

# Exploration and Practice of Application-Driven Teaching Reform in the Course Principles and applications of Linux

Wenxuan Fan, Zhihui Li\*, Xiaoshuo Jia and Shaoze Zhang

College of Computer Science, Guangdong University of Science and Technology, Dongguan  
523000, China

## Abstract

Under the strategic background of cultivating application-oriented talents, the development of engineering practice ability, system application competence, and innovative thinking among computer science students has become a core training objective. Drawing on the experience of curriculum cluster development and teaching reform achievements in computer-related majors at universities, an application-driven teaching reform scheme is proposed. This scheme encompasses the construction of an application-oriented teaching resource repository, innovation in theoretical and experimental teaching methods, the establishment of a whole-process supervision system, and the reconstruction of a diversified assessment mechanism. It aims to build a teaching model centered on "practice as the core, cases as the guide, and students as the primary" that is, a student-centered approach. Through reform practice, students' learning initiative and enthusiasm for exploration have been effectively stimulated, and their practical Linux system operation skills, engineering application competence, and teamwork abilities have been significantly enhanced. The reform provides a replicable

## Keywords

Application-Driven Teaching Reform; Linux; Exploration and Practice.

## 1. Introduction

"Linux Principles and Applications," a core applied course for science and engineering majors such as Computer Science and Technology, Software Engineering, Internet of Things Engineering, Network Engineering, and Electronic Information Engineering in universities, bears the important mission of cultivating students' abilities in Linux system management, configuration, development, and engineering applications. It serves as a crucial foundation for students' subsequent learning of professional skills such as embedded development, server maintenance, cloud computing, and big data processing. In the era of deep integration between the digital economy and information technology, Linux, with its open-source, stable, and efficient characteristics, is widely used in key areas such as server clusters, embedded devices, and artificial intelligence platforms. Enterprises have an increasingly urgent need for technical talents with solid Linux practical skills and engineering application thinking. Therefore, the teaching quality of this course directly affects students' core professional competitiveness and their receptiveness to subsequent professional courses.

However, the traditional teaching and assessment model of the "Linux Principles and Applications" course has long suffered from numerous drawbacks, making it difficult to meet the requirements for cultivating application-oriented talents: First, the teaching logic is centered on theoretical lectures, with a large amount of class time spent explaining pure theoretical knowledge such as system principles, command syntax, and configuration file formats. Practical exercises are mostly textbook verification operations, which are seriously out of touch with actual industrial application scenarios. Second, the teaching model is teacher-

centered, with students in a passive state of receiving information. They lack initiative in self-learning and practical exploration, often resulting in the dilemma of "being able to recite commands but not being able to operate them" and "understanding principles but not being able to solve practical problems." Third, the practical cases are rigid and outdated, mostly verification content for single knowledge points, lacking comprehensiveness, innovation, and practicality, and failing to cultivate students' ability to solve complex problems. Fourth, the assessment method emphasizes results over process and written tests over practical operation, making it difficult to comprehensively measure students' system operation ability, engineering practice literacy, and innovative thinking. This leads to the widespread phenomenon of "high scores but low ability," which is seriously out of touch with the talent needs of enterprises.

The application-driven teaching model is guided by real-world application scenarios and uses real-world project cases as a vehicle to closely integrate theoretical knowledge with engineering practice. It emphasizes "learning by doing and applying what you learn," guiding students to build a knowledge system by completing practical application tasks, aligning with the "competency-oriented" talent cultivation requirements of engineering education in the new era. Integrating the application-driven concept into the teaching and assessment reform of the "Linux Principles and Applications" course effectively breaks through the limitations of traditional teaching, shifting the focus of teaching from "knowledge transmission" to "competency development," and achieving the teaching goals of "learning through application, improving skills through practice, and promoting practical application through assessment." In the process of completing practical application cases, students not only master the operation methods and application skills of the Linux system but also develop engineering thinking and teamwork awareness, laying a solid foundation for their future career development.

## **2. Current Status and Problems of Course Teaching and Assessment**

"Linux Principles and Applications" is a highly practical engineering course. Its core teaching objective is to cultivate students' ability to transform theoretical knowledge into practical skills, enabling them to independently solve Linux system configuration, operation, and development problems in real-world industrial scenarios. However, currently, the teaching and assessment of this course in most Chinese universities still suffer from a reversal of priorities, specifically manifested in the following four aspects:

### **2.1. Imbalance in teaching content: Emphasis on theoretical instruction and neglect of practical application**

Currently, most universities adopt a teaching model that prioritizes classroom theory teaching and supplements it with experimental teaching. Course implementation is teacher-centered and driven by the delivery of theoretical knowledge. In classroom teaching, teachers spend over 60% of their time explaining purely theoretical content such as Linux system architecture, command syntax rules, and configuration file parameters. Practical teaching is completed only through simple command demonstrations, lacking systematic cultivation of practical application skills such as system configuration, troubleshooting, and project development. For example, when teaching the "Shell Script Programming" chapter, teachers focus more on explaining syntax rules, neglecting training in the application of scripts in real-world scenarios such as automated operation and maintenance and data processing. Students mechanically memorize command formats and theoretical knowledge points but cannot understand their application logic in actual work, often encountering the dilemma of "understanding in class but not being able to use it afterward," failing to transform abstract theoretical knowledge into practical application skills, and significantly reducing learning effectiveness.

## **2.2. Rigid practical cases: lack of innovation and practicality, limiting the improvement of capabilities.**

Traditional courses often present experimental questions and practical cases that are merely verification content from textbooks. These are simplistic in format and function, and haven't been updated in years, resulting in a severe disconnect from real-world business applications. Practical tasks only require students to follow predetermined steps to input commands and verify results, without considering the complexity of real-world scenarios, system stability, or operational standardization. This rigid case-based approach allows students to simply apply textbook templates to complete tasks, neglecting complex problem analysis, solution design, and innovative thinking. Consequently, students' programming thinking, engineering application skills, and innovative awareness are not effectively cultivated and enhanced.

## **2.3. Lack of teaching supervision: Significant differences in students' academic levels make it difficult to implement personalized tutoring.**

"Linux Principles and Applications," a fundamental course for computer science majors, is often taught in large classes, frequently exceeding 50 students. This results in significant differences in students' computer skills, learning abilities, and self-study awareness. Some students possess a certain level of computer literacy and can quickly master Linux system operations; however, others lack basic operating system knowledge, making the learning process more challenging. After classroom instruction, teachers, limited by time and energy, are unable to effectively monitor each student's practical work: they cannot confirm whether students have completed the practical exercises independently, nor can they grasp the specific problems encountered during practice. Student assignments often consist of screenshots of the final results, making it impossible for teachers to trace the operational process, debugging strategies, and problem-solving methods, thus hindering the assessment of students' true mastery. Personalized tutoring for students with different backgrounds is difficult to implement. Students with weak foundations easily fall into learning difficulties, gradually lose interest, and ultimately give up on practical exercises; while students with stronger foundations lack challenging tasks, failing to fully stimulate their learning enthusiasm, and their practical skills development becomes merely a formality.

## **2.4. The assessment mechanism is flawed: it emphasizes results over process, and the proportion of practical assessment is too low.**

The course assessment system has significant flaws, making it difficult to comprehensively and objectively evaluate students' overall abilities. On the one hand, the assessment method relies primarily on a closed-book final exam, supplemented by regular coursework. Regular coursework largely depends on attendance and assignment submission (usually accounting for no more than 40%), lacking comprehensive monitoring of the student's learning process. The final exam focuses on testing theoretical knowledge memorization and command syntax identification (accounting for over 60%), severely neglecting the assessment of practical Linux system skills, troubleshooting abilities, and engineering application skills. For example, the final exam mainly uses multiple-choice, fill-in-the-blank, and short-answer questions to test command formats and theoretical knowledge points, while focusing less on practical aspects such as server configuration, Shell script development, and system troubleshooting. On the other hand, the assessment content is limited to verifying single knowledge points, lacking comprehensive assessment questions that connect to real-world engineering scenarios; the evaluation criteria are rigidly "standard answer-based," lacking tolerance for students' personalized operational approaches and innovative application solutions. This assessment model leads students to focus excessively on final exam scores, neglecting daily practical exercises and application attempts. As a result, students' engineering practice literacy and

practical application ability are not effectively trained, making it difficult to achieve the course's talent training goals.

### **3. Reform measures for course teaching and assessment methods**

Guided by application-driven and constructivist theories, the teaching model of the "Linux Principles and Applications" course will be transformed from a "theory-based" model to a "learning-by-doing" model that emphasizes "experimental practice leading theory, with practice taking precedence over theory." This model stresses practical application throughout the entire teaching process, adopts an application-driven teaching approach, and constructs a harmonious learning environment centered on students. Through practical case studies, students will be guided to independently build a complete knowledge system. Specific reform measures include four aspects: teaching resource development, teaching method optimization, enhanced process supervision, and restructuring of the assessment system.

#### **3.1. Build an application-oriented teaching resource database to solidify the foundation for reform.**

Teaching resources are the core support for implementing the application-driven teaching model. Adhering to the principles of "aligning with industry standards, meeting job requirements, and conforming to the teaching syllabus," an application-oriented teaching resource library for "Linux Principles and Applications" has been constructed, encompassing three major modules: case resources, online platforms, and supplementary materials.

**Case Resource Library Construction:** Practical cases suitable for teaching are extracted from real-world enterprise projects, covering five typical application scenarios: server configuration and management, embedded Linux development, automated operation and maintenance, system troubleshooting, and Shell scripting. Representative practical cases are also selected from past exam questions in competitions such as the National Computer Rank Examination, the Blue Bridge Cup, and the Software Cup to enrich the case types. All cases follow the principles of "moderate theory, emphasis on practice, and application orientation," covering the main knowledge points of the course and possessing both demonstrative and practical value.

Based on the Learning Tong online teaching platform, an online teaching resource center for "Linux Principles and Applications" has been built, uploading teaching videos, practical case analyses, command manuals, project development documents, assessment questions, and other resources to support students' pre-class preparation, post-class review, and self-study. The platform features an online Q&A area, case study discussion area, and results display area to facilitate interaction between teachers and students and collaboration among students. Simultaneously, leveraging the platform's learning data statistics function, teachers can monitor students' resource learning time, assignment completion, and case study discussion participation in real time, providing objective evidence for formative assessments.

**Supplementary Materials:** The platform compiles commonly used Linux operation techniques, troubleshooting solutions, project development standards, and real company interview questions into a "Linux Engineering Practice Handbook" for students' after-class learning. The handbook includes practical content such as "Server Security Configuration Guide," "Shell Script Optimization Techniques," and "Common Troubleshooting Flowcharts," helping students connect with real-world enterprise work scenarios.

#### **3.2. Optimize teaching methods to achieve deep integration of theory and practice**

Breaking away from the traditional teaching logic centered on theoretical instruction, this curriculum constructs a teaching system of "theoretical knowledge + applied cases + project-

driven learning." It employs a three-step practice-led teaching model: pre-class preparation – in-class discussion – post-class practice. All steps follow the principle of practice first, then theory, and finally innovation, cyclically permeating the entire teaching process:

**Pre-class preparation:** Teachers post applied cases and pre-class assignments on an online teaching platform, requiring students to independently explore the cases using platform resources. Through pre-class preparation, students achieve initial independent knowledge construction and cultivate self-learning abilities.

**In-class discussion:** Teachers guide students through the problems encountered during pre-class preparation, explaining theoretical knowledge and practical methods, integrating abstract theory into the case-solving process. Students then participate in group discussions to collaboratively solve complex problems within the cases. Teachers provide targeted answers and clarification, ensuring authoritative confirmation of knowledge. Classroom teaching reduces pure theoretical lecture time (to within 30%) and increases interactive elements such as classroom practice, group discussions, and case analysis, guiding students to actively participate in the knowledge construction process.

**Homework:** Assign extended practical tasks, requiring students to optimize the case studies presented in class or complete similar application cases to consolidate and innovate their knowledge. For example, after completing the "Personal Website Server Setup" case study, students are required to expand upon it by implementing functions such as "Multi-user Website Deployment," "Automatic Log Backup," and "Automatic Fault Alarm," thereby cultivating innovative thinking and engineering application skills.

### **3.3. Strengthen process supervision and differentiated guidance, while taking into account individual student differences.**

Leveraging online teaching platforms and Linux hands-on training platforms, a tiered after-class practice system and a full-process supervision mechanism are constructed to cater to the learning needs of students with varying levels of prior knowledge and ensure the effectiveness of practical teaching:

**Tiered Practice Task Design:** Weekly practical tasks are assigned in three levels: "Basic Questions (Consolidating Knowledge Points) + Advanced Questions (Strengthening Application) + Extension Questions (Stimulating Innovation)." Basic questions are for all students, focusing on consolidating core knowledge points; advanced questions are for students with a strong foundation, focusing on strengthening application skills; and extension questions are for students with extra capacity, focusing on cultivating innovation. Students can choose to complete the appropriate level of tasks based on their own abilities, and teachers provide additional incentives to students who complete the extension questions.

**Full-Process Supervision Mechanism:** Utilizing the Linux hands-on training platform, data such as students' practice time, operation steps, command input records, problem-solving processes, and task completion quality are recorded in real time. Teachers use backend data analysis to accurately identify students who did not participate in practice, practiced perfunctorily, or experienced significant difficulties. For example, students who fail to submit tasks multiple times are given online reminders and one-on-one communication to understand their situation; students who frequently make command errors during practical exercises receive targeted basic tutoring.

A diversified tutoring system is established: a three-tiered tutoring mechanism of "teachers + lab assistants + student assistants". Course team teachers serve as lab assistants to each other, providing cross-tutoring during lab classes; students with strong practical skills are selected from the class to serve as lab assistants, assisting teachers in providing classroom practical guidance and answering questions after class; online Q&A groups and platform message boards provide students with 24-hour response to questions. Common problems are addressed and

demonstrated in the next class; individual problems are addressed through one-on-one tutoring to ensure timely feedback and that every student receives effective guidance.

### 3.4. Reconstruct the assessment system and build a "four-in-one" comprehensive evaluation model.

This innovative assessment approach breaks away from the traditional "one exam determines your future" model, constructing a comprehensive four-in-one evaluation system, as shown in Table 1: "Self-directed learning effectiveness (20%) + Classroom performance (20%) + Experimental results (30%) + Final exam (30%)". This system comprehensively covers the entire learning process, strengthening the assessment of practical and application skills. Specific assessment content is as follows:

Self-directed learning effectiveness (20%): Quantitative evaluation based on online teaching platform data. This objectively reflects students' self-directed learning attitude and ability, guiding them to value pre-class preparation and post-class extension.

Classroom performance (20%): Recorded in real-time by the instructor. Students who actively participate in classroom interaction and demonstrate excellent practical skills receive bonus points, encouraging active participation in classroom activities.

Experimental results (30%): As the core of the assessment, teachers score based on functional completeness, technical rationality, and document standardization.

Final exam (30%): A combination of "computer-based practical exam + project defense," with the computer-based practical exam accounting for 20% and the project defense accounting for 10%. The evaluation criteria adopt a "basic score + optimization score + innovation score" model, encouraging students to use novel methods and optimize functional design to break free from the constraints of "standard answers".

Table 1. Composition and Proportion of Comprehensive Grades

Assessment Category: Final Exam Grade	Process-based assessment score (70%)			Final exam (30%)	
Assessment Items	Self-directed learning	Classroom performance	Experimental results	Computer-based exam	Project defense
Assessment ratio	20%	20%	30%	20%	10%

## 4. Implementation effect

The application-driven teaching and assessment reform of the "Linux Principles and Applications" course was implemented in the 2022 cohort of Software Engineering and Network Engineering students in the School of Computer Science at our university. Through application-based case teaching, tiered practical tasks, and comprehensive assessment throughout the entire process, students' learning initiative and classroom participation have significantly improved. The course teaching quality and students' practical application abilities have been effectively enhanced. Specific implementation results are reflected in the following aspects:

### 4.1. Students' initiative in learning has been significantly enhanced.

After the reform, the student resource learning rate on the online teaching platform increased from 65% to 92%, the average submission rate of the three-level after-class practical tasks increased from 70% to 96%, and the participation rate in classroom practical exercises and group discussions increased from 55% to 90%. More students actively consulted Linux

industry materials, participated in online technical community discussions (such as CSDN and Stack Overflow), and independently completed extended practical tasks, forming a positive learning atmosphere of "active thinking, enjoyment of practice, and courage to innovate." In the course feedback survey, 85% of students believed that "application-based cases made learning more goal-oriented," and 78% of students said that "differentiated tasks could take into account their own learning foundation," indicating that students' self-learning ability has been effectively cultivated.

#### **4.2. Students' practical and application skills have been greatly improved.**

Through application-based case studies and hands-on assessments throughout the entire process, students have broken free from their reliance on textbook templates and are able to independently complete practical tasks such as Linux server configuration, system maintenance, and simple troubleshooting. After the reform, students' average score in the final computer-based practical exam increased from 68 to 83 points, and some students were even able to develop small embedded Linux application projects based on course knowledge.

#### **4.3. The fairness and objectivity of course assessments have been significantly improved.**

The comprehensive assessment system covers the entire learning process for students. Online platform data and practical operation records provide objective evidence for the assessment, effectively avoiding the randomness of "one exam determining the grade" in traditional assessments and reducing plagiarism and perfunctory work. After the reform, the standard deviation of course assessments decreased from 15.2 to 10.01, and the distribution of grades became more reasonable, more accurately reflecting students' actual learning situation and ability level. In course evaluations, 88% of students believed that "the new assessment method better reflects true ability," effectively avoiding the problem of "high scores but low ability," and has gained widespread recognition from students and peers.

#### **4.4. The achievement rate of course teaching objectives has been significantly improved.**

After the reform, the overall pass rate of the course assessment increased from 85% to 98%, with the percentage of students scoring 80 or above rising from 20% to 45%. Students' understanding of Linux principles and their practical application abilities both met the course's talent development goals. Course evaluation results showed that student satisfaction with the course increased from 72 to 91 points, and peer reviews ranked the course among the top in the school for indicators such as "course's practical orientation" and "effectiveness in cultivating students' abilities."

## **5. Conclusion**

This application-driven reform of the "Linux Principles and Applications" course addresses the core shortcomings of traditional teaching methods—namely, an overemphasis on theory and neglect of practice, a focus on results over process, outdated and simplistic case studies, and an imperfect assessment mechanism—by building an application-oriented teaching resource library, optimizing theoretical and experimental teaching methods, strengthening process supervision and tiered guidance, and reconstructing a comprehensive "four-in-one" assessment system. Teaching practice demonstrates that this reform effectively mobilizes students' initiative and enthusiasm for learning, significantly improves their Linux system practical skills, engineering application literacy, and innovative thinking, and achieves a shift in teaching focus from "knowledge transmission" to "ability cultivation." This aligns with the core requirements of cultivating application-oriented talents and lays the foundation for training

more computer application-oriented technical personnel who meet industry needs. implementation of the assessment mode reform in this course has significantly enhanced students' learning initiative through problem oriented teaching, layered practical tasks, and process based assessment incentives. After the reform, the average submission rate of students' online programming tasks increased from 75% to 95%, the classroom interaction participation rate increased from 60% to 88%, and the proportion of students who independently searched programming materials and participated in online programming community discussions increased significantly, forming a good learning atmosphere of "active thinking and willingness to practice". After grading the final exam papers and summarizing the data, the students' final paper scores and overall evaluation scores are obtained, and the final paper scores and overall evaluation scores are analyzed.

## Acknowledgements

This work was financially supported by GKZLGC2023034, GKZLGC2024153, GKZLGC2024180 fund.

## References

- [1] Alharbi, A., & Zhang, L. (2021). Application-Driven Teaching Reform for Linux Courses: Enhancing Practical Competence in Engineering Students. *Journal of Engineering Education Transformations*, 34(3), 67-82.
- [2] Kim, S., Park, J., & Lee, H. (2022). A Blended Learning Model for Linux System Administration: Integrating Application Cases and Online Platforms. *Computer Applications in Engineering Education*, 30(4), 987-1005.
- [3] Rodriguez, M., & Gonzalez, A. (2023). Project-Based Learning in Linux Courses: Bridging the Gap Between Theory and Industry Needs. *IEEE Transactions on Education*, 66(2), 145-153.
- [4] Sharma, R., & Singh, S. (2024). Formative Assessment for Linux Practical Courses: A Data-Driven Approach Using Online Programming Platforms. *European Journal of Engineering Education*, 49(1), 189-206.
- [5] Chen, W., & Wang, Y. (2024). Enhancing Student Engagement in Linux Courses Through Application-Driven Case Studies and Collaborative Projects. *Education and Information Technologies*, 29(3), 2871-2890.
- [6] Ferreira, J., & Silva, M. (2023). Reforming Linux Course Assessment: From Knowledge-Centered to Competency-Based Evaluation. *Journal of Computing Sciences in Colleges*, 38(6), 98-110.
- [7] Li, Z., & Jia, X. (2023). Target Detection Based on Taylor Expansion and Bilateral Symmetric Network. *IEEE the 3rd International Conference on Computer Communication and Artificial Intelligence*, 456-462.
- [8] Xie, B., & Liu, Y. (2022). Exploring the Path of "Dual-Teacher" Faculty Construction in the Context of School-Enterprise Cooperation. *International Core Journal of Engineering*, 8(4), 78-85.