

限制优先次数的优先级调度算法

钱光明,刘弢

(湖南师范大学 数学与计算机科学学院,湖南 长沙 410081)

摘要: 优先,可以说这个世界无处不有。银行排队的时候,先来的客户应该先得到服务;交钱多的人往往比普通人优先;领导较员工优先;女士比男人优先;孩子比成人优先;老人比年青人优先,等等例子实在太多。在信息和计算机系统中,尤其是多任务和/或多用户环境中,也常常采用优先调度策略,这些策略的目的在于尽量合理地利用系统资源,以满足系统的设计要求。该文首先回顾和比对了几个经典的、代表性的优先级调度策略,然后详细提出并描述了一种新的方案,即限制优先次数的优先级调度算法。

关键词: 优先调度;优先级调度策略;限制优先次数

中图分类号: TP311 **文献标识码:** A **文章编号:** 1009-3044(2013)34-7765-02

1 代表性的优先级调度策略回顾

普通的固定优先策略。每一个任务固定分配一个优先级,每次调度时,从就绪队列中选最高优先级的那一个任务来运行。该算法不需要去追究是谁提出的,因为普通人均能想到,长期以来在计算机中实际采用。如上世纪 PDP-11 系列计算机中采用的 RSX-11M,用的就是该策略,这个例子当代许多大学生可能都不知道。又如,生于 20 世纪 90 年代且仍大量应用的 $\mu\text{cos-II}$ 及 $\mu\text{cos-III}$ 操作系统,也采用了该固定优先策略^[1]。

RM(Rate Monotonic) 调度策略。也是一个静态的、固定的优先策略^[2]。只不过按每个任务的周期分配其优先级,周期越短,优先级越高,永远如此。只要一开始排好队,就可以一劳永逸地永远调度下去,实现方便,因此较早地就被吸收进 Linux 中。与上一算法一样,该算法不能动态地依据实际情况调整优先级。并且,它只根据固定的周期确定任务的优先级,难道周期短的任务一定就该优先吗?现实世界中并非总是如此。

最早截止期优先策略。即 EDF(Earliest Deadline First)^[3]。有别于上面的静态优先策略,该算法属动态优先策略。它认为每一个任务的每一次作业均有一特定时间点,本次作业必须在该点之前完成,该时间点被称作截止期。算法提出时主要针对实时系统,即对时间有要求的系统,如工业控制场合。而实际环境中,很多其它普通场合,任务的执行时间同样是人们所要求或期盼的。例如,要连接某一个网站浏览网页时,如果过了几十秒仍未打开,多数人可能无法忍受,浏览器也设置了一个超时时间限制。因此,EDF 思想具有的意义不只是局限在实时领域。

如果任务数量和周期固定,且每个任务作业的截止期与周期相等,EDF 算法已被充分研究,具有不错的性能。但是,对于任务数或参量变动的模型,该算法仍然有相当的研究空间和难度。需要指出的是:该算法虽能动态地调整优先级,但只依据截止期,与实际场合的对应非常有限。另外,在实际编程的时候,因为要依据运行时刻来动态调整调度队列,需要设置计数器,因此 EDF 比 RM 实现难度要大不少,尤其是系统中任务数量较多的时候。

加权的公平排队策略。该策略主要用于网络数据包发送。不同来源端口的数据包被分配不同的权值,依据权值的不同,端口数据被调度器赋予不同的服务时间。该算法的关键在于权值的选择。数据量大和/或优先级高(如交钱多的)的用户,可分配较大的权值。在类似 ATM 这种以信元为单位的网络中,权值的大小实际上就是时间片的多少。而在以不定长数据包为发送单位的网络上,先是依据权值求出各来源端口数据包可能的完成时刻,然后将这些完成时刻排队,先发送最早完成时刻对应的数据包。

通过引入不同的权值,该算法虽然比 EDF 策略更能反映用户的身份差别,但也不能包打天下。例如会出现这样一种优先级被反转的情况:如果在某一时刻,优先级高的任务(设为任务 A)数据包很长,优先级低的任务(设为任务 B)数据包很短,那么,即使任务 A 的权值大于任务 B,但如果任务 B 的完成时刻早于任务 A,优先级低的任务 B 会比优先级高的任务 A 先得到服务。

实时 **RR** 策略和 **SCHED-OTHER** 策略。这是 Linux 中两个著名调度策略,实时 **RR** 指的是某实时任务用完自己的时间片之后,将被排到调度队列的尾部,等待下一轮调度机会^[4]。**SCHED-OTHER** 主要对普通任务,它是依据任务的动态优先级来选择任务,实际上是依据一个静态值和动态值之和(权值)进行决策。权值最大的任务将被投入运行。随着任务的执行,其动态值慢慢减小,当其时间片用完之时,动态值变为 0。实时 **RR** 和 **SCHED-OTHER** 较好地兼顾了公平性与优先级。但是,它们均是基于时间片的策略。基于时间片设置计数器,在系统中需要一定的开销。况且,以时间片来衡量实际任务的执行时长并非总是那么完善。例如,一个特定任务很可能在一个新的时间片刚到来后就执行完了,时间片设置过长时尤其如此。时间片设置过短又会增加开销。

收稿日期:2013-09-04

作者简介:钱光明,男,教授,主要研究方向为嵌入式和实时系统;刘弢,男,硕士研究生。

本栏目责任编辑:谢媛媛

■■■■■■■■■■ 软件设计开发 ■■■7765

