文章编号:1008-0570(2007)02-2-0063-02

基于 Keil C51嵌入式系统多任务实现技术

Achieving technique of multitask based on Keil C51 embedded operation system

(1.山东科技大学;2.上海大学)薛庆军[1] 韩 进[1,2]
XUE QINGJUN HAN JIN

摘要:基于 Keil C51 嵌入式操作系统的开发,把整个应用程序划分成多个任务,可大大减少错误率,加快产品开发速度。Keil C51 生成的目标代码效率高,而在 C51 下的所有函数的局部变量都放在 RAM 的一片共同的区域里,给实时操作系统下的应用程序开发带来一定的难度。文章提出了单任务法、变量覆盖分析法、寄存器变量法、利用重入栈四种解决方法,结合实例验证了其有效性和正确性。

关键词:嵌入式操作系统;Keil C51;局部变量;全局变量;多任务
中图分类号:TP302 **文献标识码:A**

Abstract:Program based on Keil C51 embedded operation system, which divides whole program into multitask, can decrease the error rate and increase program speed.Keil C51 can make highly validity object codes,but all its local variables are deposit in common area of RAM,which bring some difficult in program based on real time operation.The four methods, that is single task,variable overly analyses,register variable and reentrant stack,are put forward in this paper.Its validity and correctness is proved with an example.
Key Words:embedded operation system,Keil C51,local variables,global variables,multitask

技术创新

引言

51系列单片机应用非常广泛, 一大批性能优越的 51 兼容单片机相继推出,如低功耗、高速度和增强型的 Philips 公司的系列产品;完美地将 Flash(非易失闪存技术)EEPROM 与 80C51 内核结合起来的 Atmel 公司的系列产品;在抗干扰性能,电磁兼容和通信控制总线功能上独树一帜,其产品常用于工作环境恶劣场合的 Siemens 公司的系列产品等。随着单片机性价比的不断提高,在系统设计上,存储器的容量不再是决定系统价格的主要因素,而开发周期短,上市时间快成为系统设计工程师的首选。

基于 Keil C51 的嵌入式操作系统的开发,可以把整个应用程序划分成多个任务由多人编写,使应用程序编写变得容易实现,大大减少了错误率,加快了开发速度。重要的是, Keil C51 生成的目标代码效率非常高,多数语句生成的汇编代码很紧凑,容易理解。在开发大型软件时,更能体现高级语言的优势。C 编译器能产生可重入代码,而且用 C 语言可以打开和关闭中断。

1 问题的提出

Keil C51 不是标准 C 语言编译器,函数的局部变量并不放在堆栈中,根据编译模式的不同,局部变量可以放在 R0-R7 寄存器中、内部 RAM 中或外部 RAM 中。放在内部 RAM 或外部 RAM 中的局部变量和全局变量一起编译, 所以局部变量成了全局变量。但和全局变量相比还

有一点不同,放在内部 RAM 中或外部 RAM 中的局部变量可以被其他函数的局部变量所覆盖,也就是说,在 C51 下的所有函数的局部变量都放在 RAM 的一片共同的区域里, 形成局部函数变量区。这就给实时操作系统下的应用程序开发带来一定的难度。不同的任务调用不同的函数,不同函数的局部变量又都存放在相同的存储器中,势必造成一个任务的函数的局部变量被另一个任务的函数修改。

可重入函数的局部变量放在堆栈中,可以被一个以上的任务调用,不必担心数据被破坏。但是在 C51 中,堆栈的资源有限, 如此众多的重入函数有可能造成堆栈溢出。Keil C51 给我们提供了重入栈,用一部分内部或外部 RAM 作为重入栈。但是重入栈的换入换出降低了系统的效率。

2 解决方案

根据系统的不同要求和操作系统的特点我们可以通过以下途径实现多任务或函数的重入。

2.1 单任务法

单任务的方法就是在一个任务的主体开始运行之前就关闭中断,让系统的任务调度无法实现。只有在这个任务主体运行完毕才让系统调度其他任务。例如:

```
void SempleTask(void)
{
ET2=0;//关闭任务调度的定时器中断
...//任务处理
ET2=1;//打开任务调度的定时器中断
}
```

任务的调度是以时钟中断作为时间片调度和以其他中断作为抢先式调度的。关闭中断,系统就无法进行调度,如果不是递归不必考虑重入函数的问题。这种方法降低了系统的适时性,在一些适时性要求不高的系统中可以采用。如:测温仪表,压力仪表等。

2.2 变量覆盖分析法

如前所述,Keil C51 如果把局部变量放在 RAM 中,在编译时会把所有函数的局部变量放在局部变量区, 造成变量覆盖。

薛庆军:讲师

如果让不同函数的局部变量占用不同的区域,不同的任务调用不同的函数就不会修改另一个函数的局部变量。但是必须保证不同的任务不能调用同一个函数,否则第二任务调用函数时会修改第一个任务调用此函数后的局部变量。这种方法实际上就是把函数的局部变量变成了全局变量。

在Keil C51实现变量覆盖的方法如下:

在Progect->Option for Target->BL51 Misc->Overlay里填入:

"SampleTask~FuctionA"

这种方法只有在保证函数不被多个任务调用的情况下使用。但是在实际应用中很难做到这一点。

2.3 寄存器变量法

Keil C51在编译时可以选择寄存器变量,操作系统在任务切换时都要保存寄存器R0-R7的值。如果把函数的局部变量放到寄存器中,可以被多个任务调用。但是如果局部变量太多,Keil C51在编译时仍然会把局部变量放到内部RAM中,造成不同函数的局部变量覆盖。

在Progect->Option for Target->C51的优化选项中选择4 Register Variable。这样编译后的局部变量都尽量(局部变量在8个字节以下)放在寄存器中。

这种方法只有保证所有的局部变量都放到寄存器中才可选用。一般应用在系统不太大的场合。

2.4 利用重入栈

利用重入栈,虽然存在栈的换入换出,造成效率的下降,但是由于51单片机性价比的提高,在产品成本提高不太大的情况下,可以充分发挥操作系统下应用程序开发的优越性,缩短开发周期。利用重入栈可以编写重入函数,在编写重入函数时,只关心共享资源的互斥。Keil C51下重入栈的应用方法:Keill C51在产生新项目之前询问是否选择STARTUP程序。如果决定拿出一部分内部RAM或外部RAM作为重入函数的重入栈,就必须选择此启动程序STARTUP.A51。

启动文件STARTUP.A51中包含目标板启动代码,可手动加入这个文件,只要复位,则该文件立即执行,其功能包括:

1)定义内部RAM大小、外部RAM大小、可重入堆栈位

2)清除内部、外部或者以此页为单元的外部存储器

3)按存储模式初使化重入堆栈及堆栈指针

4)初始化8051硬件堆栈指针

5)向main()函数交权

也可以修改以下数据对系统初始化,如下表:

常数名	意义
IDATALEN	待清内部RAM长度
XDATA START	指定待清外部RAM起始地址
XDATALEN	待清外部RAM长度
IBPSTACK	小模式重入堆栈指针需初始化标志,"1"为需要;"0"为缺省。
IBPSTACKTOP	指定小模式重入堆栈顶部地址
XBPSTACK	大模式重入堆栈指针需初始化标志,"0"为缺省
XBPSTACKTOP	指定大模式重入堆栈顶部地址
PBPSTACK	是否Compact重入堆栈指针,需初始化标志,"0"为缺省。
PBPSTACKTOP	指定Compact模式重入堆栈顶部地址
PPAGEENABLE P2	初始化允许开关
PPAGE	指定P2值
PDATASTART	待清外部RAM页首址
PDATALEN	待清外部RAM页长度

根据应用系统的大小和对RAM的使用情况,合理选择模式。此处的模式选择必须跟编译器的模式一致。不管哪种模式,一旦选择使用重人栈必须根据硬件系统的RAM设计设置好栈顶。重入栈只是保存重入函数的局部变量。在Keil C51里,函数的返回地址仍然保存在内部堆栈中,所以在编写程序时仍然要考虑函数的嵌套次数,否则造成堆栈内部堆栈溢出函数无法返回。重人栈是递减的,重入函数的局部变量都保存在重入栈中,在设置重入栈顶的时候充分考虑有可能用到存储器的大小,否则重入栈会覆盖其他变量。

3 实例分析

```
void SubA(void)
{
unsigned int x;
...
}
void SubB(void)
{
unsigned int y;
...
}
void TaskA(void)
{
for(;;)
{
...
SubA();
...
}
void TaskB(void)
{
for(;;)
{
...
SubB();
...
};
}
```

在此实例中有两个任务TaskA、TaskB,两个任务分别调用两个过程SubA、SubB,在两个任务中分别存在两个局部变量x、y,在Keil C51的编译过程中,如果不是选择寄存器变量,变量x、y会分配在内部RAM或外部RAM的同一个单元。当任务TaskA在调用过程SubA时,如果任务TaskB得到执行时间,并且调用了SubB,就会造成局部变量y值的改变,由于x和y在同一个单元,x的值也被改变了,当TaskA再次执行时,SubA就会按照改变的值继续处理,造成错误结果。特别是在选用分时抢占式多任务操作系统时,问题变得尤为突出。我们可以采用变量覆盖分析法,让x,y分配到不同的单元;或者采用寄存器变量,因为一般的操作系统在任务切换时会保存当前任务的R0-R7的值,当再次切换回原来的任务时会恢复R0-R7的值,使得原来任务的寄存器变量值保持不变。

4 结论

根据Keil C51的特点,合理的安排任务和函数,可以完全避免函数局部变量的覆盖,充分发挥嵌入式操作的优点,加快产品的开发。(下转第77页)

技术创新

或者漫游时，系统会不停的进行 I/O 操作，会大大降低数据加载的效率。因此，格网划分的标准是相对的，要视具体的图幅范围和数据量而定。

通过实验，我们发现对于郑州市实验数据，格网数划分为 64×64 时，数据加载的效率比较理想。

对于普通 PC 机而言，将数据全部读入内存，会提高数据的访问速度和图形绘制速度，但通过实验却发现，PDA 的情况恰好相反。究其原因，主要是因为 PDA 的数据访问机制与 PC 机有很大不同，PDA 的内存与外存（FLASH 卡）是共用的，从内存访问数据与从外置存储设备上访问数据是没有区别的，将数据全部读入，反而占用了大量的 RAM 资源，降低了程序的执行效率。正因如此，数据分块的空间矢量数据模型非常适合在 PDA 上使用。另一方面，由于 PDA 的 CPU 处理能力有限，因此，与 PC 相比，PDA 上图形绘制的时间会占用很大比重。因此，我们只有使用适应性好的数据模型，才能有效的弥补 PDA 在硬件上的不足。

本文作者创新点：针对 PDA 硬件配置较低的特点，设计了一种具有较好适应性的面向对象分层的矢量数据模型，有效解决了 PDA 导航软件对大数据量的管理；采用空间分幅、图幅分块、要素分级的策略建立联合索引，提高了导航过程中地图数据动态调阅的效率和浏览的流畅性；提出了一种适于 PDA 的矢量数据多尺度分级显示的算法，实现了不同尺度、不同层次的空间信息服务；

参考文献：

[1]Site Characterization Utilizing GIS and Technology. Joint Services Environmental Management Conference August 2004.

[2]Hector Gracia - Molina,Jeffery D.Ullman,Jennifer Widom Database System Implementation.

[3]Donald Hearn, M.Pauline Baker,蔡士杰等译 Comper Graphics with OpenGL Third Edition.

[4]Spatial Technologies Assessing Rural Septic Systems. October 20- 21.2003.

[5]Douglas Boling Programming Windows CE, 北京博彦科技发展有限公司译,北京大学出版社.

[6]桂玲,吴舒辞,向诚.一种利用 LabVIEW 在 PDA 上设计数据文件存储与回放的方法[J].微计算机信息,2005,11- 1:74- 75.

[7]夏启兵.基于关系数据库的地理数据库引擎.解放军信息工程大学硕士学位论文 2002.

作者简介：文江，男，解放军信息工程大学测绘学院，硕士研究生；朱宝山，男，解放军信息工程大学测绘学院，副教授；蔡文涛，男，河南省商务中等职业学校电教中心；李韶芳，男，解放军信息工程大学测绘学院，硕士研究生。

(450052 河南郑州 解放军信息工程大学测绘学院)文江 朱宝山 李韶芳

(450011 河南郑州 河南省商务中等职业学校电教中心)蔡文涛

(Institute of Surveying and Mapping, Information Engineering University;66 Longhai Road, Zhengzhou 450052,China)Wen Jiang Zhu Baoshan Li Shaofang

(Henan Commercial Junior Vocational School; NO. 6 BoSong Road, northern Wenhua road, Zhengzhou 450011,China)Cai Wentao

通讯地址：(450052 河南郑州 郑州陇海中路 66 号测绘学院六系二队)文江

(收稿日期：2006.9.26)(修稿日期：2006.10.23)

(上接第 59 页)

[2]王中明,武树斌,刘贤德等.在 Linux 平台下 BACnet 路由器的实时性研究[J]华中科技大学学报(自然科学版)2005,33(3):60- 62

[3]吴姣梅,李红艳,吴保荣等. 改善嵌入式 Linux 实时性能的方法研究[J].微计算机信息,2006,1- 2:72- 74

作者简介：刘文龙(1981-)，男，辽宁人，华中科技大学光电子科学与工程学院硕士，主要研究方向：基于嵌入式系统的应用程序开发. E- mail:liuwenjohn@gmail.com;徐海峰(1965-)，男，江苏人，华中科技大学光电子科学与工程学院副教授，博士，主要研究方向：智能楼宇；刘贤德(1938-)，男，湖北人，华中科技大学光电子科学与工程学院教授，博士生导师，主要研究方向：智能楼宇，BACnet 控制网络。

Biography:Liu Wenlong (1981-),male,LiaoNing,Graduate student in Institute of Optoelectronics Science and Engineering of Huazhong University of Science and Technology, majoring in application program developing based on Embedded System.

(430074 武汉 华中科技大学光电子科学与工程学院)刘文龙 徐海峰 刘贤德

(Institute of Optoelectronics Science and Engineering, Huazhong University of Science and Technology,Wuhan Hubei 430074,China)Liu Wenlong Xu Haifeng Liu Xiande

通讯地址：(430074 武汉 武汉市华中科技大学主校区西 16 舍 311 室)刘文龙

(收稿日期：2006.9.26)(修稿日期：2006.10.23)

(上接第 64 页)

本文作者创新点：文章创新地提出了用单任务法、变量覆盖分析法、寄存器变量法、重入栈四种方法，并结合实例验证了其有效性和正确性。

参考文献：

[1]陈明计.嵌入式实时操作系统 Small RTOS51 原理及应用[M].北京航空航天大学出版社,2004.

[2]李善平.LINUX 与嵌入式系统[M].清华大学出版社,2003.

[3]陈晓莉.keil C51 单片机仿真器的设计[J].微计算机信息,2006,2- 2:19- 20.

[4](韩国)金炯泰.如何使用 KEIL 8051 C 编译器[M].北京航空航天大学出,2002.

[5]尹永.Vision2 单片机应用程序开发指南[M].科学出版社,2005.

作者简介：薛庆军(1971-)，汉族，男，讲师，研究方向：计算机应用技术。

Biography:Xue Qingjun (1971 -),lecturer,Research field is Computer Application Technology.

(266510 青岛市山东科技大学信息科学与工程学院)薛庆军 韩 进

(200072 上海市上海大学机电工程与自动化学院)韩 进

(College of Information Science and Engineering,Shandong University of Science and Technology,Qingdao,Shandong 266510)Xue Qingjun Han Jin

(School of Mechanical & Electronic Engineering and Automation,Shanghai University,Shanghai 200072)Han Jin

通讯地址：(266510 青岛市山东科技大学信息科学与工程学院)薛庆军

(收稿日期：2006.12.17)(修稿日期：2007.1.15)