(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102063325 A
(43)申请公布日 2011.05.18

(21)申请号 201110001117.8

(22)申请日 2011.01.06

(71)申请人 哈尔滨工业大学
地址 150001 黑龙江省哈尔滨市南岗区西大直街 92 号

(72)发明人 彭宇 陈叶富 刘大同

(74)专利代理机构 哈尔滨市松花江专利商标事务所 23109
代理人 张宏威

(51)Int.Cl.
G06F 9/45(2006.01)
G06F 9/48(2006.01)

权利要求书 3 页 说明书 8 页

(54)发明名称

一种嵌入 51 系列单片机的多任务实时操作系统的实现方法

(57)摘要

一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，涉及一种嵌入单片机内部的操作系统。本发明解决了现有嵌入单片机内部的操作系统的代码量大、以及任务单一的问题。它采用基于 keil 的 C51 编程语言编辑、采用 keil 编译器编译实现，并采用 _task_ 关键字区别不同的任务，该操作系统的多任务切换过程为：在接收到新任务信号之后，关中断、启动任务切换模块进行任务切换：首先判断当前任务是不是空闲任务，如是，则不保存堆栈信息，否，则保存；然后判断当前任务是不是最高优先级任务，如是，则不做堆栈调整，否，则做堆栈调整；最后把最高优先级的任务设置为当前任务，如该当前任务是空闲任务，则直接返回，否，则需要判断是否需要出栈操作然后才能返回。

CN 102063325 A

1. 一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，它是采用基于 keil 的 C51 编程语言编辑、并采用 keil 编译器编译实现，并采用 task 关键字区别不同的任务，该操作系统的多任务切换过程为：

在接收到新任务信号之后，关中断，然后启动任务切换模块，该模块实现切换的过程为：

首先，发生任务切换的时候，它首先判断当前正在运行的任务是不是空闲任务，如果是空闲任务，则不保存堆栈信息，否则保存堆栈信息；

然后，判断当前任务是不是最高优先级任务，如果当前任务就是最高优先级任务，则不做堆栈调整，否则做堆栈调整；

最后，当堆栈调整结束之后，把最高优先级的任务设置为当前任务，然后观察当前任务是不是空闲任务，如果是空闲任务，就直接返回，否则需要判断是否需要出栈操作然后才能返回。

2. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，所述保存堆栈信息的过程为：

步骤 Q1、判断当前任务是否是空闲任务，如果是，则执行步骤 Q2，否则执行步骤 Q3；

步骤 Q2、从任务堆栈数组中加载栈顶指针到 SP；然后执行步骤 Q4；

步骤 Q3、保存堆栈信息 SP 到任务堆栈数组。

3. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，所述堆栈调整的过程为：

步骤 Q4、从任务就绪表 lqRdyTbl 中找出就绪表中的最高优先级的任务；然后执行步骤 Q5；

步骤 Q5、判断就绪表中的最高优先级任务是否是当前任务，如果判断结果为是，执行当前任务；否则执行步骤 Q6，进行堆栈调整；

步骤 Q6、获取就绪表中的最高优先级任务的栈顶位置；

步骤 Q7、判断当前任务的优先级是否高于就绪表中的最高优先级任务的优先级，如果判断结果为是，则执行步骤 Q10；否则执行步骤 Q8；

步骤 Q8、根据任务堆栈数组中加载栈顶的地址与最高优先级任务的栈顶地址之间的差值，获得移动的空间长度，并把位于任务堆栈数组中加载栈顶的地址与就绪表中的最高优先级任务的栈顶地址之间的存储器中所存储的数据转移到当前任务的栈底地址处；

步骤 Q9、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值，调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值，使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID 号对应的任务堆栈栈底值之间；堆栈调整结束；

步骤 Q10、根据就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的差值，获得移动的空间长度，并把位于就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的存储器中的所有数据移动到任务堆栈数组中加载栈顶的地址处；

步骤 Q11、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值，调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值，使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID

对应的任务堆栈栈底值号之间；堆栈调整结束。

4. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，所述把最高优先级的任务设置为当前任务，然后观察当前任务是不是空闲任务，如果是空闲任务，就直接返回，否则需要判断是否需要出栈操作然后才能返回的具体过程为：

步骤 Q12、将就绪表中的最高优先级的任务作为当前任务；

步骤 Q13、判断当前任务是否为空闲任务，如果判断结果为是，则执行步骤 Q16；否则执行步骤 Q14；

步骤 Q14、判断当前任务是否被中断切换，如果判断结果为是，则执行步骤 Q15，否则执行步骤 Q16；

步骤 Q15、将任务切换类型设置为非中断切换类型，并执行寄存器出栈操作；

步骤 Q16、开中断，完成任务调用。

5. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，

所述操作系统中的全局变量有：

变量模块包括的全局数据变量有：任务状态表 unaigned char data lqTaskState[lqMaxID]、任务就绪表 Unsigned char data lqRdyTb1、任务切换类型表 unsigned char data lqSwitch Type、中断号 unsigned char data lqIntNum、标志量数据 unsigned char data lqCtr、信号量数据结构 unsinged char data lqSemData[lqSemMax*2]、消息邮箱数据结构 unsinged char datalqMsgData[lqMsgMax*2]，其中：

任务状态表 unaigned char data lqTaskState[lqMaxID] 用字节的位表示任务状态，所述字节中的每个位表示信息为：

第 7 位 K_MSG 是邮箱事件标志位，该位置位表示当前任务在等待邮箱；

第 6 位 K_SEM 是信号量事件标志位，该位置位表示当前任务在等待信号量；

第 5 位 K_FLG 是标志量事件标志位，该位置位表示当前任务在等待标志量；

第 4 位 K_TMO 是延迟等待事件标志位，该位置位表示当前任务在延迟等待；

第 0-3 位是任务等待事件索引号，用于记录任务等待事件的索引号；

任务就绪表 Unsigned char data lqRdyTb1 的第 i 位表示 i 号任务处于就绪状态；

任务切换类型表 unsigned char data lqSwitch Type 的第 i 位置位表示第 i 号任务通过中断被切换出去，否则，表示第 i 号任务被更高优先级任务切换出去；

中断号变量 unsigned char data lqIntNum，高三位记录中断嵌套数，低 5 位记录中断类型号；

标志量数据 unsigned char data lqCtr，最高位表示当前标志量是否有效，其余七位以位的形式记录等待该标志量的任务；

信号量数据结构 unsinged char data lqSemData[lqSemMax*2]，该数据结构的第一个字节表示信号量值，第二个字节以位的形式记录等待第一个字节所表示的信号量值对应的任务；

消息邮箱数据结构 unsinged char data lqMsgData[lqMsgMax*2]，该数据结构中的第

一个字节表示该邮箱的消息值，第二个字节的最高位表示该邮箱是否有消息，如果该位置位，则表示该邮箱有消息，否则表示该邮箱没有消息；第二个字节的低 7 位以位的形式记录等待该邮箱的事件的任务信息；

中断程序入口地址数组 unsigned int code lqISREnter[]，用于记载各个中断服务子程序的入口地址。

6. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，将中断服务程序采用汇编语言编制。

7. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，

对于中断处理过程中，进入用户中断子程序的过程为：

关中断；存储当前中断号；进入用户中断服务子程序；

所述用户中断服务子程序的操作过程为：

进行当前寄存器压栈处理，保护寄存器；

修改任务切换类型表 unsigned char data lqSwitch Type，标记当前正在执行任务通过中断被切换；

当退出该用户中断服务子程序时，恢复当前任务需要执行出栈操作，恢复寄存器。

8. 根据权利要求 1 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，

所述系统中断服务程序，用于：首先根据中断变量 lqIntNum 检查当前中断是否处于中断嵌套中，如果出处于中断嵌套中，则执行被嵌套的中断；如果不是处于中断嵌套中，则执行任务切换。

9. 根据权利要求 7 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，进入用户中断子程序的具体过程为：

获得当前用户中断程序的入口地址；

并将该地址数据压入堆栈中，

获取中断变量 lqIntNum 的高 3 位，同时清零低 5 位，然后将所述高 3 位加 1，然后更新中断变量 lqIntNum 数据；

开中断，返回执行启动的用户中断子程序。

10. 根据权利要求 7 所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，其特征在于，退出用户中断服务子程序的具体过程为：

关中断，执行返回命令 RTEI；

获得中断变量 lqIntNum 的高 3 位，如果所述高 3 位不是零，则将所述高 3 位减 1；然后执行系统中断服务子程序；否则执行任务表中更高优先级的、并且准备就绪的任务。

一种嵌入 51 系列单片机的多任务实时操作系统的实现方法

技术领域

[0001] 本发明涉及到嵌入式编程技术，具体涉及一种嵌入单片机内部的操作系统。

背景技术

[0002] 随着软件技术的发展，现有嵌入式实时操作系统已经获得了广泛的应用。在单片机中植入嵌入式操作系统，能够实现单片机产品的智能化。但，由于一般的操作系统的代码量都比较长，使得所需要的存储空间比较大，不适用于内存空间有限的单片机。

[0003] 现有技术中，嵌入单片机内部的操作系统有 ucosii、small rtos 51 和 rtx51-tiny，其中 ucosii 可移植到许多类型的单片机内，但由于 51 系列单片机的内存比较小，该种操作系统不适用于 51 系列单片机。small rtos 51 是针对 51 单片机而开发的一种操作系统，该操作系统的优点是内核针对 51 单片机设计，内核更为优化，同时提供了与 ucosii 比较相近的、比较丰富的系统任务，但该系统有以下缺点：没有充分利用针对 keil 的 C51 编程语言的 _task_ 中的关键字进行设计，因此不能利用 keil 编译器的代码优化功能，甚至是需要人为的编译器内设置，使其阻止代码优化；另外，该系统的代码对于 51 系列单片机来说还是有点长。rtx51-tiny：是 keil 公司开发的针对 51 单片机的操作系统，该系统有效地克服了 smallrtos 51 系统的缺点，但是，该系统没有 small rtos 51 的优点。

发明内容

[0004] 为了解决现有嵌入单片机内部的操作系统的代码量大不适于 51 系列单片机使用，以及现有迁入 51 系列单片机的操作系统的代码长、任务单一的问题，本发明提出一种嵌入 51 系列单片机的多任务实时操作系统的实现方法。

[0005] 本发明的嵌入 51 系列单片机的多任务实时操作系统的实现方法是采用基于 keil 的 C51 编程语言编辑、并采用 keil 编译器编译实现，并采用 _task_ 关键字区别不同的任务，该操作系统的多任务切换过程为：

[0006] 在接收到新任务信号之后，关中断，然后启动任务切换模块，该模块实现切换的过程为：

[0007] 首先，发生任务切换的时候，它首先判断当前正在运行的任务是不是空闲任务，如果是空闲任务，则不保存堆栈信息，否则保存堆栈信息；

[0008] 然后，判断当前任务是不是最高优先级任务，如果当前任务就是最高优先级任务，则不做堆栈调整，否则做堆栈调整；

[0009] 最后，当堆栈调整结束之后，把最高优先级的任务设置为当前任务，然后观察当前任务是不是空闲任务，如果是空闲任务，就直接返回，否则需要判断是否需要出栈操作然后才能返回。

[0010] 所述保存堆栈信息的过程为：

[0011] 步骤 Q1、判断当前任务是否是空闲任务，如果是，则执行步骤 Q2，否则执行步骤

Q3 ;

[0012] 步骤 Q2、从任务堆栈数组中加载栈顶指针到 SP ;然后执行步骤 Q4 ;

[0013] 步骤 Q3、保存堆栈信息 SP 到任务堆栈数组。

[0014] 所述堆栈调整的过程为 :

[0015] 步骤 Q4、从任务就绪表 lqRdyTbl 中找出就绪表中的最高优先级的任务 ;然后执行步骤 Q5 ;

[0016] 步骤 Q5、判断就绪表中的最高优先级任务是否是当前任务,如果判断结果为是,执行当前任务 ;否则执行步骤 Q6,进行堆栈调整 ;

[0017] 步骤 Q6、获取就绪表中的最高优先级任务的栈顶位置 ;

[0018] 步骤 Q7、判断当前任务的优先级是否高于就绪表中的最高优先级任务的优先级,如果判断结果为是,则执行步骤 Q10 ;否则执行步骤 Q8 ;

[0019] 步骤 Q8、根据任务堆栈数组中加载栈顶的地址与最高优先级任务的栈顶地址之间的差值,获得移动的空间长度,并把位于任务堆栈数组中加载栈顶的地址与就绪表中的最高优先级任务的栈顶地址之间的存储器中所存储的数据转移到当前任务的栈底地址处 ;

[0020] 步骤 Q9、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值,调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值,使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID 号对应的任务堆栈栈底值之间 ;堆栈调整结束 ;

[0021] 步骤 Q10、根据就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的差值,获得移动的空间长度,并把位于就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的存储器中的所有数据移动到任务堆栈数组中加载栈顶的地址处 ;

[0022] 步骤 Q11、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值,调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值,使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID 对应的任务堆栈栈底值号之间 ;堆栈调整结束。

[0023] 所述把最高优先级的任务设置为当前任务,然后观察当前任务是不是空闲任务,如果是空闲任务,就直接返回,否则需要判断是否需要出栈操作然后才能返回的具体过程为 :

[0024] 步骤 Q12、将就绪表中的最高优先级的任务作为当前任务 ;

[0025] 步骤 Q13、判断当前任务是否为空闲任务,如果判断结果为是,则执行步骤 Q16 ;否则执行步骤 Q14 ;

[0026] 步骤 Q14、判断当前任务是否被中断切换,如果判断结果为是,则执行步骤 Q15,否则执行步骤 Q16 ;

[0027] 步骤 Q15、将任务切换类型设置为非中断切换类型,并执行寄存器出栈操作 ;

[0028] 步骤 Q16、开中断,完成任务调用。

[0029] 本发明的操作系统同时具有 small rtos 51 系统与 rtx51-tiny 系统的优点,该操作系统充分利用了 keil 的 C51 编程环境中的 _task_ 的关键字进行设计,充分发挥了 keil 编译器的代码优化功能,使编译后的代码量少,该操作系统的系统任务丰富。

具体实施方式

[0030] 具体实施方式一：本实施方式所述一种嵌入 51 系列单片机的多任务实时操作系统的实现方法，是采用基于 keil 的 C51 编程语言编辑、并采用 keil 编译器编译实现，并采用 _task_ 关键字区别不同的任务，该操作系统的任务切换过程为：

[0031] 在接收到新任务信号之后，关中断，然后启动任务切换模块，该模块实现切换的过程为：

[0032] 首先，发生任务切换的时候，它首先判断当前正在运行的任务是不是空闲任务，如果是空闲任务，则不保存堆栈信息，否则保存堆栈信息；

[0033] 然后，判断当前任务是不是最高优先级任务，如果当前任务就是最高优先级任务，则不做堆栈调整，否则做堆栈调整；

[0034] 最后，当堆栈调整结束之后，把最高优先级的任务设置为当前任务，然后观察当前任务是不是空闲任务，如果是空闲任务，就直接返回，否则需要判断是否需要出栈操作然后才能返回。

[0035] 具体实施方式二：本实施方式是对具体实施方式一所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法的进一步限定，本实施方式中所述的保存堆栈信息的过程采用下述过程实现：

[0036] 步骤 Q1、判断当前任务是否是空闲任务，如果是，则执行步骤 Q2，否则执行步骤 Q3；

[0037] 由于空闲任务堆栈内的数据全是无用数据，所以空闲任务不需要保存堆栈信息；如果不是空闲任务，堆栈内的数据为有用信息，在任务切换时必须保存堆栈信息。

[0038] 步骤 Q2、从任务堆栈数组中加载栈顶指针到 SP；然后执行步骤 Q4

[0039] 步骤 Q3、保存堆栈信息 SP 到任务堆栈数组。

[0040] 该步骤完成的动作是保存堆栈信息。

[0041] 具体实施方式三：本实施方式是对具体实施方式一所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法的进一步限定，本实施方式中所述的堆栈调整的过程采用下述过程实现：

[0042] 步骤 Q4、从任务就绪表 lqRdyTbl 中找出就绪表中的最高优先级的任务；然后执行步骤 Q5；

[0043] 步骤 Q5、判断就绪表中的最高优先级任务是否是当前任务，如果判断结果为是，执行当前任务；否则执行步骤 Q6，进行堆栈调整；

[0044] 如果当前任务是就绪表中的最高优先级的任务，那么不需要做堆栈调整，否则需要堆栈调整，加载就绪表中的最高优先级任务的堆栈。

[0045] 步骤 Q6、获取就绪表中的最高优先级任务的栈顶位置；

[0046] 步骤 Q7、判断当前任务的优先级是否高于就绪表中的最高优先级任务的优先级，如果判断结果为是，则执行步骤 Q10；否则执行步骤 Q8；

[0047] 当前任务是正在执行的任务，而“最高优先级任务”就是把当前任务切换出去，转而运行的“就绪表内的最高优先级任务”。发生这种情况有可能是两种原因导致的：一、当前正在运行的任务，在运行的过程中激活了一个优先级更高的任务，那么必须发生任务切换，这时“最高优先级任务”的优先级肯定大于当前任务；二、当前任务在运行的过程中需要外

界给出某个资源，而这个资源一时半会儿又没办法获得，那么当前正在运行的任务就被挂起，以等待外界提供它所需要的资源，当前任务在被挂起的时候，就要执行任务切换，把就绪表内的“最高优先级任务”切换进来，此时“最高优先级任务”的优先级肯定低于当前任务的优先级。

[0048] 步骤 Q8、根据任务堆栈数组中加载栈顶的地址与最高优先级任务的栈顶地址之间的差值，获得移动的空间长度，并把位于任务堆栈数组中加载栈顶的地址与就绪表中的最高优先级任务的栈顶地址之间的存储器中所存储的数据转移到当前任务的栈底地址处；

[0049] 步骤 Q9、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值，调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值，使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID 号对应的任务堆栈栈底值之间；堆栈调整结束；

[0050] 对于 ID 号在当前任务与最高优先级任务之间的任务，它们的栈底值需要调整。

[0051] 步骤 Q10、根据就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的差值，获得移动的空间长度，并把位于就绪表中的最高优先级任务的栈顶地址与当前任务的栈底地址之间的存储器中的所有数据移动到任务堆栈数组中加载栈顶的地址处；

[0052] 步骤 Q11、根据当前任务的栈底地址与任务堆栈数组中加载栈顶地址之间的差值，调整堆栈数组中的所有任务的 ID 号对应的任务堆栈栈底值，使得所有任务的 ID 号对应的任务堆栈栈底值都在就绪表中的最高优先级任务 ID 号对应的任务堆栈栈底值到当前任务的 ID 对应的任务堆栈栈底值号之间；；堆栈调整结束。

[0053] 具体实施方式四：本实施方式是对具体实施方式一所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法的进一步限定，本实施方式中所述的，把最高优先级的任务设置为当前任务，然后观察当前任务是不是空闲任务，如果是空闲任务，就直接返回，否则需要判断是否需要出栈操作然后才能返回的具体过程为：

[0054] 步骤 Q12、将就绪表中的最高优先级的任务作为当前任务；

[0055] 步骤 Q13、判断当前任务是否为空闲任务，如果判断结果为是，则执行步骤 Q16；否则执行步骤 Q14；

[0056] 此时，由于当前任务已经是就绪表中的最够优先级任务了，所以，如果当前任务是空闲任务时，直接开中断返回即可。

[0057] 步骤 Q14、判断当前任务是否被中断切换，如果判断结果为是，则执行步骤 Q15，否则执行步骤 Q16；

[0058] 执行到这一步的时候，当前任务已经是就绪表中的最够优先级任务了，如果当前任务是空闲任务，因为空闲任务的所有信息均是无效信息，那么直接开中断返回即可。

[0059] 步骤 Q15、将任务切换类型设置为非中断切换类型，并执行寄存器出栈操作；

[0060] 步骤 Q16、开中断，完成任务调用。

[0061] 上述各实施方式所述的方法中，使用 _task_ 关键字来区别不同的任务，可以使得被不同任务调用的不同函数即使没有相互调用，他们的局部变量也不会相互覆盖。

[0062] 具体实施方式五：本实施方式是对具体实施方式一至四任一实施方式所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法的进一步限定，本实施方式是对所述操作系统中的全局变量做进一步限定，该系统的全局变量有：

[0063] 变量模块包括的全局数据变量有：任务状态表unaigned char data lqTaskState[lqMaxID]、任务就绪表Unsigned char data lqRdyTbl、任务切换类型表unsigned char data lqSwitch Type、中断号unsigned char data lqIntNum、标志量数据unsigned char data lqCtr、信号量数据结构unsinged char data lqSemData[lqSemMax*2]、消息邮箱数据结构unsinged char datalqMsgData[lqMsgMax*2]，其中：

[0064] 任务状态表unaigned char data lqTaskState[lqMaxID]用字节的位表示任务状态，所述字节中的每个位表示信息为：

[0065] 第7位K_MSG是邮箱事件标志位，该位置位表示当前任务在等待邮箱；

[0066] 第6位K_SEM是信号量事件标志位，该位置位表示当前任务在等待信号量；

[0067] 第5位K_FLG是标志量事件标志位，该位置位表示当前任务在等待标志量；

[0068] 第4位K_TMO是延迟等待事件标志位，该位置位表示当前任务在延迟等待；

[0069] 第0-3位是任务等待事件索引号，用于记录任务等待事件的索引号；

[0070] 所述第四位K_TMO可以与K_MSG、K_SEM、K_FLG中的任何一个位同时置位，表示在规定时间内等待某个事件，如果在规定的时间内该事件发生，则此时K_TMO被清零，如果在规定的时间内，该时间未发生，则该任务被激活，此时K_TMO被置位。

[0071] 本实施方式中的任务等待事件索引号是四位二进制码，可记载16个事件。

[0072] 例如，当任务状态表unaigned char data lqTaskState[lqMaxID]的参数为10010010B时，表示当前任务是在规定的时间内等待2号邮箱事件。

[0073] 任务就绪表Unsigned char data lqRdyTbl的第i位表示i号任务处于就绪状态；

[0074] 任务切换类型表unsigned char data lqSwitch Type的第i位置位表示第i号任务通过中断被切换出去，否则，表示第i号任务被更高优先级任务切换出去。

[0075] 根据任务切换类型表中的标志，当有任务是通过中断被切换时，要执行压栈操作，保存寄存器：当有任务是被更高优先级任务切换出去时，不执行寄存器压栈操作。

[0076] 中断号变量unsigned char data lqIntNum，高三位记录中断嵌套数，低5位记录中断类型号；

[0077] 所述中断号变量中高三位用于记录中断嵌套数，本操作系统最多能够有7重中断嵌套。

[0078] 中端号中的低5位的中断类型号，是用于获得中断任务的入口地址，本操作系统最多能够支持32个中断。

[0079] 标志量数据unsigned char data lqCtr，最高位表示当前标志量是否有效，其余七位以位的形式记录等待该标志量的任务。

[0080] 该标志量数据的最高位有效时，激活该标志量数据的低7位所记录的任务。在该操作系统中，标志量数据的初值在初始化时可以给定为0x80或0x00。

[0081] 信号量数据结构unsinged char data lqSemData[lqSemMax*2]，该数据结构的第一个字节表示信号量值，第二个字节以位的形式记录等待第一个字节所表示的信号量值对应的任务，

[0082] 在该系统中，该信号量数据结构的初始值，第一个字节在0-255之间，第二个字节

是0。当信号量事件有效时，等待信号量的最高优先级任务就绪。如果没有人在等待这个信号量，那么当前信号量的信号值加1。

[0083] 消息邮箱数据结构unsinged char data lqMsgData[lqMsgMax*2]，该数据结构中的第一个字节表示该邮箱的消息值，第二个字节的最高位表示该邮箱是否有消息，如果该位置位，则表示该邮箱有消息，否则表示该邮箱没有消息；第二个字节的低7位以位的形式记录等待该邮箱的事件的任务信息；

[0084] 该消息邮箱数据结构在系统初始化时，第一个字节在0-255之间；第二个字节的初值是0x80或0x00。

[0085] 当邮箱中有消息时，可以再向该邮箱发送消息，当消息发送失败时，该邮箱内的原消息不会被覆盖。当邮箱事件有效时，把消息邮箱数据结构中的消息值返回给在等待该消息的优先级最高的任务。

[0086] 中断程序入口地址数组unsigned int code lqISREnter[]，用于记载各个中断服务子程序的入口地址；

[0087] 例如，第一个数据记载外部0号中断子程序的入口地址，第2个数据记载定时器0函数的入口地址，第3个数据记载外部1号中断子程序的入口地址，第4个数据记载定时器1函数的入口地址，第5个数据记载串口中断SPI函数的入口地址，第6个数据记载定时器2函数的入口地址。

[0088] 本方法中，将操作系统的入口函数在初始化堆栈时，把？RTX？TASKENT？S数组内的任务函数入口地址填入堆栈，该堆栈中有多有用户任务的入口地址、一个空闲任务的入口地址。

[0089] 例如：假设单片机的RAM区的最高地址是0xFF，进入main时SP寄存器的值为0x40，一共有4个用户任务，再加一个空闲任务，那么运行main函数之后，堆栈为：

[0090] 堆栈示意图：

[0091]

ROM地址	地址内存储的值
0XFF	空闲任务入口地址高8位
0XFE	空闲任务入口地址低8位
0XFD	3号任务入口地址高8位
0XFC	3号任务入口地址低8位
0XFB	2号任务入口地址高8位
0XFA	2号任务入口地址低8位
0XF9	1号任务入口地址高8位

0XF8	1 号任务入口地址低 8 位
0X43	0 号任务堆栈的最多空间，
-	lqSPtemp ＝ 0XF7
0XF7	
0X42	0 号任务入口地址高 8 位
0X41	0 号任务入口地址低 8 位

[0092]

[0093] 此时 SP ＝ 0X42，同时入口函数 main 初始化 0 号任务定时器并开中断，最后执行 RET 指令，把 0 号任务的入口地址出栈到 PC 寄存器，执行 0 号任务。

[0094] 具体实施方式六：本实施方式是对具体实施方式一至五任一实施方式所述的一种嵌入 51 系列单片机的多任务实时操作系统的实现方法的进一步限定，本实施方式是对中断服务程序做进一步说明：本方法中，将中断服务程序采用汇编语言编制，可以在实现功能的前提下节省代码量。

[0095] 本方法中，对于中断处理过程中，进入用户中断子程序的过程为：

[0096] 关中断；存储当前中断号；进入用户中断服务子程序。

[0097] 所述用户中断服务子程序的操作过程为：

[0098] 进行当前寄存器压栈处理，保护寄存器；

[0099] 修改任务切换类型表 unsigned char data lqSwitch Type，标记当前正在执行任务通过中断被切换；当退出该中断服务子程序时，恢复当前任务需要执行出栈操作，恢复寄存器。

[0100] 把系统中断服务程序的入口地址压入堆栈。确保从中断返回后，执行该子程序。

[0101] 将系统中断服务程序的入口地址压入堆栈，是为了确保系统在执行完用户中断服务子程序之后能够继续执行该系统中断服务程序。

[0102] 所述系统中断服务程序，用于：首先根据中断号变量 lqIntNum 检查当前中断是否处于中断嵌套中，如果处于中断嵌套中，则执行被嵌套的中断；如果不是处于中断嵌套中，则执行任务切换。获得当前用户中断程序的入口地址，并将该地址数据压入堆栈中；

[0103] 根据中断号变量 lqIntNum 获得当前中断的入口地址的过程为：

[0104] 即：根据获得的中断变量 lqIntNum 从中断程序入口地址数组 unsigned int codelqISREnter[] 获得中断服子程序的入口地址，

[0105] 获得中断变量 lqIntNum 的高 3 位，同时清零低 5 位，然后将所述高 3 位加 1，然后更新中断变量 lqIntNum 数据。所述中断变量 lqIntNum 的高 3 位存储的是中断嵌套数，每进入一次中断，该数据加 1，表示有一层中断，最多能够有 7 重中断嵌套。

[0106] 开中断，返回执行启动的用户中断子程序。

[0107] 当用户中断子程序执行结束，退出该用户中断子程序的过程为：

[0108] 关中断，执行返回命令 RTEI ；

[0109] 获得中断变量 lqIntNum 的高 3 位，如果所述高 3 位不是零，则将所述高 3 位减 1 ；然后执行系统中断服务子程序 ；否则执行任务表中更高优先级的、并且准备就绪的任务。

[0110] 当所述高 3 位为 0 时，表示有更高优先级的任务处于就绪状态，如果该高 3 位不是 0，则表示当前中断时从低优先级的中断中抢过 CPU 控制权，所以执行出栈操作，开中断并返回低优先级的中断。

[0111] 当用户中断服务子程序结束之后，开始执行系统中断服务子程序。