A Novel Teaching Solution for "Operating Systems"

Tunhua Wu
School of Information and Engineering
Wenzhou Medical College
Wenzhou, China, 18957768018
wth@wzmc.edu.cn

Baogang Bai
School of Information and Engineering
Wenzhou Medical College
Wenzhou, China
bbg@wzmc.edu.cn

Ping Wang
Department of Environmental Science
Wenzhou Medical College
Wenzhou, China, 0577-85696377
fruitful@xmu.edu.cn

***Abstract*—Traditional teaching process of "Operating Systems" course has many defects, such as divorce between theory and practice, outdated experimental platform, which result in low learning interest and effect on the course. A novel teaching solution for "Operating Systems" course was proposed in this paper. The solution is divided into four layers. First, construct a novel experimental platform based on Ubuntu Linux and Qt Creator. Students can easily design any program with GUI interface under Linux system by applying Qt Library. Second, simulate some critical algorithms in the textbooks. Third, trace Linux kernel code utilizing Qemu virtual machine and Eclipse. By doing so, students may insight about the structure of Linux kernel. Final, modify part of the kernel code and rebuild the kernel. Compared with traditional teaching model, this solution not only provide attractive experimental platform, but also give the chance for looking into Linux kernel. Teaching practice show that this solution can dramatically improve students' learning interest and practical ability.**

Keywords- Operating System; education reform; Linux; kernel; Qt Library

I. INTRODUCTION

"Operating Systems" is not only a core course of computer science, but also a compulsory of some computer-related professionals in general, such as e-commerce, information management and information systems [1]. It is an important basis for learning computer principle and the underlying programming technology. However, there are many defects in the teaching process of "Operating Systems" course. Traditional teaching solutions emphasize too much on the mastery of concepts and algorithms, and the description of system is usually too simplistic and abstract. By traditional teaching solutions, students usually feel that the course is boring and the knowledge is far from real kernel, and then students may lose interest in the course [2, 3]. By referencing advanced teaching experiences in the world, we found that the key point of this course is practice [4, 5]. Hence, analysis of Linux kernel was integrated into our teaching solution. The purpose is to lead students to grasp principles of operating systems and the ability for developing and maintaining the kernel. In addition, an attractive experimental platform was constructed for improving students' learning interest.

II. ANALYSIS OF CURRENT TEACHING SOLUTIONS FOR "OPERATING SYSTEMS"

By classroom observation, watching videos and reading various teaching materials, we found that there are two main defects in the teaching process of "Operating Systems" course. And the defects are described as follows.

A. Divorce Between Theory and Practice

The ultimate goal of studying "Operating Systems" course is to learn how to design or develop a real operating system [6]. Students should at least be able to participate in maintaining or updating some open source operating system, such as Linux and Minix [7]. To achieve the goal, student should be familiar with the principles and code structure of the system kernel. However, in traditional teaching model, usually only principles are introduced in detail, while the code structure is rarely mentioned (even in some classical textbooks). As a result, even the students had mastered all the concepts and critical algorithms, they could not be directly involved into operating system design or maintenance. So, the divorce between theory and practice often leads the students to feel that the course is "on paper", "not practical" or even "useless".

B. Outdated Experimental Platform

Most "Operating Systems" textbooks take Unix system as prototype to introduce the principles [8, 9]. But Unix lacks GUI (Graphical User Interface) desktop which was universally applied to Linux and Windows systems, and the Shell interface is not friendly and attractive to current students. In addition, as Unix had been turned to closed-source system, it is impossible to modify and rebuild the kernel code. So, Unix is not suitable to continue as a teaching system.

The defects mentioned above often cause much confusion to students. On one hand, they know the importance to learn the course. But on the other hand, they have low interest in studying the teaching materials, and their unique learning goal is to pass the exam. In order to lead students to really enjoy this course, the teaching content should be adjusted on a large scale. It requires teacher to explain important concepts and algorithms while introducing the structure and critical data structure of various kernel modules in limited time. By doing so, students can directly participate into developing or maintaining certain open source operating system, and the programming skills may also be improved. But it is hard to find the intersection between theoretical knowledge and kernel modules. In addition, how to introduce the kernel module in a simple way is very difficult. Therefore, the selected kernel code should be representative enough for demonstrating the principle, and size of the code should be as concise as possible.

In sum, the current teaching mode of "Operating Systems" course should be reformed. It is very important to combine theoretical knowledge with kernel modules. In this way, students' learning motivation and practical ability can be promoted, and it may be helpful for satisfying the urgent requirement of competent employees from companies.

III. Research Goal and Content of Education Reform on "Operating Systems"

The research goal of this education reform is to propose a novel teaching solution for "Operating Systems" course. The solution should combine the theoretical knowledge with real kernel modules. By applying the solution, students' learning interest and practical ability may be improved. And it may stimulate students' enthusiasm on developing or maintaining kernel.

Research content can be divided into four parts which were described as follows.

A. Reform on Algorithm Description

Not only portrait the algorithm using pseudo-code, but also to present a more realistic and attractive pattern to students. So students will understand how the algorithm was implemented and applied in actual operating system.

B. Reform on Concepts Description

Let the students to be familiar with current hot technologies which should be described in detail, while the outdated or rarely used technologies should be neglected.

C. Reform on Experimental Platform

Provide simulation environment for classical algorithms, and to create the conditions for tracing or modifying kernel code.

D. Reform on Teaching Method

Promote the harmonious interactions between teacher and students, and to obtain the students' learning status and timely adjust teaching content and progress.

IV. Novel Teaching Solution for "Operating Systems"

Our teaching solution is divided into four layers in accordance with the principle of gradual and orderly progress. The layers, from low to high, are: attractive experimental platform, algorithms simulation, kernel tracing and kernel modification. The experimental system is Ubuntu 11.10 with latest Linux kernel 3.0.3. Integrated developing tool is Qt creator (based on Qt Library, offer rapid and visual programming functions). And the tools for tracing, debugging, rebuilding kernel are Qemu (a Linux-oriented virtual machine) and Eclipse CDT (a plugin for C++ programming under Eclipse).

A. Attractive Experimental Platform

Ubuntu is the most mainstream Linux distribution having Windows-like friendly interface and rich hardware and software support. As most "Operating Systems" textbooks take Unix/Linux system as prototype to introduce the principles, using Ubuntu as an experimental system is conducive to linking theory with practice. Qt Creator is a C++ developing environment based on Qt Library (open source) under Linux system. And it has the advantages of high speed and less resource. It is quite easy to design a program with GUI interface utilizing Qt Creator. In order to help students master Qt Creator, a MySQL based database demo program for hospital management was elaborately designed and then introduced to students. The attempt of taking Ubuntu and Qt Creator as experimental platform not only greatly improves students' programming efficiency but also makes the originally dull algorithm simulation becomes vivid and clear. In addition, the novel experimental platform enhances the relationship between "Operating Systems" course and some other important courses, such as "Data Structure", "Object-Oriented Programming" and "Database Principles".

B. Algorithms Simulation

This layer is the focus of the traditional teaching, and its task is to simulate important algorithms. Take LRU page replacement algorithm for example. Suppose the number of access to process pages is n, and the number of frames allocated to the process is m. Firstly, design a stack with m units on the form, and each unit stores the number of latest accessed page. Secondly, if a page is to be accessed and the page number is in the stack, just pull the page number out of stack and push it back to the stack. Otherwise, remove the bottom number and push the new page number into stack. The whole process was demonstrated on GUI interface, so students can grasp the algorithm intuitively.

C. Kernel Tracing

In order to help students deeply understand kernel code structure, Eclipse CDT and Qemu were used to realize remote tracing and debugging. First, execute "Make" command on kernel code to generate kernel configure file (*.config). Then, construct a project based on kernel code in Eclipse, and generate a kernel image "bzImage" by executing "Make" command too. Next, put the kernel image into Qemu and start the kernel in debug mode. Namely, the kernel cannot continue to run until it had received the instruction from remote debugger (i.e. Eclipse). At the last, students can trace the executing flow by setting breakpoints.

D. Kernel Modification

Modification and rebuild of kernel can lead students to participate in developing or maintaining kernel. Similar to kernel tracing, a project based on kernel code was constructed firstly. Then, the kernel code was modified and rebuilt. Finally, the new kernel image was moved to "/boot", and the settings about kernel image in “/etc/lilo.conf” was modified. Restart the system and the results will appear.

V. CONCLUSIONS

A novel teaching solution for “Operating Systems” was proposed in this paper. Main goal of the solution is to promote students’ learning interest and practical ability, and it has three innovations described as follows. First, the combination between theoretical knowledge and actual kernel modules can not only greatly promote students’ learning interest but also improve their practical ability. In this way, students can directly participate in developing and maintaining the kernel module, and then it may promote the development of open source systems. Second, the friendly and attractive experimental platform can not only provide simulation environment for classical algorithms but also create the conditions for tracing or modifying kernel code. Third, the supplementation of latest course-related technologies makes teaching content keep pace with the times. Teaching practice shows prove the effectiveness of this solution. Nevertheless, for the teaching time is rather limited, how to further refine course content and enhance teaching efficiency is the next exploring target.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of China (No.11005081, No. 21207103), Natural Science Foundation of Zhejiang Province (No.Y1110322, No. 2012C31025), Scientific Research Project of Zhejiang Education Department (No.Y201016244), Scientific Research Project of Wenzhou (No.G20110004, S20110011) and Special Education Reform Project of Zhejiang Undergraduate Computer Education Steering Committee (No.2011-4).

REFERENCES

[1] T. H. Wu, “Teaching Practice on ‘Synchronization and Mutual Exclusion’ Problem,” The Science Education Article Collects, vol. 12, pp. 36-37, December 2010.

[2] T. H. Wu, “Teaching Reform of Operating Systems Course in E-Commerce Professional,” Market Modernization Magazine, vol. 17, pp. 170-170, June 2011.

[3] L. N. Li, “Teaching Reform of Operating Systems Course,” Information Systems Engineering, vol. 9, pp. 126-126, September 2010.

[4] Z. G. Sun, P. H. Tang, Z. J. Huang and Z. G. Sun, “Exploration of the Practical Educational Schemes in Operating System,” Computer Education, vol.8, pp. 146-148, April 2010.

[5] Q. L. Zhang and B. Han, “Exploration on Reform of Experimental Teaching of the Operating System Course,” Computer Education, vol. 7, pp. 91-93, April 2010.

[6] Y. Y. Xu, “Teaching Reform of Windows and Linux Operating Systems in Higher Vocational Education Based on Project-driven,” Computer Education, vol. 18, pp. 25-27, September 2010.

[7] B. Xu and J. Dai, “Experiment Teaching Reform on Operating System Principle,” Computer Education, vol. 17, pp. 85-87, September 2011.

[8] Y. Ying, “Practical Teaching Research Based on Linux for Application,” Computer Education, vol. 9, pp. 139-142, May 2010.

[9] X. L. Wang and S. P. Zhao, “Exploration on Experiment Teaching Method of Operating System Course Based on Kernel Analysis,” Computer Era, vol. 4, pp. 48-49, April 2011.