操作系统课程教学方法和内容的优化研究

邢丽莉,韩　莹,鹿玉红

(防灾科技学院信息工程学院,河北三河　065201)

[摘　要]本文分析了当前操作系统课程教学中存在的问题,剖析了操作系统课程中的底层原理、概念等知识对上层应用软件开发的重要性;强调在授课中应设置合理的教学内容,采用多种教学方法对底层原理、概念等知识进行充分挖掘,帮助学生建立知识的“连连看”,完善学生的专业知识体系,为其他计算机专业课程的学习打下基础;最后,以操作系统课程中的进程和线程两个重要概念为例,设计了教学案例,综合运用多种教学方法激发学生的学习兴趣和动力,提升他们的专业技能和竞争力。

[关键词]操作系统课程;教学方法;教学内容;教学案例;进程;线程

[作者简介]邢丽莉(1983—),女,河北唐山人,防灾科技学院信息工程学院副教授,硕士,研究方向:计算机应用。韩莹(1979—),女,河南信阳人,防灾科技学院信息工程学院副教授,硕士,研究方向:计算机应用。鹿玉红(1977—),女,河北唐山人,防灾科技学院信息工程学院副教授,硕士,研究方向:计算机应用。

[基金项目]本文系防灾科技学院教育研究与教学改革项目“操作系统课程教学方法及实验教学内容的优化研究”(项目编号:JY2024B21)。

[DOI]https://doi.org/10.62662/kxwxz0202012

[本刊网址]www.oacj.net

操作系统课程是面向计算机类专业开设的一门学科基础课,该门课程以计算机基础、数据结构等课程为前驱课程,重在讲授操作系统的实现原理,培养学生的系统设计和优化能力。通过本门课程的学习,学生能够了解如何在复杂的系统中进行资源管理和调度,从而设计出更高效、更稳定的系统架构。这对于开发大型软件、构建云计算平台等工作是非常有帮助的。其次,操作系统课程还能让学生更好地理解并发和并行计算。掌握这些技术,能让学生在开发高性能计算应用、处理大规模数据时更加得心应手,提升学生的专业技能和竞争力,为未来的职业发展打下坚实的基础。

本文从操作系统课程教学目前存在的问题出发,强调操作系统课程教学改革的重要性,并从建立该课程与其他课程之间的联系角度,强调教学方法和教学内容优化的重要性,最后以进程、线程为例,设计了教学案例,综合运用多种教学方法激发学生的学习兴趣和动力,提升他们的专业技能和竞争力。

一、操作系统课程教学中的问题

目前,在操作系统课程教学中,存在着一些问题:

虽然随着计算机技术的飞速发展,新型操作系统不断涌现,但操作系统课程的教学内容重在讲授操作系统的设计思想和实现原理,将多种操作系统的共性作为教学核心内容,不能激发学生的兴趣。

操作系统课程的内容庞杂、原理和概念抽象难懂,加之不能像其他课程一样学习后立竿见影,学生在学习过程中目标感和成就感不强,甚至会觉得学习此课程无用,学习到的原理、方法等无用武之地,导致学生学习兴趣和动力不足。

另外,操作系统课程往往与其他计算机相关课程存在割裂现象,缺乏整体性和系统性。学生在学习过程中更是缺少对知识的“连连看”,单一地学习操作系统基本原理,缺少对底层原理、概念的挖掘利用,面对不同环境中的软件设计需求,不能灵活、合理地选择软件设计方法。如何加强课程的整合与衔接,完善知识体系,建立多门课程的联系,在操作系统课程教学中通过实验项目、引导等方式提高

学生的综合应用能力和应变能力，是操作系统课程教学需要思考的问题。注重培养学生的跨学科思维和创新能力也是一个亟待解决的问题。

二、操作系统课程教学方法和内容的优化

很多高校教师致力于课程的教学改革研究，提出了融合“三创”教育的工科类实践课程研究与探索，强调通过教学重组、内容优化等方法建立多导师联合培养机制来提升人才培养质量；通过学赛同步的方式改革教学内容；将“PAD+OBE”模式、PBL教学理论、多种教学方法应用于操作系统课程教学中；分析突出工程实践能力培养在该门课程教学中的重要性。也有的高校从操作系统课程中存在的问题出发，增加了讲座分享来激发学生的学习兴趣；在操作系统教材建设中强化系统能力培养；从教学内容、教学设计、教学方法、考核评价等多方面改善教学，文献中探讨了操作系统课程知识体系建设及知识点关联性研究。

要想改善操作系统课程的教学效果，在多导师、多课程、多知识点之间建立连接，进行教学方法和内容的优化势在必行。

（一）建立操作系统课程与其他计算机专业课程之间的联系

如果将计算机专业课程体系比作武术，那么操作系统课程就好比是九阳真经，它可以修炼学生的“内功”，而软件设计就好比是各种招式和拳脚，它们是“外功”的展现。练成了九阳神功，普通拳脚也能使出极大攻击力。

操作系统课程不能像其他课程一样，学习后立竿见影。例如学了高级程序设计语言，就会编写程序，学习了操作系统，却很难运用所学的知识去编写一个操作系统程序。但它可以使学生理解计算机底层机制，掌握操作系统管理资源的方法；更好地理解和使用系统调用，编写更高效的程序以提升编程能力；学习操作系统设计并实现其中的算法和数据结构，提高解决问题的能力。这些正是在提升学生的内在素质和能力，在修炼“内功”，使学生具备更强的竞争力、适应能力和创新能力。

“内功”的修炼可以提高身体的内在力量，提高身体的反应速度和灵敏度。内在力量的提升，为“外功”的发挥提供了坚实的基础，使“外功”的施展更加得心应手，动作更为准确和迅速。只有掌握了操作系统课程中的“内功”精华，才能更好地服务于软件设计这类“外功”，使“外功”的施展更为敏捷，能够面对不同的软件开发需求和环境，快速、准确地给出合理的开发方案。

学生在软件开发“外功”的磨炼过程中也在检验“内功”的成效，不断加深对操作系统课程知识“内功”的理解和运用。只有通过软件开发“外功”的反复练习，才可以更好地将操作系统“内功”的力量发挥出来，形成内外合一的整体效果。

总之，操作系统课程不是孤立的，它需要与其他课程的知识进行“连连看”，在学习过程中进行有机结合，才能使学生的“武功”强大。

（二）丰富教学方法，优化授课内容，促进知识的“连连看”

在操作系统课程授课过程中，一方面要丰富教学方法，在授课过程中引导学生主动深层次地思考、探索、总结“内功”知识。只有修炼好“内功”，“外功”的施展才会事半功倍。如果只是孤立地学习操作系统底层知识，对原理、概念的理解也会停留在表面。没有深度的思考、探索、总结，“内功”就得不到有效的修炼，对“外功”的激活程度不够，使学生在不同环境中的“外功”比试中就不能灵活、随机应变地选择合理的“拳脚招式”来应对。例如，进程和线程是操作系统课程中的重要概念，进程与线程各有怎样的特点？又有怎样的区别？如果不引发学生的深层次思考，学生对其理解就仅仅会停留在表面，很难让其和软件开发“外功”的增强建立起联系。丰富教学方法、促进学生探索的主动性至关重要。

另一方面，要优化授课内容，通过讲授和实践扩充操作系统和其他课程的结合面，从而使“内功”和“外功”互相促进，合二为一，增强“武功”。还是以进程和线程的特点和区别为例，学习了这些知识，如何促进“外功”呢？这时就需要设置合理的实验内容来建立其与软件开发的联系，通过实验，去思考、讨论并总结在特定环境中的软件开发应选择多进程还是多线程的开发。如果仅仅为了实现软件功能急于写代码，而不去考虑软件的性能，例如运行稳定性、速度、时空开销等，这不亚于在“武功”的比试过程中，不能快速分析对手的特点而随机应变，缺少“外功”的敏感度，胡乱使出拳脚招式。对不同的对手使出一样的拳脚招式，显然是不可取的。学生在暑期实习和毕业设计中的表现更为明

显,对实际需求不能进行合理的分析,只是一味地认为只要写出代码实现软件功能来施展“外功”即可,殊不知采用不同的设计方法完成的程序质量和时空开销是完全不一样的。归根结底,学生对底层的概念和特点理解不够,对操作系统这一“内功”修炼不够。

因此,在操作系统课程的教学中,要尽可能让枯燥的原理、概念与其他计算机专业课程建立连接,运用知识的“连连看”来帮助学生真正地内化这些底层概念和原理,同时促进其他课程的学习。

三、操作系统课程教学案例设计——以进程、线程为例

在操作系统课程教学中,应让学生感知“内功”的修炼对“外功”的促进,并在“外功”的练习中,不断加深对“内功”的理解和应用,在提高学习兴趣的同时,享受“内外功”结合提高自身整体能力所带来的成就感。

进程、线程是操作系统课程中的重要概念,接下来以“进程、线程的特点”这一知识点为例,将实验课和理论课相结合,在实验课中加深对理论内容的理解和掌握,并引发学生深层次的思考,激发“外功”的提高。授课过程中不仅仅要注重授课目标和授课方法,对学生的引导也尤其重要,它使学生将学到的知识点真正内化,将学习的过程上升为“内功”的修炼,从而激发学生面对不同的软件开发时选择多进程还是多线程开发的灵敏度。如果没有进一步的引导,部分学生仅仅学习了进程和线程的特点,只是理解了这些底层概念,但很难建立它们与软件设计方法的联系,导致课程间的脱节。

(一)教学内容设计

在 Linux 环境中运行以下 3 个程序,观察分析运行结果,并分析原因。

程序 1、程序 2、程序 3 分别如图 1、图 2、图 3 所示:

```
#include<sys/types.h>
#include<unistd.h>
#include<stdio.h>
int main()
{
    pid_t pid;
    int i = 0;
    pid = fork();
    if(pid<0)
        printf("error in fork!\n");
    else if(pid == 0)
    {
       i=i+1;
       printf("I am the child process, %d\n",i);
    }
    else
    {
        i=i+1;
        printf("I am the parent process, %d\n",i);
    }
    return 0;
}
```

图 1　多进程并发执行程序示例

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
void task1(void);
void task2(void);
int sharedi=0;
int main()
{
pthread_t thrd1,thrd2;
int ret;
ret=pthread_create(&thrd1,NULL,(void *)task1,NULL);
if(ret)
{
perror("pthread_create :task1");
exit(EXIT_FAILURE);
}

ret=pthread_create(&thrd2,NULL,(void *)task2,NULL);
if(ret)
{
perror("pthread_create :task1");
exit(EXIT_FAILURE);
}
pthread_join(thrd1,NULL);
pthread_join(thrd2,NULL);
printf("sharedi=%d\n",sharedi);
}
void task1(void)
{
long i;
for(i=0;i<1000000;i++){
sharedi=sharedi+1;
}
}
void task2(void)
{
long i,tmp;
for(i=0;i<1000000;i++){
sharedi=sharedi+1;
}
}
```

图 2　多线程并发执行程序示例(没有加互斥锁)

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
void task1(void);
void task2(void);
int sharedi=0;
pthread_mutex_t mutex=PTHREAD_MUTEX_INITIALIZER;
int main()
{
pthread_t thrd1,thrd2;
int ret;
ret=pthread_create(&thrd1,NULL,(void *)task1,NULL);
if(ret)
{
perror("pthread_create :task1");
exit(EXIT_FAILURE);
}
ret=pthread_create(&thrd2,NULL,(void *)task2,NULL);
if(ret)
{
perror("pthread_create :task1");
exit(EXIT_FAILURE);
}
pthread_join(thrd1,NULL);
pthread_join(thrd2,NULL);
printf("sharedi=%d\n",sharedi);
}
void task1(void)
{
long i;
for(i=0;i<1000000;i++){
pthread_mutex_lock(&mutex);
sharedi=sharedi+1;
pthread_mutex_unlock(&mutex);
}
}
void task2(void)
{
long i,tmp;
for(i=0;i<1000000;i++){
pthread_mutex_lock(&mutex);
sharedi=sharedi+1;
pthread_mutex_unlock(&mutex);
}
}
```

图 3　多线程并发执行程序示例(加互斥锁)

(二)教学目标

真正内化进程、线程的特点和区别,从最底层概念上理解进程和线程的本质不同,进而将“内功”的学习晋升到对“外功”招式的展现上,尝试分析多进程开发与多线程开发的区别。能够针对不同的环境和需求,定制软件的开发方式,增强应变能力。

(三)教学方法

授课过程中,综合运用多种教学方法:案例法、直观演示法、讨论法、启发式教学、探究式教学、引导和比较法。

教学方法的应用及具体的教学过程如图 4 所示:

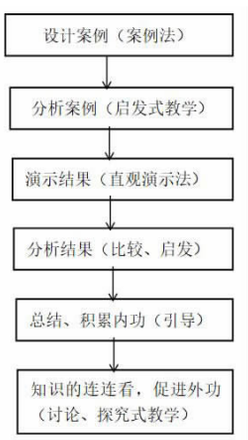

图 4　教学方法的应用

(四)教学过程

首先,使用案例法,设计能够反映理论知识的经典案例,来加深对理论课程知识的理解和掌握。

其次,结合理论课堂所讲授的知识点,对案例进行演示前的分析,深化掌握和理解理论课程中所学习到的进程和线程相关知识,进行启发式教学:程序会是怎样的执行结果?程序执行的过程是怎样的?进程、线程的创建、执行是怎么进行的?在整个程序执行过程中,各进程、线程的状态变化是怎样的?程序运行过程中,对内存会有怎样的影响?

接着,进行程序的直观演示,通过程序的直观演示,观察程序的执行结果,结合比较法、引导式教学:程序的执行结果和预想结果是否一致?

运行程序 1,分析运行结果:父子进程中输出的 i 的值分别为多少?

引导思考问题:两条输出语句中的 i 是否为同一个变量?父进程创建子进程后,父子进程并发执行,体现出了哪些特点?引导学生从时空开销、运行方式上评价多进程的开发。

运行程序 2,分析运行结果:子线程 thrd1 和 thrd2 执行过程中的 sharedi 是否为同一个变量?如果为同一个变量,在两个子线程中分别加了 100000,为什么主程序中输出的 sharedi 结果有可能不是 200000?

引导思考问题:线程间的并发执行体现出了哪些特点?引导学生从时空开销、运行方式上评价多线程的开发。

运行程序 3,分析运行结果:增加了 5 行代码后,为什么主程序中输出的 sharedi 为 200000?

引导思考问题:当多个线程对同一个变量进行访问时,为了保证结果的确定性,应注意什么问题?

最后,在以上教学方法的基础上,进行知识的“连连看”,将其作用于“外功”。向学生抛出问题:进程与线程在资源分配、独立性、开销、并发性、健壮性、执行过程等方面存在哪些差异?这些差异使得它们在不同的应用场景下具有哪些优势和局限性?进行探究式教学,引导学生分组讨论:以多进程与多线程开发的不同软件可能会体现出哪些特点?哪种情形下的软件开发适合选择多进程?哪种情形适合选择多线程?

四、结论

操作系统课程概念、原理多,内容枯燥,只有进一步丰富教学方法和优化教学内容,在操作系统课程教学中,与其他课程建立连接,强调“内外功”的相互促进,将枯燥的概念、原理进行迁移,才能促进知识的消化、吸收,才能提高学生的学习动力和兴趣,才能更好地服务于整个计算机体系知识结构的理解和掌握,提高学生的应变能力。这样不仅能让学生更深入地了解计算机系统,还能提升学生的专业技能和竞争力,为未来的职业发展打下坚实的基础。

参考文献:

[1]魏碧霞,陈飞,黄彬. 融合“三创”教育的工科类实践课程研究与探索[J]. 黑龙江教育(理论与实践),2023(11):75-77.

[2]蔡迅华. 新工科背景下操作系统原理课程教学探索[J]. 广西广播电视大学学报,2023,34(5):25-28.

[3]张静,王丽. 新工科背景下“PAD+OBE”模式在操作系统课程教学中的应用[J]. 现代职业教育,2024(24):109-112.

[4]张伟,朱志良,李传文,等. PBL 教学理论在操作系统

课程中的应用[J]. 计算机教育,2024(9):102-106.

[5]胥桂仙,李苏娟,何敬元. 融入多种教学方法的操作系统混合教学模式改革[J]. 中央民族大学学报(自然科学版),2022,31(4):90-94.

[6]张赛男,郑长友,胡斌,等. 突出工程实践能力培养的操作系统课程改革[J]. 计算机教育,2024(6):227-231.

[7]宁振虎,张腾,薛菲. 计算机操作系统课程体系建设探索[J]. 计算机教育,2024(9):137-140.

[8]罗宇,文艳军. 强化系统能力培养的操作系统教材建设[J]. 计算机教育,2024(11):18-21.

[9]李征. "一流课程"教学创新探索与实践——以"操作系统"课程为例[J]. 西部素质教育,2024,10(20):24-27.

[10]陈向群. "101 计划"操作系统课程知识体系建设及知识点关联性研究[J]. 计算机教育,2024(5):16-19.

Research on the Optimization of Teaching Methods and Content of Operating System Courses

XING Li-li, HAN Ying, LU Yu-hong

(School of Information Engineering, Institute of Disaster Prevention, Sanhe Hebei 065201, China)

Abstract: This paper analyzes the existing problems in the current teaching of operating system courses, and dissects the importance of underlying principles, concepts, and other knowledge in operating system courses for the development of upper-level application software; emphasizes the need to set reasonable teaching content during instruction and adopts various teaching methods to fully explore underlying principles, concepts, and other knowledge, helping students establish a "knowledge connection" and improve their professional knowledge system; finally, taking the two important concepts of processes and threads in operating system courses as examples, designs teaching cases and comprehensively utilizes various teaching methods to stimulate students' interest and motivation in learning, thereby enhancing their professional skills and competitiveness.

Key words: operating system courses; teaching methods; teaching content; teaching cases; process; thread