采用插桩技术的可视化虚拟实验室研究与实现①

胡志刚 李 勇 (中南大学 信息科学与工程学院 湖南 长沙 410083)

摘 要: 操作系统原理课程普遍存在实践教学设备的缺乏和低效，为此，提出一种采用插桩技术的可视化操作系统虚拟实验室 VOSLS(A Visual Operating System Virtual Lab Using Stub Method)。采用插桩方案调试用于实验的操作系统内核，将复杂的 GDB 调试协议简化为简单的插桩通信协议；以软盘或硬盘映像文件为媒介，与运行于虚拟机上的被实验操作系统通信，获取其运行信息，并采用可视化图形引擎技术将获得的运行信息以图形的方式呈现给用户。实际应用表明，借助该虚拟实验室，可有效提高实验教学效果。

关键词: 操作系统实验; 虚拟机; 虚拟实验室; Linux 内核; 插桩技术

Research and Realization of a Visual Virtual Lab Using Stub Method

HU Zhi-Gang, LI Yong

(School of Information Science and Technology, Central South University, Changsha 410083, China)

Abstract: In order to address the lack and inefficiency of teaching tools of the operating system principles course, a VOSLS was proposed. It used Stub method to debug the experimental operating system kernel, and reduce the complex GDB debugging protocols to simple Stub communication protocols. It communicated with the experimental operating system running on a Virtual Machine through the floppy or hard disk image files. Then the running operating system' s information was graphically and vividly delivered to the users by using visualization graphics engine technology. Practice has proved that this system can significantly improve the experimental teaching effect of the operating system principles course.

Keywords: operating system experiment; virtual machine; virtual lab; linux kernel; stub method

1 引言

操作系统原理既是大学计算机专业的核心课程，也是基础软件领域内的重点研究课题。由于实验条件限制及操作系统原理自身的特点，在大多数操作系统教学过程中，学生通常以理论化、静态化的方式学习，很难开展有效的实践活动[1]。目前，现有的大部分操作系统教学工具都只是对一些操作系统原理和算法做一些简单的动态演示，如 RCOSjava 等；少数能提供真实的实验环境，但实验步骤过于繁琐，如 Nachos 等[2]。针对已有不足，本文研究了采用插桩技术的可视化操作系统虚拟实验室，借助该环境，学生不仅可以构建、调试自己的操作系统，而且可以实现操作系统内部运行过程的“可视化”。其主要特点有：

(1)采用通用虚拟机技术，使得实验环境接近真实的硬件；

(2)采用裁剪后的 Linux 内核源码；

(3)简单易用的插桩调试操作系统内核方案；

(4)操作系统内部运行状态可视化呈现；

(5)基于 WEB 浏览器的实验平台。

2 基本概念

2.1 虚拟实验室技术

虚拟实验室(Virtual Lab)概念，最早于 1989 年由美国的 WilliamWolf 教授提出。美国国家研究委员会的定义是：虚拟实验室是一个无墙的中心。研究人员能在其中从事科学研究和工程设计，不必顾及地理

① 收稿时间:2010-03-04;收到修改稿时间:2010-06-21

位置的限制，实现同行间、同事间的互动；共享仪器、设备、数据、计算资源以及数字图书馆的信息。

2.2 虚拟机技术

虚拟机 VM (Virtual Machine)被定义为一台真正机器的高效率的和隔离的重复。虚拟机环境通过虚拟机监控器 VMM (Virtual Machine Monitor)创建，也称为“操作系统中的操作系统”。虚拟机监控器在单台真正的机器上创建一个或多个虚拟机。每个虚拟机为应用程序或者“客户操作系统”提供执行环境。虚拟机典型的用途包括：开发和测试新操作系统，实验领域[3]。现实世界中存在各类虚拟机 VM (Virtual Machine)，如：Bochs, Vmware。

2.3 Linux 内核

Linux 内核主要完成与计算机硬件进行交互和调度硬件资源，实现对硬件部件的编程控制和接口操作，调度对硬件资源的访问，并为用户程序提供一个高级的执行环境和对硬件的虚拟接口。本文采用 Linux-0.11 版内核，由 5 个模块构成，分别是：进程调度模块、内存管理模块、文件系统模块、进程间通信模块和网络接口模块，整个内核源代码只有 325K，但却包括了 Linux 的核心模块，而其它版本的内核不具备该特点[4]。

3 VOSLS研究与实现

3.1 整体框架

VOSLS 的硬件架构遵循典型的 WEB 三层架构，如图 1 所示。数据服务器、操作系统虚拟实验室服务器群、WEB 服务器和防火墙均部署在同一个局域网，它们一起为通过 Internet 连接 VOSLS 的不同用户提供相应的服务[5]。

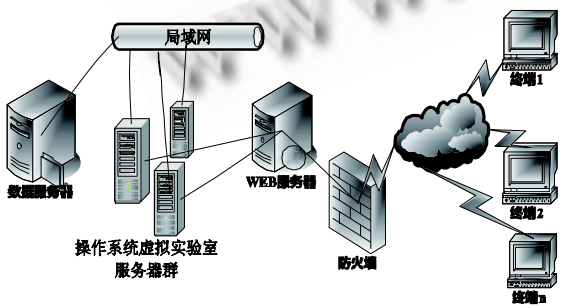

图 1　VOSLS 硬件架构

用户通过运行在终端上的 WEB 浏览器向 WEB 服务器发送服务请求。服务器的服务过程是：每新增加一个用户，则为该用户建立一个用户服务进程，由该服务进程启动一台虚拟机提供服务。相当于每一个用户都有一台无限接近实际计算机的虚拟机为其提供实验环境。

VOSLS 的主要组成模块见图 2。

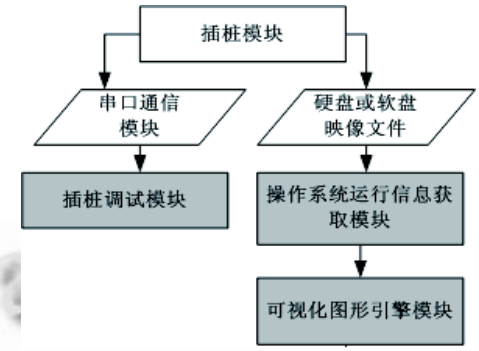

图 2　VOSLS 组成模块示意图

图 2 中，插桩模块以 Linux 内核态常驻内存。插桩调试模块通过串口通信模块从插桩模块获取所需调试信息[6]。插桩模块也可将操作系统运行信息写入虚拟机的硬盘或软盘映像文件中，操作系统运行信息获取模块则从硬盘或软盘映像文件中读取该信息，然后可视化图形引擎模块将这些信息以图形方式呈现给用户。灰色模块为 VOSLS 主要功能模块，下文将详细介绍。

3.2 VOSLS 插桩调试模块

操作系统内核调试的调试目标是操作系统的内核，对于内核调试而言，将调试目标中断到调试器意味着将中断操作系统内核，以接受调试器的分析和检查。由于内核负责整个系统的调度和执行，一旦被停止，系统中的所有进程和线程都将停止运行。为此，才需要在另一个系统中运行，并需要找到一种方案保持调试器与这个静止的内核通信和联系。

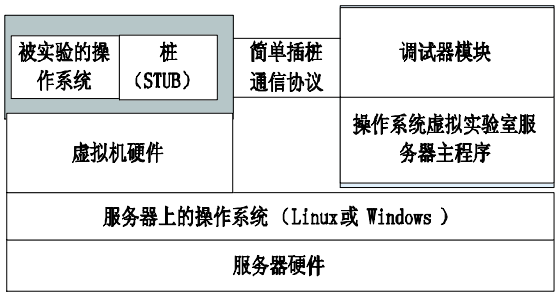

图 3　插桩调试内核

VOSLS 采用的方案是在系统内核中加入调试支持，当需要中断到调试器中时，只保留这部分支持调试的代码继续运行，内核的其它部分则被停止。该方法需要调试器运行在另一个系统中，支持调试的代码和调试器通过串口通信交换信息。

VOSLS 调试环境的构成如图 3 所示。

VOSLS 的调试环境包括 3 个部分：① 服务器既是运行调试器的平台，也是运行虚拟机的平台，相当于宿主机；② 虚拟机，被实验操作系统机运行在此虚拟机上，相当于目标机；③调试器模块和桩，调试器模块是操作系统虚拟实验室服务器主程序的一部分，桩(STUB)是被实验操作系统的内嵌模块，两者之间按简单插桩通信协议通过串口通信 。简单插桩通信协议的数据报格式如图 4。

图 4 简单插桩通信协议的数据报格式

BF、EF 分别是数据报开始和结束标志，占一个字节；LEN 是整个报文长度，占两个字节；TYPE 是报文类型，占一个字节；CS 是校验和；DATA 是实际数据，占据的宽度依照 TYPE 而定。VOSLS 插桩调试模块通过此插桩通信协议实现对被实验操作系统的调试。

3.3 VOSLS 操作系统运行信息获取模块

该模块使用基于软盘或硬盘映像文件的数据交互技术与插桩模块通信。调试操作系统内核的数据交换量较小且交换频率低，故通过串口进行数据传递可行，但如果要记录操作系统内部的运行状态，如内存分配，串口通信便不堪重负，基于软盘或硬盘映像文件的数据交互方法则能满足数据交换量大且交换频率高的要求。

图 5 FAT12 软盘映像文件格式

本文以基于软盘映像文件的交互技术为例。图 5 是 FAT12 文件格式软盘的存储格式，当被调试操作系统运行时，桩(STUB)将系统运行信息写入文件 system.info，服务器主程序通过直接读取 system.info 获得被实验操作系统的运行信息。从 FATA12 格式的软盘映像文件(亦称虚拟软盘)中读取 system.info 文件的过程描述如下：首先在根目录区中找到文件名为 system、文件后缀名为 info 的根目录项，然后根据该根目录项的索引值找到 FAT1 或 FAT2 中的 FAT 项，每个 FAT 项本身的序号对应于数据区中的一个扇区(sector)，而 FAT 项的值是下一个 FAT 项的序号值，如该值是 0xfff，则指示文件结束，依次类推便可找到包含 system.info 文件数据块的所有扇 区[7]。

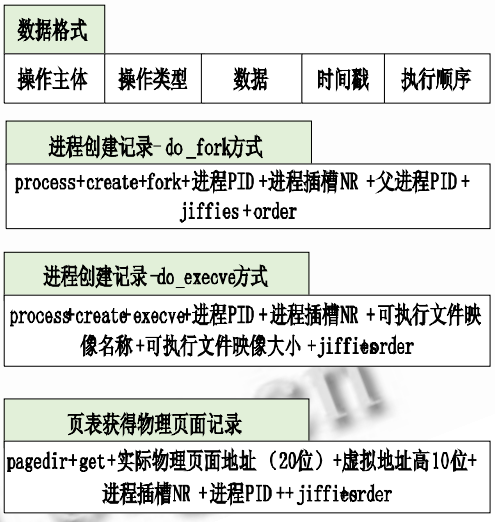

图 6 部分操作系统运行信息记录格式

system.info 文件中包含的数据具有固定的数据格式，用来记录操作系统的运行信息。见图 6。

本文以进程创建记录-do_fork 的记录格式为例进行说明。操作主体为 process 字段，操作类型为 create fork 字段，时间戳为 jiffies 字段，执行顺序为 order 字段，数据部分为进程 PID+进程插槽 NR+父进程 PID。本例中系统获得的部分操作系统运行信息如表 1 所示。

表 1 的第一列到第五列依次对应操作系统运行信息记录的操作主体、操作类型、数据、时间戳、执行顺序。表 1 中的数据使用十六进制。将图 6 和表 1 中数据进行对比可知，表 1 每行数据都记录了操作系统

在特定时刻的行为。获取上述部分信息的主要代码片段见图 7。

表 1　截取的部分操作系统运行信息

pagedir	creak fork	ffa027 2f 2 2	26e	da
process	create exit	2	271	db
page	free	ffb000	273	dc
process	slot free	ffb000 2	275	dd
page	get	ffb000	278	de
page	get	ffa007	27a	df
pagedir	get	ffa007 20 4 2	27c	e0

```
struct process_create{//进程创建信息的数据结构
        int type;
        unsigned long pid;
        unsigned long slot_nr;
        unsigned long father_pid;
        char *exec_name;
        unsigned long exec_size;
};
struct process_page_slot{//进程分配的内存页的数据结构
        int type;
        unsigned long phyical_address_20;
        unsigned long pid;
};
struct process_create my_pc;
struct process_page_slot my_pps;
 //宏定义，控制是否需要获取该部分操作系统运行信息
#ifdef _GET_INFO_PROCESS_
    set_process_create(&my_pc, p->pid,nr, p->father, 0,0, 0);
    print_process_create(&my_pc);
    set_process_page_slot(&my_pps,(unsigned long)p, p->pid, 0);
    print_process_page_slot(&my_pps);
#endif
```

图 7　获得操作系统运行信息的部分代码

该段代码内嵌在 Linux0.11 内核代码的 fork.c 代码文件中，用以获取进程创建信息以及为该进程分配的内存页(该内存页用以存储进程本身数据结构)信息，set_process_create() 函数获取进程创建信息，set_process_page_slot()函数获取为该进程分配的内存页信息。

3.4 VOSLS 可视化图形引擎模块及程序截图

借助可视化图形引擎模块，并根据从操作系统运行信息获取模块获得的被实验操作系统各个部分的运行信息，可实现被实验操作系统运行过程的动态表达。该图形引擎采用 JAVA 语言实现，具有良好的跨平台性。图 8 是该图形引擎的部分类图。

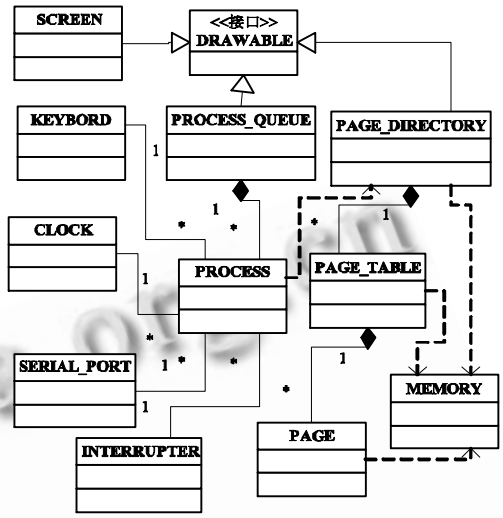

图 8　图形引擎的部分类图

PAGE_DIRECTOR 代表页目录对象，PAGE_TABLE 代表页表对象，PAGE 代表基本页对象，其它的类由英文名可知其含义。DRAWABLE 是绘图接口，继承了该接口的类可以在本文介绍的系统中用图形的方式表达出来。这些类之间的基本关系在图 8 的 UML 图中予以简单表述。同时在图 10 中给出程序截图。

图 10　VOSLS 内存分配实验模块截图

3.5 VOSLS 实验流程

(1) 用户选定一个实验项目，按实验要求决定是否需要提交实验代码。

(2) VOSLS 自动完成实验的准备工作。即根据实验要求，修改配置文件，调用编译工具重新编译操作系统内核，并将编译好的操作系统内核映像写入虚拟软盘，启动虚拟机运行该操作系统内核。

(3) 进入实验的主要阶段。用户通过 WEB 浏览器给 VOSLS 发送命令，可以是调试命令，用以查看操作系统的各类信息(如特定变量的值)，也可以是可视化命令，图形引擎根据命令以图形化的方式显示信息(如进程队列信息、内存分页信息)。

(4) VOSLS 对获得的信息进行处理之后，启用内置图形引擎，按时间顺序将操作系统内部运行过程以图形的方式呈现给用户。

4 结束语

实验环节能使学习操作系统原理的学生对操作系统的认识由抽象变为具体。VOSLS 由于采用了插桩方案调试操作系统内核，可以让学生如同调试普通应用程序一样调试操系统内核。另外 VOSLS 使用基于软盘或硬盘映像文件的数据交互技术，按照本文介绍的信息记录格式来记录操作系统运行数据，再借助可视化图形引擎，可将被实验操作系统的运行情况以图形的方式呈现给用户。

虽然目前 VOSLS 所设计提供的实验项目还不多，仅包括进程调度、内存分配等，但从实验方式和其所搭建起来的这个实验平台来说，已为操作系统辅助实验教学工具的研发提供了一个很好的参考。

参考文献

1 胡志刚,徐益海.基于虚拟机的操作系统教学工具的架构设计.企业技术开发, 2007,26(7):27－29.

2 Procter SJ, Anderson TE. The Nachos instructional operating system. Proceedings of the Winter 1993 USENIX Conference, 1993:481－489.

3 Li Peng,Lunsford Philip,Mohammed Tijjani,et al. 114th Annual ASEE Conference and Exposition, 2007,7:12－14.

4 赵炯.Linux 内核完全剖析.北京:机械工业出版社, 2008:141－185.

5 陆炜妮,庞竣.基于 Internet 的计算机网络虚拟实验环境架构.计算机工程, 2007,33(13):284－285.

6 Hongwei Li, Fangsheng Wu1, Yaping Xu1, et al. Research of "stub" remote debugging technique. 24th International Conference on Computer Science and Education, 2009:990－994.

7 于渊.一个操作系统的实现.北京:机械工业出版社, 2009:120－166.