问题:模拟堆栈结构和 51 系列单片机小 RAM 缺陷的解决。

问题 1 解决方案:在任务切换时,uC/OS-II 模拟堆栈的结构一定要和中断堆栈结构一致,这取决于单片机的体系结构和使用的编译器。不同体系结构的微处理器所拥有的寄存器不同,并且各种编译器在中断发生时寄存器入栈顺序不同,在设计时一定要参考所使用的编译器的用户手册。

问题 2 解决方案:MCS-51 系列微控制器结构最大缺陷是其内部数据 RAM 非常小,无法应用在多任务环境下。在实际中,作者采用了“堆栈映射”的方法有效地解决了上述难题,使得 MCS-51 系列微控制器在多任务环境下的应用成为可能。具体思路是:在外部数据 RAM 中为每个任务建立一个用户堆栈。在任务切换时,将当前硬件堆栈和可重入堆栈的数据全部复制到当前任务的用户堆栈中,同时将高优先级任务的用户堆栈中的内容复制到硬件堆栈和可重入堆栈中,通过这种方式来完成的任务切换。具体用户堆栈数据结构的设计如图 2 所示。在多任务环境下,函数的可重入性非常重要。Keil uVision3 通过在数据 RAM 中专门建立一块特殊区域,用来模拟硬件堆栈,在发生任务切换时,保存函数的参数和内部变量,从而实现函数的可重入性。这块特殊的区域称为可重入堆栈。Keil uVision3 使用变量 C_XBP 来保存可重入堆栈顶部的指针。

采用“堆栈映射”方法在一定程度上会影响任务切换的速度,对于低速微控制器,可能会导致任务切换时的实时性变差,因此该方法仅适用于高速微控制器。

3 uC/OS-II 的调试

操作系统内核的稳定性直接决定了整个控制软件系统的稳定性。在硬实时控制环境下,操作系统一个小小的错误,可能导致灾难性的后果。因此,在完成 uC/OS-II 移植之后,一定要对

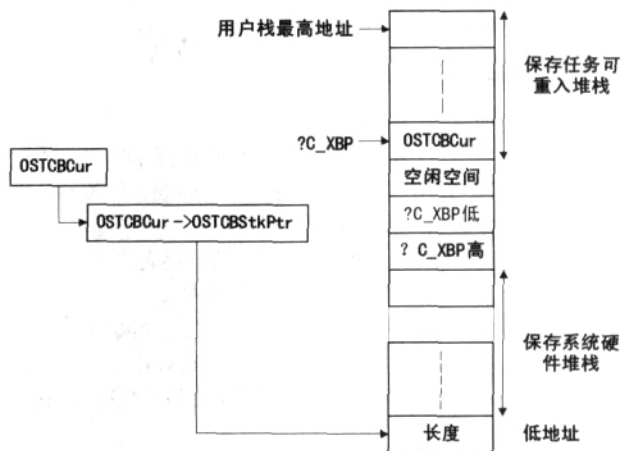


图2 用户堆栈结构

移植后的内核进行全面和细致的调试。在实际项目中,笔者采用了如下调试方法:

- 1) 确保 C 编译器、汇编器及链接器正常工作

首先验证 C 编译器、汇编器和链接器是否能够产生正确的指令。这一步,需要用户来解决各种编译器、汇编器及连接器的错误和警告信息。所使用的测试程序如下:

```
#include "includes.h"
```

```
void main(void)
```

```
{
```

```
OSInit();
```

```
OSStart();}
```

2) 验证 OSTaskStkInit() 和 OSStarHighRdy() 函数

首先修改 OS_CFG.H 文件, 设置 OS_TASK_STAT_EN 为 0, 以禁止统计任务, 使空闲任务: OS_TaskIdle() 成为 uC/OS-II 中的唯一任务。将程序下载到调试器, 单步执行 main() 程序, 跳过 OSInit(), 单步进入 OSStart()。一直单步运行到调用 OSStartHighRdy(), 然后单步进入 OSStartHighRdy()。由于该函数是用汇编编写, 编译器自动切换到汇编模式。此时将调用 uC/OS-II 唯一任务: OS_TaskIdle(), 继续单步执行, 同时观察各个寄存器中的内容, 如果 OSStartHighRdy() 能够将 OSTaskStkInit() 压入堆栈的寄存器数据, 按照相反顺序弹出, 并退出中断后调用 OS_TaskIdle() 第一条指令, 则说明 OSTaskStkInit() 和 OSStartHighRdy() 函数是正确的。否则, 对这两个函数进行修改。

3) 验证 OSCtxSw() 函数

在这一步测试中, 先添加一个应用程序:

```
Void TestTask(void)
```

```
{
```

```
While(1){
```

```
OSTimeDly(1);}
```

```
}
```

将程序下载到调试器, 单步执行程序, 并观察寄存器内容变化和程序执行顺序。如果任务 TestTask() 成功调用 OS_TaskIdle(), 则说明 OSCtxSw() 函数是正确的。否则, 对它进行修改。

4) 验证 OSIntCtxSw() 和 OSTickISR() 函数

这是测试的最后一步。测试方法与验证 OSCtxSw() 类似, 加入含有 OSTimeDly() 延时指令的用户任务, 然后进行单步执行, 观察程序能否在用户任务和 OS_TaskIdle() 之间进行切换。在测试之前, 应该保证时钟中断向量向时钟节拍中断服务子程序, 然后, 初始化时钟节拍并开中断。

4 结束语

本文作者创新点主要体现在:

1、针对 MCS-51 系列微控制器硬件堆栈空间小的缺陷, 提出“堆栈映射”的方法, 很好的解决了这个问题, 使 uC/OS-II 在 MCS-51 系列单片机上的移植成为可能。

2、根据 uC/OS-II 函数调用关系, 结合笔者在实际项目中的经验, 给出了 uC/OS-II 内核完整的调试方法。

本文给出的 uC/OS-II 移植方法, 已经成功地在 Cygnal C8051F120 微控制器上实现。经测试, 内核运行稳定、可靠, 任务切换速度快, 而且已经运用到激光切割机控制系统项目当中。

参考文献:

- [1] Jean J. Labrosse. The Real-Time kernel Second Edition. 北京航空航天大学出版社, 2003
- [2] Keil 公司. Keil C51 handbook, 2001
- [3] 张俊谟. 单片机中级教程—原理与应用. 北京航空航天大学出版社, 2003
- [4] 张培仁. 基于 C 语言编程 MCS-51 单片机原理与应用. 清华大学出版社, 2003
- [5] 杜树春. 单片机 C 语言和汇编语言混合编程实例详解. 北京航

空航天大学出版社, 2003

作者简介: 田志鑫, 男, 1980 年生, 硕士研究生, 研究方向: 嵌入式系统、控制理论。

Biography: Tian Zhixin, Gender: male, Date of Birth: 1980, Position: postgraduate, Research: embedded system design.

(110016 沈阳 中国科学院沈阳自动化研究所) 田志鑫 张雷 赵明扬

(100039 北京 中国科学院研究生院) 田志鑫 张雷

通讯地址: (110016 沈阳 中国科学院沈阳自动化研究所)

田志鑫

(收稿日期: 2007.9.13)(修稿日期: 2007.11.15)

(上接第 99 页)

本文作者创新点是将小波变换、数学形态学理论和现代图像处理技术相结合实现了指纹图像预处理和细化, 为指纹特征的提取和指纹识别提供了技术保证。



图 6 灰度图像



图 7 细化图像

参考文献

- [1] 赵松年, 熊小芸. 子波变换与子波分析. 电子工业出版社, 1996.
 - [2] Carre, P.; Fernandez- Maligne, C.. Use of the angle information in the wavelet transform maxima for image de-noising. Image and Vision Computing, 2000, 18(13): 1055- 1065.
 - [3] 朱菊华 杨新等. 基于纹理分析的保细节平滑滤波器. 中国图像图形学报, 2001, 6(11): 1058- 1064.
 - [4] Feels, et, al. Multidimensional Morphological Edge Detection. SPIE, 1987, 845: 285- 292
 - [5] 王亚, 吕新华, 王海峰. 一种改进的小波阈值降噪方法及 Matlab 实现. 微计算机信息, 2006, 2- 3: 259- 261.
- 作者简介: 李媛(1970-), 女(汉族), 辽宁省锦州市人, 北京联合大学自动化学院副教授, 博士, 2002 年毕业于北京理工大学, 主要研究方向是智能控制与模式识别。
- Biography: Li Yuan (1970-), Female (Han ethnic), born in Jinzhou, Liaoning Province, China. Associate professor, doctor degree. Her major research interests are intelligent control and model recognition.
- (100010 北京 北京联合大学自动化学院) 李媛
- (Automation School, Beijing Union University Beijing 100010) Li Yuan
- 通讯地址: (100010 北京 北京联合大学自动化学院) 李媛
- (收稿日期: 2007.9.23)(修稿日期: 2007.11.25)