



融合知识图谱和大模型的
操作系统学习平台

OPERATING SYSTEMS LEARNING
PLATFORM BASED ON KNOWLEDGE
GRAPHS AND LLMs

学 院： 计算机科学与技术学院
专 业： 计算机科学与技术
姓 名： 王善路
学 号： 210860119
指 导 教 师： 王璿
提 交 日 期： XX年X月

东华大学

毕业论文（设计）学术诚信声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。本论文（设计）为本人亲自撰写，对所写的内容负责。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期： 年 月 日

东华大学

毕业论文（设计）版权使用授权书

本毕业论文（设计）作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权东华大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业论文（设计）。

保密☐，在____年解密后适用本授权书。

本论文属于

不保密☐。

（请在以上方框内打“√”）

指导教师签名：

作者签名：

日期： 年 月 日

日期： 年 月 日

融合知识图谱和大模型的操作系统学习平台

摘 要

近年来,传统操作系统教学面临知识点碎片化、内容抽象难懂以及个性化学习支持不足等问题,学生在学习过程中难以系统整合分散的教学资源,且缺乏即时有效的问题解答渠道。为解决这些问题,本文设计实现了融合知识图谱与大模型的操作系统学习平台。

本文首先分析了操作系统学习平台的需求,提出了知识结构化、模块化设计、知识增强的智能问答及前后端分离四大设计原则。其次,构建了包含前端展示层、后端服务层和数据存储层的三层系统架构,详细介绍了系统的技术选型与工作流程。

在功能架构方面,系统包含用户账户、智能问答、知识图谱和学习资源四大模块。其中,知识图谱模块负责构建操作系统领域知识体系;智能问答模块提供融合知识图谱增强的大模型问答服务;学习资源模块整合教学视频和习题资源,为学生提供全方位的学习支持。开发环境采用 Windows 系统,使用 Vue3、Node.js、FastAPI 等技术框架,以及 MySQL、Neo4j 存储用户数据和知识图谱。该系统的应用将有效改善操作系统教学中知识分散、难以理解及缺乏个性化支持等问题,为学生提供智能化、个性化的学习环境。

关键词: 知识图谱, 大语言模型, 操作系统, 智能学习平台, 智能问答

OPERATING SYSTEMS LEARNING PLATFORM BASED ON KNOWLEDGE GRAPHS AND LLMs

ABSTRACT

In recent years, traditional operating system education has faced challenges such as fragmented knowledge points, abstract and difficult-to-understand content, and insufficient personalized learning support. Students struggle to systematically integrate scattered teaching resources and lack immediate and effective channels for answering questions. To address these issues, this paper designs and implements an operating system learning platform that integrates knowledge graphs and large language models.

The paper first analyzes the requirements for an operating system learning platform and proposes four design principles: knowledge structuring, modular design, knowledge-enhanced intelligent Q&A, and separation of front and back ends. Next, it constructs a three-layer system architecture including a front-end presentation layer, back-end service layer, and data storage layer, with detailed introductions to the system's technology selection and workflow.

In terms of functional architecture, the system includes four main modules: user accounts, intelligent Q&A, knowledge graph, and learning resources. The knowledge graph module is responsible for constructing the knowledge system in the operating system domain; the intelligent Q&A module provides large model question-answering services enhanced by knowledge graphs; the learning resources module integrates teaching videos and practice exercises to provide comprehensive learning support for students. The development environment uses Windows system, with technology

frameworks such as Vue3, Node.js, FastAPI, as well as MySQL and Neo4j for storing user data and knowledge graphs. The application of this system will effectively improve issues in operating system teaching such as scattered knowledge, difficulty in understanding, and lack of personalized support, providing students with an intelligent and personalized learning environment.

Key words: Knowledge Graph, Large Language Model, Operating System, Intelligent Learning Platform, Intelligent Q&A

目 录

1	绪论	1
1.1	研究背景与意义	1
1.2	研究现状	2
1.3	研究内容	4
2	融合知识图谱与大模型的操作系统学习平台总体设计	5
2.1	需求分析	5
2.2	系统设计原则	5
2.3	系统技术架构	6
2.4	系统功能架构	8
2.5	系统开发环境与工具简介	8
2.6	数据库设计	10
3	基于大模型的操作系统知识图谱设计与实现	12
3.1	操作系统知识图谱系统设计	12
3.1.1	知识图谱建模方法论	12
3.1.2	知识图谱本体设计	12
3.2	基于大语言模型的知识图谱构建	14
3.2.1	基于大语言模型的教学资源实体识别与属性抽取	14
3.2.2	基于大语言模型的知识关系提取与验证	18
3.2.3	操作系统知识图谱的质量评估与优化	19
3.3	本章小结	21
4	操作系统知识图谱智能问答系统的设计与实现	22
4.1	知识增强问答算法设计与实现	22
4.1.1	知识图谱检索算法实现	22
4.1.2	Bing Search API 集成与互联网知识获取	25
4.1.3	大模型问答增强机制	27
4.1.4	可溯源问答结果生成	29



4.2 系统功能实现	30
4.2.1 问答功能	30
4.2.2 查看聊天记录功能	32
4.2.3 知识溯源功能	33
4.2.4 操作系统知识图谱可视化功能	34
4.3 本章小结	36
5 总结与展望	37
参考文献	38
致谢	40

1 绪论

1.1 研究背景与意义

随着信息技术的飞速发展，教育领域正经历深刻的变革。传统的教学模式正逐步向数字化、智能化方向转型，以满足个性化、精准化的学习需求^[1]。这种转型推动了人工智能、大数据等技术与教育领域的深度融合，为教学方法的创新和教育资源的高效利用带来了新的契机。其中，知识图谱作为一种强大的知识表示与组织方式，凭借其直观的图形化表达与语义关联能力，成为教育技术领域的研究热点，广泛应用于知识管理、教学辅助、学习分析等场景。

知识图谱能够将知识点及其相互关系以图形化的形式呈现，帮助学习者直观地理解知识结构，探索知识之间的关联，从而促进深度学习的发生。在教育场景中，特别是在复杂的学科知识教学中，知识图谱的作用尤为显著。例如，操作系统课程作为计算机科学与技术专业的核心课程，涵盖了进程管理、内存管理、文件系统、输入输出系统等诸多知识点，同时知识点之间存在复杂的逻辑联系和依赖关系。学生在学习过程中，常常面临知识点繁杂且难以系统掌握的困难。基于知识图谱的学习系统能够为学生提供清晰的知识结构，通过关联性、层次化的图形化展示，帮助学生构建完整的知识体系，并加深对复杂概念的理解，从而显著提升学习效率与效果。

此外，随着人工智能技术的快速发展，大语言模型（如 GPT-4）展现出强大的自然语言处理与知识推理能力，为教育智能化提供了更多可能性^[2]。通过将大语言模型与知识图谱相结合，可以进一步增强教育系统的智能化水平。例如，大语言模型能够通过自然语言理解实现智能问答、自动化题库管理、智能化知识点推理、以及个性化学习路径推荐等功能。这些功能不仅满足了学生个性化学习需求，还能根据学习行为数据提供实时的学习反馈与建议，显著提升学生的学习体验。同时，教师也可以借助这些工具进行教学资源的智能生成与优化，推动教学

质量的全面提升。

在操作系统课程的教学场景中,通过引入大语言模型与知识图谱的融合技术,不仅可以实现知识点的图形化表示,还能够提供交互式学习功能,让学生以自然语言的方式与知识系统进行实时交互。例如,王芳等^[3]构建了 C++ 语言的知识图谱方便学生学习,学生可以提出问题并获得基于知识图谱的智能回答,也可以在学习中发现知识点的关联性与层次关系,从而优化学习路径,弥补知识盲区。此外,通过对学习行为数据的分析,还可以生成个性化学习报告,为教师和学生提供针对性的教学建议。这种智能化学习系统,不仅具有重要的理论研究价值,还在教学实践中展现出广阔的应用前景。

因此,设计并实现一个基于知识图谱的操作系统学习系统,融合大语言模型的能力,不仅能够解决传统教学中的诸多痛点,还可以推动智能教育技术的发展。该系统将结合知识图谱的结构化知识表示能力与大语言模型的智能化交互能力,为学生提供一个功能丰富、高效智能的学习平台,同时为教育研究者探索人工智能与教育深度融合提供一个典型的应用场景。

1.2 研究现状

当前在教育领域,知识图谱(Knowledge Graph, KG)作为一种强大的知识表示与组织方式,已成为研究热点,特别是在个性化教育和智能化教学系统的设计中。知识图谱能够帮助学生直观地理解复杂知识结构,实现学习路径推荐、动态知识查询和个性化评估,为教育领域带来了深刻变革。

知识图谱的构建是核心技术之一。传统方法多依赖人工规则和手动输入,虽然在部分领域仍有应用,但存在更新缓慢和成本高的问题。近年来,基于自然语言处理(NLP)和深度学习的自动化构建方法逐渐主流。Zhong 等^[4]综述了自动构建的研究进展,提出基于文本挖掘和关系抽取的方法,取得了良好效果。Liang 等^[5]研究了大语言模型(LLM)增强的知识图谱构建技术,在复杂产品设计领域展现了其高效性和准确性。Su 等^[6]进一步提出了一种基于关系数据的自动化方法,实现多领域高效应用。

基于深度学习的技术展现出显著潜力。Meyer 等^[7]利用大语言模型(如 ChatGPT)辅助知识图谱工程,证明 LLM 在构建与推理中提升了效率和质量。Zhu 等^[8]详细分析了 LLM 在知识图谱生成、更新及推理中的应用,并探讨其未

来发展趋势。

在教育领域，知识图谱的应用集中于知识管理、学习辅助与教学优化。知识图谱能帮助学生理解知识点间关系，尤其适用于操作系统和人工智能等复杂学科。Lin 等^[9]提出基于知识图谱的学习路径推荐方法，而 Chen 等^[10]设计了基于知识推理的学习模型，通过智能问答和个性化推荐为学生提供精准支持。

知识图谱与 LLM 的结合成为教育智能化的重要趋势。Abu-Salih 等^[11]系统综述了这一方向，强调其在智能答疑、内容生成及学习路径推荐中的潜力。Trajanoska 等^[12]的研究表明，使用先进的大语言模型可以提高从非结构化文本创建知识图谱的准确性，甚至可以探索使用基础大语言模型进行自动本体创建，从而生成更相关、更准确的知识图谱。

Sen 等^[13]的研究表明，利用知识图谱事实增强语言模型提示可以显著提高复杂问答性能，特别是在需要计数、交叉或多跳推理操作的复杂问题上。Guo 等^[14]开发的基于大语言模型和知识图谱的医学问答系统，通过语义匹配与命名实体识别以及知识图谱增强，充分利用问题和答案中实体之间的关系，在多个数据集上的实验证明该方法相较于强基线模型具有一致的改进。

Zhang 和 Hao^[15]利用大语言模型构建了中医知识图谱，通过收集和嵌入大量领域相关数据，生成精确的表示并转化为知识图谱格式，实验评估证实了所构建图谱的准确性和有效性，为学科知识的学习、研究和应用提供了坚实基础。Baek 等^[16]提出的知识增强语言模型提示框架(KAPING)实现了零样本知识图谱问答，通过在输入中直接增强知识，避免了大型语言模型依赖内部知识可能产生的事实错误，并且无需模型训练，完全实现零样本学习。Jiang^[17]提出的"思想知识图谱"使大语言模型能够利用知识图谱进行更有效的推理，为教育领域的智能问答提供了新思路。

刘敏^[18]研究了基于深度学习的多源信息融合知识图谱智能化构建技术，为复杂领域知识图谱的构建提供了新思路。随着 LLM 技术发展，其在教育领域的应用前景愈加广阔，可为教师和学生提供实时支持。

然而，这一领域仍面临挑战，包括知识点提取、关系抽取、图谱更新等技术问题，以及确保图谱准确性和推理链有效性的难点。未来，随着技术进步和数据积累，知识图谱与 LLM 的结合将为个性化教育和自适应学习系统提供更智能化

支持。因此本项目拟采用知识图谱与大语言模型的融合技术，针对操作系统教学中知识点繁杂、难以系统掌握的痛点，通过构建操作系统知识图谱，结合深度学习、自然语言处理与图数据库，实现知识结构的可视化、动态问答等功能。项目将使用 Neo4j 进行知识存储，结合 Vue3 和 FastAPI 开发交互式学习平台，显著提升学习效率与效果，并推动教育智能化发展。

1.3 研究内容

本文的研究内容主要包括三个部分，首先研究了利用大语言模型自动化构建操作系统知识图谱的方法，本文针对知识图谱手动构建效率低、一致性难以保证的问题，设计了基于大模型辅助的知识图谱自动化构建流程。该流程通过文本挖掘与实体识别技术，从操作系统教材、题库和教学视频中自动提取知识点、定义和关系，构建包含章节节点、子章节节点、概念节点、题目节点和视频节点的结构化知识体系，大幅提高了知识图谱的构建效率和质量。

其次，设计并实现了基于知识增强的智能问答系统，本文提出了结合知识图谱检索和大语言模型的可溯源问答方案，解决了传统大模型因无法溯源而产生"幻觉"的问题。系统通过精准匹配知识图谱中的相关概念节点，提取知识点语料作为背景知识输入给大模型，并通过 Bing Search API 实时检索互联网信息作为补充，显著增强了回答的准确性和全面性，为操作系统学习提供了强有力的智能辅助工具。

最后，为满足操作系统学习的系统性和可视化需求，本文构建了全面的操作系统知识图谱可视化系统。该系统包含章节节点、子章节节点、概念节点、题目节点和视频节点五种核心节点类型，形成完整的知识结构体系，支持用户根据章节和子章节筛选查看相关知识结构，通过交互式节点探索深入了解具体知识点内容，实现了知识的可视化呈现与系统化学习，帮助学生全面把握操作系统学科体系。

2 融合知识图谱与大模型的操作系统学习平台总体设计

2.1 需求分析

本文通过对操作系统学习平台的调查研究，结合高校教师和学生的实际需求，对融合知识图谱与大模型的操作系统学习平台进行设计与开发。

经过研究，首先了解到传统的操作系统教学模式往往以教材和课堂讲授为主，学生在学习过程中常常面临知识点碎片化、内容抽象难懂、个性化学习支持不足等问题。其次，现有的教学资源分散在书本、视频、习题等多种载体中，学生难以系统地整合这些资源进行学习。此外，学生在自主学习过程中遇到问题时，往往缺乏即时有效的解答渠道。

在听取了教师和学生的建议后，本文提出设计一个融合知识图谱与大模型的操作系统学习平台来满足上述需求。通过构建操作系统领域的知识图谱，实现知识的结构化表达与可视化呈现，帮助学生建立系统性认知；利用大语言模型与知识图谱相结合的方式，提供精准、可溯源的智能问答服务，解决学生自主学习过程中的疑惑；同时整合教材、视频、习题等多种学习资源，为学生提供全方位的学习支持。

经过分析不难得出，整个系统的核心功能模块可分为知识图谱构建与管理模块、智能问答模块、学习资源整合模块三大部分。知识图谱模块负责构建和可视化操作系统领域知识体系；智能问答模块提供融合知识图谱增强的大模型问答服务；学习资源整合模块关联相关的教学视频和习题。

2.2 系统设计原则

融合知识图谱与大模型的操作系统学习平台的主要设计原则如下：

(1) 知识结构化设计原则

系统采用知识图谱技术对操作系统领域知识进行结构化表达，从章节、子章节、概念、题目到视频等多个维度构建完整的知识网络。这种结构化设计使知识点之间的关系清晰可见，有助于学生形成系统性认知，同时也为智能问答提供了可靠的知识基础。

(2) 模块化设计原则

系统采用模块化设计思想，将复杂的操作系统学习平台划分为知识图谱构建与管理、智能问答、学习资源整合等多个功能模块。每个模块相对独立，职责明确，既能够单独开发和测试，又能够通过接口协同工作，形成完整的系统功能。这种设计方式有利于提高开发效率，简化系统维护，并支持未来的功能扩展。

(3) 知识增强的智能问答原则

系统的智能问答功能不仅依赖大语言模型的自然语言处理能力，更融合了结构化的知识图谱和互联网检索结果，形成知识增强的问答机制。这一设计原则确保了系统回答的准确性、可靠性和可溯源性，避免了单纯依赖大模型可能带来的"幻觉"问题。

(4) 前后端分离的开发原则

系统采用前后端分离的开发架构，前端负责用户交互界面的实现，后端负责业务逻辑处理和数据访问。这种设计使得前后端开发可以并行进行，提高开发效率；同时也提高了系统的可维护性和可扩展性。

2.3 系统技术架构

本系统采用前后端分离的架构，基于 Vue3、FastAPI、Neo4j 等框架和工具，构建了包含前端展示层、后端服务层和数据存储层的三层架构体系，如图 2-1 所示。三层架构的具体作用如下：

前端展示层是系统与用户直接交互的界面，采用 Vue3 框架开发，使用 HTML、CSS 和 JavaScript 等技术实现。主要功能包括知识图谱可视化展示、智能问答交互界面、学习资源展示等。前端通过 Axios 进行 HTTP 请求，与后端服务进行数据交互；同时通过 WebSocket 实现实时的流式问答功能，提升用户体验。

后端服务层采用 Python 语言开发，基于 FastAPI 框架构建 Web 服务。主要负责处理前端请求，实现业务逻辑，包括知识图谱检索、大模型调用、资源推荐等核心功能。后端服务通过 API 与大语言模型服务和 Bing Search API 等外部服务进行交互，获取问题解答和互联网信息；同时通过数据访问层与数据库交互，获取和存储系统数据。

数据存储层由 Neo4j 图数据库和 MySQL 关系型数据库组成。Neo4j 主要用于存储操作系统知识图谱，包括章节、子章节、概念、题目和视频等节点及其关

系；MySQL 则用于存储用户信息、聊天历史记录等结构化数据。两种数据库相互补充，共同支持系统的数据需求。

系统工作流程如下：用户通过前端界面发起问答请求，后端服务接收请求后，首先从 Neo4j 知识图谱中检索相关知识，同时调用 Bing Search API 获取互联网补充信息，将这些信息作为上下文与用户问题一起传给大语言模型。大模型生成的回答通过 WebSocket 实时传送给前端，以流式方式呈现给用户。同时，系统根据问题内容，推荐相关的题目和视频资源，提供全方位的学习支持。

这种技术架构设计充分利用了知识图谱的结构化优势和大模型的自然语言处理能力，为操作系统学习提供了智能化、个性化的支持环境。

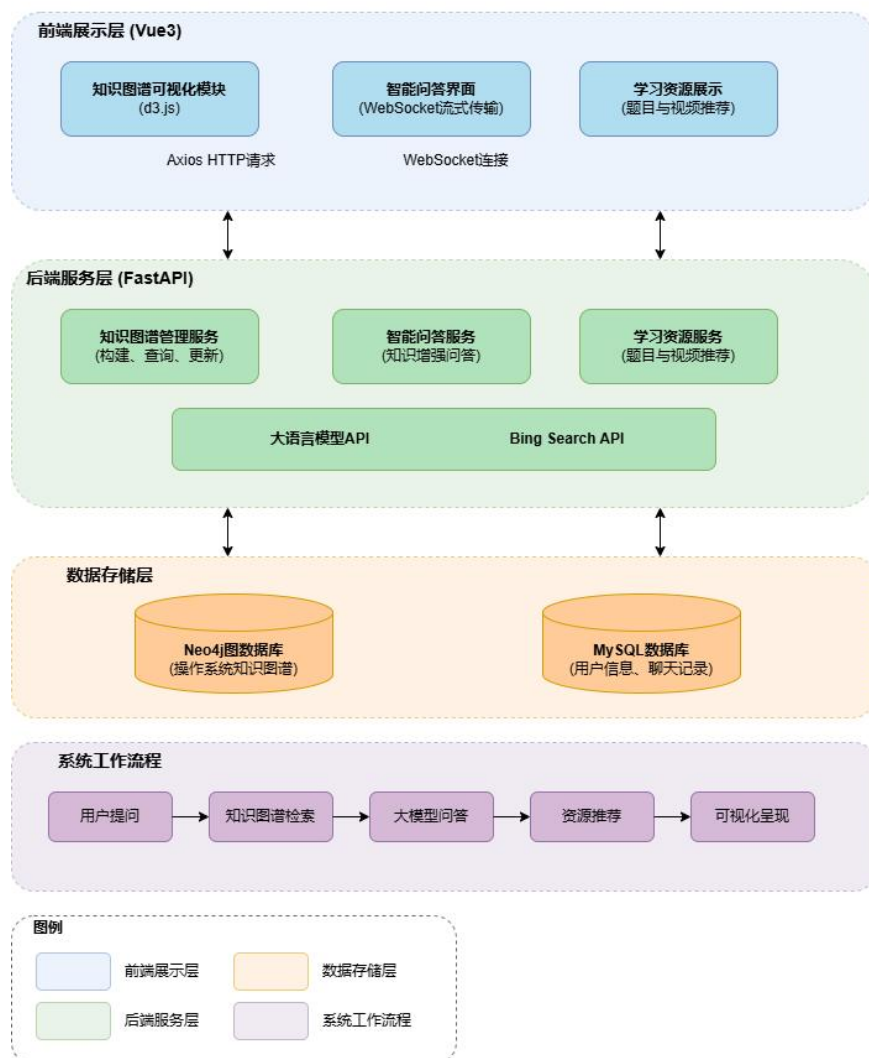


图 2-1 系统总体设计架构

2.4 系统功能架构

操作系统学习平台功能模块主要分为用户账户模块、智能问答模块、知识图谱模块和学习资源模块四大模块。用户账户模块主要负责管理用户的登录、注册和个人资料，包括用户认证功能、个人信息管理和学习历史记录功能；智能问答模块提供基于大模型的交互式学习体验，包括知识图谱增强问答、互联网检索增强问答和学习资料检索增强问答三大功能；知识图谱模块主要负责展示操作系统知识的结构化内容，包括知识图谱可视化浏览、知识节点交互查询和知识关系探索功能；学习资源模块负责整合和推荐相关学习材料，包括相关视频推荐、练习题目推荐和下载功能，系统功能架构图如图 2-2 所示。

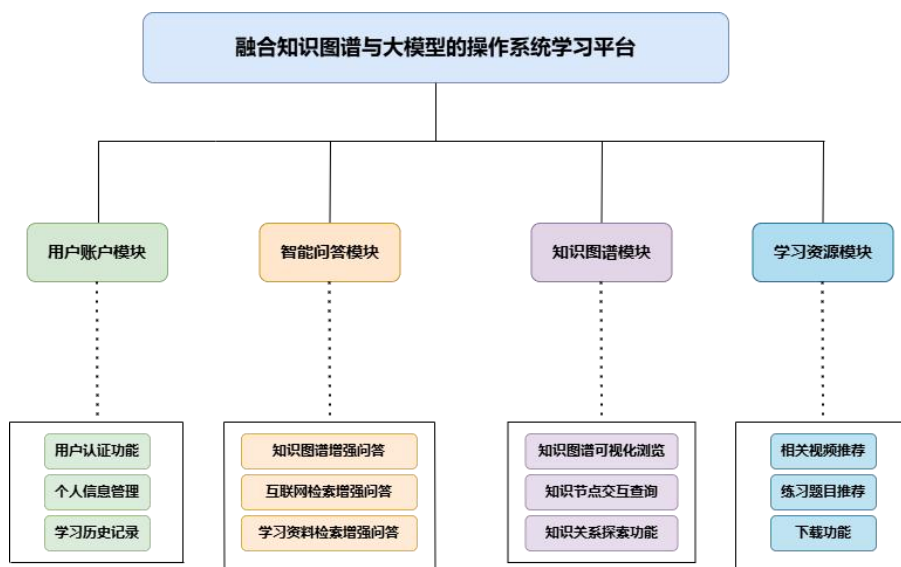


图 2-2 系统功能架构

2.5 系统开发环境与工具简介

融合知识图谱与大模型的操作系统学习平台在 Windows 系统环境下进行开发，采用 Vue3 与 Node.js 作为核心框架，Visual Studio Code 作为开发工具。数据存储层面采用 MySQL 5.7 关系型数据库和 Neo4j 5.13 图数据库，辅以相应的数据库管理工具。

(1) Vue3 是本系统前端实现的基础框架。选择 Vue3 主要考虑其组件化开发特性，可将复杂界面分解为独立组件，简化开发流程。在实际应用中，Vue3 的 Composition API 提供了更灵活的代码组织方式，改善了代码可维护性。系统的知识图谱可视化与实时聊天界面实现过程中，Vue3 的响应式系统有效降低了状

态管理复杂度，使界面能够准确响应各类用户交互。另外，Vue3 对 TypeScript 的良好支持也为代码质量提供了保障，减少了潜在错误。

(2) 服务器端技术采用 Node.js 作为运行环境。Node.js 在处理并发连接和非阻塞 I/O 操作方面表现出色，尤其是在实现 WebSocket 连接支持系统实时聊天功能时，其事件驱动模型提供了稳定性能。前后端技术栈统一使用 JavaScript 语言，减少了技术切换成本，有利于将更多精力集中在业务逻辑实现上。此外，Node.js 丰富的包管理生态系统提供了众多可靠的第三方模块，加快了开发进度。

(3) 开发工具采用 Visual Studio Code 作为主要代码编辑环境。实际开发中，通过配置专用插件如 Vue Language Features 完成 Vue 组件的语法检查，ESLint 与 Prettier 保证代码风格一致性，Python 插件则为 FastAPI 后端开发提供支持。VS Code 集成的 Git 版本控制功能便于跟踪代码变更，特别是在频繁调整知识图谱结构的开发阶段，可靠地记录各版本间的差异，确保开发过程可追溯。调试功能的灵活性也大大提高了问题定位效率。

(4) 系统选用 MySQL 5.7 存储用户数据与聊天历史记录。数据库设计阶段，针对用户信息、对话记录等数据进行了表结构规划，设计了合理的字段类型与索引策略。特别对高频查询操作的聊天记录表建立了复合索引，优化查询性能。MySQL 的事务管理机制确保了数据一致性，特别是在处理并发用户访问时，能够有效避免数据冲突问题。通过规范的数据库设计，系统实现了高效的数据访问与存储功能。

(5) 数据库管理工具选用 Navicat 进行 MySQL 数据操作。开发过程中，Navicat 的表设计功能辅助数据结构实现，其查询编辑器便于执行与测试复杂 SQL 语句。数据导出功能用于定期备份开发数据，保障数据安全。Navicat 的直观界面减少了数据库管理学习成本，提高了开发效率。在系统调试阶段，Navicat 的数据可视化功能对验证数据流转、检查数据完整性提供了直观支持。

(6) 操作系统知识图谱的构建与存储采用 Neo4j 5.13 图数据库。图谱设计过程中，通过分析操作系统教材内容，抽取章节、子章节、概念、题目和视频五类节点类型，并定义了相应的关联关系。Neo4j 基于图的存储模型适合表达知识间的复杂联系，Cypher 查询语言的表达能力支持实现各类知识检索需求。与传统关系数据库相比，Neo4j 在处理多层级关联查询时具有明显优势，如查找特定概

念的相关资源时，可通过简洁的图查询语句一次性获取所有关联内容，减少了查询复杂度。

(7) 后端框架采用 Python 的 FastAPI 实现。FastAPI 基于现代 Python 特性开发，提供自动文档生成、请求验证等功能，简化了 API 开发流程。系统中大模型调用与知识图谱查询等核心功能，通过 FastAPI 的异步处理机制得以高效实现。后端架构设计遵循 REST 原则，清晰划分了资源路径与操作方法，使 API 接口具有良好的可读性与可维护性。FastAPI 的性能表现也满足了系统的并发处理需求。

(8) 智能问答功能基于阿里云通义千问大模型的"qwen-plus"模型实现。针对操作系统领域知识，设计了知识增强的提示策略：先从 Neo4j 图谱中检索相关概念节点，提取节点内容作为背景知识，与用户问题共同组成提示输入。这种方法有效弥补了大模型在专业领域知识的不足，提升了回答准确性。通过比较不同检索算法与提示模板的效果，最终确定了现有实现方案，为系统提供了可靠的智能问答能力。

2.6 数据库设计

本系统使用 MySQL 5.7 作为数据库系统，来完成有关数据的存储与操作。具体的数据库设计以数据表形式说明，详情如下：

(1) 用户信息表 用户信息表存储系统用户的基本信息，包含用户 ID、昵称、邮箱、密码、性别、管理员标识及头像信息等。该表是系统的核心表之一，其中用户 ID 作为主键，同时对邮箱和昵称建立了唯一索引以确保数据完整性。具体如表 2-1 所示。

表 2-1 用户信息表(user_info)

列名	中文名称	数据类型	长度	允许空
user_id	用户 ID	varchar	15	Not null
nick_name	昵称	varchar	20	Null
email	邮箱	varchar	150	Null
password	密码	varchar	50	Null
sex	性别	tinyint	1	Null
is_admin	管理员标识	int	11	Null
image	头像	varchar	255	Null

(2) 对话主表 对话主表用于存储系统中所有用户创建的对话会话信息，包含主键 ID、对话 ID、对话标题、用户 ID 及创建时间等字段。该表通过外键约

束与用户信息表关联，确保数据一致性。对话 ID 设置为唯一索引以提高查询效率，同时对用户 ID 建立索引便于按用户检索对话记录。具体如表 2-2 所示。

表 2-2 对话主表(master_chat)

列名	中文名称	数据类型	长度	允许空
id	信息序号	int	11	Not null
chat_id	对话标识符	varchar	50	Not null
title	对话标题	varchar	255	Null
user_id	用户 ID	varchar	15	Not null
created_time	创建时间	datetime	0	Not null

(3) 消息记录表 消息记录表存储系统中所有对话的详细内容，包括消息 ID、角色类型、消息内容、网络参考、知识图谱参考、相关主题、用户 ID、对话 ID 及创建时间等字段。该表通过外键约束与用户信息表关联，同时对对话 ID 和用户 ID 建立索引以优化查询性能。消息内容字段使用 text 类型以支持存储较长文本，而知识图谱参考信息则使用 longtext 类型以适应可能的大量数据。具体如表 2-3 所示。

表 2-3 消息记录表(master_message)

列名	中文名称	数据类型	长度	允许空
id	信息序号	int	11	Not null
role	角色类型	varchar	20	Not null
content	消息内容	text	0	Null
web_reference	网络参考	text	0	Null
gene_reference	知识图谱参考	longtext	0	Null
relevant_topics	相关主题	text	0	Null
user_id	用户 ID	varchar	15	Not null
chat_id	对话 ID	varchar	50	Not null
created_time	创建时间	datetime	0	Not null

数据库设计中，各表之间建立了合理的外键关联。master_chat 表和 master_message 表均通过 user_id 字段与 user_info 表关联，确保数据引用完整性。此外，master_message 表中的 chat_id 字段虽未设置外键约束指向 master_chat 表，但在应用层面保证了其引用的对话记录存在性。

为提升查询效率，在高频访问的字段上建立了适当索引。例如，在 master_chat 表的 chat_id 字段上建立唯一索引，在 user_id 字段上建立普通索引；在 master_message 表的 chat_id 和 user_id 字段上分别建立普通索引，以加速按对话或用户检索消息的操作。

3 基于大模型的操作系统知识图谱设计与实现

3.1 操作系统知识图谱系统设计

操作系统作为计算机科学的核心课程，其知识体系庞大而复杂。构建操作系统知识图谱不仅有助于梳理知识结构，还能为学生学习和教师教学提供直观的参考工具。本章将详细阐述基于大模型技术构建操作系统知识图谱的系统设计方案与实现过程。

3.1.1 知识图谱建模方法论

本研究采用自顶向下的构建方法，其优势在于可以先确定知识图谱的整体框架，再逐步填充具体内容，保证知识体系的完整性和结构化程度。这种方法论主要包含以下步骤：

- 1) 定义知识图谱的节点类型、关系类型及属性
- 2) 收集和预处理操作系统教材资料
- 3) 利用大模型进行实体抽取
- 4) 利用大模型进行实体属性抽取
- 5) 利用大模型进行实体关系抽取
- 6) 基于 Neo4j 构建知识图谱并可视化

为确保知识的全面性和权威性，本研究采用多元化的数据来源：操作系统专业教材如汤子瀛、哲凤屏、汤小丹编著的《计算机操作系统》（第四版）和 Andrew S. Tanenbaum 的《现代操作系统》（第 4 版）；网络知识库包括百度百科和维基百科中关于操作系统的条目以及 CSDN 等技术社区中的操作系统相关文档；此外还参考了 B 站上的操作系统教学视频系列。

3.1.2 知识图谱本体设计

本体设计是知识图谱构建的关键环节，它决定了图谱的结构和表达能力。基于操作系统领域的特点，我们系统设计了实体类型、关系类型和属性体系。

3.1.2.1 实体类型设计

根据操作系统课程的特点，设计了以下五种节点类型：

- 1) ChapterNode（章节节点）：表示教材中的主要章节，如"进程管理"、"内存管理"、"文件系统"等，用于构建知识图谱的顶层结构。
- 2) SubChapterNode（子章节节点）：表示教材中的二级或三级章节，如"进程调度算法"、"死锁检测"、"页面置换策略"等，用于构建知识图谱的中层结构。
- 3) ConceptNode（概念节点）：表示操作系统领域的核心概念、术语和技术，如"进程"、"虚拟内存"、"信号量"等，是知识图谱的基本构建单元。
- 4) TopicNode（题目节点）：表示与操作系统相关的习题和问题，便于学习巩固
- 5) VideoNode（视频节点）：表示 B 站上的操作系统教学视频，提供多媒体学习资源

这种分层设计既符合教材的组织结构，又满足了不同学习场景的需求，为知识图谱提供了清晰的骨架结构。

3.1.2.2 关系类型设计

基于节点之间的逻辑关联，设计了四种关系类型：

- 1) HAS_SUBCHAPTER：章节包含子章节的关系，体现知识的层次结构
- 2) HAS_CONCEPT：子章节包含特定概念的关系，表明知识点的归属
- 3) HAS_TOPIC：概念节点关联特定题目的关系，支持习题练习
- 4) HAS_VIDEO：概念关联教学视频的关系，提供多媒体学习途径

这些关系类型构成了操作系统知识的网络结构，能够清晰地表达不同知识点之间的联系，便于系统化学习和教学。

3.1.2.3 属性体系设计

为了丰富知识图谱的内容，为各类节点设计了以下属性：

- 1) ChapterNode 属性：chapter_name, description
- 2) SubChapterNode 属性：subchapter_name, description
- 3) ConceptNode 属性：concept_name, description, color

4) TopicNode 属性: topic_name, topic_description, topic_type, topic_answer, topic_answer_reason

5) VideoNode 属性: video_name, url

这些属性提供了节点的详细信息,使知识图谱不仅能表达知识之间的关系,还能包含丰富的知识内容,提高了图谱的应用价值。

3.2 基于大语言模型的知识图谱构建

大语言模型具有强大的文本理解和生成能力,本研究充分利用这一优势,将其应用于知识图谱构建的各个环节,显著提高了构建效率和质量。

3.2.1 基于大语言模型的教学资源实体识别与属性抽取

实体是知识图谱的基本组成单元,通过大语言模型可以高效识别和抽取操作系统领域的关键实体。为保证抽取质量,本研究设计了迭代式实体抽取方法,通过传递已抽取实体信息,保证命名一致性并避免重复抽取。

3.2.1.1 面向课本文本的实体抽取与分类

本研究以汤子瀛、哲凤屏、汤小丹编著的《计算机操作系统》(第四版)和 Andrew S. Tanenbaum 的《现代操作系统》(第 4 版)作为主要知识源,系统性地抽取操作系统领域的核心实体。实体抽取过程采用分批迭代的方式,确保实体命名的一致性和提取的完整性。

为确保实体抽取的质量和一致性,本研究采用了以下迭代抽取流程:

初始语料处理时,我们将两本教材内容按章节划分为适当大小的文本段落作为大语言模型的输入。随后设计结构化提示模板,明确指导模型识别特定类型的实体,包含任务描述、实体类型定义、抽取规则和输出格式。采用增量式抽取方法,每次处理完一段文本后,将新抽取的实体合并到已有实体列表中,并在下一轮抽取时传递给大语言模型。对抽取的实体进行去重处理和标准化,统一表述不一致但指代相同概念的实体,如将"虚存"和"虚拟内存"统一为"虚拟内存"。每轮抽取后,对结果进行人工审核,确保实体类型分配正确,并修正潜在错误。结构化提示模板如下表 3-1 所示。

表 3-1 实体提取 Prompt

请阅读以下关于操作系统的文本：'[文本内容]'

你是一位操作系统领域的专家，需要帮我从文本中抽取实体，用于构建操作系统知识图谱。

请识别并抽取以下类型的实体：

1. ChapterNode: 主要章节节点，如"进程管理"、"内存管理"等
2. SubChapterNode: 子章节节点，如"进程调度算法"、"死锁检测"等
3. ConceptNode: 核心概念节点，如"进程"、"虚拟内存"等

我已经从之前抽取了以下实体：

ChapterNode: [已抽取的章节实体列表]

SubChapterNode: [已抽取的子章节实体列表]

ConceptNode: [已抽取的概念实体列表]

请从新文本中抽取上述类型的实体，但需要注意：

1. 如果遇到与已有列表中相似但表述略有不同的实体，请使用已有列表中的表述保持一致性
2. 只输出新发现的、不在已有列表中的实体
3. 确保新抽取的实体与已有实体在概念层级上保持一致
4. 为每个实体提取相应的描述或定义字段

请以三元组形式输出结果：[(实体类型 1, 实体名称 1, 实体描述 1), (实体类型 2, 实体名称 2, 实体描述 2), ...]

其中实体描述应尽量简洁地概括该实体的定义、功能或特点，从原文中直接提取或适当归纳。

通过对两本教材的系统性分析，共抽取出 ChapterNode 9 个，SubChapterNode 78 个，ConceptNode 362 个，涵盖了操作系统领域的主要知识点。实体抽取过程中发现，大语言模型在识别明确的专业术语方面表现出色，但对于概念边界模糊的实体，如"资源分配"与"资源管理"这类相近概念的区分上存在一定挑战。通过迭代式抽取方法，我们有效地解决了这一问题，确保了实体命名的一致性。

表 3-2 实体抽取结果

节点类型	节点数量	节点示例
ChapterNode	9	进程与线程
SubChapterNode	78	进程调度方式
ConceptNode	362	抢占调度

实体抽取的质量直接影响知识图谱的构建效果。通过采用大语言模型辅助的迭代式抽取方法，本研究有效地从操作系统教材中提取了结构化的实体信息，为后续的关系抽取和知识图谱构建奠定了坚实基础。

3.2.1.2 面向题库资源的题目节点自动识别与构建

本研究针对操作系统题库资源，设计了自动化的题目节点识别与构建流程。为了构建高质量的操作系统知识图谱，题目节点的精确识别与结构化至关重要。我们采用基于大语言模型的方法，从不同来源的操作系统题库中提取结构化的题目信息。

为确保题目节点识别的准确性和一致性，本研究采用了以下工作流程：题库预处理阶段收集来自多种来源的操作系统题库，包括教材配套习题集、历年考试题、在线题库等，并进行格式标准化处理。随后设计针对题目节点识别的专用提示模板，引导大语言模型准确识别题目的各个组成部分，明确定义了需要提取的字段及其规范。最后使用设计好的提示模板，对预处理后的题目进行批量处理，自动化提取关键节点信息。

表 3-3 题目节点抽取 Prompt

<p>我正在构建操作系统题库知识图谱，需要从题目文本中提取题目的各个组成部分：</p> <p>## 输入文本</p> <p>请分析以下操作系统题目内容：'[题目文本]'</p> <p>## 提取要求</p> <p>请从题目文本中提取以下节点信息：</p> <ol style="list-style-type: none">1. topic_name: 题目名称/标题2. topic_type: 题目类型（如判断题 1、选择题 2、解答题 3 等，用数字表示）3. topic_description: 题目描述/题干
--

4. topic_answer: 题目答案（如 A、B、C、D 或其他正确答案）

5. topic_answer_reason: 答案解析/解释

提取规则

- 每个节点必须从文本中明确提取，不要增加不存在的内容
- 保留原文中的格式特征（如换行、标点等）
- 若某节点信息不存在，请标记为"未提供"
- 确保提取的内容完整、准确，特别是答案和解析部分

输出格式

请以 JSON 对象形式返回结果：

```
{ "topic_name": "提取的题目名称", "topic_type": "提取的题目类型",  
  "topic_description": "提取的题目描述", "topic_answer": "提取的题目答案",  
  "topic_answer_reason": "提取的答案解析" }
```

通过对收集的操作系统题库资源进行系统化处理，共提取并构建了 424 个结构化题目节点。

题目节点的类型分布情况如表 3-4，题目示例如表 3-5。

表 3-4 题目抽取结果

题目类型	数量	占比
选择题	189	44.5%
判断题	111	26.2%
简答题	124	29.3%

表 3-5 抽取题目示例

topic_answer	A
topic_answer_reason	程序是静态的，它是存储在外部存储设备中的一组指令或代码，而进程是程序的执行实例，它在系统运行时由操作系统创建并分配资源。程序是一个静态的、持久的对象，而进程是一个动态的、暂时的执行状态。因此，A（程序是动态的，进程是静态的）是错误的。其余选项均为正确，程序是永久存在的，而进程是由操作系统创建和销毁的临时实体；一个程序可以对应多个进程，尤其在多任务操作系统中，一个程序可以同时由多个进程并发执行；而一个进程不可能同时包含多个程序，因为一

	个进程通常执行一个程序的代码。
topic_description	下列关于程序和进程的说法中，错误的是： A、程序是动态的，进程是静态的 B、程序是永久的，进程是暂时的 C、一个程序可对应多个进程 D、一个进程可包括多个程序
topic_name	题目 4:进程管理
topic_type	2

3.2.2 基于大语言模型的知识关系提取与验证

基于已抽取的 ChapterNode、SubChapterNode 和 ConceptNode，本节将详细阐述如何同时提取两类核心关系：章节间的层次关系(HAS_SUBCHAPTER)和章节与概念的从属关系(HAS_CONCEPT)。

3.2.2.1 章节体系内部层级关系的构建与验证

为同时高效提取两类关系，我们设计了一个综合性关系提取框架。输入层包含三部分核心输入：已抽取的实体集合(ChapterNode、SubChapterNode、ConceptNode)、操作系统教材原文文本片段以及关系定义与提取规则。输出层则生成结构化的关系三元组，包括 HAS_SUBCHAPTER 关系集合、HAS_CONCEPT 关系集合和关系置信度评分。

本研究采用"文本增强的关系提取"方法，通过引入原始教材文本片段，提高关系识别的准确性。首先进行文本分段与映射，将操作系统教材按章节分割为文本片段，与实体建立映射关系。然后进行关系提取上下文构建，为每次关系提取任务动态选择相关文本片段作为上下文。接着实施融合式提取，同时向大语言模型提供实体列表和文本上下文，提取两类关系。最后采用递进验证，利用已提取的 HAS_SUBCHAPTER 关系辅助验证 HAS_CONCEPT 关系。具体提取两类关系的提示模板如表 3-6 所示。

表 3-6 关系抽取 Prompt 示例

请作为操作系统领域专家，基于以下信息提取知识图谱关系：

【已抽取的实体】

ChapterNode: [章节节点列表, 如"操作系统概述", "进程管理", "内存管理"等]

SubChapterNode: [子章节节点列表, 如"进程概念", "进程调度", "虚拟内存"等]

ConceptNode: [概念节点列表, 如"进程控制块", "时间片轮转算法", "页面置换"等]

【操作系统知识文本】

[相关操作系统教材文本片段]

请同时提取以下两类关系:

1. HAS_SUBCHAPTER: 表示章节包含子章节的关系, 反映知识的层次结构

2. HAS_CONCEPT: 表示子章节包含特定概念的关系, 表明知识点的归属

输出格式:

HAS_SUBCHAPTER 关系: [(ChapterNode, HAS_SUBCHAPTER, SubChapterNode), 置信度]

HAS_CONCEPT 关系: [(SubChapterNode, HAS_CONCEPT, ConceptNode)]

注意事项:

1. 充分利用提供的文本内容分析实体间的关系
2. 当某概念可能属于多个子章节时, 请列出所有可能的关系

最后成功提取 78 个 HAS_SUBCHAPTER 关系, 同时识别 362 个 HAS_CONCEPT 关系, 将这 362 个 ConceptNode 归类到适当的 SubChapterNode 中, 实现了知识体系的完整结构化。

3.2.3 操作系统知识图谱的质量评估与优化

知识图谱的质量直接影响其应用效果, 本研究设计了全面的质量评估体系, 包括定量与定性评估方法, 并基于评估结果进行多轮优化。

本研究设计了两层评估框架, 包括实体层评估、关系层评估。

实体层评估包含三个方面: 实体覆盖度, 评估知识图谱是否包含了操作系统领域的核心实体; 实体正确性, 评估抽取的实体是否准确描述了操作系统概念; 命名一致性, 评估相同实体在不同上下文中是否保持一致命名。

关系层评估同样关注三个维度：关系正确性，评估抽取的关系是否符合操作系统领域知识；关系完整性，评估重要实体间的关系是否被完整捕获；关系一致性，评估相似概念间的关系模式是否一致。

为了进行客观评估，本研究构建了操作系统知识图谱的标准数据集，数据来源从《计算机操作系统》（第四版）手动抽取核心概念和关系，规模包含 100 个核心概念节点、30 个章节/子章节节点和 135 个关系，通过多轮交叉验证确保数据准确性，并使用 Qwen2.5-Max、GPT-4o 和 DeepSeek-R1 三个模型进行了评估分析，重点对实体抽取进行了结果评估。评估结果如图 3-7 和图 3-8。

表 3-7 实体抽取评估结果

模型	Precision(%)	Recall(%)	F1(%)	命名一致性(%)
GPT-4o	83.52	80.16	81.81	89.35
Qwen2.5-Max	81.24	78.93	80.07	85.67
DeepSeek-R1	79.47	76.83	78.13	84.51

表 3-8 关系抽取评估结果

模型	Precision(%)	Recall(%)	F1 (%)	关系一致性(%)
GPT-4o	73.45	85.62	79.06	84.21
Qwen2.5-Max	70.16	83.47	76.25	81.93
DeepSeek-R1	68.24	81.53	74.32	80.12

结果分析表明，GPT-4o 在所有评估维度上表现最佳，特别是在实体命名一致性和关系抽取的召回率方面具有显著优势。Qwen2.5-Max 表现次之，但与 GPT-4o 的差距相对较小。DeepSeek-R1 虽然整体指标略低于前两者，但其性能与 Qwen2.5-Max 相近，特别是在实体识别的精确度和关系一致性方面表现出了强劲的能力。

鉴于 GPT-4o 在各项评估指标中均表现出最佳性能，本研究最终采用 GPT-4o 作为主要知识抽取工具完成了操作系统知识图谱的构建。GPT-4o 不仅在实体识别准确性和关系抽取完整性方面表现优异，其生成的结果通过多轮专家验证也证

实了较高的可信度和一致性。此外，GPT-4o 在处理专业领域知识时展现出的理解深度和推理能力，为构建高质量的操作系统教学知识图谱提供了坚实保障。最终构建的知识图谱不仅保持了操作系统知识的学科内在逻辑，还通过合理的层次结构和关联关系，为后续的教学应用和知识推理奠定了基础。

最终构建的操作系统知识图谱利用 Neo4j 数据库存储，图谱示例如下：

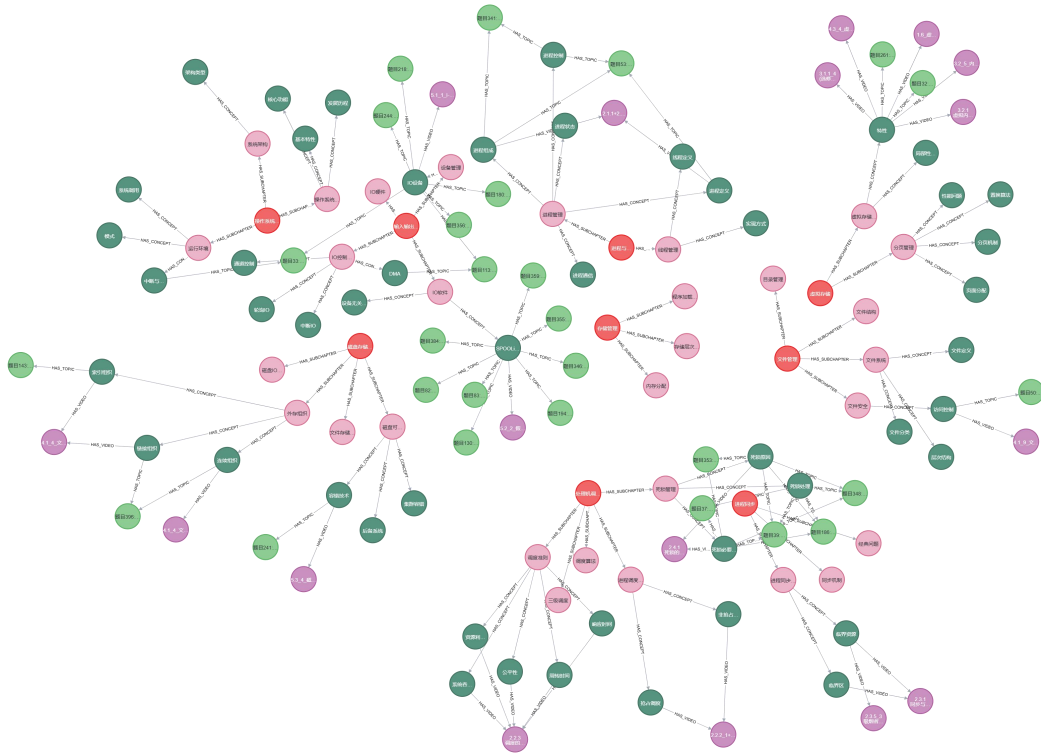


图 3-1 操作系统知识图谱示例

3.3 本章小结

通过基于大模型的知识图谱构建方法，结合严谨的评估与优化流程，本研究成功构建了高质量的操作系统知识图谱。该图谱具有结构清晰、连通性高、领域覆盖度广等特点，为操作系统教学提供了有效的辅助工具，可用于章节概念关联分析、概念视频关联和题目解析辅助等多种教学场景。实验结果表明，大模型在实体、关系抽取任务中表现优异，特别是采用迭代式抽取策略并结合专家指导的方法，能够显著提高知识图谱的质量和实用价值。

4 操作系统知识图谱智能问答系统的设计与实现

本章将详细介绍操作系统知识图谱智能问答系统的设计与实现。该系统通过融合知识图谱检索、互联网搜索与大模型问答能力，构建了一个能够提供准确、专业且可溯源的操作系统领域问答平台。

4.1 知识增强问答算法设计与实现

知识增强问答是本系统的核心，它通过多种知识源的有机结合，弥补了单一知识来源的局限性，提升了回答的专业性与准确性。本节将详细阐述知识增强问答算法的设计思路与实现方法。

4.1.1 知识图谱检索算法实现

操作系统知识图谱作为系统的核心知识库，承载了操作系统领域的结构化知识。为了有效利用这一资源，我们设计了一套多层次、高效率的知识图谱检索算法。

在知识图谱检索的入口函数中，首先通过向量相似度计算找出与用户问题最相关的概念节点，然后基于这些节点展开更广泛的知识图谱检索，最终提取语料作为回答依据。核心算法流程如表 4-1 所示。

表 4-1 查找相关概念节点代码

```
async def ragFromNeo4j(user_question: str):  
    # 首先通过计算向量相似度来识别相关的概念节点  
    relevant_concept_nodes = await  
rerank_and_extract_conceptNode(user_question)  
    # 基于已识别的概念节点检索知识图谱数据  
    relevant_nodes_links = await  
retrieve_knowledge_graph(relevant_concept_nodes)  
    # 从检索到的节点中提取语料文本  
    corpus_text = [node["value"] for node in relevant_nodes_links["nodes"]]  
    combined_corpus = "".join(corpus_text)
```




```
# 记录检索到的信息用于调试

logger.info(f"Corpus retrieved from knowledge graph:
{combined_corpus}\n")

logger.info(f"Retrieved nodes and links information:
{relevant_nodes_links}")

# 返回语料文本和结构化图谱数据

return combined_corpus, relevant_nodes_links
```

在向量相似度计算环节，系统利用预训练的向量模型，将用户问题与知识点描述转化为高维向量，通过余弦相似度等指标衡量它们之间的语义关联程度。这种方法能够捕捉问题与知识点之间的深层语义联系，而不仅限于表面的关键词匹配。为了确保检索质量，系统设置了相似度阈值，只有超过阈值的概念节点才会被纳入后续检索范围，文本向量化与相似度计算流程如图：

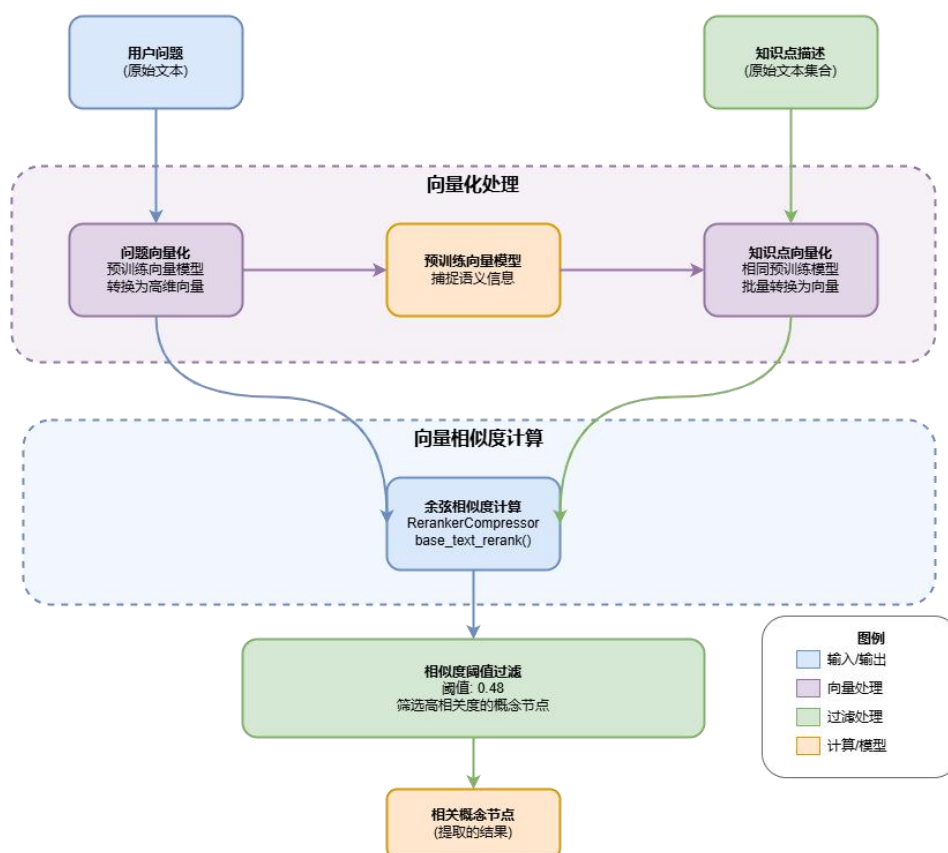
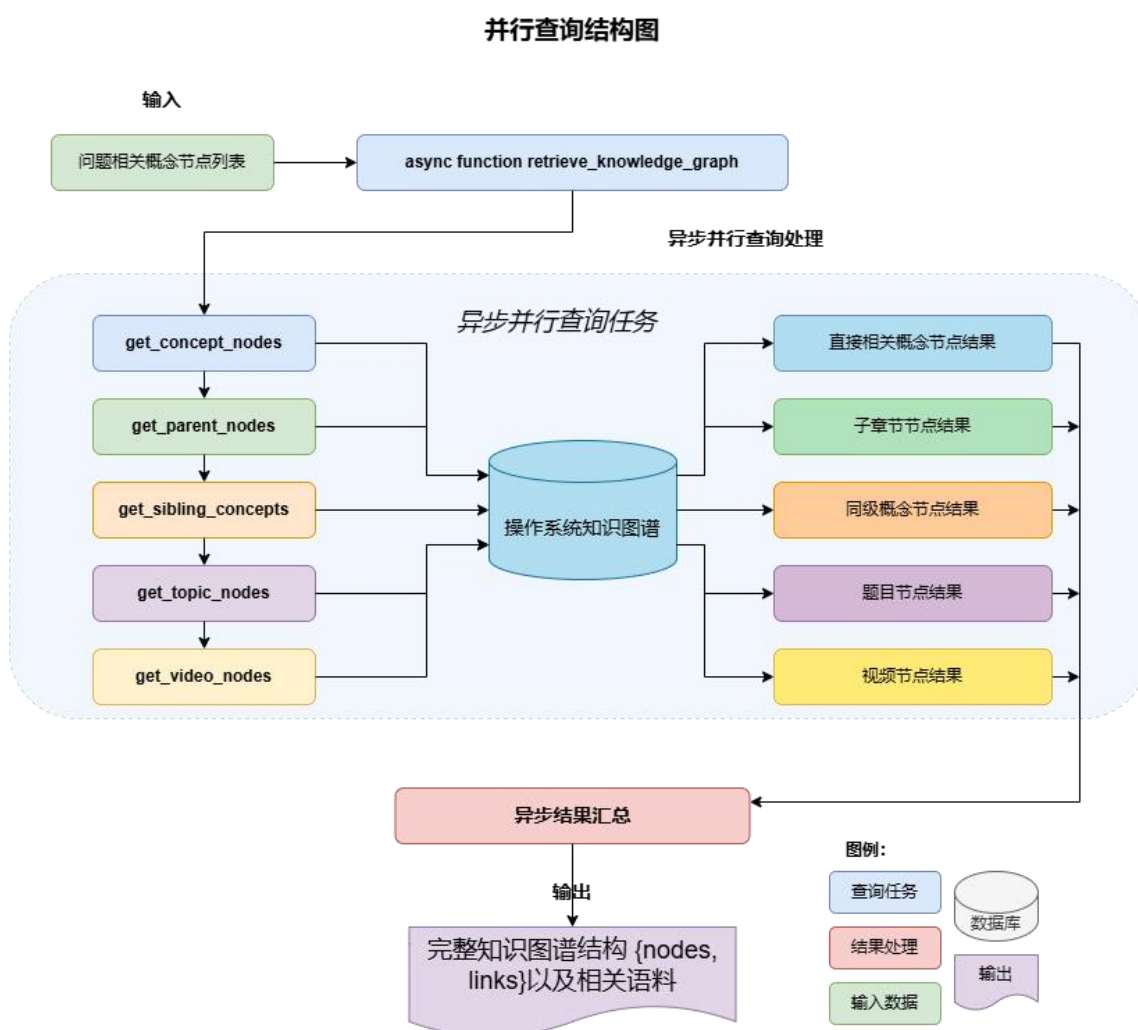


图 4-1 文本向量化与相似度计算流程图

获取相关概念节点后，系统开始执行多维度的知识图谱检索。与传统的单一路径检索不同，本系统采用了并行多路径检索策略，同时从概念节点出发，查询其父节点（知识章节）、同级概念、相关主题以及视频资源。这种全方位的检索方式能够构建一个更加完整的知识网络，为后续的问答提供丰富的背景信息和专业知识。

在知识图谱检索的实现中，充分利用了异步并行处理技术，同时执行多个图数据库查询任务：



这种异步并行设计不仅提高了系统响应速度，还使复杂的图结构检索变得高效可行。在处理节点数据时，系统为每个节点分配唯一标识符，避免重复添加，并根据节点类型进行分类标记，构建起完整的知识网络结构。

知识图谱检索算法的核心价值在于，它能够从结构化的知识库中提取出与用户问题最相关的知识片段，并保留知识间的关联关系，为大模型提供高质量的背景信息，从而生成更加准确、专业的回答。

4.1.2 Bing Search API 集成与互联网知识获取

尽管知识图谱包含了大量专业知识，但它难以覆盖所有最新信息和边缘案例。为此，将互联网搜索作为知识图谱的重要补充，通过集成 Bing Search API 实现了对互联网知识的实时获取与处理。

互联网知识获取模块采用异步设计模式，避免网络请求延迟对系统整体响应速度的影响。其相关代码如表 4-2 所示。

表 4-2 获取互联网知识代码

```
async def search_bing(query_term: str, max_results: int = 6):  
    try:  
        # 准备 API 请求参数  
        subscription_key = settings.BING_SEARCH_API_KEY  
        endpoint = "https://api.bing.microsoft.com/v7.0/search"  
        headers = {"Ocp-Apim-Subscription-Key": subscription_key}  
        params = {"q": query_term, "count": max_results, "textFormat":  
"HTML"}  
  
        # 发送请求  
        async with aiohttp.ClientSession() as session:  
            async with session.get(endpoint, headers=headers,  
params=params) as response:  
                if response.status != 200:  
                    logger.error(f"Bing 搜索错误: {response.status}")
```



```
        return {"contents": "", "results": []}

    # 处理结果
    search_results = await response.json()
    results = []
    contents = ""

    # 提取结果内容
    for result in search_results.get('webPages', {}).get('value', []):
        results.append({
            'title': result.get('name', ""),
            'url': result.get('url', ""),
            'content': result.get('snippet', ""),
            'score': result.get('relevanceScore', 0)
        })
        contents += f'{result.get('name',
    "")}\n{result.get('snippet', "")}\n\n"

    return {"contents": contents.strip(), "results": results}

except Exception as e:
    logger.error(f'Bing 搜索错误: {str(e)}')
    return {"contents": "", "results": []}
```

在搜索结果处理环节，系统不仅提取了每个结果的核心内容，还保留了原始的结构信息，便于后续在回答中进行准确引用。这种设计使系统能够在回答中明确标注信息来源，增强了回答的可信度和透明度。

互联网知识获取模块还实现了完善的错误处理机制，对非正常响应状态、网络异常、解析错误等情况进行了妥善处理，确保即使在搜索失败的情况下，系统仍能继续运行并提供基于其他知识源的回答。

4.1.3 大模型问答增强机制

系统的核心问答能力依赖于大模型与知识增强的有机结合。设计了一套灵活的知识融合与提示词生成机制，使大模型能够充分利用多源知识，生成高质量的专业回答。

根据用户选择的知识来源，系统动态生成不同的提示词模板。相关代码如表 4-3。

表 4-3 多源问答代码

```
def generate_prompt(search_options, user_question, internet_info,
knowledge_graph_info):
    # 从搜索选项中获取知识来源偏好
    knowledge_graph_option = search_options.get("knowledgeGraph", 0)
    internet_option = search_options.get("internet", 0)

    # 组合设置创建选项键
    option_key = f'{knowledge_graph_option}{internet_option}'

    # 根据知识来源选择适当模板
    # "00": 仅使用模型知识
    # "01": 互联网增强模板
    # "10": 知识图谱增强模板
    # "11": 综合增强模板
    prompt_templates = {
        "00": base_template,
        "01": internet_template,
        "10": knowledge_graph_template,
        "11": combined_template
    }

    # 获取当前选项的模板
```

```
selected_template = prompt_templates.get(option_key,
prompt_templates["00"])

# 如启用互联网搜索则准备引用指令
citation_text = citation_instruction if internet_option == 1 else ""

# 用实际数据填充模板
filled_prompt = selected_template.format(
    question=user_question,
    internet_info=internet_info,
    knowledge_graph=knowledge_graph_info,
    citation_instruction=citation_text
)

return filled_prompt
```

针对不同的知识来源组合，精心设计了四种提示词模板：基础模板仅依靠大模型自身知识；互联网增强模板融合搜索结果并规范引用格式；知识图谱增强模板引入结构化专业知识；全面增强模板则同时整合两种外部知识源。每个模板都包含了专业领域界定、回答指南和具体的知识融合指示，确保大模型能够理解任务要求并生成符合预期的回答。当同时使用互联网检索与知识图谱检索时相关的 Prompt 如表 4-4 所示。

表 4-4 问答 Prompt

```
"11": """"# 操作系统专家助手指令
作为操作系统领域的专业 AI 助手，你的任务是提供高质量、深度技术解答。
## 知识来源
### 互联网实时数据
{internet_info}
### 专业知识图谱
{knowledge_graph}
{citation_instruction}
```

```
## 用户查询
```

```
{question}
```

```
## 回答要求
```

1. 精确性：确保技术内容准确无误
2. 专业深度：展示操作系统领域的专业洞见
3. 引用规范：仅在直接使用互联网信息时添加[citation:n]标记
4. 知识整合：有效融合互联网数据与知识图谱结构化信息
5. 清晰度：使用专业却易懂的语言解释复杂概念
6. 引用审慎：宁可不引用也不误引用

```
请始终保持操作系统专家的专业水准，确保回答既有技术深度又易于理解。
```

```
""""
```

在提示词中，特别设计了引用规范，要求大模型使用标准格式（如[citation:n]）标注互联网信息来源，并明确引用内容与来源的对应关系。这种设计不仅增强了回答的可信度，还使用户能够方便地追溯信息来源，验证回答的准确性。

通过这种动态的提示词生成机制，系统能够根据不同问题类型和用户需求，选择最适合的知识增强策略，在保证回答质量的同时，提供灵活多样的知识服务。

4.1.4 可溯源问答结果生成

在传统问答系统中，用户往往只能看到最终的回答结果，无法了解回答的依据和推理过程。为了增强系统的透明度和可信度，实现了可溯源的问答结果生成机制。

可溯源机制的核心是将检索到的知识与最终回答关联起来，使用户能够追根溯源。具体来说，系统通过两种方式实现可溯源：一是将检索到的知识图谱节点以可视化方式呈现给用户；二是在回答中明确标注互联网信息来源并提供原始链接。

知识图谱可视化展示采用了图形化界面，将检索到的概念节点、章节节点、主题节点和视频节点及其关系以直观的方式呈现。不同类型的节点使用不同的颜色和形状表示，关系则以连线方式展示。这种可视化方式不仅使用户能够一目了然

然地了解问题相关的知识结构，还能够通过交互式探索深入了解感兴趣的知识点。

对于互联网信息，系统在回答中使用标准的引用标记，并在引用列表中提供完整的 URL 地址，使用户可以直接访问原始信息来源，获取更详细的内容。这种设计既保证了回答的简洁性，又为用户提供了深入了解的途径。

系统还实现了交互式溯源机制，用户可以通过点击引用标记查看具体引用内容，通过知识图谱可视化界面探索相关知识点，查看与问题相关的视频资源链接，或访问原始网页获取更详细的信息。这种多层次、多角度的溯源机制，大大增强了系统回答的透明度和可信度。

4.2 系统功能实现

4.2.1 问答功能

本系统的核心问答功能设计了多种检索增强方式，旨在为用户提供全面且准确的学习辅助。如图 4-3 所示，在问答页面中，用户可根据实际需求选择不同的检索功能来提升回答质量。知识图谱检索作为系统的基础功能，能够从已构建的专业知识库中提取相关信息，确保回答的专业性与精确度。当用户需要了解最新研究进展或补充资料时，可启用互联网检索功能，系统会通过 Bing 搜索 API 获取网络上的最新资源，使回答内容更加丰富且与时俱进。为了进一步支持用户的学习过程，系统还提供学习资料检索功能，能够为用户匹配相关的练习题和教学视频，形成完整的知识学习闭环。

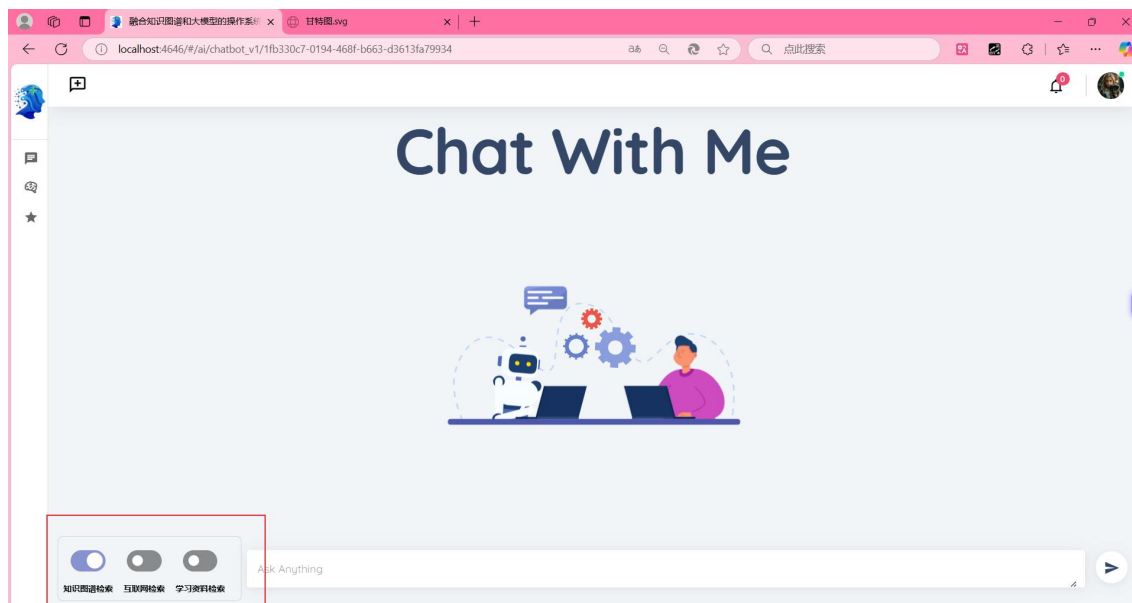


图 4-3 核心问答界面

如图 4-4 所示，系统采用 WebSocket 技术实现了回答内容的流式传输，使生成的回答能够以 Markdown 格式实时呈现在页面上，大幅提升了用户的交互体验。这种即时反馈的机制让用户无需等待完整答案生成，可以在内容逐步显示的过程中及时获取信息，有效减少了等待时间，提高了系统的响应性能。同时，页面设计考虑了信息层次展示，在系统生成回答后，用户可通过界面底部的功能按钮进行深入探索，包括查看详细的知识图谱可视化、相关练习题等。页面还贴心地设置了视频参考链接和互联网信息来源，用户只需点击即可跳转至原始资料，便于进行更深入的学习与研究。

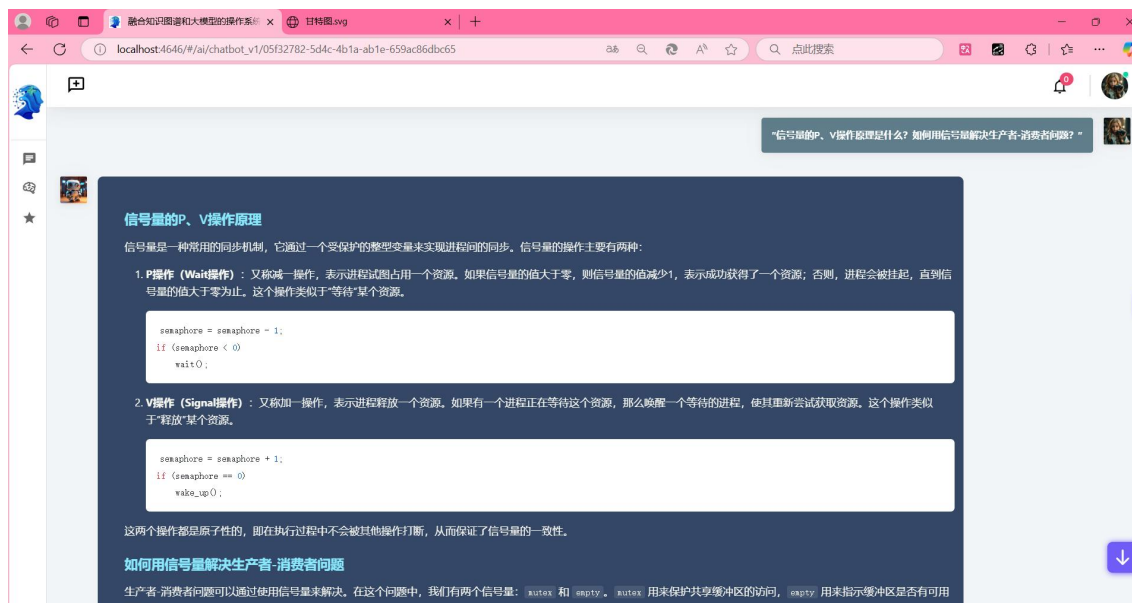


图 4-4 系统回答界面

4.2.2 查看聊天记录功能

如图 4-5 所示，系统实现了完整的聊天记录存储与查看功能，为用户的学习过程提供了连续性支持。所有用户与系统之间的对话内容均被完整记录并存储在 MySQL 数据库中，采用合理的数据结构设计确保了信息的可靠保存与高效检索。这种设计使用户可以随时回顾过去的问题与答案，不仅有助于知识的巩固与复习，也为学习过程提供了完整的追溯能力。

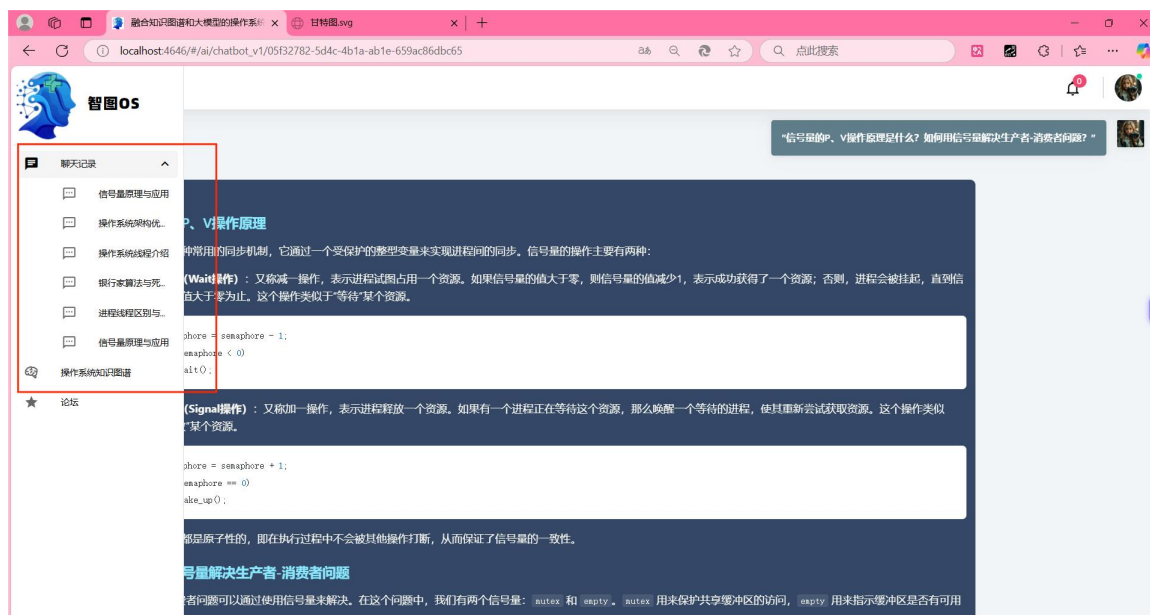


图 4-5 聊天历史界面

4.2.3 知识溯源功能

知识溯源功能是本系统的重要特色，它通过可视化技术使抽象的知识关联变得直观可见。如图 4-6 和图 4-7 所示，当用户点击问答界面中的"知识图谱可视化"按钮后，系统会自动提取与当前问题相关的知识节点，并以图形化方式清晰展示在页面中。这种展示方式使用户能够直观理解知识之间的结构关系，实现了回答内容的可溯源性，大幅提升了学习效果。

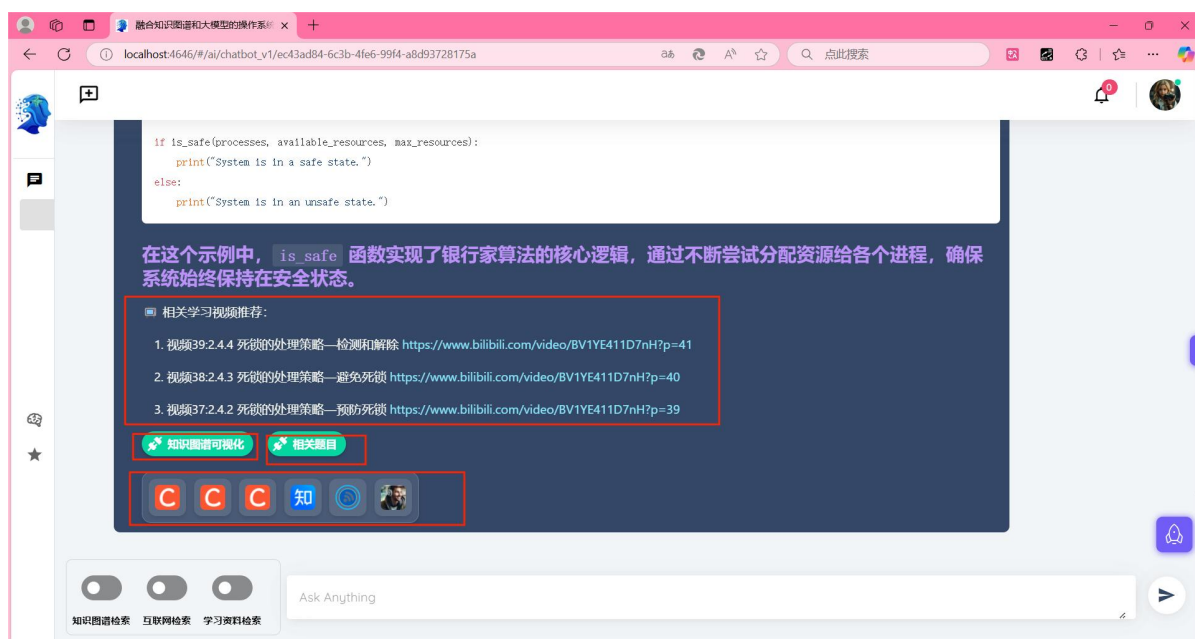


图 4-6 资源溯源界面

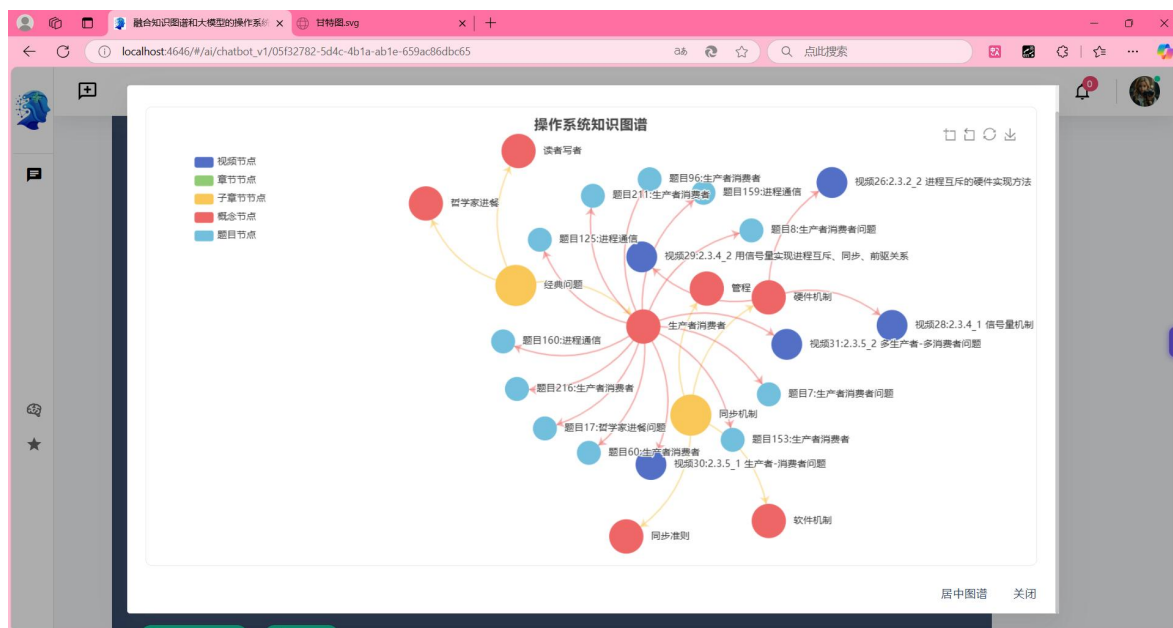


图 4-7 知识图谱溯源界面

如图 4-8 所示,系统的"相关题目"功能进一步增强了知识应用环节,为用户 提供与当前问题密切相关的练习题,帮助用户巩固所学知识。这些练习题经过精 心筛选,覆盖了不同难度和知识点,满足用户多层次的学习需求。特别值得一提 的是,系统支持将这些练习题一键下载为 Word 文件的功能,极大地方便了用户 进行离线学习与复习。通过这种知识溯源与练习相结合的方式,系统不仅告诉用 户"是什么",还引导用户理解"为什么",形成了完整的知识学习闭环。



图 4-8 题目推荐界面

4.2.4 操作系统知识图谱可视化功能

操作系统知识图谱可视化是本系统的第二个核心功能,它为用户提供了一种 全新的知识导航方式。在此功能模块中,系统将操作系统的专业知识体系化地呈 现给用户,如图 4-9 所示,用户可以根据章节和子章节进行筛选,灵活查看不同 粒度的知识结构。这种可视化展示方式帮助用户全面把握操作系统学科的整体知 识框架,理解各知识点之间的内在联系,从而构建起系统性的知识网络。



图 4-9 操作系统知识图谱可视化界面

知识图谱界面的设计注重交互性与实用性，如图 4-10 和图 4-11 所示，用户可以通过拖拽、缩放等操作自由探索知识空间。特别是双击节点查看详细信息的功能，让用户能够在宏观把握知识结构的同时，便捷地深入了解具体知识点内容。系统还智能地标识了重点和难点知识，引导用户进行有针对性的学习。这种兼具宏观与微观视角的知识展示方式，极大地提升了学习的系统性和效率，使操作系统这门复杂的课程变得更加清晰易懂，有效克服了传统线性学习方式的局限性。



图 4-10 操作系统知识图谱查看详细节点界面

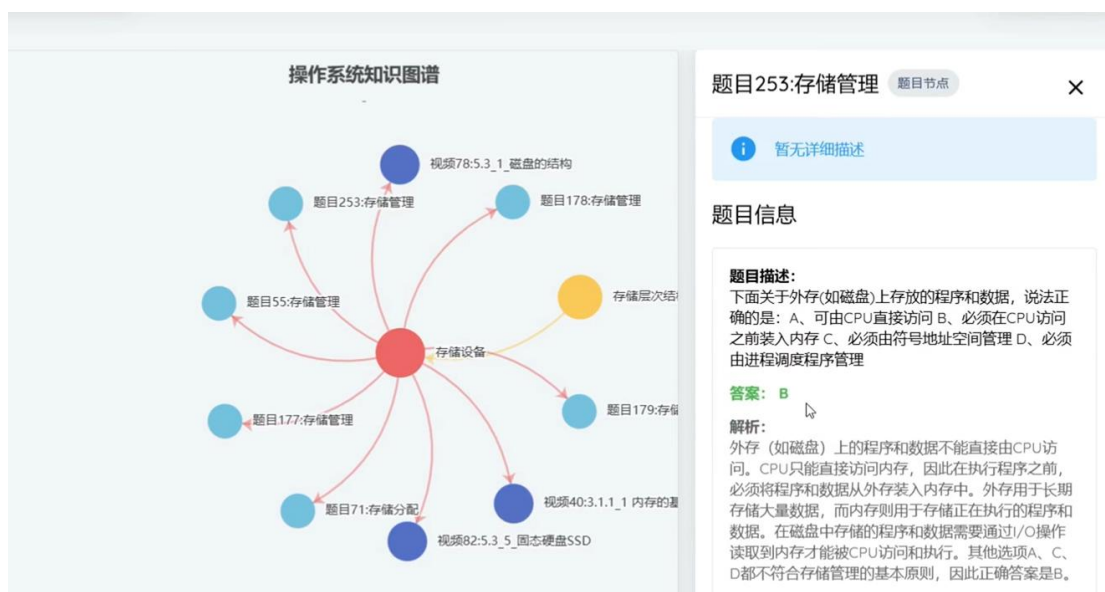


图 4-11 操作系统知识图谱查看详细节点界面

4.3 本章小结

本章详细介绍了操作系统知识图谱智能问答系统的设计与实现过程。该系统通过融合知识图谱检索、互联网搜索与大模型问答能力,实现了一个能够提供准确、专业且可溯源的操作系统领域问答平台。

系统的核心是知识增强问答算法,它包含三个关键组件:知识图谱检索算法、Bing Search API 集成的互联网知识获取机制以及大模型问答增强机制。知识图谱检索算法采用多层次的检索策略,通过向量相似度计算找出相关概念节点。互联网知识获取模块作为知识图谱的补充,提供了获取最新信息的能力。大模型问答增强机制则通过动态生成的提示词模板,实现了多源知识的融合与专业回答的生成。

系统实现了四项核心功能:问答功能、查看聊天记录功能、知识溯源功能以及操作系统知识图谱可视化功能。问答功能支持多种检索增强方式,实现了回答内容的流式传输。知识溯源功能使抽象的知识关联变得直观可见,还提供了相关练习题功能。操作系统知识图谱可视化功能为用户提供了一种新型的知识导航方式。

5 总结与展望

本文详细地描述了融合知识图谱与大模型的操作系统学习平台从设计到实现的过程。下面是对本文工作的一个总结：

融合知识图谱与大模型的操作系统学习平台运用了模块化的设计思想，通过将这个系统划分为知识图谱构建与管理模块、智能问答模块、学习资源整合模块三大部分，实现了分模块的开发设计。系统采用前后端分离的三层架构体系，包括前端展示层、后端服务层和数据存储层。前端展示层采用 Vue3 框架开发，负责知识图谱可视化展示、智能问答交互界面和学习资源展示等功能；后端服务层基于 FastAPI 框架构建，处理知识图谱检索、大模型调用和资源推荐等核心功能；数据存储层由 Neo4j 图数据库和 MySQL 关系型数据库组成，分别存储操作系统知识图谱和用户信息、聊天历史等数据。

从功能架构上看，系统分为四大模块：用户账户模块负责管理用户的登录、注册和个人资料；智能问答模块提供基于大模型的知识增强问答服务；知识图谱模块展示操作系统知识的结构化内容；学习资源模块整合和推荐相关学习材料。通过知识结构化表达、知识增强智能问答、学习资源整合等核心功能，解决了传统操作系统教学中存在的知识点碎片化、内容抽象难懂、个性化学习支持不足等问题，为学生提供了智能化、个性化的学习环境。

由于时间与技术条件所限，系统仍有许多能够改善的地方。知识图谱方面可以扩充覆盖范围，引入更复杂的语义关系；智能问答能力可以引入多模态大模型，支持图像、代码等多种形式的问答交互，优化知识检索算法；学习功能方面可以增加个性化学习路径规划与推荐、协作学习功能和学习评估与反馈系统；技术层面可以优化大模型调用效率和知识图谱查询性能，增强系统并发处理能力。此外，平台模式可以扩展到其他学科领域，打造更加通用的智能学习平台。这些都是未来系统可以改进和扩展的方向。

参考文献

- [1] 郑永和, 王一岩. 教育与信息科技交叉研究: 现状、问题与趋势[J]. 中国电化教育, 2021, (07): 97-106.
- [2] Kasneci E, Sessler K, Küchemann S, et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn Individ Differ*, 2023, 103: 102274.
- [3] 王芳, 孔维琦, 齐浩哲, 等. 基于知识图谱的 C++ 程序设计知识体系构建[J]. 计算机教育, 2024, (12): 194-199. DOI: 10.16512/j.cnki.jsjy.2024.12.038.
- [4] Zhong L, Wu J, Li Q, et al. A comprehensive survey on automatic knowledge graph construction[J]. *ACM Computing Surveys*, 2023, 56(4): 1-62.
- [5] Liang X, Wang Z, Li M, et al. A survey of LLM-augmented knowledge graph construction and application in complex product design[J]. *Procedia CIRP*, 2024, 128: 870-875.
- [6] Su Z, Hao M, Zhang Q, et al. Automatic knowledge graph construction based on relational data of power terminal equipment[C]//2020 5th International Conference on Computer and Communication Systems (ICCCS). IEEE, 2020: 761-765.
- [7] Meyer L P, Stadler C, Frey J, et al. Llm-assisted knowledge graph engineering: Experiments with chatgpt[C]//Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow. Wiesbaden: Springer Fachmedien Wiesbaden, 2023: 103-115.
- [8] Zhu Y, Wang X, Chen J, et al. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities[J]. *World Wide Web*, 2024, 27(5): 58.
- [9] Lin J, Zhao Y, Huang W, et al. Domain knowledge graph-based research progress of knowledge representation[J]. *Neural Computing and Applications*, 2021, 33: 681-690.
- [10] Chen X, Jia S, Xiang Y. A review: Knowledge reasoning over knowledge graph[J]. *Expert systems with applications*, 2020, 141: 112948.
- [11] Abu-Salih B, Alotaibi S. A systematic literature review of knowledge graph construction and application in education[J]. *Heliyon*, 2024.
- [12] Trajanoska M, Stojanov R, Trajanov D. Enhancing knowledge graph construction using large language models[J]. *arXiv preprint arXiv:2305.04676*, 2023.
- [13] Sen P, Mavadia S, Saffari A. Knowledge graph-augmented language models for complex question answering[C]//Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE). 2023: 1-8.
- [14] Guo Q, Cao S, Yi Z. A medical question answering system using large language models and knowledge graphs[J]. *International Journal of Intelligent Systems*, 2022, 37(11): 8548-8564.

- [15]Zhang Y, Hao Y. Traditional Chinese medicine knowledge graph construction based on large language models[J]. Electronics, 2024, 13(7): 1395.
- [16]Baek J, Aji A F, Saffari A. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering[J]. arXiv preprint arXiv:2306.04136, 2023.
- [17]Jiang J H A. Knowledge Graph of Thoughts: An LLM Using a Knowledge Graph to Reason[D]. ETH Zurich, 2024.
- [18]刘敏. 基于深度学习的多源信息融合知识图谱智能化构建技术[J/OL]. 自动化技术与应用, 1-5[2024-12-27]. <http://kns.cnki.net/kcms/detail/23.1474.TP.20241223.1322.066.html>.

致谢

当落笔至此时，代表着我的大学生涯即将告一段落。以后的生活仍然充满着未知和挑战，但不论如何，我都十分感谢陪我一起走过这一段人生路的人们。

首先我要向我的导师王璿老师表示感谢。在本科期间我曾选修过王老师的操作系统课程，在学习的过程中多次遇到问题，每次向王老师请教，王老师总是和蔼可亲、不厌其烦地向我伸出援手。后来在毕业设计时又一次有幸得到王老师的指导，毕业设计的内容对我来说还是颇有挑战性，很多知识是本科时未曾深究的，这也导致在进行毕业设计的过程中我多次遇到了难以解决的问题，而王老师仍然是我在代码的汪洋里迷惘时的一盏明灯，无论我何时向老师请教，老师总会向我提出宝贵的意见。王老师事务繁多却对我的设计定期探问，这使我非常感动，也更加敬重王老师。

然后，我要向我的辅导员刘欣怡老师致谢。在本科期间，刘老师无数次地为同学们费心指导，无论在生活上还是学习上我认为刘老师都是一位负责又认真的好导师。在毕业设计期间，面对各种挑战与压力，刘老师始终给予我充分的理解与支持，为我提供了许多学业与择业方面的指导，我十分感谢。

同时我要向我的室友们致谢。四年时光转瞬即逝，感谢这一路的陪伴，使我校园中的日日夜夜都不孤单。虽然专业不同，但在一起度过的快乐回忆不会更改，祝愿 3023 的各位都有美好的前程！特别感谢在我完成毕业设计期间给予我帮助与鼓励的室友们，是你们的支持让我能够克服困难，顺利完成这一重要任务。

最后，我要向我的家人致以最深沉的谢意。感谢你们在我人生的二十余年中为我遮风挡雨，才使我能健康快乐地成长至今。大学毕业在即，希望我能早些为你们减轻负担，让你们更加快乐地生活。

在此，祝愿大家一生和乐安康，幸福美满！