Journal of Contemporary Educational Research, 2023, Volume 7, Issue 8
http://ojs.bbwpublisher.com/index.php/JCER
ISSN Online: 2208-8474
ISSN Print: 2208-8466

Experimental Design for Computer System Platform Course

Li Zhang*, Qing Zhu, Xuejun Yu

School of Software Engineering, Faculty of Information Technology, Beijing University of Technology, Chaoyang District, Beijing 100124, China

***Corresponding author:** Li Zhang, zhangli828@bjut.edu.cn

Abstract: The computer system platform course is a comprehensive platform course formed by modifying and optimizing the content of the relevant basic courses of computer major to meet the requirement for basic computer knowledge in digital media technology major. The scope and extent of the course content make its experimental design more challenging. By considering the scope of contents, the difficulty of experiments, and the rationality of intrinsic logic, the experiments covering from operating system to computer network are designed. The relevance and continuity of the experiments are ensured, so as to achieve the goals of promoting students' understanding of key concepts, improving their hands-on ability, cultivating their interest, and guiding their self-learning within the limited class time.

Keywords: Digital media technology; Experimental design; Computer fundamentals; Intrinsic logic

Online publication: August 29, 2023

1. Introduction

Digital media technology major is an emerging interdisciplinary field focusing on technology and supplemented by art [1]. It is a major that integrates and develops disciplines related to digital media art and computer technology [2]. Designing the digital media technology experimental teaching to enhance students' practical skills and comprehensive literacy has become a key issue that educators need to break through in the practical exploration [3].

Computer technology is the foundation of the digital media technology major, and these technologies are rooted in the corresponding computer theory, thus the corresponding computer theory and practice courses are needed for the digital media technology major. However, the basic computer knowledge requirements of this major are different from those of computer science major, and it is not feasible to copy the curriculum of computer science major. Most universities and colleges have set up computer-based professional foundation courses for digital media technology major, while the course categories, credit hour arrangements, and content organization are very different since it is an emerging major [4]. Overall, the demand for fewer hours and wider content coverage is encountered when the basic computer courses for digital media technology major are considered [5]. A significant problem is the lack of integrated consideration for the design of the experimental system in digital media technology courses [6]. It may result in a lack of knowledge or a disjointed logical connection within the course content, hence it cannot effectively support the core knowledge of the professional direction if computer science courses are simply utilized or modified for digital media technology major [7].

Computer system platform course is a basic platform course tailored for the digital media technology major. The course is based on the professional training objectives, formed by sorting out the needs of basic computer knowledge in the major, modifying and optimizing the content of relevant basic courses in computer science. The course content covers computer composition, operating system principles, and computer network fundamentals. The operating system principles and computer network fundamentals are the key core contents of the course. Experimental sections have been set up in order to help students to understand the key concepts involved in the course content from a perceptual perspective, while also training their practical skills, and laying a necessary foundation for subsequent other professional courses.

Due to the wide coverage and scope of the course content, the design of experimental content is more challenging. It should be completed within a limited class hour, while also play a role in coordinating theoretical knowledge, promoting the understanding and mastery of concepts, and improving practical skills. The content should also be connected in an orderly manner, with a reasonable intrinsic logic, and can be organically combined to form a course's experimental system. Through a detailed analysis of course objectives, content, and professional needs, experimental content of the course is designed under the consideration of the following aspects. The experimental content should aim to achieve the goal of promoting students' understanding of key concepts, improve hands-on skills, cultivating students' interests, and laying a solid foundation for further self-learning within the limited experimental class hours.

2. Broad content coverage meets the professional requirement

Based on the characteristics of the digital media technology major, the computer system platform experiments cover a series of program from operating system usage, core concepts, to computer network programming, although the experimental class hours are limited.

In terms of computer fundamentals, students majoring in digital media technology first need to master the use of operating systems, and further understand the functions and behaviors of operating systems. This can help students to use digital media related software on the operating system platform, such as video editing and production. It can also help students to design program, such as game program development. Therefore, the first exercise is set up as shell programming. Through this exercise, students are familiarized with the Linux operating system interface and proficient in using the Linux operating system. At the same time, it is hoped that students will have a perceptual understanding of the concept of offline interfaces.

For digital media technology major, a large part of programming works may be implemented by scripts, therefore mastering script programming is crucial for students of this major. Shell programming exercises enable students to learn scripts writing grammatically, practice the core loop structure and branch structure, and understand the running process of script programs.

Process is the core concept of the operating system. Multi-process or multi-threaded programming is a crucial foundation for digital media applications. The theoretical concept of processes and process scheduling is too abstract for students, and the dynamic nature of process makes it more difficult for students to master. On the surface, students seem to understand the concept of process, but they easily confuse it with static programs and do not understand the meaning of dynamic and independent scheduling units. Therefore, the experimental content for creating multi-process program was set up. The student who finished multi-process program expressed that this experiment provided a deeper understanding of the methods, principles, and effects of process scheduling. Many groundbreaking problems were encountered in this experiment. However, in the process of continuously solving problems, a lot of knowledge was learned and completed this experiment was completed by consulting a lot of materials. Some students believed that this experiment was not only a test of knowledge, but also a test of self-learning ability. Other students expressed that knowledge about processes was gained, such as fork() function in process replication, different return values of parent and child processes, the relationship between parent and child processes, and that child process can also have child process, which is similar to cell division. Child process can use exec() function family to load and run new programs. In order to achieve the experimental purpose, new knowledge points were learned, such as getpid(), getppid(), sleep(), wait(), waitpid(), and main functions with formal parameters. The subject's knowledge is a part of completing the experiment, but to fully achieve the goal, one needs to search for information on their own, which can help improve self-learning. Overall, the experiment was very rewarding, in which new knowledge was gained, Linux systems were familiarized, and logical thinking and self-learning were exercised.

Students majoring in digital media technology need to have a basic understanding of computer networks because they need to design networking application in the future, such as online games, smart applications, and so on. Students should first master the concept of networking application structure, such as Client/Server (C/S) structure, Browser/Server (B/S) structure, and Peer-to-Peer (P2P) structure. The communication of the applications is implemented with the support of the transport layer. To effectively use the interfaces of the transport layer, students need to have a basic understanding of the transport layer behavior. Students need to know the basic support provided by the transport layer to the application layer, understand transport layer mechanism, and then master the functions and usage methods of the network interface. This requires students to first understand the basic concepts of network layering and protocols. To this end, the motivation for network layering and an introduction to the network reference model were designed in the course content with a focus on the implementation mechanism of the transmission control protocol (TCP). In order to help students to understand these concepts and lay the foundation for students to design networking application, an experiment to implement a simplified version of the file server and client is designed in which the concept of socket is introduced. Through this, students can understand the basic form of the transport layer interface and the means of network communication, and practice the structure of the C/S mode networking application.

3. Moderate experimental difficulty based on students' foundation

As the experiments are for the basic course of digital media technology, the depth and difficulty of the experimental content should be intermediate, and cannot be copied from that of computer science major directly. In our design, the experiments start with the simplest shell programming. Through a simple script experiment, students can learn the knowledge of computer systems. Some students said that it was their first time using a Linux system for the experiment, thus it was a process of constantly trying and exploring from the most basic functions. Becoming familiarized with the system provides a great sense of achievement. They found Linux extremely difficult to use at first due to the unfamiliarity with the system, now they have gained a basic understanding of shell programming through experiments. Through the experiment, students also gained some understanding of obtaining process information and filtering processes. This part of the experiment involved many unfamiliar knowledge, such as the use of "|" and the command "ps," "grep," etc. Students felt that they gained a deeper understanding of the content learned in previous courses about shell programming, commands, operations, conditions, and loop statements. "Through the problems encountered in the experiment, students also deepened their understanding and expanded their knowledge in practice."

Creating a multi-process program experiment is very simple in terms of content, which is just to use system calls to create a specific number of processes. Students will feel that it is relatively easy to complete and will easily understand the concept of progress. However, the completion of this experiment is not as easy as it appears. By setting a specially designed requirement for the number of creation processes, students are unable to achieve the experimental objectives at the beginning. They need to gradually understand and continuously revise the program in a seemingly simple process to achieve the experimental objectives. Through the control of the process of loop creation, students can understand the relationship between child process and parent process, the concept of child process created dynamically, the state of child process after it is created, and the process program execution. For example, when four sub-processes are required to be created, some students directly put the system calls to create the sub-processes into a four-fold loop. The program did not produce four sub-processes as students intended, but countless child processes were seemingly created. This prompted students to further reexamine the process and understand that child process copies all the parent process code, and the child process runs from the statement just created. The child process was located in a layer of the four loop, thus the number of child process generated was not the number they intended. From the running sequence of multiple processes, students can understand the scheduling and meaning of the operating system process, enable them to shift from the original idea of single process running mode to the relationship among multiple processes, and realize that processes are task-scheduling entities. Process scheduling is a task of the operating system, which cannot be controlled by the user programs. For example, students may find that the newly created child process is not executed in the order of creation when they execute the single process program in sequence, and they first thought about where the problem occurred. Through further thinking, they can realize that each child process and the parent process are equal, and the running of the process is determined by the scheduler of the operating system. When students encounter this, they will be asked the method to implement it if users want to control the running order of multiple sub-processes. In students' understanding, these processes are within the same program and can be achieved through control statements. However, the students found that the results were not under their control. Through this experiment, students can understand that the running order of processes is under the control of the operating system scheduler and is not influenced by the users. At the same time, students can understand that child process is not a part of the original process although the it is created by the parent process, it is an independent entity competing for central processing unit (CPU). Through this experiment, students can intuitively understand the meaning of process scheduling and independent CPU competition. In addition, during the experiment, students will be guided to think about how programmers should control the order in which processes run, and understand that this requires the use of process synchronization methods provided by the operating system. This can deepen their understanding of the theoretical content related to mutual exclusion and synchronization.

Networking application designing is a relatively complex process, it requires students to constantly improve in their subsequent professional courses. A simple file server design experiment is set up as the basis. The implementation of the file server requires the students to practice the main network programming interface and C/S application architecture, which is the basis of the networking application. Through the experiment, students can experience the complete architecture and the design algorithm of networking application. However, the overall logic and functions of the server are relatively simple, matching with students' current foundation and cognition.

4. Intrinsic logic design for broad content coverage

The first experiment is set up as shell programming, and the subsequent process program creation experiment is a C language programming. This design follows the principle of gradual progress. Through the previous shell programming, students are familiarized with the operating system, and based on this, they can further understand and use the programming environment and tools. Shell programming experiments are the foundation of process creation experiments. Through these two experiments, students can understand the difference between running a script program and a compiled binary program.

The simplified version of the file server and client experiment requires students to solve server scalability issues through concurrency. Server performance issues can be resolved by creating a process for each client request. The creation of a service process is based on the previous process creation experiment. The file server experiment requires students to draw a multi-process execution flowchart for the server. Through the flowchart, students' understanding of the independence of processes and the meaning of process scheduling is improved, and the knowledge points of process creation experiments are consolidated.

The design of the server involves the processing of zombie child process by signal mechanism. This provides students with a deeper understanding of inter-process communication mechanisms from a practical perspective. At the same time, in the process of experiment design and debugging, students can understand the differences between the design and debugging of networking applications and stand-alone applications, as well as the different design considerations of server and client programs. In the process of data interaction between servers and clients, the focus is to emphasize the concept of flow-based services to students. Through the implementation of specific statements, students can understand the implementation methods of reliable services and the concept of connection-based services.

Through this series of design considerations, the content of key curriculum theory was effectively supported by the seemingly simple and limited experimental content. The experimental design enhanced students' practical skills, and improved their self-learning through the experimental process.

Funding

This paper is funded by the Beijing University of Technology teaching reform project "New Engineering" background software engineering professional teachers' ability assessment and training curriculum system exploration and practice project (ER2022GLB01), and the Beijing University of Technology teaching reform project "online offline integration+segmented learning" software development comprehensive practice teaching mode exploration and research project (ER2022SJA01).

Disclosure statement

The authors declare no conflict of interest.

References

[1] Jia H, 2021, Construction of Curriculum System for Digital Media Technology Major. Technology and Education, 2021(02): 143–145.

[2] Li Q, Luo T, Wang J, et al., 2020, Exploration and Practice of Teaching Reform in Digital Media Major Under the Background of New Engineering. Packaging Engineering, 41(S1): 140–143.

[3] Zhu J, 2021, Research on Experimental Teaching Strategies for Digital Media Technology Courses in Universities. Science Enthusiasts (Education and Teaching), 2021(02): 11–12.

[4] Yang Z, 2011, Discussion on the Construction of Curriculum System for Digital Media Technology Majors. Modern Educational Technology, 2011(08): 88–91.

[5] Liao J, 2021, Exploration of Curriculum Design for Digital Media Technology Major. Digital Communication World, 2021(01): 228–229.

[6] Hu S, Zhou G, Chen J, 2022, Practice and Reflection on Digital Media Technology Simulation Experimental Teaching. Fujian Computer, 38(01): 108–111.

[7] Cao K, 2019, Exploring the Curriculum and Practical Teaching Mode of Digital Media Technology. Computer Knowledge and Technology, 2019(29): 114–115.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.