

基于多核 ARM 的数控系统实时性研究^①

王颖慧^{1,2}, 郭锐锋², 韩卫光², 尹震宇², 邓昌义^{1,2}, 彭阿珍^{1,2}, 吴昊天^{1,2}

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

摘要: 随着数控系统的不断发展, 数控系统对精度和速度有了更高的要求, 这都对当前普遍采用的单核处理器平台提出了严峻的挑战. 基于同构的四核 ARM 处理器平台, 在添加了实时补丁的 Linux 系统环境下, 根据数控系统中硬实时任务、软实时任务、非实时任务并存的特点, 把数控系统任务合理的分配到多核 ARM 上运行, 并采用改进的 RM 算法进行调度, 最后对新的任务调度算法进行实时性能测试. 本调度模型能够解决了数控系统混合任务调度的问题, 并能满足数控系统对实时性的要求.

关键词: 多核 ARM; 实时调度; RM 算法; IRM 算法

Real-Time Performance of CNC System Based on Multi-Core ARM

WANG Ying-Hui^{1,2}, GUO Rui-Feng², HAN Wei-Guang², YIN Zhen-Yu², DENG Chang-Yi^{1,2}, PENG A-Zhen^{1,2}, WU Hao-Tian^{1,2}

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: With the continuous development of computer numerical control (CNC) system, there are higher requirements of accuracy and speed for CNC system, which brings grave challenge for the current most widely used single-core processor platform. In this paper, based on isomorphic quad core ARM processor platform which is under the Linux system environment with the real-time patch, according to the characteristic of hard real-time tasks, soft real-time tasks and non real-time tasks running parallel in the CNC system, the CNC system tasks are distributed to multi-core ARM reasonably. The improved RM algorithm is used to schedule. Finally execute the real-time performance test for the new scheduling algorithm. This scheduling model can solve the problems of mixed tasks scheduling of CNC system admirably, can well meet the real-time requirement of real-time control of CNC system.

Key words: multi-core ARM; real-time scheduling algorithm; rate-monotonic scheduling algorithm; IRM algorithm

1 引言

目前基于嵌入式芯片的数控系统普遍采用单核处理器平台, 随着对数控系统功能和性能要求的不断提高, 对处理器平台的性能要求也越来越高, 在单核平台上运行高档数控系统软件遇到了严峻的挑战. 单纯通过提高 CPU 的主频和指令的执行效率来提高性能, 不仅会带来巨大的功耗问题, 而且还面临着技术和工艺方面的瓶颈. 多核处理器的出现, 大大提高了处理器的性能, 满足了人们对处理器性能提高的要求, 成

了商业化处理器的发展趋势, 但也给体系结构、软件、功耗和安全性设计等方面带来了巨大挑战, 多核系统线程调度算法就是其中一个重要的研究方向. 随着数控技术的发展, 加工对象的复杂度增加, 处理器处理运动控制插补运算的高速度高精度要求更高, 要解决这一问题, 除了更换更好性能的 ARM 芯片, 多核 ARM 是其另一个重要的发展方向.

数控系统具有实时性和多任务的特点^[1], 如当数控系统正处于加工控制状态时, 各数控加工程序段的

① 基金项目: 国家科技重大专项(2014ZX04009031)

收稿时间: 2016-08-03; 收到修改稿时间: 2016-09-05 [doi: 10.15888/j.cnki.csa.005714]

预处理、插补计算、位置控制和各种辅助控制任务都要及时进行。且数控系统是一个实时任务系统,其任务按是否具有实时性要求和周期性要求可分为:实时周期性任务、实时突发性任务和非实时任务。数控系统任务的复杂性,使得单核处理器不再适应数控系统对速度和精度的要求,而将多核 ARM 应用于数控系统可以根据任务的不同性质将任务合理的分配到不同的核,减少所有任务都等待同一个处理器而增加的延时,从而更好的适应数控系统的要求。目前,对多核处理器实时调度算法的研究主要为了提高多核处理器系统的负载平衡,本文采用改进的 RM 调度算法,主要针对数控系统混合任务的特点,可减小插补周期的延迟,提高数控系统加工工件的精度和速度。

2 基于数控系统的RM调度算法及其改进算法

2.1 RM 调度算法的概述

1973年 Liu 和 Layland 提出的单调速率调度算法(RM)^[5]是一种静态优先级调度算法,已被证明是可抢占的硬实时周期性任务调度的最优算法。RM 算法应用于周期任务,根据任务的周期来分配优先级,周期越短,优先级越高。理想的 RM 调度模型是基于一系列假设的基础上的,这些假设是 RM 调度算法可执行的必要条件, RM 调度模型的基本假设如下:

(A1) 所有的任务请求都是相隔固定时间触发的即任务具有周期性,且为硬实时任务,任务必须在规定的时限内完成;

(A2) 任务的周期即为任务的截止期,即任务必须在下一次任务到来之前完成;

(A3) 任务之间是相互独立的,每个任务的请求和执行与其他任务的请求和执行无关;

(A4) 每个任务的运行时间是不变的,这里任务的运行时间是指处理器在无中断情况下用于处理该任务的时间;

(A5) 系统调度和任务的上下文切换的时间不计;

(A6) 任务之间是可抢占的,高优先级任务可以抢占低优先级任务;

(A7) 默认 CPU 是任务唯一需要争夺的资源,任务所需的其他资源都是可满足的。

给定一个独立的任务集 $S=\{t_1, t_2, \dots, t_n\}$, 集合中任务 $t_i=\{E_i, T_i, D_i\}$ ($i=1, \dots, n$), 其中, E_i 表示任务 t_i 的执行时间, T_i 表示任务 t_i 的周期, D_i 表示任务 t_i 的截止期限。如

果这 n 个任务的 CPU 利用率满足下面的条件:

$$U = \sum_{i=1}^n \frac{E_i}{T_i} \quad (1)$$

$$U < n(2^{1/n} - 1) = L(n) \quad (2)$$

则该任务集 S 用 RM 算法是可调度的。 $L(n)$ 表示这 n 个任务的 CPU 利用率的最小上界。当任务的数量 n 趋向于无穷大时, $L(n)$ 的最大值约为 0.69, 即系统的 CPU 利用率最大值为 69%。图 1 为 RM 调度算法的任务调度图。



图 1 RM 调度算法任务调度

由于 RM 调度算法的可执行条件的限制,导致 RM 调度算法在面对数控系统的实时环境方面存在一些不足。首先,数控系统是混合任务系统,不仅有周期任务还有非周期任务,而 RM 调度算法规定实时任务必须为周期任务;其次, RM 调度算法是固定优先级调度算法,它的优先级只取决于任务的周期,任务周期越短,优先级越高,但是数控系统中的实时任务除了实时周期任务,还有实时突发任务,这些任务周期不固定且具有很高的实时性,需要系统及时的响应和处理。使用 RM 调度算法可能会使得一些重要的实时突发任务得不到调度,从而超过任务的时限,给系统带来不可挽回的错误。

2.2 基于 RM 调度算法的改进算法

基于以上提到的 RM 调度算法的不足,文献[3]提出一种基于 RM 调度算法的单调速率计算能力调度算法^[3],该算法是应用于云环境任务调度。因数控系统有其自身的特点,例如:混合多任务、高要求的实时性等。本文在该算法的基础上进行改进,提出一种能更好的适用于数控系统任务环境的调度算法 IRM 算法,使其更好的满足数控系统对实时性的要求。

文献[3]在 RM 调度算法的基础上引进参数 I , I 表示一个任务的重要程度,数控系统操作者根据经验静态的给不同的实时任务赋一个 I 值,则数控系统任务的优先级为:

$$P_i = \xi I + (1 - \xi) D_i \quad (3)$$

其中, D_i 为任务的截止时间, 周期任务的 $T_i=D_i$, ξ 根据任务的性质设定, P 值越大表示任务的优先级越高. 本文在该算法的基础上引入一个计数值 N , N 的值表示数控系统实时任务还可以等待多长时间, $N=D-E-C$ (C 表示任务已经等待的时间), 这可以判断哪些任务不能在该核上继续等待, 否则会错过截止时间, 通过对 N 值的判断可以将不能继续等待的任务迁移到其他核运行, 从而避免这些任务错过截止时间. 设置两个队列 Q 、 R 分别存放数控系统实时任务.

IRM 调度算法描述如下:

- (1) 调度开始;
- (2) 通过(3)式计算各个任务的优先级, 按优先级从高到低进行排列;
- (3) 设置两个队列 Q 和 R , 且为每一个实时任务设置一个 N 值, 计算并保存 N 值;
- (4) 按优先级从高到低选取实时任务, 使用(1)(2)式来计算实时任务集是否可调度, 选取最大可调度集, 将此可调度集任务加入队列 Q 中, 将剩余任务加入队列 R 中;
- (5) 选取队列 Q 中的优先级最高的任务调度执行, 同时每个时钟周期将任务的 N 值减一, 若某个任务的 $N < 1$, 则将此任务迁移到其他核执行;
- (6) 判断队列 R 是否为空, 如果为空, 执行第 7 步, 如果不是空队列, 则执行第 4 步;
- (7) 判断队列 Q 是否为空, 如果是空执行第 8 步, 如果不为空跳回到第 5 步;
- (8) 调度结束.

由此看出, 改进的 RM 调度算法, 任务的优先级不再仅仅取决于任务的周期, 而是任务的重要度和截止期的二元函数. 同时为每一个任务维持一个表示还能等待多长时间 N 值, 保证任务不会由于等待而错过截止时间. 这样在多任务混合的数控系统中, 不仅能够保证周期任务的正常运行, 而且能够使得非周期任务能及时响应. 由此可以看出, IRM 调度算法比 RM 调度算法更加适用于多种实时任务并存的数控系统, 能更好的满足数控系统对实时性的需求.

3 多核处理器下数控系统的架构及实验结果分析

3.1 多核处理器下数控系统的架构

按照对实时性要求的不同, 数控系统的任务可分为硬实时任务、软实时任务和非实时任务. 在单核处

理器时代, 其主要解决方法为在保证实时任务能够在周期内完成的前提下, 在处理器空闲的时间才执行非实时任务. 由于处理器的个数限制, 使得非实时任务的等待时间大大的增加, 甚至导致非实时任务得不到运行. 同时也可能由于多个实时任务的周期相近而导致实时任务不能在截止期内完成.

轨迹插补是数控系统最重要的计算任务, 轨迹插补具有实时性和精度的要求, 数控系统如果需要提高加工零件的精度, 则需要提高插补的精度, 插补精度可以通过减小步长获得, 但减小步长将导致计算量成倍地增加, 极大地限制机床进的进给速度的提高, 甚至可能造成数控系统无法进行实时加工. 本文基于以上考虑, 从任务分配和任务调度两个方面来缩小插补周期, 提高数控系统的加工精度, 同时保证数控系统的其他任务能够更加高效的运行. 基于数控系统对实时性的需求, 本文基于四核的 ARM 硬件结构, 采用开源的 Linux 为软件系统来搭建数控系统.

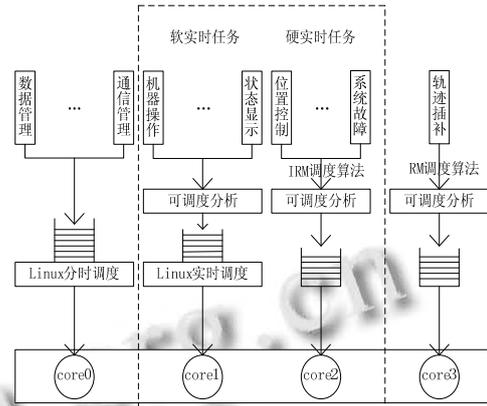


图 2 数控系统任务分配

图 2 中, 数控系统非实时任务分配到 core0 上执行, 使用 Linux 的分时调度器调度. 软实时任务在 core1 核上调度, 采用 Linux 的实时调度器进行调度. core2 核上调度数控系统的硬实时任务, 采用 IRM 调度算法进行调度, 当 core2 核上某个任务的剩余等待时间 $N < 1$ 时, 则将此任务赋予一个最高优先级, 迁移到 core1 核上运行. core3 核上运行数控系统的插补任务, 使用 RM 调度算法进行调度, 单独为插补任务分配一个内核运行, 有利于插补任务周期的缩短, 能有效提高数控系统的加工精度. 此任务分配模型既可以保证数控系统重要的实时任务能够得到更快的执行, 也可以保证非实时任务能够及时的响应, 有效的解决了数控系统多任务混合的调度问题.

3.2 仿真实验与结果分析

本文基于四核 ARM 结构,操作系统为添加了实时内核的 Linux 系统,实时内核为 Xenomai.

3.2.1 Linux 系统实时性能测试

本文针对数控系统的延迟时间进行了测试,分别就 Linux 系统和添加了 Xenomai 实时内核后的系统进行了 20000 次实验,以下是测试结果.

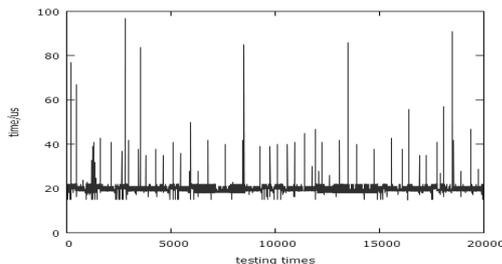


图3 未添加实时补丁的系统延迟

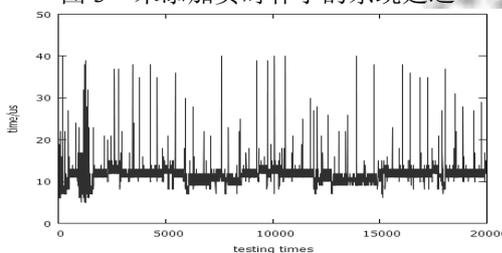


图4 添加实时补丁的系统延迟

图3为标准Linux内核的延迟,延迟的最大值为97us,最小值为15us,平均值为20us,延迟的波动性较大,延迟时间较大,系统的实时性能不佳.图4为添加了实时补丁后的系统延迟,其最大延迟为40us,最小延迟为5us,延迟的时间比较稳定,平均延迟为11us.从结果可以看出,增加了实时补丁的Linux系统延迟明显减小,系统延迟更加稳定,能更好的提高系统对任务的响应速度,使得增加了实时补丁的系统比之前的系统在实时性能方面有很大的提高,能更好的满足数控系统对实时性的要求.

3.2.2 数控系统实时任务调度测试

本文基于搭建的数控系统仿真环境进行性能测试,插补算法选用逐点比较法^[10],直线插补算法是在本地进行模拟测试,设插补周期为1ms和0.8ms,分别用C语言对直线插补进行仿真,把交叉编译后的程序在目标平台上运行,测试结果表明该插补算法可以在目标平台上运行并绘制出一条直线,插补运算都能在其周期内完成,从而说明数控系统的实时性能基本能够满足数控系统插补运算的要求,能够明显的缩短数控系统的插补周期.

4 总结

由于单核处理器在性能上遇到的瓶颈问题,多核处理器的出现有效的提高了系统的性能和效率,将多核处理器应用到数控系统能有效的提高数控系统的效率,满足数控系统对速度和精度的要求.本文基于四核的ARM处理器,搭建添加了实时补丁的Linux操作系统,将数控系统任务合理的分配到每个核上运行,很好的提高了数控系统的效率.同时使用改进的RM调度算法来调度数控系统的实时任务,能够进一步提高数控系统的加工速度和精度.接下来的进一步任务是进行大量的实验,根据这些实验的经验确定合理的插补周期,既能保证数控任务精度的要求也能使插补运算能够在周期内完成.

参考文献

- 1 王保胜,马跃,吴文江,秦承刚.开放式数控系统任务调度模型.控制与检测,2010,(6):37-39,42.
- 2 丁万夫,郭锐锋,高甜容,秦承刚.面向数控系统的三级调度模型研究.小型微型计算机系统,2011,3(3):439-443.
- 3 王琪元,闫宏印.基于改进RM算法的云环境任务调度研究.计算机测量与控制,2013,21(6):1612-1614.
- 4 郑玉虎,王品,陆小虎.基于双核处理器平台的数控系统软件架构设计.组合机床与自动化加工技术,2013,(8):62-65.
- 5 Liu CL, Layland J. Scheduling algorithms for multi-programming in a hard real-time environment. Journal of the ACM, 1973, 20(1): 46-61.
- 6 张修棋.基于Linux多核进程调度的研究[硕士学位论文].成都:电子科技大学,2012.
- 7 张晓龙.实时系统性能测试方法的研究及应用[硕士学位论文].北京:中国科学院大学,2014.
- 8 姚鑫骅,潘雪增,傅建中,陈子辰.数控系统的混合任务模型及其最优调度算法研究.浙江大学学报,2006,40(8):1315-1319.
- 9 王涛,王太勇,许爱芬,赵丽,杨洁.嵌入式实时操作系统在数控系统设计中的应用.计算机工程,2008,34(4):250,251,268.
- 10 金中波,张百臣,韩霞,代洪庆.逐点比较法直线插补原理及其改进算法分析.机械工程师,2009,3:126-128.
- 11 Laplante PA, Ovaska SJ. Real-Time Systems Design and Analysis. IEEE Press, 2012: 237-253.
- 12 韩守谦,裴海龙,王清阳.基于Xenomai的实时嵌入式Linux操作系统的构建.计算机工程与设计,2011,32(1):96-98,102.
- 13 Zheng YH, Wang P, Lu XH. CNC Software architecture design based on dual-core CPU platform. Modular Machine Tool & Automatic Manufacturing Technique, 2013, (8): 62-65.