第34卷 第19期 计 算 机 工 程 2008年10月
Vol.34 No.19 Computer Engineering October 2008
·开发研究与设计技术· 文章编号：1000—3428(2008)19—0261—03 文献标识码：A 中图分类号：TP391

基于 RTAI-Linux 的飞行仿真实时管理系统

曾 炜，沈为群
(北京航空航天大学自动化科学与电气工程学院，北京 100083)

摘 要：探讨基于 Linux 的飞行仿真实时管理系统需要解决的实时控制问题。研究 Linux 实时扩展的开源项目 RTAI 及其用户态硬实时控制方案 LXRT，结合系统的开发研制分析如何利用 RTAI/LXRT 实现关键任务的实时控制以及实时和非实时任务的管理。最终实现的多任务飞行仿真管理系统具有良好的实时性，在 1 ms, 5 ms, 10 ms 仿真周期下，关键任务的最大单步误差均小于 30 μs。
关键词：飞行仿真；实时；Linux 操作系统

Real-time Management System of Flight Simulation Based on RTAI-Linux

ZENG Wei, SHEN Wei-qun
(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】 This paper explores how to address the real-time control problem of real-time management system of flight simulation based on Linux. For this purpose, RTAI, a community project for real-time extension of Linux, and its hard real-time solution in user mode, LXRT are analyzed, From the view of research and development of the management system, it discusses how to use RTAI/LXRT to realize hard real-time control of critical mission and how to manage real-time and non real-time tasks. The ultimate realization of the multi-task flight simulation management system has a good performance in real-time control, the biggest single step deviations of critical tasks in 1 ms, 5 ms, 10 ms cycle simulation are all less than 30 ms.
【Key words】 flight simulation; real-time; Linux operating system

1 概述

飞行模拟器是现代飞机设计和飞行员训练领域的重要工具，通常是由多台高性能计算机和各种外围设备组成的复杂综合系统，而飞行仿真管理系统则是飞行模拟器的中枢神经，在负责人机交互的同时，还要对各分系统进行实时调度、管理和同步，其软件复杂程度高，而且具有硬实时要求[1]。因此，开发平台的选择尤为重要。

目前，国内在建的各类飞行模拟器基本上都是直接购买并使用国外的商用实时操作系统作为其飞行仿真管理系统的实时应用平台，如 VxWorks, QNX，这样做可以缩短开发时间，而且实时性和可靠性有很好的保障，但开发成本高且核心技术无法掌握。本文利用 Linux 操作系统开源的特点，结合 Linux 实时性扩展的开源补丁 RTAI(Real-time Application Interface)，包括界面开发工具在内，全部采用免费的开源软件，开发出成本低且开放性好的飞行仿真实时管理系统。

2 系统功能与结构

本文研究的飞行仿真实时管理系统主要为工程飞行模拟器提供人机交互、飞机建模、数据管理、曲线显示、模型计算、网络通信、任务管理等功能，按功能划分的结构如图 1 所示。管理系统分布在 3 台 X86 结构的高性能计算机上，其中人机界面、建模软件和数据库程序作为控制中心模块运行在 PC1 上，开发平台为 Fedora Core 6；模型计算、网络通信、任务管理等程序作为实时计算模块运行在 PC2 上，开发平台为实时改造后的 Fedora Core 6(内核版本为 2.6.19.1)；曲线和参数显示程序作为图形监控模块运行在 PC3 上，开发平台为 Fedora Core 6。3 台计算机之间通过高速以太网进行通信，分别采用 TCP 和 UDP 协议传输命令参数和计算数据。

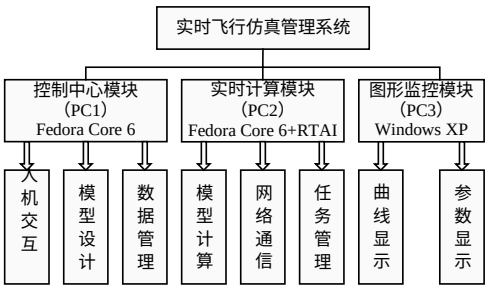

图 1 系统结构

管理系统各模块的主要功能和实现方式如下：

控制中心模块提供控制界面，实现人机交互，向实时计算模块提供各种初始设置和控制命令，包括飞行参数、仿真环境、仿真周期等的设置以及准备、运行、暂停、停止、复位、退出等控制命令[2]；对Linux下的各种应用程序进行统一管理，包括开源建模软件Scilab、RTAI-Lab、开源数据库MySQL等；并负责将建模软件生成的飞机模型的可执行代码下传至实时计算模块。所有界面均采用开源的可视化编程工具Glade3.0 和图形界面工具库GTK+2.0 实现。

实时计算模块负责接收控制中心模块的初始设置和控制

作者简介：曾 炜(1983－)，男，硕士研究生，主研方向：计算机实时控制与仿真；沈为群，副研究员
收稿日期：2007-11-15 **E-mail：**anticternn@yahoo.com.cn

命令，对下载的模型进行实时解算，通过高速网络与运动系统、操纵系统、视景系统、图形监控模块等子系统进行数据交换，对所有实时和非实时任务进行同步和管理。采用 RTAI 对实时任务进行硬实时控制，并利用 RTAI 提供的实时 FIFO、实时 Mailbox 和实时 IPC 等通信机制实现实时任务间的通信。

图形监控模块负责对各子系统的变量分别以参数和曲线的形式动态显示，实现了监控变量的动态增删、曲线的缩放和游动、坐标点的读取等功能。该模块采用 XML 标记语言和 GTK+2.0 实现。

3 RTAI 和 LXRT

RTAI 是由意大利米兰理工学院开发的一个开源项目，其目的是实现 Linux 对任务的硬实时控制，同 RT-Linux 一样是双内核架构的典型代表，由于其移植难度小，发展和更新速度快，且一直坚持开源，目前已成为最流行的 Linux 硬实时化改造方案。

双内核方案最早是由RT-Linux提出的，而后来其设计者转向了RT-Linux商业版本的开发，为该方案申请了专利，RTAI的开发者为了避免引发纠纷，将RTAI移植到了ADEOS(Adaptive Domain Environment for Operating Systems)上[3]，实现了可移植性和开放性更好的新版本，并在此基础上提出了用户态下的硬实时控制方案LXRT。

ADEOS是免费的开源软件(FOSS)，其目的是提供一个灵活的环境，实现不同操作系统或者同一操作系统内不同程序对相同硬件资源的共享，其核心机制是域和中断管道机制。RTAI移植到ADEOS就是将普通Linux和RTAI定义成不同的域，并将RTAI域设置为优先级最高的域，然后利用它在中断管道中的入口优先接收和处理系统中断，而系统所有的中断都必须通过中断管道传递[4]，因此，RTAI域具有整个系统的中断裁决权，如图 2 所示，正是该中断处理机制帮助RTAI实现了对中断的快速响应和对任务的硬实时控制。

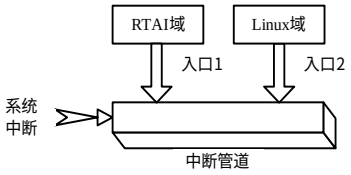

图 2　RTAI 的中断处理机制

在ADEOS的基础上，RTAI开发团队提出了LXRT——用户态下硬实时控制的方案，LXRT利用宏RTAI_PROTO为用户态下的进程提供了调用内核函数接口的捷径，但使用该方案时仍然需要谨慎进行系统调用，因为系统调用有可能在内核中阻塞。由于进程的执行上下文会在用户态和内核态之间进行切换，因此相对于内核态方案可能会有几微秒的延　时[5]，但LXRT提供的实时控制仍能够达到较高的精度。其主要实现机制是将每一个实时任务与一个实时伺服进程配对，实时任务可以先在用户态下的普通进程内实现，然后在需要硬实时控制时将其转入实时伺服进程进行实时调度，处理完毕再将控制权交还给普通进程。LXRT的主要优点是能够利用几乎所有的Linux系统调用，且无须进行内核模块编程，可以很方便地进行开发。

4 实时和非实时任务管理的实现

4.1 实时和非实时任务的划分

为了对系统的计算资源进行合理分配，并保证对实时任务的快速响应，首先将实时计算模块的所有任务严格划分为实时任务和非实时任务，任务都以 Linux 下普通线程的方式实现，如图 3 所示。

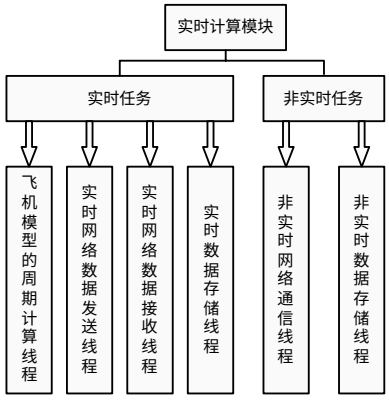

图 3　任务划分

4.2 任务的具体实现

Linux 下线程的开销非常小，适于采用单进程/多线程模式，为了使程序结构清晰化，也为了更方便地进行硬实时控制，每个任务都用一个线程来实现。各个任务的具体实现方式如下：

(1)飞机模型计算线程。负责以 5 ms为周期(默认周期)执行模型计算程序，并以查询标志变量的方式接收控制中心的控制命令，时间步长的误差大小直接影响计算结果的准确性，所以，此线程对实时性的要求非常高。创建的过程中将该线程的优先级设为最高实时优先级 1，采用Linux提供的实时调度算法SCHED_FIFO[6]，并利用LXRT提供的API在线程的初始化阶段创建一个实时伺服进程，该伺服进程仍然采用最高实时优先级 1 和SCHED_FIFO调度算法，不同于普通进程的是它处于RTAI域内。初始化完毕就调用mlockall(MCL_CURRENT | MCL_FUTURE)将线程锁入内存并禁止内存扩展，然后调用LXRT提供的rt_make_hard_real_time()使实时伺服进程替代当前线程获得RTAI域的调度，由于RTAI域具有系统中断的裁决权，因此可以保证该线程在周期为 5 ms的硬件中断到来时得到快速响应执行。

(2)实时网络数据发送线程。负责将飞机模型计算线程的计算结果通过网络传输至运动平台和操纵系统，对其进行硬实时控制的目的在于从软件的角度保证各子系统在各自的运行周期内得到响应(硬件部分的快速响应能力依赖于硬件的特性)。该线程的实时优先级为 2，低于飞机模型计算线程，硬实时控制的方法和过程同上。

(3)实时网络数据接收线程。负责接收运动平台和操纵系统的返回参数，操纵系统的返回参数将用于模型计算，对该线程进行硬实时控制的目的在于尽量减小操纵系统对飞机模型的作用延时。该线程的实时优先级为 2，硬实时控制的方法和过程同上。

(4)实时数据存储线程。负责将飞行方程的解算结果写入共享内存区域，对其进行硬实时控制的目的在于准确记录每个仿真周期的仿真结果，防止丢帧。该线程的实时优先级为 3，硬实时控制的方法和过程同上。

2 个非实时线程主要通过网络接收控制中心的命令，向视景系统、控制中心模块和图形监控模块发送部分仿真结果参数，并将共享内存区域的仿真结果数据转储至文件内。

4.3 任务的管理和同步

实时线程之间采用 LXRT 提供的函数 rt_task_suspend()

和 rt_task_resume()进行同步，rt_task_suspend()负责每个周期对所在线程进行一次阻塞(不包括飞机模型计算线程)，而 rt_task_resume()负责唤醒下一个要执行的线程。利用此机制进行任务切换的平均时间约为 50 ns。

飞机模型计算线程是主线程，也是每个周期最先执行的线程，其运行周期由函数 rt_task_make_periodic()设置，设置完成后系统每隔 5 ms 产生一次外部中断，RTAI 域接收到此中断后立刻调度该线程投入运行，每次执行至 rt_task_wait_period()语句时就将自己挂起，等待下一个周期来临时再次运行，每次挂起前它都会调用 rt_task_resume()唤醒下一个要执行的线程(创建后立刻阻塞)。每个周期内所有实时线程循环运行一次消耗的时间小于 5 ms，剩余的时间基本被非实时线程占用。

非实时网络通信线程的运行周期为 10 ms，非实时数据存储线程的运行周期为 50 ms，这 2 个线程通过等待最后一个实时线程发送的信号获得运行，在运行过程中随时可能被飞机模型计算线程抢占。所有线程的同步模型如图 4 所示。

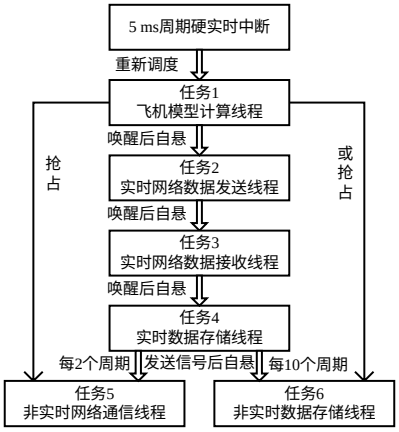

图 4　线程同步模型

5　系统实时性评测

系统搭建完成后，为了验证 RTAI/LXRT 的硬实时控制效果，对实时管理系统在期望仿真周期分别为 1 ms, 5 ms, 10 ms 的情况下进行实际仿真周期的测量，整体运行 1 h，取得的最大单步误差分别为 13 μs, 18 μs, 24 μs，该测试结果略微高于以 VxWorks 为平台的飞行仿真实时管理系统，但完全可以满足需求。测量结果如表 1 所示。

表 1　实时性测量结果

期望周期/ms	平均周期/ms	平均单步误差/μs	最大单步误差/μs
1	0.999 999 917	4	13
5	4.999 998 750	7	18
10	9.999 954 220	9	24

6　结束语

本文在充分了解国内外 Linux 实时改造成果的前提下，利用 RTAI 在用户态下的硬实时控制方案 LXRT 和 Linux 下丰富的自由软件实现了功能强大且实时性良好的飞行仿真实时管理系统，相比同类产品具有成本低、开放性好等特点，为飞行模拟器的实时应用提出了新的解决方案，也证明了 Linux 和 RTAI 的开源组合完全可以满足大型工程项目的实时应用需求，具有较好的推广价值。

参考文献

[1] 彭　华，沈为群，宋子善. 一种基于 VxWorks 的飞行仿真实时管理系统[J]. 系统仿真学报, 2003, 15(7): 966-968.

[2] 吴　超, 沈为群, 潘顺良, 等. 某直升机工程飞行模拟器控制中心的研究与实现[J]. 计算机仿真, 2006, 23(9): 294-297.

[3] Dozio L, Mantegazza P. Real Time Distributed Control Systems Using RTAI[J]. Object-oriented Real-time Distributed Computing, 2003, 14(16): 11-18.

[4] Bucher R, Balemi S. Scilab/Scicos and Linux RTAI: A Unified Approach[C]//Proceedings of the 2005 IEEE Conference on Control Applications. Toronto, Canada: [s. n.], 2005: 1121-1125.

[5] Magma. RTAI API Documentation[Z]. (2006-06-14). https://www.rtai.org/ RTAILAB/RTAI-Lab-tutorial.pdf.

[6] Barabanov M, Yodaiken V. Real-time Linux[J]. Linux Journal, 1997, 34(2): 1-7.

（上接第 255 页）

虽然图 4 中的第 5 层近似信号($a5$)也可以看出信号发生突变的时刻，但对比图 3 和图 4 可以发现以下 2 点不同：

(1)图 3 中的第 5 层近似信号(a5)其突出的部分更接近方波，而图 4 中的第 5 层近似信号(a5)还是原来的正弦波，可见提升小波对信号奇异点的检测更为明确。

(2)图 3 中的第 5 层近似信号(a5)在 t=65 ms 时刻检测到信号奇异点，而图 4 中的第 5 层近似信号($a5$)在 t=68 ms 时刻检测到信号奇异点，可见提升小波对信号奇异点的检测更迅速。

以上 2 点进一步说明了提升小波非常适合电力系统准确和快速的实时操作要求。

5　结束语

根据以上的理论分析和实验仿真结果得到以下结论：

(1)提升小波变换可以准确而有效地检测输电线路短路故障信号奇异点，并为下一步故障测距提供有利依据。

(2)与傅里叶变换相比，提升小波变换具有时频局部化性质，能够有效地分析非平稳信号，因此，非常适合输电线路发生短路故障后暂态信号的检测。

(3)与第 1 代小波变换相比，提升小波比 db5 小波快 3 ms 检测到输电线路短路故障信号的奇异点，可见提升小波变换计算速度更快，因此，更适合于电力系统的快速实时操作。

参考文献

[1] 覃　剑, 陈祥训, 郑健超, 等. 利用小波变换的双端行波测距新方法[J]. 中国电机工程学报, 2000, 20(8): 6-10.

[2] 李弼程, 罗建书. 小波分析及其应用[M]. 北京: 电子工业出版社, 2003-05.

[3] Jung C K, Kim K H, Lee J B, et al. Wavelet and Neuro-fuzzy Based Fault Location for Combined Transmission Systems[J]. Electrical Power and Energy Systems, 2007, 29(6): 445-454.

[4] Sweldens W. The Lifting Scheme: A Construction of Second Generation Wavelet[J]. SIAM J. Math. Anal., 1998, 29(2): 511-546.

[5] 马永翔, 王世荣, 于　群. 电力系统继电保护[M]. 北京: 中国林业出版社, 2006-08.