国 防 科 技 大 学 学 报
第21卷 第4期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol. 21 No. 4 1999

DEC UNIX 操作系统的实时特性及评测*

毛羽刚 金士尧 凌云翔

(国防科技大学计算机学院 长沙 410073)

摘 要 实时操作系统是复杂实时系统的一个重要组成部分,它必须是可预测的。本文首先简要阐述了DEC UNIX 4.0 操作系统的实时特性;然后开发了一种简单有效的实时操作系统性能评测方法,并给出了我们在DEC Alpha 系列机上对 DEC UNIX 4.0B 进行评测的过程和结果。

关键词 实时操作系统,可预测性,延迟时间

分类号 TP302

DEC UNIX Real-Time Operation System Performance and Test

Mao Yugang Jin Shiyao Ling Yunxiang

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract Real-time operating system is one important component of complex real-time system. The operating system itself must be predictable. This paper first analyzes and describes the real-time functions of DEC UNIX 4.0. And then we secondly developed a effective test approach of real-time operation system performance. Finally we give the test process and results of DEC UNIX4.0B running on DEC Alpha series computers.

Key words real-time operating system, predictability, latency time.

实时系统在航空、航天、国防等领域的应用越来越广泛。它最重要的一个性质是要求系统响应时间可预测,包括:硬件延迟,操作系统开销和应用软件的最大执行时间等。其中硬件是基础,在硬件性能一定的条件下,系统响应时间主要由实时操作系统来保证。中断响应时间,进程之间的切换、抢占、同步与通信等时间是衡量操作系统实时性能的重要指标。由于操作系统结构和运行过程的复杂性,很难精确测量这些数据。通常评测的方法有:一是对有关源代码进行分析和测试;二是在系统中插入一些监控点或跟踪进程来观察和测量。这些方法都需要测试者对操作系统内部结构有比较深入的理解,测试过程复杂。本文根据 DEC Rhealstone 测试框架和软件执行时间测试的有关思想,设计和实现了对 DEC UNIX4.0 操作系统实时性能测试的方法和代码。并对文献[1]中所述系统的 DEC Alpha 系列机器进行了测试。该方法简单、有效,测试结果对实时系统的设计和论证具有重要参考价值。

1 DEC UNIX 的实时功能

DEC UNIX4.0 是当前应用比较广泛的一种操作系统,它在通用分时 UNIX 基础上增加了强大的实时功能。既有通用操作系统标准的编程接口、丰富的开发工具和较好的人机界面,又有良好的实时性能。它遵循 POSIX 标准,包括系统程序设计接口标准 P1003.1,线程标准 P1003.1_c,实时扩展程序设计接口标准 P1003.1_b,使得应用程序可移植性好。具体有如下几个主要实时功能。

1.1 固定优先级调度策略

DEC UNIX4.0 支持两种调度接口:nice 和实时接口。实时接口支持两个固定优先级的抢占调度策略:FIFO 和循环策略。进程优先级和调度策略由应用程序设计者决定,而不是由系统动态改变。所有进程优先级可分为三类:最高优先级留给实时应用;中间和最低优先级由系统和非实时进程使用。超级用

* 国家部委基金项目资助
1998 年 12 月 22 日收稿
第一作者:毛羽刚,男,1966 年生,博士生

户可调用实时函数修改进程的调度策略和优先级。但硬件中断不受进程优先级的影响。这些机制对保证高优先级实时任务的响应时间提供了强大支持。

1.2 抢占式内核

DEC UNIX4.0 内核具有实时抢占能力,允许高优先级进程中断低优先级进程的执行,而不管低优先级进程处于用户方式还是内核方式。内核支持进程同步概念,可保证数据完整。内核的最大进程抢占延迟是指:内核维护系统数据完整且抢占正运行进程所用的时间。一般最坏抢占延迟也只是毫秒级的。而非抢占内核的延迟常要以秒为单位来计算。

1.3 实时时钟和定时器

系统时钟是每个进程定时器的定时基础和同步源。实时应用设计者可用定时器控制每个进程的动作。DEC UNIX4.0 系统时钟精度为1/1024秒。定时器可由用户进程创建和设置,并与信号机制关联触发一定的动作。它可被设为一次性或周期性的,定时时间可被设为绝对或相对时间。DEC UNIX4.0 的时钟和定时器功能与非实时操作系统的主要区别是:精度高、灵活、易用。

1.4 存储器加锁

存储器加锁是 DEC UNIX4.0 实时应用设计者用于减小延迟的主要工具之一。它允许实时应用程序将进程地址空间锁入内存中,包括当前和将来需要使用的地址空间。通常,实时进程必须被锁定在内存中,对大多数实时任务来说,由于页交换而造成的等待是难以接受的。

1.5 异步 **I/O**

传统UNIX 的I/O 进程在等待I/O 操作时必须停止执行,直到I/O 操作完成。DEC UNIX4.0 异步I/O 操作允许:一旦某个 I/O 操作进入队列,调用该I/O 操作的进程就可立即恢复执行。当 I/O 操作完成时可通过信号机制通知该进程。因此,应用程序可在不被中断的情况下同时启动多个 I/O 设备并发操作。这对许多实时应用来说十分必要。

1.6 进程间通信机制

进程间通信是指两个或多个进程间的信息交换。主要有如下几种形式:

(1) 共享存储器:是进程间通信的最快形式。只要一个进程在共享存储区中写入数据,就可被其他使用同一共享存储器的进程访问。

(2) 信号机制:信号常被称为软中断,一般用于表示某一事件的发生,例如定时器超时、异步 I/O 完成等。信号可被用户进程、核心进程或驱动程序发送给另一用户进程或用户进程发送给自己。接收进程可对信号采取三种动作:忽略、缺省动作或调用事先定义的处理例程。DEC UNIX4.0 信号机制使用了一些特殊的数据结构使得信号的传递异步、快速和可靠。

(3) 信号灯:用于控制进程同步访问系统资源。这些资源包括全局变量和硬件资源等。DEC UNIX 支持POSIX 程序设计实时扩展接口标准 P1003.1b 定义的计数信号灯机制。

(4) 消息:合作进程可通过访问系统消息队列进行通信。任何进程无论与其它进程是否关联,都可通过同步或异步访问消息队列来交换信息。进程在等待接收信息时可以执行其他工作。同一消息队列可以存取多个消息,也可被多个进程存取。

2 测试过程和结果

本文测试的实时操作系统版本为 DEC UNIX 4.0B。测试的基本思想来源于对短程序代码执行时间的测试,可用如下伪码表示:

```
time0 = current_clock_time;                    /* read time now* /
for ( j = 1; j ≤ LOOP; j + + ) {do}            /* run empty loop* /
time1 = current_clock_time;
loop_time = time1 -time0;
time2 = current_clock_time;
for ( j = 1; j ≤ LOOP; j + + ) {function_to_be_tested};
```

```
time3 = current_clock_time;
total = (time3-time2-loop_time) / LOOP;
```

根据该思想和 DEC Rhealstone 测试框架,我们编制和实现了测试代码。测试在两个实时进程之间进行。先用超级用户权限将其设为系统中两个最高的实时进程,并锁定在内存中,不会被任何进程中断。然后计算 empty 时间,它包括循环指令和有关实时原语的调用开销。再计算 elapsed 时间,它既包括循环指令和实时原语调用开销又包括有关的延迟时间。各循环 LOOP 次以保证测试精确。最后两者相减并除以 LOOP 或 2* LOOP,即为实时原语的调用延迟。测量精度可通过下式计算:

$$\eta = \frac{\pm C_{clock}}{\tau^* \ LOOP} * 100\%$$

Cclock 是时钟粒度,τ为所测代码的延迟时间。下面我们给出在 DEC Alpha 433 机器上对各种延迟分别进行测试的过程,测试结果如表 1 所示(共测试 5 次,LOOP= 10^6)。

2.1 进程切换延迟的测试

首先创建父进程,它循环执行进程切换原语 Sched_yield() LOOP 次(此时系统只有一个实时进程,并不发生任何实际切换),并记录循环过程的执行时间。然后创建子进程,并调用 Sched_yield(),将执行权切换给子进程,子进程再调用 Sched_yield(),将执行权切换给父进程,如此循环 LOOP 次。并记录该过程执行时间。最后,两次记录时间相减并除以 2* LOOP 就是进程上下文切换延迟。

2.2 进程抢占延迟的测试

首先创建父进程,它循环执行优先级设置原语 Set_priority_highest 和 Set_priority_highest-1 LOOP 次,记录时间。然后创建子进程,调用 Set_priority_highest-1,子进程抢占父进程后,将父进程优先级设为highest,循环 LOOP 次,记录时间。两次记录时间相减并除以 2* LOOP 就是所求。

表 1 测试结果(ns)

Table 1 Test results

	1	2	3	4	5
进程切换时间	1792	1745	1720	1735	1696
进程抢占时间	4181	3672	3562	3664	3609
消息传递时间	7905	7898	7838	8006	7730
信号灯切换时间	48330	55368	48767	54027	54333
中断响应延迟	2690	1589	5680	7776	6541

2.3 任务间消息传递延迟的测量

首先创建父进程,它循环调用 Send_msg(),Receive_msg(),Sched_yield() 函数 LOOP 次,并记录时间。然后创建子进程,并向子进程发送一条消息,然后将控制权切换给子进程,子进程接收消息后,将执行权切换给父进程,循环 LOOP 次,并记录时间。两次记录的时间相减并除以 LOOP 就是所求。

2.4 信号灯请求延迟

首先创建父进程,循环执行进程切换原语 LOOP 次,并记录时间。然后创建子进程,申请并占用信号灯,再将执行权切换给子进程。子进程申请同一信号灯但被阻塞。父进程运行,释放信号灯,又将执行权切换给子进程,子进程获得信号灯后,将执行权交给父进程,父进程申请信号灯但被阻塞。子进程运行并释放信号灯,又将执行权交给父进程。如此循环 LOOP 次并记录时间。最后,两个时间相减并除以 2* LOOP 就是所求。

2.5 中断响应时间

该延迟是指从内核接收中断到中断处理例程第一条指令执行之间的时间。测试时,先将一随机负数加载到实时时钟计数器中。时钟计数到零时产生中断,并继续计数。用中断处理例程的第一条指令读时钟值,即为中断延迟时间。

3 结果分析和比较

根据需要,本文还在其他两种DEC Alpha 平台上进行了测试,并与DEC 公司提供的,在 DEC3000/

500平台上给出的数据进行了比较。从测试结果看出: UNIX4.0B 在其系列平台上运行时的几种主要实时性能指标是微妙量级的, 并且机器速度越快, 延迟时间越小, 系统反应越快。虽然所测结果是统计平均值, 但对文献[1]所述系统来说, 它大大的小于实时任务的执行时间(ms 级)。如果实时软件系统有良好的结构, 则实时操作系统的开销是完全可预测的。

表 2 结果比较(ms)

Table 2 Compare of results

	DEC3000/500/150 MHz	Alpha 274/274MHz	Alpha 333/333MHz	Alpha 433/433MHz
进程切换时间	15.9	8.1	2.3	1.7
进程抢占时间	27.3	17.6	5.4	3.5
消息传递时间	49.0	40.9	11.2	7.9
信号灯切换时间	261.0	152.9	53.2	48.7
中断响应延迟	5.8—28.1	4.3- 21.4	2.8- 12.6	1.5- 7.8

4 结束语

由于计算机技术的迅速发展, 实时系统越来越受到重视。实时操作系统是实时系统获得可预测性的一个关键部分。本文分析了 DEC UNIX4.0 操作系统的实时性能, 并给出了测试方法和结果。这对实时系统的设计和时间验证等多方面的工作具有积极意义。

参考文献

1 金士尧, 王志英, 胡华平. 强实时高可靠性群机系统的研究. 计算机工程与科学, 1997, 19(A1): 1～5

2 赵龙, 胡宁. 对 UNIX 实时扩充的一点研究. 计算机工程与科学, 1998, 20(4): 76～80

3 DEC OSF/1 实时编程指南. 华北计算所超级小型机系统编辑部

4 Ramamritham K, Stankovic J A. Scheduling Algorithms and Operating Systems Support for Real-Time Systems. Proceedings of the IEEE, Vol. 82, No. 1, January 1994: 55～66