

1991年的天空

欢迎访问我的博客

博客园

首页

新随笔

联系

订阅

管理

随笔 - 84 文章 - 0 评论 - 24

昵称: sky1991

园龄: 4年8个月

粉丝: 28

关注: 0

+加关注

2017年3月						
日	一	二	三	四	五	六
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

最新随笔

1. 安卓使用WIFI调试程序

2. 一步步写STM32 OS【四】OS基本框架

3. 一步步写STM32 OS【三】PendSV与堆栈操作

4. 一步步写STM32 OS【二】环境搭建

5. 一步步写STM32 OS【一】序言

6. [转]关于WM_NCHITTEST消息

7. [转]ARM QT实现多点触摸

8. W3C SVG标准Web颜色值列表

9. [转]SYSZUXpinyin中文输入法的移植(到QT)

一步步写STM32 OS【一】序言

一直想写个类似uCOS的OS, 近段时间考研复习之余忙里偷闲, 总算有点成果了。言归正传, 我觉得OS最难的部分首先便是上下文切换的问题, 他和MCU的架构有关, 所以对于不同的MCU, 这部分需要移植。一旦这个问题解决了, 整个OS算是成功了一半了, 当然, 是对于简单的OS。

好了, 要写一个OS, 首先需要开发板和仿真器。我的开发板是STM32F4DISCOVERY, 自带ST-LINK V2仿真器, 板载MCU为STM32F407VGT6, 支持FPU, 32位ARM Cortex-M4F核, 1024KB FLASH, 192 KB RAM, 总之很强大。对STM32其他系列, 本OS几乎不需修改修改就可使用。开发环境为IAR for ARM 6.5, 如果是MDK的话, 也是大同小异, 汇编部分需要修改。

研究了一下UCOS-II的Cortex-M4的Port部分, 觉得很好, 就直接拿来用了, 修改的很少。首先我们来看一下这一部分几个比较重要的函数, 打开os_cpu_a.asm文件, 定位到下面的地方, 注释我改成中文了。当OS初始化完毕后, 执行OSStart, OSStart最后调用OSStartHighRdy函数, 注意在此之前的线程模式和异常模式的堆栈都是MSP, 在此之后线程模式的堆栈是PSP, 异常模式的堆栈仍是MSP。

```

OSStartHighRdy
    LDR    R0, =NVIC_SYSPRI14      ; 设置PendSV的异常中断优先级
    LDR    R1, =NVIC_PENDSV_PRI
    STRB  R1, [R0]

    MOVS  R0, #0                  ; 初始化PSP=0
    MSR   PSP, R0

    LDR    R0, =OS_CPU_ExceptStkBase ; 初始化异常堆栈MSP地址
    LDR    R1, [R0]
    MSR   MSP, R1

    LDR    R0, =OSRunning          ; 置OSRunning = TRUE
    MOVS  R1, #1
    STRB  R1, [R0]

    LDR    R0, =NVIC_INT_CTRL      ; 触发PendSV异常 (引起上下文切换)
    LDR    R1, =NVIC_PENDSVSET
    STR   R1, [R0]

    CPSIE I                       ; 开启中断, 于是进入PendSV异常

OSStartHang
    B     OSStartHang             ; 正常情况下, 不应运行到这里

```

其中最核心的函数应该是OS_CPU_PendSVHandler了, 它处理PendSV中断, 完成上下文切换。

```

OS_CPU_PendSVHandler
    CPSID I                       ; 关中断
    MRS   R0, PSP                  ; 获得PSP
    CBZ  R0, OS_CPU_PendSVHandler_nosave; PSP为0跳到OS_CPU_PendSVHandler_nosave, 即不保存上文, 直接进入下文。

    ; 问什么呢, 因为首次调用, 是没有上文的。
    ; 保存上文
    SUBS  R0, R0, #0x20            ; 因为寄存器是32位的, 4字节对齐, 自动压栈的寄存器有8个, 所以偏移为8*0x04=0x20
    STM   R0, {R4-R11}           ; 除去自动压栈的寄存器外, 需手动将R4-R11压栈

    LDR  R1, =OSTCBCur            ; 保存上文的SP指针 OSTCBCur->OSTCBStkPtr = SP;
    LDR  R1, [R1]
    STR  R0, [R1]

```

10. [转]交叉编译tslib1.4
我的标签
stm32(5)
OS(4)
MSP430(3)
调试(2)
蜂鸣器(1)
汉字(1)
汇编(1)
驱动(1)
时钟(1)
位带(1)
更多

随笔分类(81)
Arm(11)
C#、QT(10)
FPGA(4)
Java(1)
Linux(3)
单片机(48)
一步步写STM32 OS(4)

随笔档案(84)
2014年3月 (1)
2013年11月 (3)
2013年10月 (1)
2013年8月 (1)
2013年6月 (1)
2013年5月 (9)
2013年4月 (5)
2013年2月 (3)
2012年11月 (5)

```

OS_CPU_PendSVHandler_nosave           ; 切换下文
    PUSH    {R14}                       ; LR压栈, 下面要调用C函数
    LDR     R0, =OSTaskSwHook           ; 调用OSTaskSwHook();
    BLX     R0
    POP     {R14}

    LDR     R0, =OSPrioCur              ; 置OSPrioCur = OSPrioHighRdy;
    LDR     R1, =OSPrioHighRdy
    LDRB    R2, [R1]
    STRB    R2, [R0]

    LDR     R0, =OSTCBCur                ; 置OSTCBCur = OSTCBHighRdy;
    LDR     R1, =OSTCBHighRdy
    LDR     R2, [R1]
    STR     R2, [R0]

    LDR     R0, [R2]                    ; R0中的值为新任务的SP; SP = OSTCBHighRdy-
    >OSTCBStkPtr;
    LDM     R0, {R4-R11}                 ; 手动弹出 R4-R11
    ADDS    R0, R0, #0x20

    MSR     PSP, R0                      ; PSP = 新任务SP
    ORR     LR, LR, #0x04                 ; 确保异常返回后使用PSP
    CPSIE   I
    BX      LR                            ; 退出异常, 从PSP弹出xPSR, PC, LR, R0-R3, 进入新任务运行

```

之后我们在此基础上构建自己的OS, 首先完成两个任务互相调用, 然后是加入SysTick的任务调度, 最后加入信号量, 邮箱等功能。

分类: [一步步写STM32 OS](#)

标签: [stm32](#), [OS](#)

好文要顶 关注我 收藏该文  

 **sky1991**
[关注 - 0](#)
[粉丝 - 28](#)
[+加关注](#)

2 0

« 上一篇: [\[转\]关于WM_NCHITTEST消息](#)
 » 下一篇: [一步步写STM32 OS【二】环境搭建](#)

posted @ 2013-10-28 13:39 sky1991 阅读(3480) 评论(1) 编辑 收藏

评论列表

#1楼 2015-06-23 16:47 Burden

感谢分享, 受益匪浅

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【免费】自开发零实施的H3 BPM免费下载
- 【推荐】Google+GitHub联手打造前端工程师课程

2012年10月 (5)

2012年9月 (4)

2012年8月 (27)

2012年7月 (13)

2012年6月 (6)

积分与排名

积分 - 81360

排名 - 3027

最新评论

1. Re:[MSP430] 蜂鸣器演唱音乐

厉害啊, music_tab是怎么写的?

--侯胜滔

2. Re:一步步写STM32 OS【三】PendSV与堆栈操作

我大四呢, 这不是选了一个坑爹的毕业设计嘛, 看到你的帖子了, 很赞!

--李可jesscia

3. Re:一步步写STM32 OS【三】PendSV与堆栈操作

@李可jesscia哈哈, 博客好久不更新了, 没想到还有评论。这里说声抱歉, 这个博客是我本科时写的, 现在已经不做那些东西了。年久失修, 多数都已忘却。所以这个还要靠你自己了, 加油。。。...

--sky1991

4. Re:一步步写STM32 OS【三】PendSV与堆栈操作

博主, 我是初学嵌入式, 看了你的内容很有用, 求解一个问题, 我想“验证中断级任务切换”, 但是不会编写“OSTickISR”汇编, 求助

--李可jesscia

5. Re:一步步写STM32 OS【一】序言

感谢分享, 受益匪浅

--逝者如斯_不舍昼夜



最新IT新闻:

- 为什么说亚马逊正在逐渐失去云服务霸主地位?
 - 你在支付宝起早贪黑种下的树 可以免费去看了
 - 微信重磅功能上线: 家长可一键禁止游戏
 - Google员工嘲讽: 都5年了Nexus/Pixel的供应依然如此糟糕
 - 董明珠: 买的技术不是最好的 所以我不买
- » 更多新闻...



最新知识库文章:

- 垃圾回收原来是这么回事
 - 「代码家」的学习过程和学习经验分享
 - 写给未来的程序媛
 - 高质量的工程代码为什么难写
 - 循序渐进地代码重构
- » 更多知识库文章...

阅读排行榜

1. [原创] linux下的串口调试工具(14528)
2. 一步步写STM32 OS【三】PendSV与堆栈操作(14397)
3. STC单片机 IAP(EEPROM)的使用(8110)
4. c语言实现sin,cos,sqrt,pow函数(6480)
5. Bresenham快速画直线算法(5461)

评论排行榜

1. stm32汇编实例(5)
2. Linux系统下烧录单片机（转）(4)
3. [MSP430] 蜂鸣器演唱音乐(3)
4. [转]MSP430另一种UART实现(3)
5. 一步步写STM32 OS【三】PendSV与堆栈操作(3)

推荐排行榜

1. 一步步写STM32 OS【三】PendSV与堆栈操作(2)
2. 一步步写STM32 OS【一】序言(2)
3. c语言实现sin,cos,sqrt,pow函数(1)
4. FPGA 状态机(FSM)的三段式推荐写法(1)
5. DS-S6D0154彩屏驱动程序(1)