文章编号:1008-0570(2007)02-2-0068-03

实时系统优先级位图调度算法的改进

Improvement of the scheduler algorithm based on priority bitmap in real-time system

(南京工业大学)刘　卫　程明霄　王晓荣
LIU WEI CHENG MINGXIAO WANG XIAORONG

摘要:实时操作系统要求具有速度快和可预测性的特点,必须保证实时任务在要求的时间内完成。本文在描述了优先级位图调度算法的机制后,然后针对其在支持任务数量,所需数据空间和存在多个相同优先级任务的处理方面,提出改进方法。

关键词:优先级;任务调度;实时系统
中图分类号:TP316.2　　**文献标识码:**A

Abstract:The real-time operating system request the speed tobe quick and the characteristic may be forecast,the real- timeduty must completed in the request time. This paper describes the theory of the scheduler algorithm based on priority bitmap, then it presents an improvement for the scheduler algorithm at the priority levels, the data storage and the tasks in the same level.

Key words:priority,task scheduler,real-time system

技术创新

引言

嵌入式实时系统的进程调度与通用计算机操作系统进程调度有很大的不同,实时操作系统要求任务的执行的确定性和可预测性。实时系统的任务调度通常采用任务队列的方式进行管理,在基于优先级的调度处理中,需要获得当前具有最高优先级的就绪任务。采用直接将就绪任务的任务控制块放在就绪队列的末尾或则将就绪队列中的任务控制块按优先级的顺序排序的调度方法都会使调度程序所花的时间与任务数量相关,具有不确定性,为提高实时内核的确定性,实时系统通常采用优先级位图的就绪任务处理算法。

1 优先级位图算法的实现机理

优先级位图算法采用含有 8 个 char 型的数组的每个二进制位对应的一个优先级的数据结构,二进制位的值为 1,表示对应数组元素中对应的优先级有就绪任务;二进制位为 0,表示对应数组元素中对应的优先级没有就绪任务。

在任务进入和退出就绪态时,根据任务的优先级改写数组的对应优先级的二进制位,改写方式采用对优先级数进行变换获得数组中对应优先级二进制位位置的方式。这样对于任何优先级的任务改写位图数据结构的时间都是相同的。

在获取进入就绪态的最高优先级时,是通过优先级判定表即查表的方法来实现的,这样就保证了获取进入就绪态任务最高优先级的执行时间是确定的。

优先级位图算法采用含有 8 个 char 型的一维数组的每个二进制位对应一个优先级的数据结构,使得其只支持 64 个优先级,而且每个优先级只能对应一个任务,不能出现多个相同优先级的任务;在获取进入就绪态的最高优先级时,采用优先级判定表的方法,解决了执行速度的确定的问题,但是这样必须事先填写好一个含有 255 个 char 型的数据表,占用了比较大的数据空间。

刘卫:硕士

2 优先级位图算法在支持优先级数方面的改进

原优先级位图算法采用 char 型一维数组的二进制位对应一个优先级,使得只支持 64 个优先级。

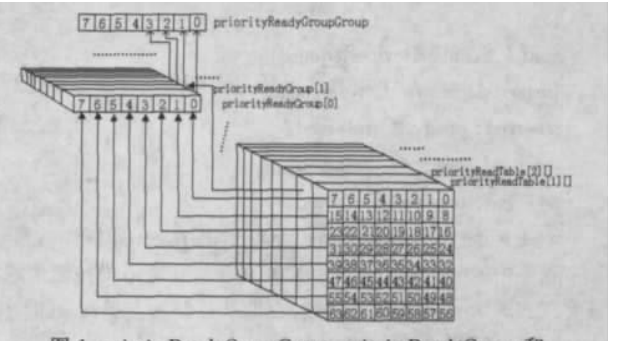

图 1　priorityReadyCroupGroup、priorityReadyCroup 和 priorityReadyTable 之间的关系

扩展数组的维数便可以增加支持的优先级数,例如采用 char 型 8x8 的二维数组的每个二进制位对应优先级则可以支持 512 个优先级;采用 8x8x8 的三维数组的每个二进制位对应优先级则可以支持 4096 个优先级。扩展数组维数后相应的其他数据结构、任务进入和退出的操作、获取进入就绪态的最高优先级的操作也需要相应更改,以二维数组支持 512 优先级数为例具体算法如下:

(1) 设置变量

char priorityReadyCroupGroup;
char priorityReadyCroup;
char priorityReadyTable;

priorityReadyCroupGroup、priorityReadyC roup 和 priorityReadyTable 之间的关系如图 1 所示。priorityReadyTable 的每个数组元素对应 512(0~512)个优先级的 8 个优先级。如 priorityReadyTable 对应的优先级为 64~71,如果对应优先级存在就绪任务,则相应的二进制位为 1,否则为 0。若 priorityReadyTable 的第 0 位为 1,则当前存在一个优先级为 64 的就绪任务;若

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

priorityReadyTable第 0 位为 0，则当前不存在一个优先级为 64 的就绪任务。

(2) 任务优先级与 priorityReadyCroupGroup、priorityReadyCroup 和 priorityReadyT

able 之间的关系

任务优先级与 priorityReadyCroupGroup、priorityReadyCroup 和 priorityReadyTab

le 之间的关系如图 2 所示。当优先级数为 512 时，优先级与 9 个二进制位相对应。在表示优先级的 9 个二进制位中，高三位为优先级在 priorityReadyCroupGroup 的二进制位位置，与 priorityReadyCroup 的数组元素下标相对应；中三位表示对应 priorityReadyCroup 元素中的值；低三位表示对应 priorityReadyTable 的数组元素的值。如对于优先级 70，二进制表示为 1000110，优先级高三位为 001，在 priorityReadyCroupGroup 中位序 1 为 1，对应 priorityReadyCroup；优先级中三位为 000，对应 priorityReadyCroup 中的位序 0，对应 priorityReadyTable；优先级底三位为 110，对应 priorityReadyTable 中的位序 6，即优先级 70。

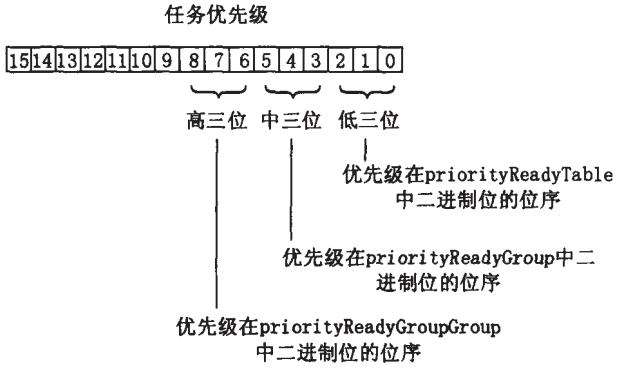

图 2　任务优先级与 priorityReadyCroupGroup、priorityReadyCroup 和 priorityReadyTable 之间的关系

(3) 任务进入和退出就绪态

首先设置优先级映射表 C 语言描述为：

```
char priorityMapTable1[]={
0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80
};
```

任务进入就绪态，需要把任务的优先级(priority)转换为在 priorityReadyCroupGro

up、priorityReadyCroup 和 priorityReadyTable 中的表示，转换过程用 C 语言描述为：

```
priorityReadyCroupGroup|= priorityMapTable[priority>>6];
priorityReadyCroup [priority>>6]|= priorityMapTable[(priority & 0x38) >> 3];
priorityReadyTable[priority>>6][(priority & 0x38) >>3]|=priorityMapTable[priority & 0x07];
```

任务进入就绪态，需要对 priorityReadyCroupGroup、priorityReadyCroup 和 priorit

yReadyTable 进行处理，清除任务优先级在 priorityReadyCroupGroup、priorityReadyCro

up 和 priorityReadyTable 中的体现，处理过程用 C 语言描述为：

```
if(priorityReadyTable[priority>>6][ (priority & 0x38) >>3]& = ~ priorityMapTable[priority & 0x07] = = 0){
priorityReadyCroupGroup & = ~ priorityMapTable[priority >> 6];
priorityReadyCroup[priority>>6] & = ~ priorityMapTable[(priority & 0x38) >> 3];
}
```

(4) 获取进入就绪态的最高优先级及相应的任务

获取进入就绪态的最高优先级是通过优先级判定表即查表的方式进行的，具体实现见后述在所需数据空间的改进。假定任务按优先级进行组织，以优先级为下标，在获得进入就绪态的最高优先级后便可以得到任务的任务控制块。

3　优先级位图算法在所需数据空间方面的改进

优先级位图算法在获取进入就绪态任务的最高优先级时，采用的是查找优先级判定表的查询方式，这样需要一个含有 256 个 char 型数据的数组。由于嵌入式系统中对存储空间要求往往较为苛刻，改进算法在分析了优先级判定表的规律，发现优先级判定表中除第一列数据外，其他列数据均相同，这样便可以改用两个含有 16 个 char 型数据的数组表示原包含 256 个 char 型数据的数组的所有信息，具体算法实现，C 语言描述如下：

(1) 设置优先级判定表

```
char priorityDecisionTable1={
0,0,1,0,2,0,1,0,3,0,1,0,2,0,1,0  /* 以 16x16 排列后，原判定表中第一行的值 */
};
char priorityDecisionTable2={
0,4,5,4,6,4,5,4,7,4,5,4,6,4,5,4   /* 以 16x16 排列后，原判定表中第一列的值 */
};
```

优先级判定表以 priorityReadyCroupGroup、priorityReadyCroup 和 priorityReadyTa

ble 数组元素的值为索引，获取 priorityReadyCroupGroup、priorityReadyCroup 和 priori

tyReadyTable 数组元素(取值为 0x00~0x0FF)对应二进制表示 1 出现的最低二进制位的序号(0~7)。

(2) 获取进入就绪态的最高优先级

获取进入就绪态的最高优先级通过查询优先级判定表来实现，处理过程 C 语言描述为：

```
#define GetValue(a) ((a & 0x01)==0 ? priorityDecisionTable2[(a&0x0f0)>>4] : \
priorityDecisionTable1[(a & 0x0f) >> 0])
high3Bit = GetValue (priorityReadyGroupGroup);
mid3Bit  = GetValue (priorityReadyGroup[high3Bit]);
low3Bit  = GetValue (priorityReadyTable[high3Bit][mid3Bit]);
priority = (high3Bit<<6)+ (mid3Bit<<3)+ low3Bit;
```

high3Bit 为就绪任务最高优先级的高三位，mid3Bit 为就绪任务最高优先级的中三位，low3Bit 为最高优先级的底三位。算法通过 GetIndex 宏计算查表的索引值，在查表的时候如果发现索引值的最底位不为 0，则以索引值直接取得判定表中前 16 个数据；如果索引值最底位为 0，则需要计算后，再查表。

在宏中“(a & 0x0f) >> 0”主要为与“(a & 0x0f0)>>4”相一致，使 a 在任何取值下执行的时间都是相同的，当然编译器可能优化“ >> 0”语句，这时需要在编译此语句时关闭编译优化选项。

? 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

4 优先级位图算法在存在多个相同优先级任务时的改进

优先级位图算法采用二进制位对应一个优先级的数据结构,这样使得每一个优先级只能对应一个任务,不能同时存在多个相同优先级的任务。

改进算法将相同优先级的任务组成一个任务队列,然后将队列指针按优先级进行组织,以优先级为下标。这样在获得最高优先级时,便可以获得这个优先级的任务队列,然后从该队列中取出一个任务执行,当任务退出运行态时,将该任务挂到这个优先级队列的队列尾,这样便可以支持多个相同优先级任务同时存在,并且相同优先级任务的调度近似时间片调度方式。

5 性能对比

原优先级位图算法支持64个优先级的任务,在判定表的实现上需要255字节的数据存储空间,不支持多个任务具有相同的优先级。改进算法可以通过扩展优先级位图数组的维数来扩展所支持优先级的数量,这需要额外的数据空间;在判定表的实现上只需要32字节的数据存储空间,这样,在查找的时候会增加计算的时间,对任务调度执行时间有细微的影响;在支持多个相同优先级任务的实现上,采用任务队列的方式解决相同优先级的任务调度,这样,也会增加系统的调度开销。

本文作者创新点:1.用扩展数组维数的方法增加了原算法的支持的优先级的数目。2.改进了原算法中的判定表的结构使所需空间减少。3.对原算法不支持多个任务具有相同优先级的问题提出改进方案

参考文献:

[1]Spae J Mullender,Pierre G Jansen.Real Time in a Real Operating System.Lucent Technologies and University of Twente.January 30,2003.

[2]Mercer C W.An Introduction to Real-Time Operating Systems: Scheduling Theory.School of Computer Science,Carnegie Mellon University,November 1992

[3]Audsley,Burns,Richardson,Wellings.Hard Real-Time Scheduling: The Deadline Monotonic Approach.IEEE Workshop on Real-Time Operating Systems.1992.

[4]Sanjeev Khushu and Johnathan Simmons.Scheduling and Synchronization in Embedded Real-Time Operating Systems.CSE 221. March 5,2001.

[5]罗蕾.嵌入式实时操作系统及应用开发.北京航空航天大学出版社.2005.

[6]Labrosse J J.嵌入式实时操作系统uC/OS-II.第二版.邵贝贝,宫辉,等译.北京航空航天大学出版社,2003.

[7]罗蕾,熊光泽.实时多任务应用最坏情况设计的研究.电子科技大学学报.1997(2).

[8]桑楠.嵌入式系统原理及应用开发技术.北京航空航天大学出版社.2002.

[9]褚敏芬.RTLinux调度策略的研究.微计算机信息.2003,11,91~92.

作者介绍:刘卫,男,1982年出生,现就读于南京工业大学自动化学院控制理论与控制工程硕士专业,研究方向为嵌入式系统智能仪表的开发与应用;程明霄,男,1950年12月生,江苏连云港人,副教授,研究方向为测控技术、智能仪器。

Biography:Liu Wei,male,College of automatization,Nanjing University of technology,research interest,real-time operation system, intelligent devices

(210009 江苏南京 南京工业大学自动化学院)刘 卫 程明霄 王晓荣

(College of automatization,Nanjing University of technology, Nanjing Jiangsu 210031,China)

通信地址:(210009 江苏南京南京工业大学丁家桥校区自动化学院测控系)程明霄

(收稿日期:2006.9.26)(修稿日期:2006.10.23)

(上接第157页)

USB移动存储设备。这类设备是USB Mass Storage类。同时,这类设备支持FAT文件系统,以文件的形式管理和存储数据。因此,作为用户软件,就需要建立一个精简的文件系统。整个文件系统通过Mass Storage的UFI命令与磁盘建立联系,实现建立FAT表、FDT表,查找文件、写文件和读文件等一系列功能。

3 结束语

USB为接口种类繁多的存储媒介与主机之间提供了方便可靠的连接。在嵌入式系统实现了USB的主机功能,实现了将USB移动存储设备应用到嵌入式系统这一领域,在实际应用中产生了数百万元的经济效益。

本文作者的创新之处在于:(1)基于USB主机控制接口芯片SL811HS,设计了一套USB Host嵌入式系统模块,使得传统意义上作为Slave端的嵌入式设备具有了Host端的功能。该系统支持USB Mass storage类,并实现了控制传输和批量传输,从而实现了与U盘的通信。(2)硬件上主处理器采用市面上流行的32位ARM核微处理器,软件框架上没有采用较流行的嵌入式操作系统,而采用中断任务调度机制,这些使得该系统在应用于其它嵌入式系统中比较方便,具有很高的参考价值。

参考文献:

[1]冉光志,赵明生.基于SL811HST和S3C4510B的嵌入式USB主机与设备的设计[J].微计算机信息,2005,11-2:77-79.

[2]Don Anderson,Dave Dzatko著,孟文译.《USB系统体系》(第二版)[M].中国电力出版社.2003.

[3]马伟.《计算机USB系统原理及其主/从机设计》[M].北京航空航天大学出版社.2005.

[4]陈启美,丁传锁.《计算机USB接口技术》[M].南京大学出版社.2005.

[5]张念淮,江浩编著.《USB 2.0硬件设计》[M].清华大学出版社.2005.

[6]Myke Predko著,陈逸 译.《PC接口技术内幕》[M].中国电力出版社.2002.

作者简介:黄樱(1972—),女(汉族),河北省乐亭县人,湖南工学院(筹)计算机科学系讲师,硕士,主要从事嵌入式技术、接口技术的研究,E-mail:hy.1972@163.com。

Biography:Huang Ying (1972-),Female (Han People),LeTing County,HeBei Province,Computer Science Department,Hunan Project College,Maste,Embeded Techology&Interface Research.E-mail:hy.1972@163.com.

(421008 湖南省衡阳市湖南工学院)黄樱 刘君 刘卉 叶晓舟

通讯地址:(421008 湖南省衡阳市湖南工学院计算机科学系)黄樱

(收稿日期:2006.12.17)(修稿日期:2007.1.15)

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net