

嵌入式实时操作系统在 ARM 上移植的分析与应用研究^①

Analysis and Implementation of Embedded Real-time Operation System Transplanting on ARM

孙育河 (新疆大学电气工程学院 乌鲁木齐 830008)

梁岚珍 (北京联合大学自动化学院 北京 100101)

摘要:以 UP-NETARM2410 嵌入式开发平台为硬件,介绍了嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ 移植条件,阐述了 $\mu\text{C}/\text{OS}-\text{II}$ 在 ARM 微处理器(以 S3C2410 为例)上的移植过程中的几个重要问题,经过测试,表明系统移植成功。

关键词:嵌入式系统 $\mu\text{C}/\text{OS}-\text{II}$ 移植 测试

1 引言

在开发嵌入式系统时,一般选择基于 ARM 和 $\mu\text{C}/\text{OS}-\text{II}$ 的嵌入式开发平台,因为 ARM 微处理器具有处理速度快、超低功耗、价格低廉、应用前景广泛等优点。将 $\mu\text{C}/\text{OS}-\text{II}$ 移植到 ARM 系统之后,可以充分结合两者的优势。嵌入式系统是先进的计算机技术、半导体技术、电子技术、和各个行业的具体应用相结合的产物,它是以计算机应用为中心、以计算机技术为基础,软硬件可裁减的专用计算机系统。在嵌入式系统中使用操作系统(嵌入式操作系统)已是大势所趋。由于 $\mu\text{C}/\text{OS}-\text{II}$ 是一个实时操作系统,所以如果将它嵌入到 ARM 处理器上,就能够进一步简化 ARM 系统的开发。

2 硬件运行平台及开发环境

本文以 UP-NETARM2410 为嵌入式开发平台,应用嵌入式开发软件 ADS 1.2 (ARM Developer suite v1.2)。采用主机——目标机交叉开发模型,主机和目标机之间通过串口线相连接,应用程序在主机的 windows 环境下编译链接,生成可执行的 system.bin 文件,下载到目标机。通过主机上的调试软件和连接到目标机上的调试设备,完成对应用程序的调试、分析。

3 $\mu\text{C}/\text{OS}-\text{II}$ 的移植与分析

3.1 $\mu\text{C}/\text{OS}-\text{II}$ 的软件体系结构

$\mu\text{C}/\text{OS}-\text{II}$ 是一个可移植、固化、裁减的占先式实时多任务内核,它使用标准的 ANSI C 语言编写,并包含一段汇编语言代码,使之可供不同架构的微处理器使用。它的主要特点如下:

- (1) 优先级可剥夺的实时多任务操作系统;
- (2) 可处理和调度 56 个用户任务,任务的优先级可以动态调整;
- (3) 提供任务间通信、同步使用的信号量、邮箱和消息队列;
- (4) 具有良好的可裁剪性,可减小系统的 ROM 和 RAM 大小。

$\mu\text{C}/\text{OS}-\text{II}$ 的软件体系结构如图 1 所示。

$\mu\text{C}/\text{OS}-\text{II}$ 的软件体系结构包括应用程序软件、与应用相关的代码、与处理器无关和相关的三个部分代码。应用程序软件是用户根据需求来编写的代码。要使同一个内核能适用于不同硬件体系,就需要在内核和硬件之间有一个中间层,这就是与处理器相关的代码,处理器不同,这部分代码也不同。

3.2 移植原理及分析

所谓移植,指的是一个操作系统可以在某个微处理器或者控制器上运行。 $\mu\text{C}/\text{OS}-\text{II}$ 在最初编写时就充分考虑了可移植性的问题,大部分的 $\mu\text{C}/\text{OS}-\text{II}$ 代码都是用 C 语言编写的,只有少量与处理器相关的代码是用汇编语言编写的,这是因为 C 语言不能直接读

^① 新疆教育基金高新技术重点项目(项目批号: XJEDU2005102)

写处理器寄存器,要使 $\mu\text{C}/\text{OS}-\text{II}$ 正常运行,处理器必须满足以下要求:

$\text{OS}-\text{II}$ 在移植时还需要用户自己修改部分代码。系统的移植原理图如图 2 所示。

对于操作者而言,需要完成的任务如图 3 所示,但实际上需要修改的部分只是与处理器相关部分的文件,即 OS_CPU.H , OS_CPU.C 和 OS_CPU.A.SM 。

(1) 设置 OS_CPU.H 中与处理器和编译器相关的代码。 OS_CPU.H 包括了用 $\# \text{defines}$ 定义的与处理器相关的常量、宏和类型定义。移植时将编译器相关的数据类型,用 typedef 转换为 $\mu\text{C}/\text{OS}-\text{II}$ 对应的基本数据类型。堆栈入口宽度设为 16 位整型。同时还修改相应的宏定义和常量,如堆栈增长方向、开中断、关中断和任务切换宏定义。

(2) 编写与操作系统相关的函数 OS_CPU.C 。重新编写 OS_CPU.C 中 6 个简单的 C 函数 OSTaskStkInit (任务堆栈结构初始化)、 OSTaskCreateHook (任务创建时功能扩展)、 OSTaskDelHook (任务删除时功能扩展)、 OSTaskSwHook (任务切换时功能扩展)、 OS2TaskStatHook (用以扩展统计功能)、 OSTimeTickHook (时钟节拍扩展)。唯一必要的函数是 $\text{OSTaskStkInit}()$, 其它 5 个函数,必须声明但没必要包含代码。

(3) 编写与处理器相关的函数 OS_CPU.A.SM 。重新编写 OS_CPU.A.SM 中 4 个简单的汇编函数。这 4 个汇编函数是 OSStartHighRdy (运行优先级最高的任务)、 OSCtxSw (任务级的切换)、 OSIntCtxSw (中断级切换)、 OSTickISR (时钟中断服务例程)。针对不同的微处理器/微控制器,重新编写引导程序。完成上述工作后, $\mu\text{C}/\text{OS}-\text{II}$ 就可以运行在 ARM 处理器上了。

4 系统测试实现

移植完成后的系统包括四个部分:1)自启动程序;2) $\mu\text{C}/\text{OS}-\text{II}$ 系统文件;3)移植代码;4)应用程序。为了验证 $\mu\text{C}/\text{OS}-\text{II}$ 移植的成功,设计了一个应用程序,在 ADS 1.2 环境下,工程编译结果如图 4 所示。

部分程序代码如下:

```
ClearScreen(); //清屏
pdc=CreateDC(); //创建绘图设备上下文
```

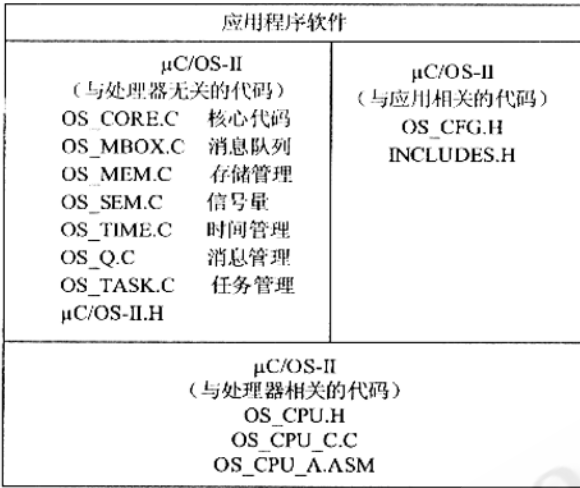


图 1 $\mu\text{C}/\text{OS}-\text{II}$ 软件体系结构

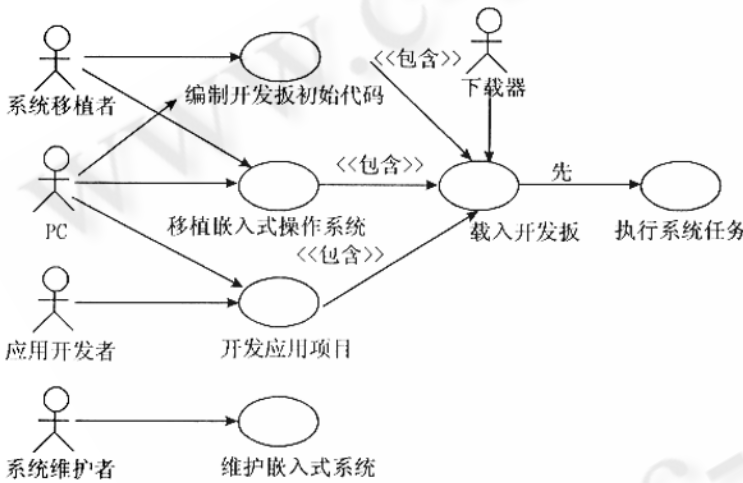


图 2 系统移植原理图

- (1) 处理器的 C 编译器能产生可重入代码;
- (2) 在程序中可以打开或者关闭中断;
- (3) 处理器支持中断,并且能产生定时中断(通常在 10Hz-1000Hz 之间);
- (4) 处理器支持能够容纳一定量数据的硬件堆栈;
- (5) 处理器有将堆栈指针和其它 CPU 寄存器读出到堆栈(或者内存)的指令。

3.3 $\mu\text{C}/\text{OS}-\text{II}$ 的移植过程

虽然 $\mu\text{C}/\text{OS}-\text{II}$ 在设计时,就考虑到了在不同的处理器上移植问题,但是实际上仅这样是不够的, $\mu\text{C}/$

```
SetDrawOrg ( pdc, LCDWIDTH/2, LCD-
HEIGHT * 2/3, &oldx, &oldy);
```

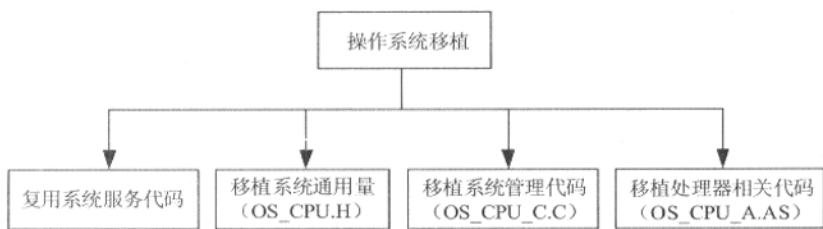


图 3 移植操作系统过程图

File	Code	Data
src	1K	256K
iic.c	896	4
main.c	1036	256K
GUI	0	0
inc	0	0
sys	0	0
drivers.h	0	0
sysconfig.h	0	0
mmu.h	0	0
drv	0	0
ucos-ii	492	0
ADD	0	0
includes.h	0	0
os_cfg.h	0	0
ucos_ii.h	0	0
arch	492	0
os_cpu.h	0	0
os_cpu_a.s	328	0
os_cpu_c.c	164	0
uhal	0	0
lib	621K	6.51M
init	492	40K
startup	244	0
Startup.s	244	0

图 4 工程编译结果

//设置绘图原点,点距离左边为屏幕 1/2,距离上边为屏幕 2/3。

```
SetPenColor(pdc, 0x0000e000); //把画笔颜色置为绿色。
```

```
MoveTo(pdc, -60, 1); //移动画点到指定坐标
```

```
LineTo(pdc, 60, 1); //画一条线到指定坐标
```

```
MoveTo(pdc, -60, 1); //移动
```

```
LineTo(pdc, -60, -120); //画线
```

```
DestoryDC(pdc); //删除绘图设备上下文
```

生成可执行的 system.bin 文件,下载并运行,在 UP-NETARM2410 嵌入式开发平台的 LCD 上显示如图 5。

从而验证移植成功!

5 结论

$\mu\text{C}/\text{OS}-\text{II}$ 具有很强的实时性, 2 已经被移植到各体系结构的微处理器上。将 $\mu\text{C}/\text{OS}-\text{II}$ 这种稳定可靠的实时嵌入式内核成功的移植到 S3C2410 上, 具有一定的应用价值。随着 ARM 体系结构和嵌入式系统得迅猛发展, $\mu\text{C}/\text{OS}-\text{II}$ 的应用将会越来越广泛。

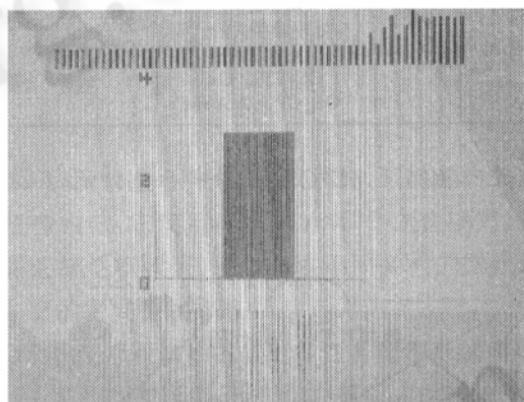


图 5 显示结果

参考文献

- 邵贝贝, 嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$, 第 2 版, 北京: 北京航空航天大学出版社, 2005.6.
- 田泽, 嵌入式开发与应用实验教程, 北京: 北京航空航天大学出版社, 2004.6.
- 陈贻, ARM 嵌入式技术实践教程, 北京: 北京航空航天大学出版社, 2005.2.
- 王田苗, 嵌入式系统设计与实例开发, 北京: 清华大学出版社, 2003.10.
- <http://www.ucos-ii.com>
- UP-NETARM2410 ($\mu\text{C}/\text{OS}-\text{II}$) 嵌入式系统试验指导书.