

嵌入式操作系统 UC/OS - II 的内核实现

王劲松, 李正熙, 夏旺盛

(北方工业大学 北京 100041)

摘要: 介绍一种源代码公开的嵌入式操作系统 UC/OS - II, 由美国人 JEAN J. LABROSSE 编写。程序主要用 C 语言编写, 已在全世界流传, 可以移植到多种 CPU 平台上, 有许多成功的应用。

关键词: UC/OS - II; 嵌入式操作系统; 任务; 用户

中图分类号: TP316.1

文献标识码: B

文章编号: 1004 - 373X (2003) 08 - 048 - 03

Kernel Design of Embedded Operation System UC/OS - II

WANG Jinsong, LI Zhengxi, XIA Wangsheng

(North University of Technique, Beijing, 100041, China)

Abstract: UC/OS - II written by JEAN J. LABROSSE is a kind of software of embedded operation system and its source code is open for anyone. The program is mainly written with C language and is spread in the world. UC/OS - II has planted in many CPU platforms. Many successful applications improve it reliable and stable.

Keywords: UC/OS - II embedded; operation system; task; user

1 引言

以前在我们一般所使用的系统中, 任务没有优先级之分。应用程序是一个无限的循环, 任务函数按在代码中的顺序运行, 处理相应的事务。时间相关性强的任务处理使用中断机制, 但是当系统比较复杂, 中断资源有限时, 中断程序只能将处理该任务的信息条件准备好后返回。当程序按顺序没有执行到该任务时, 该任务的执行必须等待, 所以将会造成任务每次的执行时间间隔不定, 不能及时处理紧急事务, 影响系统的运行。这种情况在要求限定时间内周期性处理事务的系统中是不允许发生的, 而且只由应用者编写的复杂程序很可能会出现 Bug。

嵌入式操作系统是实时操作系统, 运行于特定的硬件平台上, 一般包括处理器、存储器及外设器件和 I/O 端口, 包括操作系统软件, 要求实时和多任务操作, 用户可以在其基础上添加应用程序。使用嵌入式操作系统的用户只需添加所需的任务到操作系统中即可, 既节省开发时间, 又提高程序的可靠性。

2 UC/OS - II 及其任务介绍

UC/OS - II (Micro Control Operation System

Two) 是一种源代码公开的嵌入式操作系统, 程序绝大部分是用 C 语言写的, 带有少量的汇编程序, 并且有详细的说明和示例, 可移植性好、易调试, 稳定性与可靠性高, 功能也比较完善。UC/OS - II 和其他大部分的嵌入式操作系统的内核都是占先式内核, 被分为最高优先级的任务一旦准备就绪, 立刻就能得到 CPU 的控制权, 可以剥夺低优先级任务的 CPU 使用权, 处理系统最紧急的事务。

UC/OS - II 的任务实际是一段程序, 执行特定的功能, 拥有自己的代码和堆栈空间 (保存该任务的寄存器、返回地址和临时参数), 一般都是空函数, 不会返回任何值。任务执行一次后, 设置延时参数 OSTCBDly, 表明在经过 OSTCBDly 个时钟周期后再次运行, 然后任务进行切换, 使其他任务运行。

例如:

```
void Task(void)
{ 参数定义
  for(;;) {
    任务执行代码
    OSTimEdly() 延时函数; 清除任务就绪标志, 设置延时的时钟周期参数, 调用 OSSched() 进行任务调度
  }
}
```

3 UC/OS - II 的内核数据结构

(1) 任务控制块——OS_TCB

```

struct os_tcb{
    unsigned int *OSTCBStkPtr; 指向该任务栈顶的指针, 保存
                               该任务的上一次程序运行地址
    struct os_tcb *OSTCBNext;
    struct os_tcb *OSTCBPprev; 任务控制块的前后链表指针
    Int OSTCBDly; 延时参数, 每次中断减一, 为 0 时, 任务准备
                就绪, 可以运行任务优先级, 状态参数和任
                务通讯功能;
}

```

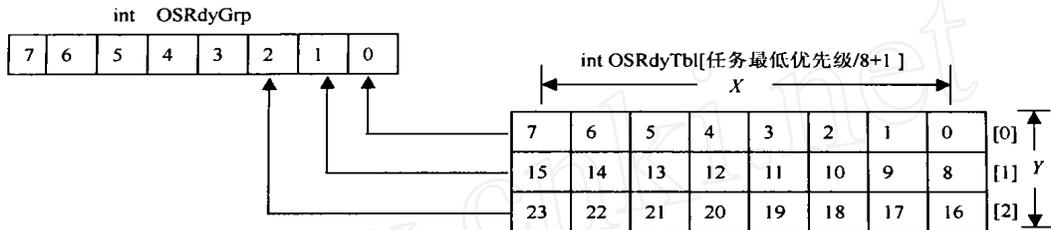


图 1 任务就绪表的结构

4 UC/OS-II 内核的运行机制

任何计算机系统都有时钟, 他是计算机世界的时间, 通过定时器的定时中断, 产生时间间隔, 每个间隔是任务执行的时间周期, 在一个间隔内所有应该执行的任务都应执行一次, 不能因为时间间隔太小而有的任务没有时间执行。因此, 任务的多少、时间周期的确定、CPU 的选择需仔细考虑, 确保在一个周期内 CPU 任务的运行时间所占比重不要太大。

主程序首先对系统进行初始化, 给各个参数赋值, 根据任务的多少来建立任务控制块链表, 并且首先建立一个最低优先级的空闲任务, 当没有其他任务需要运行时, 对一个变量进行累加计算来计算空闲时间。任务创建函数分配的一块内存, 保存寄存器的值和该任务的代码地址, 再把堆栈的地址存入任务控制块, 从而把任务程序代码、任务堆栈和任务控制块联系在一起, 操作系统内核可以通过任务控制块找到任务堆栈, 从堆栈中取得任务代码地址。基本结构如图 2 所示。

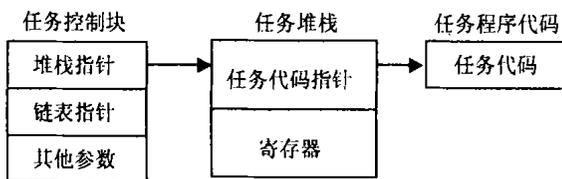


图 2 内核运行基本结构

任务的切换发生在 2 个时候, 第 1 个是当某一任务运行完毕时就调用延时函数, 使自身延时一个或几个时钟周期, 进行任务切换, 运行就绪的最高优先级最高任务; 第 2 个是每一次时钟周期中断后, 中断程序处理事务时, 可能使一些任务运行准备就绪, 然后重新整理和搜索任务就绪表, 进行任务切换(如图 3 所示), 选

(2) 任务就绪表——OSRdyTbl[]和OSRdyGrp
每个任务的就绪状态标志都放入就绪表中(如图 1 所示), OSRdyTbl[] 中的每一位为 1 表示相应优先级的任务处于就绪状态, OSRdyGrp 中的每一位为 1, 表示每 8 个为一组的任务中至少有 1 个已经就绪, 值越小优先级越高。

出其中的高优先级任务运行。

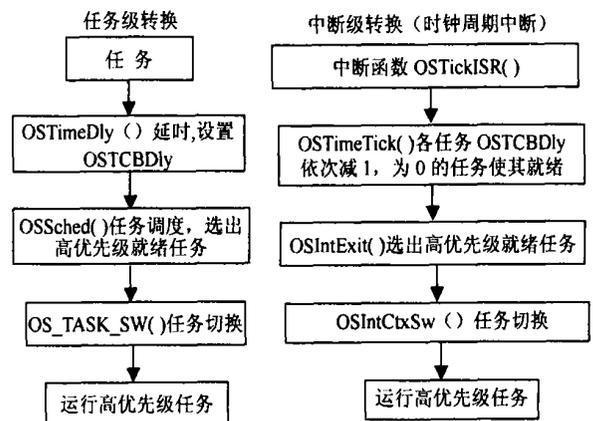


图 3 任务切换

嵌入式操作系统是多任务的, 任务切换是其核心技术, 由汇编语言编写。任务切换函数 OS_TASK_SW() 被设置成中断函数, 调用时使用的是软件中断指令, 这样在进入中断程序之前, 当前任务的代码地址和状态字就自动保存在当前任务的堆栈顶部。下面用 8086 的程序举例切换函数:

(1) OS_TASK_SW (或 OSIntCtxSw) PROC FAR 中断程序。

(2) PUSH 指令 该指令针对 OS_TASK_SW, 保存当前任务寄存器到该任务的堆栈, 代码地址已经保存。

(3) ADD SP, n 该指令针对 OSIntCtxSw, 因为该函数由中断程序调用, 运行他之前曾调用一些函数, 堆栈发生变化, 所以需要调整, 去掉前几个函数的保存地址和参数, 而代码地址在发生中断时就已经保存。

(4) 保存该任务的当前堆栈地址到该任务控制块 OS_TCB, 即保存 SS 和 SP。

(5) 调入已就绪最高优先级任务控制块, 取出其原来保存的任务堆栈地址, 即赋值给 SS 和 SP。

(6) POP 指令 恢复新任务的寄存器。

(7) RET 中断指令返回时, CPU 从堆栈顶部即 SS: SP 指向的位置取出以前所保存的代码地址和状态字, 并从此地址恢复原来任务的运行。

5 程序举例

一个最简单的嵌入式操作系统 8086 程序例子: 在主程序运行完毕后, 各任务开始轮流执行。

```
void main(void)
{
    OSInit(); 系统的初始化, 检测周边器件, 初始化各种参数和数据结构
    Vectset(0x80, OS_TASK_SW); 将 OS_TASK_SW() 设为中断函数
    OSTaskCreate(); 创建系统所需任务
    OSStart(); 根据已经设定好任务的优先级, 运行优先级最高的任务
    void Taskn(void)
    { 参数定义;
      for(;;) {
          任务执行代码;
          OSTimeDly();
      }
    }
}
```

作者简介 王劲松 北方工业大学机电学院检测技术与自动化装置在读硕士研究生。主要从事智能仪表设备的研制和开发。
李正熙 北方工业大学教授、硕士研究生导师, 北京市现场总线技术和自动化重点实验室主任。
夏旺盛 北方工业大学信息学院计算机专业在读研究生。

参 考 文 献

- [1] (美)Jean Labrosse J. UC/OS-II 源码公开的实时嵌入式操作系统[M]. 北京: 中国电力出版社, 2001.
- [2] (美)Jean Labrosse J. 嵌入式系统构件[M]. 北京: 机械工业出版社, 2002.
- [3] (美) Maurice Bach J. UNIX 操作系统设计[M]. 北京: 机械工业出版社, 2000.
- [4] 屠立德. 操作系统基础[M]. 北京: 清华大学出版社, 2000.

6 结 语

UC/OS-II 还有时间管理、内存分配的功能, 并且使用了信号量、邮箱和信息队列, 使各任务之间可以互相通讯, 协调对各种事务的管理, 适用于小型的高端 CPU。已经有许多公司以其为核心, 进行功能扩展, 从而开发自己的嵌入式操作系统, 并且由于其代码简练, JEAN J. LABROSSE 专门写书对其讲解, 也可以非常好地用于教学。

产业看台

美芯片商 Xilinx 将出售全新 90 纳米可编程芯片

美国芯片设计业者 Xilinx 表示, 将其新一代可编程芯片的样品出售给客户, 新产品系采用能降低成本的全新制程。

该芯片可允许电子、汽车等制造商为特定用途进行编程。Xilinx 表示, 新芯片采用新的 90 纳米制程, 将会在今年第二季量产。半导体制造商正竞相提供 90 纳米制程的新产品, 该制程可在愈来愈小的芯片上纳入更多的晶体管, 以带来显著的成本优势。

德州仪器和英特尔均已宣布, 将在今年内推出采用 90 纳米制程的芯片。但与可编程芯片不同的是, 这两家公司的产品主要是针对无特殊用途芯片编程需求的客户。

在可编程芯片市场与 Xilinx 竞争的 Altera 表示, 该公司自去年已经拥有测试版的 90 纳米芯片, 并计划在 2004 年上半年出货。

Xilinx 目前正与 IBM 携手进行 90 纳米技术研发, 并将由台湾联电进行代工。Xilinx 表示, 采用新制程技术的芯片, 体积将比现有可编程芯片缩小达 80%, 价格也更为低廉。

该公司称, 将使用 12 寸晶圆量产 90 纳米芯片, 而目前其样品芯片是采用 8 寸晶圆制作。Xilinx 计划在下月发表采用 90 纳米制程的特制产品。

(Chinabyte)