**中图分类号：TN929**

论文编号：10006ZY1206216



硕士学位论文

**面向能力分析的教学平台设计与实现**

作者姓名 高磊磊

学科专业 计算机系统结构

指导教师 高小鹏教授

培养院系 计算机学院

**Design and Implementation of Course Management System Oriented to Ability Analysis**

**A Dissertation Submitted for the Degree of Master**

**Candidate： Gao Leilei**

**Supervisor：** **Prof. Gao Xiaopeng**

School of Computer Science and Engineering

Beihang University,Beijing,China

**中图分类号： TN929**

**论文编号：10006ZY1206216**

硕 士 学 位 论 文

**面向能力分析的教学管理平台设计与实现**

作者姓名 高磊磊 申请学位级别 工程硕士

指导教师姓名 高小鹏 职 称 教授

学科专业 计算机系统结构 研究方向 数据挖掘

学习时间自 年 月 日 起至 年 月 日止

论文提交日期 年 月 日 论文答辩日期 年 月 日

学位授予单位 北京航空航天大学 学位授予日期 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：        日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

摘 要

随着互联网技术发展、数据挖掘技术的成熟，数据挖掘与深度学习被应用在很多领域来发现有价值的信息。目前在教学领域，教学平台所能提供的功能单一，无法全面的获取学生在学习过程中产生的数据，因此无法全面的，客观的分析学生在学习过程中的能力变化。

本文以Moodle开源社区为基础，设计与实现了面向学生能力分析的教学管理平台。该教学平台采用B/S模式，教师通过Web端发布作业，作业要求，测试用例等。学生通过Web端提交作业，进行测试，查看测试结果，进行申诉等。为了方便教师与学生在课堂后的交流，平台开发了课程讨论区，同时为了促进学生之间的交流互助，系统设计实现了互助区，促使学生之间进行交流探讨。同时是为了满足不同课程的不同需求，课程功能是可选择配置的，教师根据课程具体形式来选择需要的功能。在功能丰富的前提下，使得课程功能不臃肿，方便较适合学生使用。

此外，本文的基于Moodle的教学平台系统，将学生在平台使用过程的所有数据进行收集，并将通过这些数据发掘的有价值信息用于课程的进一步提高和改进上。期望随着使用该平台的课程的增加，所收集数据的丰富通过使用数据挖掘，深度学习等现有技术发掘出教学数据中更有价值的信息和模式，从而进一步为课程改进提供数据支撑。

本文调研分析了目前国内外教学平台系统的发展情况，对目前现有几种主流开源教学平台进行了对比分析，最终确定了基于Moodle平台进行面向学生能力分析的教学平台设计开发的技术框架。最终实现了功能丰富，可定制，关注学生学习的过程数据的教学平台系统。

本文设计实现的教学平台系统目前已经在本科生的面向对象课程，软件工程，编译原理，计算机组成原理课程，研究生的高等软件工程等课程上进行使用。针对不同课程的特殊需求进行了定制化开发，从而帮助老师更好的进行教学工作。目前的工作进展，使用情况得到老师的好评，平台运行平稳。平且随着使用该平台的课程的增多，数据量的增加，以后可以进行更加有意义的信息挖掘和模式发现。

**关键词：**教学平台；Moodle；数据挖掘； 能力分析

**Abstract**

With the development of Internet technology and data mining, data mining and deep learning has been applied in many areas to find valuable information. At present, in the field of education, the course management system can only provide simple functions, cannot get the data generated by students during their learning activities, it cannot comprehensive and objective analysis the changes of students ability during their learning process.

Based on the Moodle community, this paper designs and implements a course management system for students' ability analysis. The course management system uses B/S frame, teacher publish homework, requirements, test cases and so on through the Website. Students can submit their jobs, test results, view test results, appeal and so on with Website. In order to enhance the communication between teachers and students after the classroom, the system developed a curriculum discussion area, and in order to promote the mutual assistance between students, the system designed and developed mutual assistance area. To meet the different needs of different courses, curriculum features are optional configuration, so teachers can select the specific desired functions. In the feature-rich premise, making the curriculum is not bloated, more convenient for students to use.

In addition, the Moodle-based course management system collects all the data of the students in the process of using the platform, and uses the valuable information excavated from the data for the further improvement of the system and the course. It is expected that the enrichment of data collected through the use of the platform will help to find more valuable information and patterns in the education data through the use of existing techniques such as data mining and deep learning, so as to provide further data support for curriculum improvement.

This paper analyzes the development of course management system platform at home and abroad, compares and analyzes several mainstream open-source course management system, and finally choose the Moodle platform as technical framework for the design and development of our course management system, which is used to analyze students' ability. And finally developed the course management system which is rich in functions, can be customized and concentrate on the data of students' learning process.

The course management system designed and implemented in this paper has been used in undergraduate courses such as object-oriented course, software engineering, compiling principle, computer composition principle course and graduate student's advanced software engineering course. The special needs of different courses were customized development, so as to help teachers do better during their teaching work. Teachers have made a lot of praise about the current work, meanwhile the platform runs smoothly and stable. With the increase in the number of courses using this platform, more data mining and pattern discovery result can be carried out in the future.

**Key words:** Course Management System (CMS); Moodle, Data Mining, Ability Analysis

# 绪论

## 研究背景及意义

近年来随着国家产业结构的不断升级以及国内企业生产的转型，市场提高了对高校毕业学生的能力要求。如何全面的定义、量化和评测学生的能力成为教学过程中面临的一个问题。传统教学使用单一的考试成绩来分析、衡量学生能力的方法包含信息量少，不能完整反映学生在完成课程学习之后，能力形成和变化情况，因此造成高分低能的现象。要对学生的能力进行分析，那么我们就需要回答两个问题：如何定义学生的能力，有哪些维度来衡量；能力分析是要分析最终能力形成的结果，还是分析能力的形成/变化过程。

首先，对于不同的课程，能力的定义维度是不同的，因为课程训练目的不同。因此能力分析的维度需要由任课老师来定义，然后通过教学过程中产生的诸如作业成绩，代码成绩，测试分数，个人项目完成情况，团队项目完成情况等过程数据来进行计算、分析。所以进行能力分析的前提就是收集学生的在课程过程中的数据。其次，关注学生能力形成/变化的过程比分析能力的形成结果更有意义。因为教学过程中，教师需要时刻关注学生的能力提升过程，以便选择最适合学生能力的教学方式，最终达到能力提升最大化的目标。因此我们需要收集学生在学习过程的中的过程数据来分析学生的能力形成/变化过程。在得到学生能力的形成和变化情况后，我们需要进一步分析影响学生能力形成和变化的因素。而从信息交流的层次来看，教学过程实际上是教师与学生，学生与学生进行知识，信息交互的过程。因此我们可以认为学生能力的获取以及变化是受学生在教学交互过程中所参与的交互活动影响的。

为了完善学生能力定义，同时得到学生在课程学习过程中的能力形成和变化情况，需要收集学生在学习过程中产生的过程数据。同时为了分析影响能力形成变化的因素，需要关注学生参与教学交互过程。因此需要一个平台来收集学生在学习过程中产生的数据，同时能将学生在教学中参与的各种交互过程管理起来。收集学生在学习过程中参与的诸如个人作业交互(作业互评)，团队作业交互(团队互评)，一般性学习交互(问题探讨)等教学交互过程中所产生的数据。以此在数据的基础上分析学生能力形成和变化情况，并且发掘不同教学交互过程特征对学生能力形成和变化影响情况。

目前教学平台在高校教学管理方面的应用并不深入，目前国内各高校的教学信息系统多数仅停留在 OLTP 的层面上[2]，这些系统仅仅针对教学任务，不重视教学过程数据的收集和学生在教学交互过程中行为的管理。这样的平台无法收集到多维度，有意义的学习过程数据，不能支持对学生能力的多维度的分析，因此无法支撑我们之前提到的工作。所以迫切需要新型教学信息平台来收集、记录教学过程中产生的数据，同时能将学生在教学中参与的各种交互过程管理起来。

本文针对当前各高校使用的教学管理系统功能简单，数据收集不足，提出设计实现新型教学平台。该平台在帮助老师更好的完成教学任务的同时增加了学生与老师，学生与学生之间的交流互动，并将这些数据收集起来形成历史数据。平台针对性收集学生的在教学交互过程中产生的数据，并且使用这些数据对学生能力进行多维度的分析衡量，得到学生在学习过程中自身能力的形成和变化情况。同时将学生在教学中参与的各种交互过程管理起来。并使用数据挖掘算法发掘教学交互过程特征与学生能力形成变化之间的关系。

## 国内外教学平台发展现状

目前国内多数高校的教学管理平台依然停留在OLTP层面，并且没有对学生的历史数据进行保留，课程之间的信息没有关联在一起，只进行事务处理，没有注意到教学过程数据的价值，没有历史存留。而在数据挖掘技术相当成熟的今天，数据的重要性不言而喻，而教学过程数据产生的交互数据对教学改进更是具有非常重要的意义。从学生角度看，它可以反映一个学生学习，知识获取的过程。通过大量的数据我们可以发现一些学习模式，互动模式。而这些模式可以反过来去指导学生调整自己的学习策略，从而提高学习效率。从老师角度看，通过数据挖掘我们可以获得学生在学习过程中遇到的难点，发觉学生能力提升的影响因素，从而调整教学重心，帮助学生更好地理解课程内容。并且教学过程不仅仅是老师与学生之间的信息沟通交流。学生与学生之间的信息交流、问题探讨在一定程度对学生的学习更加有促进作用，但是目前国类的教学管理平台没有注意到学生之间交流数据的价值，也没有相应的工具去管理这些交互过程。有些课任老师会创建课程交流群，来与学生进行课下交流。但是这种消息记录无法保存，问题解答层次混乱，逻辑容易被打乱，查找不易，不能形成一种稳定有效地交流方式。

目前国外一些高校已经开始使用一些商业版或开源的平台(Course management systems)CMS来帮助教师更好地进行教学管理，帮助老师更好地完成教学目标，帮助学生更好的提升自身能力。同时也开始注意到教师与学生，学生与学生之间的信息分享，交流的重要性。典型的教学管理系统允许老师进行信息共享，课程主要内容生成，在线发布课程作业和学生作业测试，加入课程讨论，支持远程教学，组队进行协作等。目前国外的主流商业版平台有Blackboard, WebCT和TopClass，主流开源平台有Moodle，ILias和Claroline。而现在使用最多的平台是Moodle(Modular Object Oriented Developmental Learning Environment)，是一个开源的，灵活的，支持课程功能自主定制的学习环境平台。该平台在使用过程中会积累大量的课程原始过程数据，而这些原始过程数据对于分析、发觉学生的学习行为具有很大的价值。它可以记录任何学生在平台中的行为，并形成行为轨迹，比如查看，编辑，提交，进行测试以及与同学之间的交流信息。但是这些平台的局限之处在于他们只能提供课程最基本的功能支持，对于一些有特殊需求的课程如需要代码测试，团队协作，同学互评等特殊需求功能则没有涉及，但是往往这些特殊定制的功能才是一门课程所需要的最为核心的功能。该平台只是进行了信息收集，并没有对信息进行合理的展示，没有将收集到的学生信息展现给任课老师，也就是说虽然该平台只是完成了信息收集，而并没有使用收集到的数据来进行相应分析，挖掘等进一步的工作。这对于一个教学平台来说是完全不够的。在线教学环境不只是仅仅要收集数据还要对数据进行合理的展示和运用，从而更好地反映教学过程中的问题，以此来指导学生和老师完善和改进教学过程，为教学改革提供依据。

目前尚没有平台收集学生交互数据以及教学交互过程数据，并且所有平台都只实现了对课程的基本支持，但是对于具有特殊功能需求的课程，现有平台无法实现支撑。且现有平台所实现的功能主要都是面向业务处理的，也就是传统的数据库表格+页面模式，面向的是具体的教学任务，使用数据库来管理具体数据，使用页面来输入和展示具体任务数据，不能满足我们从多种交互任务的交互特征以及多维数据进行学生能力分析的需求。并且现有平台对学生之间交流没有太多的关注，没有实现对教学交互过程的管理。从而导致学生之间交互数据无从收集，无法对学生之间的交互进行分析和信息发掘。

## 研究内容及目标

根据以上分析，本文的主要目标是设计实现面向教学交互过程的教学管理平台，并使用该平台收集到的数据进行数据挖掘，价值发现。同时对衡量学生的能力给出更加客观的标准，使用平台的多维数据来描绘学生能力的变化。针对学生能力分析，需要设计实现灵活的，可进行功能定制的在线教学平台，该平台可以收集学生在教学交互过程中产生的数据，并将数据进行处理后展示给任课老师，使得老师能随时了解学生的学习状况，并将课程中学生的讨论数据内容进行收集整理形成课程知识库。对学生与老师，学生与学生之间的交流内容进行内容分析提取，进行热点挖掘。根据积累历年作业形成的作业数据库来对学生的作业进行查重。开发特定的功能模块来满足特定课程的功能需求，如学生代码互测，团队协作，学生互评等功能。

平台围绕教学交互过程设计相应的交互流模型，以此来支持提取数据和开展能力分析工作。从多种交互任务的交互及流程层次来研究平台架构和相应的实现技术，教学平台的目标就是将教学中的各种交互管理起来。针对目前各高校对学生能力分析不完善的情况，我们使用学生在学习过程中产生的多维数据来对学生的能力进行完整定义，分析，衡量。在得学生的能力分析数据后，根据学生在教学交互过程中数据来发掘教学交互过程特征对学生的能力形成的影响，以此来为教学过程改进提供数据依据。根据学生在学习过程中产生的数据绘制学生能力变化表，将学生能力变化情况提供给老师，以便老师改进教学过程。从信息传递角度看教学过程其实就是知识、经验的传递交互，其中包括师生时间的交互，学生之间的交互。从交互参与形式来看又可分为个人作业交互(如作业互评)，团队交互(团队协作)，一般性学习交互(问题讨论)。而学生能力的形成和变化就是在这些交互过程中产生的。因此使用数据挖掘算法来发现教学交互过程特征和学生能力形成之间的关系能清晰的反映出学生能力获得与所参与的交互活动之间的关系。这对于改进教学环节，进行教改是具有非常重要的意义的。

学生作为一个特殊群体，在学校围绕课程，班级，寝室等形成不同的社交圈。我们知道在社交过程中情绪会通过人与人之间的交流而进行传播。在社会中，人们通过私下，网上，阅读等来获得或者了解别人的情绪，同时传播自己的情绪。而一旦某一种情绪获得大多数人的响应以及一定数量的人参与进来，那么这种情绪就会成为对于某件事情的主流情绪，而参与人数会呈现爆发式增长，从而在社会上形成一个热点话题。这些热点话题的主流情绪会影响大多数人对这件事情的看法，甚至一定程度上使得大部分人失去自我判断的能力，而只是从众，符合而已。而在课程学习中，这种情况也同样存在。在课程学习中学生通过知识交流，闲聊，跟帖等形式来传播各自的正面的，负面的情绪。而一旦某一种情绪得到大多数人的响应，并且获得一定数量学生参与进来，那么这种情绪就会成为主流情绪并且会影响大部分同学对事情的客观评价。

而在像面向对象这样的课程中我们需要同学之间进行互测，这样就涉及到学生与学生之间的对立，而这种对立很难要求所有同学都具有很高的素质，从而使得该过程公平合理。因此很多遭遇不公平待遇的同学不可避免的就产生了负面的情绪，而学生作为一个特殊的社交团体，这种负面情绪不可避免的会进行传播。因此我们希望可以通过收集到的学生的数据来分析学生之间负面情绪的传播，以及负面情绪对学生行为的影响。

## 本文研究限制

首先由于系统是基于开源平台的基础进行开发，并且有大的功能模块加入且要求系统运行平稳，数据安全，同时具有极好的响应速度。这需要在本身了解平台框架的前提下，进行重构。以便使系统的功能扩展更加方便，新功能的加入不会影响到已有功能的稳定性和数据的完整性。同时由于教学平台根本上将依然是一个信息交互平台，且本系统的重点是关注学生在学习过程中多产生的数据，所以对数据的安全和完整性具有很高的要求。因此我们需要在原平台的基础上进行数据结构的重构，数据库的重构以满足本系统的特定需求。

由于该教学平台刚开始设计实现并从一开始就被应用在本科生课程中。因此本文的主要工作点在功能实现以及维护保证系统的平稳运行。功能实现上，针对目前计算机学院几门重点课程的特定功能需求进行了定制化设计开发，涉及到的功能模块有讨论区，互助区，作业互评，申诉去，团队协作，课程博客，移动客户端消息推送等。因此本文的主要内容是系统的设计实现和系统的维护。

由于系统刚开始设计实现，且尚未在全院推广使用。因此系统收集到的数据量太小，无法进行真正意义上的数据挖掘，即通过海量数据来发现学生的学习模式，能力变化曲线以及不同课程对学生能力的要求等对课程改革具有指导性意义的隐藏信息的分析和发现。且由于课程的特殊情况（无法进行试验，一年只有一次）因此数据在维度，数量上都无法达到信息发掘的要求。

在本文中，由于数据量太小，所以在信息发现方面我们使用系统已收集到的多维数据来客观，实时的描绘学生的能力。并将对学生能力的衡量应用到作业互测模块中，用来达到最优分配目的。同时见学生在课程中的交互文本进行自然语言处理分析，进行学生情绪检发现，期望等数据量达到一定程度、数据维度足够多的时候可以进行情绪传播途径的发现，以此来辅助老师更好的进行教学工作。

## 本文的组织结构

本文共分为五章，具体内容如下：

第一章 绪论。首先介绍了课题的研究背景和意义，接着介绍了国内外教学平台的发展现状，并对此分析了目前主流教学平台的功能短板以及本文期望的教学平台所能实现的功能，最后给出了本文的研究目标及内容。

第二章 相关技术以及原理。首先分析了插桩技术、软件调试技术，特别应用于本文中的交叉调试技术，接着详细介绍了本文的目标操作系统ACoreOs的内部实现机制，最后给出了执行重放过程中所面对的挑战和问题。

第三章 执行重放方案的设计。首先指出了重放本文目标系统上应用程序存在的问题，然后根据存在的问题具体给出了相应的解决方案，接下来，给出了本文提出的执行重放整体设计框图和各个模块的大致设计思路。最后，本文针对相关挑战给出了相应的解决方案。

第四章 执行重放方案的实现。本章分三部分，分别给出了执行重放方案中三大模块：信息记录模块、日志解析过滤模块、重放模块的实现细节。

第五章 系统验证。详细介绍了对本文提出的执行重放方案的测试和验证。并对引入的插桩开销进行了度量。

总结与展望。首先对本文的工作进行了总结，然后指出来本文工作不足，最后给出了对后续工作的展望。

# 研究现状相关技术

## 教学平台发展现状

学习管理系统是用于建立学习资源、跟踪和管理学习进程的软件，一般由课程开发系统、教学系统、管理系统组成。学习管理系统发展至今天，管理范围和功能不断扩大，已经涵盖了组织中一切学习活动的管理，包括在线学习管理、面授培训管理以及非正式学习管理等[1]。常见的学习管理系统有Moodle、 Sakai、 ILIAS、 Claroline 等。

Moodle 是由澳大利亚的马丁•多格玛斯博士主持开发的基于社会建构主义教学思想的开放源码的学习管理系统。 Moodle功能强大，发展十分迅速，全球有两百多个国家 5 万余个学校与机构均有使用，如牛津大学、剑桥大学等，在我国高校与中小学也广泛使用。 Moodle 的主要功能有：创建课程页面、组织课程资料和课程活动、发布和评判作业、成绩管理等。Sakai 源于美国印第安纳大学、密歇根大学、斯坦福大学和麻省理工学院发起的一项开放源代码的课程管理系统开发计划， Sakai 在英语国家拥有大量的大学用户群，如斯坦福大学、耶鲁大学、加州大学伯克利分校等，我国的香港科技大学也有使用该系统。 Sakai 包括课程管理工具、项目工具等组件，并可对图书馆资源进行连接。ILIAS 支持学习内容的管理、协作、交流、评价和评估。ILIAS 在欧洲国家如德国、奥地利、匈牙利、法国以及意大利等国家有大量的用户群，涉及用户有大学、教育培训机构等。Claroline是由比利时鲁汶大学开发的一个免费的开源学习管理系统，中文译名为课廊。 Claroline 小巧，访问速度快，设计思路较为符合中小学教师习惯，是一款优秀的学习管理系统。

## 主流教学平台分析

在本文中，选择 Moodle、 Sakai、 ILIAS、 Claroline 为比较分析对象，以实际安装平台以及应用这些平台的典型网站为例进行比较分析，从学习者个性化设置、基本支持功能、学习支持工具以及学习者必备条件对这四款软件进行比较，比较维度如上图 1 所示。其中 Moodle 平台以 Moodle 中国[2]（版本 Moodle1.9）为例， Sakai 平台以 Sakai Test Drive[3]（版本 Sakai2.5）为例， ILIAS 平台以 ILIAS 3.10 EvaluationPlatform[4]（版本 ILIAS 3.10.）为例， Claroline 平台以课廊中国[5]（版本 Claroline 1.8.11）为例。

1. **功能可配置**

在本研究对各个系统的界面布局、结构设计、学习者的界面功能模块设置以及学习者的个人设置进行对比。

1. 学习者个人界面布局

四款软件均支持个性化外观与风格，个人桌面的功能的增加与减少，均可以根据使用者的爱好与习惯调整。

1. 学习者课程界面

四款软件的学习者课程界面功能模块设置大同小异，均体现了学习管理系统的基本功能与软件自身特色，它们的学习者功能设置如表 1 所示。

1. 学生个人设置

四款软件都可以编辑学生注册的个人基础信息与登录密码等，个人设置功能比较完善的 Moodle、 Sakai 都能支持个性图像的上传，而 Claroline 则拥有特色的账户统计功能，可以统计该用户的登录次数、下载文档的次数、论坛发帖时间等，可以让学生更好地了解自己的学习进程。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统 | Moodle | Sakai | ILIAS | Claroline |
| 学习者界面  功能设置 | 活动（包括资源、讨论区）； 搜索论坛 （附带高级搜索功能）；课程管理；我的课程（所选课程）；资源（课程提供的资源列表）；课程新闻；提醒（日历、事件）；课程动态 | 课程提纲； 时间表；告示；模块；资源；分配；测验、提问；测试中心；成绩单；资源上传与下载；讨论与个人信息；聊天室；站点统计；播客；投票、调查；讨论区；博客；介绍 | 邮件；新闻；个人设置；书签；搜索（具有高级搜索功能）；日历；学习进程；在线用户列表； Tag；便签 | 课程简介；课程计划；课程公告；课程讲义；课程练习；学习路径；作业；学习讨论；学习小组；用户列表；协作资源；聊天答疑 |

1. **功能覆盖范围**

基本功能是指除了学习工具外的其它的对学习者的基本支持，包括导航与定位、多语言支持、学习者帮助、学习者权限管理四方面：

1. 导航与定位

四款软件所提供的导航与定位功能都能为学习者提供明确的导航。 Moodle 的导航条为链接菜单，其它功能只能通过工 具 栏 进 入 ； Sakai 的 导 航 条 菜 单 按 钮 设 置 了 “MyWorkspace”、“ User Sites”以及所选的课程页面，左侧为工具栏； ILIAS 的导航条有两级，一级是“ Personal Desktop”、“ Resources”、“ Search”、“ Mail”、“ Administration”，二级导航是进入这些网站之后的栏目导航，并具备书签功能，学生可以在任何材料中加入书签； Claroline 的导航条分为“我的课程”、“我的日程”、“我的账户”、“退出账户”可以根据用户需求添加其它栏目；进入课程之后左侧有工具导航；右侧还设置了一个下拉式的导航菜单从细节设计方面 Sakai、ILIAS 比 Moodle、合学习者操作的要求，除了可以明确定位自己所在的位置，还可以很方便地使用系统所提供的工具。

1. 多语言支持

四款软件均支持英文、中文、法文、德文等多种语言，从 Moodle 在中国的流行程度可以看出 Moodle 对中文的支持是最好的， Sakai、 ILIAS、 Claroline 中文支持并不完善，如Sakai 个别页面未译成中文， ILIAS 只有一些工具译成了中文。

1. 学习者帮助

Moodle、 Sakai、 ILIAS 提供“如何使用软件”的帮助，Sakai 提供了十分方便的进入方式，在每个工具模块的标题栏右上方都有一个问号标志进入该软件的帮助， Moodle 和 Sakai都提供帮助工具的搜索功能，而 Claroline 中文版未见有这方面的帮助。另外， Sakai、 ILIAS、 Claroline 帮助并未完全翻译成中文。

1. 学习者权限管理

四款软件对于学习者的权限管理差异不大，学生的权限全部由教师或者管理员掌握，只要教师或者管理员授权，便可以使用课程界面的所有内容与软件提供的各种工具以及更改个人信息等。 而使用 Sakai的学习者可以建立自己的个人站点，使用个人站点所提供的各种工具。

1. **教学支撑模块**

本文把学习支持工具分为独立学习工具与协作学习工具两类。其中独立学习工具包括同步/离线学习支持工具、教师在线帮助、课程管理工具、学习评价工具、个人学习资料管理、课程内容、学习进程管理和其它工具；协作学习工具包括 e-mail、 BBS、小组协作工具、实时聊天工具、文件交换共享工具、博客/播客、白板工具和其它工具。

1. 独立学习工具的比较

四款软件均支持同步与离线学习，教师在线帮助则是通过聊天工具、论坛以及 e-mail 来实现。在课程管理功能、学习评价工具的设计、课程内容管理这三方面，四款软件没有实质的差别，学生在被授权的情况下可以加入、退出课程，均具有测试功能与作业编辑、提交与评分功能，支持各种题型的测试，学生可以查看个人成绩。通过表 2 也可以看到Sakai、 ILIAS 对独立学习工具的开发要优于 Moodle 与Claroline，如 Sakai 和 ILIAS 的学习者个人资料管理功能均支持学生在网站上建立个人的文件夹，并提供了在线笔记功能。对于课程内容的提醒，四款软件都有明确的提示， Moodle、Sakai、 ILIAS 支持 RSS 订阅更新。在学习进程管理方面， Sakai的日历工具提供了很多不同类型的标记， ILIAS也有类似的功能设置，该项功能非常个性化。

表2 独立学习工具对比分析

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 系统  工具 | Moodle | Sakai | ILIAS | | Claroline | |
| 同步/离线学习支持 | 支持在线同步学习；在拥有权限的前提下可以下载教师上传的课程讲义与课程内容 | | | | | |
| 教 师 在 线帮助 | 学生只能通过实时聊天、论坛留言、站内 e-mail 的方式寻求教师帮助 | | | | | 在课程工具中有“聊天答疑”工具， 可以为学生提供在线帮助答疑， 类似于实时聊天 |
| 课程管理 | 课 程 内 有 搜 索 功能，可以订阅课程动态 | 每个加入的课程都会在导航栏中显示课程网站 | | 可对学生正在学习的课程和已经学过的课程进行信息保留，学生随时根据自己需要对课程进行检索，或者对自己学习课程的信息进行管理 | | 可以查看所有课程列表以及查看加入的课程 |
| 学 习 评 价工具 | 具 有 测 试 功 能 与作业编辑、提交与评 分 功 能 可 选 择查 看 “ 总 体 成绩 ” 和 “ 用 户 成绩” | 具有测试功能与作业编辑、提交与评分功能支持教师开放自主评分给学生 | 具有测试功能与作业编辑、提交与评分功能，支持各种题型的测试，学生可以查看个人成绩 | | 学生可通过习题、测试为自己的学习进程评价， 可查看学生个人统计， 包括论坛发帖数量、提交作业数量等 | |
| 个 人 资 料管理 | 不 支 持 在 课 程 网站 上 建 立 个 人 文件夹 | 提供个人资料管理器，学习者可以上传个人文档资料 | 支持建立个人文档以及上传个人文档资料 | | 不支持在课程网站上建立个人文件夹 | |
| 课 程 内 容管理 | 可以使用 RSS 订阅更新 | | 在课程页面中有显示 | | 课程内容有更新会在课程标题前用小红点表示提示 | |

1. 协作学习工具的比较

表3 协作学习工具对比分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统  工具 | Moodle | Sakai | ILIAS | Claroline |
| e-mail | 均拥有内部 E-mail 系统，学生可以使用内置 E-mail 功能发送电子邮件给个人或小组可以选择把电子邮件转发到外部地址 | | | |
| BBS | 支持分组讨论，可在讨论区上传与下载附件资料；可订阅论坛 RSS 具备搜索功能 | 支持分组讨论；可在讨论区上传与下载附件资料；具备搜索功能 | 支持分组讨论；支持分组讨论，讨论可跨课程跨小组共享， 可在讨论区上传与下载附件资料可订阅论坛 RSS | |
| 小组协作 | 需要教师开放权限，学生可以自由加入小组，每小组可以设置属于该小组的讨论区、聊天室或白板，可以小组协作完成任务；学生可以创建在线俱乐部、兴趣小组和学习小组 | | | |
| 实时聊天 | 支持一定数量的同步聊天室，支持不限数量的分组讨论 | 支持一定数量的同步聊天室，并且为所有的聊天室建立聊天记录，可以查看在线用户 | 支持不限数量的小组聊天，可以查看在线用户 | 支持不限数量的实时聊天，并且为所有的聊天室建立聊天记录 |
| 文 件 交 换共享 | 使用投件箱上传作业；共享的文件可以直接打开或者下载打开，可以连接到外部资源 | 使用投件箱上传作业；支持学生在有权限的情况下使用资源工具在课程网站的普通主题中上传、共享材料，资源工具支持多文件上传到小组的投件箱共享，文件可以在个人或者小组的博客、 Wiki、 邮箱、信息工具或是讨论组中共享 | 使用投件箱上传作业；支持学生可以设置个人文件夹共享；上传的多个文件自动保存为 zip压 缩 文 件 ； 支 持WebDAV | 使用投件箱上传作业；每个课程都有其上传工具，学生可在“协作资源”中上传下载 |
| 博客/播客 | 可以内嵌博客与播客工具 | 拥有博客与播客工具；博客系统支持超链接和文件上传 | 拥有自带的播客工具 | 可以内嵌博客与播客工具 |
| 白板工具 | 支持添加如 DimDim 网络会议、 Elluminate 实时教学系统等软件 | 支持在平台上整合 Elluminate实时教学系统、 Breeze Live 等商业产品 | 支持 iLink 等第三方白板工具 | 可内嵌第三方白板工具 |

由表 3 可见这四款软件均拥有内部 E-mail 系统，提供相似功能的论坛，支持小组协作与实时聊天，文件交换均通过投件箱共享，白板工具都需要通过第三方模块支持，而 Sakai提供的博客与播客服务， ILIAS 也拥有播客功能。Sakai、 ILIAS均支持 Wiki，是软件的特色功能之一。

1. **各平台自身适应情况**

每一个学习者的要求都会根据其习惯而有所不同，因此并不可能开发出适合所有学习者的学习管理系统，课程开发者可根据具体需要，对开放源代码的学习管理系统的功能进行扩充与整合，以开发出适合该课程学习者的学习管理系统。对于不同的学习群体、教师与管理者以及不同的学习内容各款软件各有其优势。在开源学习管理系统的选择上，对于中小学教师，可以选择为中小学教师提供一套符合其习惯模板的 Claroline；对于大学的教学，可以使用以课程为中心的Moodle；而 Sakai 与 ILIAS则更适合建设大型的学习社区。

## 现有推荐算法介绍

当前被广泛使用的推荐系统的非形式化概念是Resnick和Varian给出的，这是一个利用电子商务网站向客户推荐商品信息和建议，帮助用户决定购买哪些产品，模拟销售人员帮助客户完成购买的过程。其中推荐部分由3个要素组成：推荐候选对象、用户、推荐方法．目前通用的推荐模型如下图1所示．用户可以主动向推荐平台提供个人信息或请求，或是有推荐平台自行采集．推荐平台可以同时使用不同的策略来进行推荐，如通过计算采集到的个人信息和对象数据来得到推荐结果，或直接使用已经建立模型的知识数据库进行推荐，然后将推荐结果反馈给用户。

图1

推荐算法是整个推荐平台中最为关键的部分，在很大程度上决定了推荐平台的类型和性能．目前，尚没有统一的标准来对推荐平台进行分类，学术界和工业上从不同角度对推荐方法有很多种不同的划分。但归根结底，主流的推荐方法基本可以概括为以下几种：协同过滤推荐、基于内容的推荐、基于知识的推荐以及组合推荐。本节我们将讨论这几种推荐算法的技术框架，并将对这几种算法进行比较。

### 协同过滤推荐

协同过滤的思想是找到与目标用户*C*usr相近(兴趣以及喜好相似)的其他用户*C*other，并且使用算法计算某一对象O对于用户*C*usr的有效值*J*(*C*other，O)，利用有效值对所有对象进行排序或者加权平均等操作，找到最接近*C*usr的对象O*’*。协同过滤算法的思想非常容易理解，即基于其他用户对统一内容的反馈(比如喜好，评分等)向目标用户进行推荐。协同过滤算法的研究成果包括Tapestry system,GroupLens,PHOAKS,Jester等。但从整体上看，协同过滤算法整体分为两类：即与模型的方法和启发式方法。

ACoreOs 机载嵌入式实时操作系统[[51](#_ENREF_51); [52](#_ENREF_52)]（以下简称 ACoreOs）是专门为嵌入式多核实时系统设计开发的操作系统，为应用程序提供高效的实时多任务调度能力、快速中断/异常事件处理以及实时的任务间通信。

ACoreOs 操作系统作为一个针对航空应用需求设计的机载嵌入式多核实时操作系统，能够完全满足机载环境所提出的强实时、高安全、高可靠、高确定、可裁剪性、可升级性等特殊要求。借助于与之配套的集成开发环境 LambdaAE，用户可以快速完成嵌入式应用的开发、调试和部署。

ACoreOs 操作系统在系统中处于硬件层和应用层之间的位置，如图 6所示，其主要功能是高效管理机载计算机软、硬件资源，并以系统接口形式为应用程序提供所需的服务。

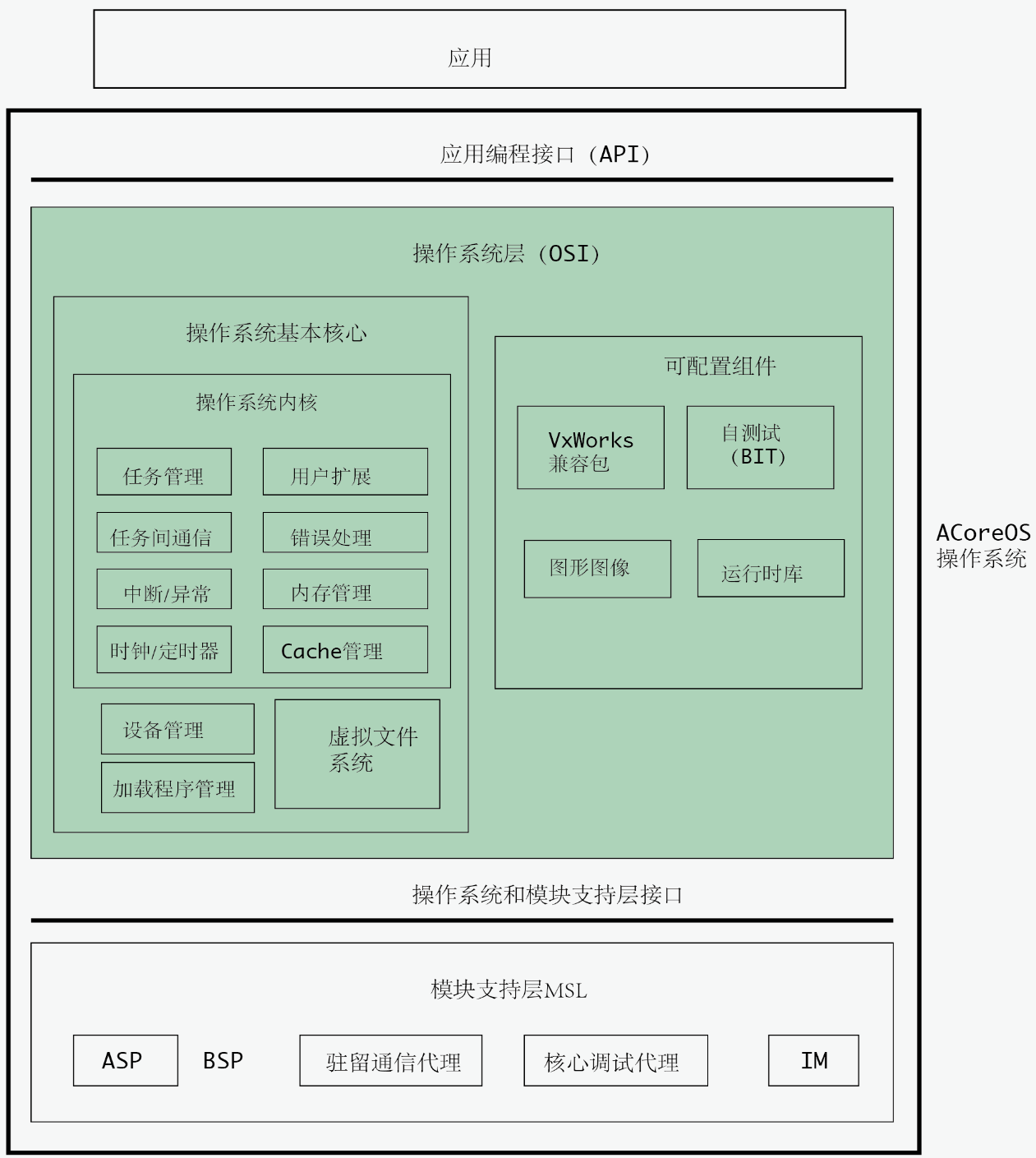


图 6 ACoreOs操作系统组成及各部分关系图

对于目标机硬件，ACoreOS 操作系统提供 MSL(Module Support Layer) 层软件，管理底层硬件层设备；对于应用软件，操作系统负责应用软件的管理，主要实现应用任务的调度、系统资源的分配，为应用软件提供可使用的组件，包括：VxWorks 兼容接口、BIT(Build In Test) 管理、C 运行时库、图形图像支持等功能，应用软件可通过操作系统提供的 API 接口访问操作系统，以支持应用软件的功能。

应用软件开发调试过程中，宿主机与目标机的连接关系如图 7所示。主机上运行的开发环境通过目标机服务器与目标机进行通信，完成应用软件的交叉调试。

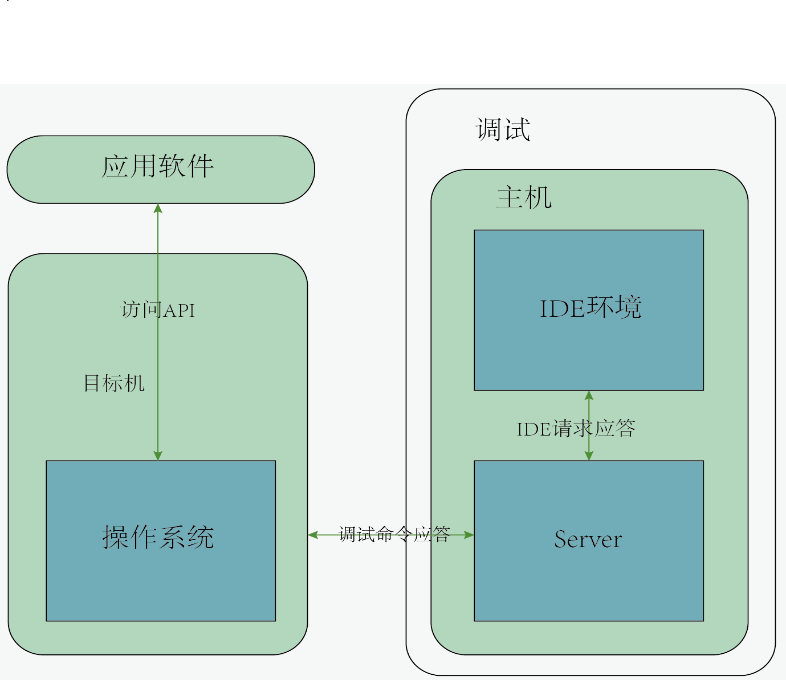


图 7 宿主机与目标机的连接关系图

ACoreOS 内核采用基于对象的方法管理任务、周期任务、用户扩展、消息队列、环缓冲、信号量、标志集和定时器。所有对象自创建起获得内核分配给它的唯一标识符，对象名字可以由用户指定，也可以由系统进行分配，但是内核不允许同类对象的名字相同，对象的名字是不能超过31个可打印字符的字符串。对象的标识符和同类对象名字全局唯一，并且同类对象标识符和名字一一对应。

### ACoreOs的任务状态转换

在 ACoreOS系统中的所有任务，其状态一定处于休眠态、就绪态、运行态、阻塞态四种状态之一。

1. 休眠态：任务已经创建或停止，并且无法接收CPU的调度；
2. 就绪态：任务已经取得除 CPU 以外的资源，可接收 CPU 的调度，一旦获得CPU控制权就进入运行态开始运行；
3. 运行态：就绪的任务成功争夺到 CPU 控制权并正在执行中；
4. 阻塞态：任务被挂起或由于所需资源目前无效而无法继续运行，该类任务不会接受CPU的调度。

任务状态在操作系统运行过程中由于资源的变化或程序指令而发生变化，各状态间的转换如图 8所示。ACoreOs上任务的创建到运行分为两个阶段，任务创建完成后状态并不是就绪态，而是休眠态，任务需要额外通过ACoreOs\_task\_start操作进入就绪态；处于就绪态的任务可被操作系统调度而获取CPU控制权进入运行态，或者因为被其他任务调用ACoreOs\_task\_stop而使其进入休眠态，或者被其他任务调用ACoreOs\_task\_suspend而使其进入阻塞态；任务在运行态可得到真正的运行，执行过程中可主动的调用ACoreOs\_task\_stop停止自身执行而进入休眠态，让出CPU，或者任务调用睡眠操作进入阻塞态，或者调用ACoreOs\_task\_suspend而挂起自身执行而进入阻塞态，或者由于调用资源获取操作但此时资源状态不可用而进入阻塞态，等待资源有效时被唤醒，或者高优先级任务就绪导致被抢占而进入就绪态等待再次得到系统的调度而进入运行态；任务处于阻塞态时，可被其他任务进行多种操作而发生状态迁移，ACoreOs\_task\_suspend操作并不会改变处于阻塞态任务的状态，ACoreOs\_task\_resume操作将唤醒阻塞态的任务，使其状态处于就绪态，或者在资源有效时资源回调函数唤醒因等待该资源而被阻塞的任务，使其进入就绪态，或者某些任务睡眠时间已到需要被唤醒，从阻塞态转换为就绪态。



图 8 ACoreOs任务状态转换图

### ACoreOs任务调度

从就绪状态的任务中，依据一定筛选规则选择一个任务使其取得 CPU 所有权，在处理器上运行，这种过程称为任务调度。与通用多核实时操作系统类似，ACoreOs也支持基于优先级的调度算法和基于时间片轮转的调度算法。

1. 优先级调度

ACoreOs系统内部集成了对基于优先级调度算法的支持，优先级调度算法是保证系统总是优先处理优先级较高的任务，在ACoreOs系统中，每一个任务都拥有一个任务优先级，主要是由程序开发人员在开发阶段手动指定的。当然，任务的优先级也不是一尘不变的，其在运行过程中可通过系统接口动态更改优先级。

ACoreOs 操作系统内核优先级调度又可以按执行过程中是否能够被更高优先级的任务抢占可分为：不可抢占的优先级调度算法和可抢占的优先级调度算法。

1. 不可抢占（non-preemptive）的优先级调度

不可抢占优先级调度算法是指一旦某个高优先级的任务被系统换入运行，会一直运行下去，直到任务执行过程中自主（如任务调用了睡眠操作）放弃CPU的控制权，系统才会重现选择当前就绪态中优先级最高的任务进行运行。当然，任务在执行过程中，可以被中断打断，但即使中断处理函数中使一个更高优先级的任务成为就绪态，在中断返回的时候，被打断的任务也不会被中断中就绪的更高优先级的任务抢占，除非被打断的任务在后续执行中自主放弃CPU控制权，更高优先级的任务才得以调度运行。如图 9所示，高优先级任务Task1获取到CPU控制权得到运行，运行过程中被中断打断，在中



图 9 不可抢占的优先级调度

断处理函数中，唤醒了比Task1拥有更高优先级的任务Task2，然而在中断处理函数退出时，Task2并没有抢占Task1的运行，较低优先级的任务Task1继续运行，直到其因为自身原因主动放弃了CPU控制权，此时更高优先级的Task2才得到系统调度获取到CPU而开始运行。

1. 可抢占（preemptive）的优先级调度

在系统任何执行时刻，都按照优先级高低来进行任务的调度，也就是，处理器上正在运行的任务一定是系统目前处于就绪态的任务中优先级最高的。

就绪态中最高优先级的任务会优先被系统调度运行，在其运行过程中，系统严格保证其对CPU的占用权，目的是尽可能让其运行完成，中途任何较低优先级的任务都不会打断高优先级任务的运行。然而，一旦最高优先的任务执行中，因为某些资源目前无法立刻获取，被阻塞而被系统换出，系统才会从就绪态任务中选择最高优先级的任务运行，但如果某时刻最高优先级任务等待的资源重新有效，最高优先级的任务又重新就绪，那么此时系统会让当前正在执行的较低优先级任务让出CPU，换入最高优先级的任务运行，也就是较高优先级的任务会抢占较低优先级的任务运行。

如图 10所示，高优先级任务Task1运行中遇到等待事件（例如执行一段时间后主动睡眠）而放弃CPU而进入阻塞状态，此时系统中处于就绪态的最高优先级的任务Task2获取到CPU控制权开始运行，运行过程中，之前由于等待事件而阻塞的Task1因等待事件到来而重新就绪，此时由于Task1的优先级高于Task2并且是系统目前就绪态优先级最高的任务，Task1将抢占Task2，获取到CPU控制权开始运行。



图 10 可抢占的优先级调度

1. 时间片轮转调度

在实时系统中，不同的任务可以共享同一个任务优先级，为了保证同优先级下的各个任务调度的公平性，优先级调度算法同时也启用了时间片轮转的策略，称之为基于优先级的时间片轮转调度。在此调度策略下，如果有多个任务共享同一个任务优先级，并且又是系统目前处于就绪态任务中优先级别最高的，那么系统会从该优先级对应的就绪链表中选择表头对应的任务进入运行，待该任务运行一段时间后，系统会重新换入另外一个同优先级的任务进入运行，而将该任务放置到同级优先级链表的链尾，等待后续调度，任务从运行到后来被系统换出换入另一个同优先级的任务运行的这段时间称之为时间片（time slicing）。

如图 11所示，任务Task1、Task2、Task3三个任务共享同一个优先级，任务Task4优先级较前者稍高，同级优先级之间系统采用的是时间片轮转调度算法，Task1消耗完自身时间片后，系统分配同样的时间片给Task运行，Task1置于同级优先级队列的尾端，Task3在Task2消耗本次时间片后得到运行，依次类推，Task1-Task3交替运行，当然在运行过程中也会被高优先级任务Task4抢占，Task4执行结束后如果当前系统中没有更高优先级任务就绪则会继续执行被抢占的任务，直到其时间片消耗完。



图 11 时间片轮转调度

### 任务间通信机制

任务间通信机制是多任务之间相互同步[[52](#_ENREF_52)]与通信的主要手段，主要用来协调各个任务的执行。在多任务的实时系统中，一项工作往往需要借助多个任务共同完成，而这些任务之间或多或少都需要相互协调，相互配合。为了满足系统中各个任务之间通信的需求，ACoreOS 内核提供了消息队列（message queue）、环形缓冲（ringbuf）、异步信号（signal）、信号量（semaphore）、事件（event）和标志集（flagset）等几种机制。

1. 信号量机制

信号量提供任务之间同步与互斥的能力，也是任务之间同步互斥最常用的一种机制，ACoreOs中将信号量细分为三种类型：

1. 二值信号量：用于实现对单个资源的访问，尤其是对临界区资源的互斥访问；其也可以用于两个任务之间的同步，一个任务获取，另外一个任务释放，但是不允许任务嵌套访问。
2. 互斥信号量：与二值信号量类似，既能提供同步功能，也能提供互斥功能，唯一不同是互斥信号量可以被嵌套访问。
3. 计数信号量：提供对使用具有多份资源的互斥机制，实现对两份或多份资源的访问保护。

以上信号量用在互斥时，可能出现优先级反转[[53](#_ENREF_53)]的问题，导致高优先级的任务因为等待某个信号量，而这个信号量需要一个较低优先级的任务释放，而这个较低优先级的任务由于被另外一个稍高优先级任务抢占而不能运行，导致高优先级的任务也无法运行，违背了系统基于优先级调度的法则，在ACoreOs中，使用优先级继承或天花板算法来解决该问题。

1. 消息队列

消息队列是一段内存缓冲区，用来存放任务之间、任务与中断之间等相互通信的数据，简称为消息，其中可以放置的消息个数和大小是可以在创建时候配置的。消息队列自身完成对内存缓冲区的维护，维护目前缓冲区中有多少消息，是否可以继续写入消息，是否可以读取消息等。

消息队列的状态有5个：（1）无消息，消息队列此时为空，里面没有存放任何有效数据；（2）无消息并且有任务等待接收消息，消息队列中没有数据，然而有任务尝试从中获取数据，消息队列可能会阻塞当前任务的运行；（3）消息队列中有消息等待被任务接收，消息队列中存放有效的消息，任务可以接收到有效的消息；（4）消息队列满，消息队列中已达到饱和，如果继续发送消息，任务会被阻塞；（5）消息队列满并且有任务等待发送消息，消息队列中已经充满消息，对于继续发送的任务，需要进行阻塞。通常，消息按照FIFO（先进先出）方式进入消息队列等待被任务接收，对于紧急消息，消息队列会将其放置到队列头部，首先被任务接收处理。

任务可以被阻塞在一个空的消息队列上，等待另外一个任务发送消息，实现两个任务之间的同步操作。任务也可以同时向一个消息队列中写入数据而实现任务之间的互斥。

1. 环缓冲

与消息队列一样，环缓冲也是一种通信机制，在环缓冲数据区域中存放任务间或任务与中断服务程序间通信的内容。

与消息队列不一样的是，环缓冲数据区域没有相互独立的消息，整个环缓冲数据区域都是用来存放通信数据的，环缓冲内部维护一个读指针和一个写指针，分别指示可用数据空间的起始和可用空闲空间的起始。对环缓冲的读和写操作可以由两个访问源同时（并发）进行，即一个访问源在读（写）环缓冲时另一个访问源可以同时写（读）环缓冲。在环缓冲上的任何操作都不会改变任务状态。为了提高效率，内核在进行数据操作时不会屏蔽中断响应和调度响应，这其间的一致性应该由用户来保证。

### ACoreOs中断处理

中断是硬件机制，其是外部设备通知CPU进行相关操作的一种手段。实时系统必须快速地对外部产生的中断请求进行响应，完成与外部环境的实时交互。ACoreOS 向 ISR（Interrupt Service Routine）提供了一些系统调用，以便 ISR 能够使用内核的部分功能。为了减少中断对实时系统性能造成的影响，必须确保关中断的时间尽可能短，因此，ISR 只能完成一些必要的操作，例如接收网卡数据以及读取其他设备输入，对中断后续处理通过代理给任务来完成。

中断发生时，ISR必须保存现场，在ACoreOs中，由于ISR只会修改部分寄存器的数值，因而只需要保存可能要修改的寄存器的数值，待中断处理完，在中断退出时，恢复修改的寄存器。其执行过程一般分为三部分。

1. 中断进入：保存 ISR 中要修改的寄存器（通过通用处理完成）；通知 ACoreOS 进入 ISR；并根据当前状态判断是否发生中断嵌套，对于不是中断嵌套的还需要进行任务堆栈和中断专用栈之间的切换
2. 用户ISR：关调度；使能中断；进入用户注册的 ISR。用户 ISR 中可以调用内核提供的部分系统接口，与任务进行通信与实现同步。如果用户没有对此中断注册相应的处理函数，默认情况下打印中断号
3. 中断退出：禁止中断；通知 ACoreOS 退出ISR；中断嵌套解除时，对于退出的是最后一级中断，完成对栈进行切换，从中断栈切换到先前被打端的任务栈；开调度；此时会触发调度器重新调度，如果系统中有更高优先级的任务处于就绪态，则切换到更高优先级的任务执行；如果当前执行任务有异步信号需要处理，则进行异步信号的处理；恢复 ISR 中修改了的寄存器的值（通用处理完成）；退出到上一级被中断的 ISR 或者被中断的任务

### 自旋锁技术

自旋锁[[50](#_ENREF_50)]是类似于互斥锁、信号量或互斥区等用于进程间或任务间同步机制。自旋锁由于不断的查询某个充当锁的变量可用性而在某些场合表现更高效而经常被用在对称多处理器操作系统内核中。自旋锁比互斥锁、信号量以及互斥区有如下的优点：

1. 任务不会被抢占而导致任务切换
2. 不用维护和修改额外的管理数据结构（目前谁拥有这个同步对象或者是否被锁，拥有该对象的时长，是否超过了设定的超时时长）
3. 不需要因为为维护公平的先进先出机制而执行额外的操作

自旋锁主要通过忙等待实现，在多核平台下，可高效用来保护互斥访问的共享资源，该共享资源临界区执行时间非常短，而且也不会触发任务切换，避免导致可能出现的死锁或者优先级反转问题，如果采用互斥锁机制保护这类资源，可能由于任务切换需要的操作时间远远大于该资源临界区执行所需要的时间，导致系统因为当前任务临时获取不到该资源触发任务切换，待该资源有效时又重新切换进当前任务，从而可能严重降低系统性能。自旋锁让任务在获取不到该类资源时不断在原地轮询资源的状态，而不用阻塞当前任务的执行，待轮询的资源被运行在别的CPU上的任务释放后，该任务可继续执行，期间并不用引入任何的系统切换开销，仅仅只有轮询开销，由于这类资源临界区执行时间很快，不会造成系统性能下降。

## 执行重放挑战及问题

本节给出了本文在提出执行重放方案时所必须面临的挑战和问题，主要包括探针效应、事件发生位置标记问题、上下文特殊性、多核事件顺序、系统特殊性以及数据竞争。

### 探针效应

无论是对目标代码插桩和源代码插桩，都会引入探针。基于软件方式的探针插入到系统中，用来监测系统行为的同时，也可能会改变被监测的系统的行为。这种对系统行为产生影响的效应称为探针效应。其存在于并发程序中并首次被Gait[[43](#_ENREF_43)]提出。

下面这个例子将很好的描述探针效应。有两个系统任务，如图 12所示，任务A和任务B共享一个资源X，这两个任务都是在临界区中对X进行访问，图 12中白色部分表示对X资源进行访问，现假设程序正确执行时访问顺序是：任务B先访问X，之后任务A访问X。根据图 12所示，正确的访问顺序没有得到满足，所以导致了系统故障。由于开发人员很疑惑系统的此次行为，在程序再次执行前加入了一个探针（例如printf语句）到程序中，在图 13中用黑色表示。

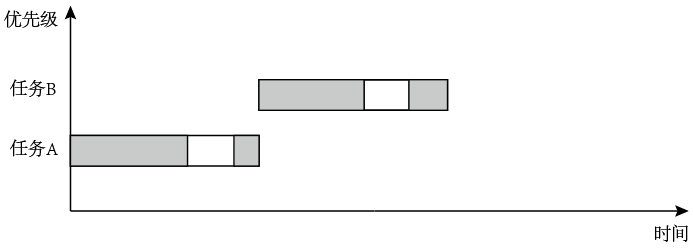


图 12 导致系统出错的执行

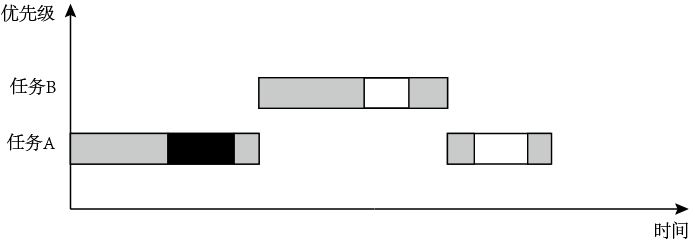


图 13 相同的执行，加入了软件探针，"躲过"了故障

第二次执行时，探针的执行会延长任务A的执行，这会导致在其进入临界区前就会被任务B抢占，这样任务B先于任务A进入临界区对资源X进行访问，而此次的访问顺序刚好满足正确的访问顺序，导致上次的故障不会再现。

由于本方案采用了软件方式的源代码插桩，从而引入了探针效应，如何解决插桩带来的探针效应是本文研究的一大难点。

### 事件发生位置标记问题

在记录执行阶段，不仅需要记录事件发生时任务的相关信息，还需要记录各个事件发生的位置。这里的位置其实指的就是事件发生时，任务执行到达的当前位置。在程序中一般用PC指针来进行标识。对于不包含循环或者递归调用的任务，通过上下文的PC指针就能够唯一确定事件发生的位置，但一旦程序中包含了循环或者递归，由于同一个循环体或者递归函数体被多次访问，从而仅仅依靠PC指针无法精确的指出事件发生的位置，因此必须引入额外的信息辅助PC指针唯一标识事件发生的位置。确定辅助信息也是本文面临的一大挑战。

### 上下文特殊性

ACoreOs是一个典型的嵌入式实时操作系统，其系统调用接口并不是采用被普通桌面操作系统(如Linux、Windows等)广泛采用的软中断机制实现的而是直接采用函数调用实现。在这种设计中，系统并不刻意区分用户态和系统态，可以认为所有的任务都运行在系统态。任务在其整个生命周期中仅存在一份上下文，其中掺杂了任务代码执行路径和系统代码执行路径，这不同于桌面操作系统，在普通桌面操作系统中，应用程序借助系统提供的接口在从用户态切换进入系统态时，操作系统会为其产生一份上下文，称为任务用户态上下文，该上下文反映了该任务在用户态的当前执行状态；一旦进入到系统态，任务将重新拥有新的一份上下文，称作任务系统态上下文，该上下文反映出任务在请求系统服务时，系统正在为其服务的当前执行状态。

ACoreOs任务在执行过程中，一旦发生切换，系统会保存该任务的上下文信息，然而，所有任务的切换都发生在系统的Thread\_dispatch函数中，系统为任务在上下文切换时保存的上下文信息基本是一样的，并且任务执行过程中被系统切换的操作是随机的，不是每次都发生在相同的场合，从而系统在任务执行中产生的一份上下文对重放执行阶段是没有任何意义的。任务整个生命周期仅有一份上下文加大了记录和重放的难度，如何正确的重现任务上下文是本文需要解决的一大挑战。

### 事件发生顺序

在多核处理器系统中，额外的不确定性来源于系统中多个CPU之间产生事件的并行性。主要包括三种情况：同步事件并行、同步和异步事件并行、异步事件和异步事件并行。

同步事件之间的并行也称为同步竞争，是内存竞争中的一种情况，多核环境下，每个核之间执行的代码是不相关的，产生的事件也是相互独立的，从而正在执行该应用中任务的CPU核心之间也会存在相互的竞争。正确的标示各个同步事件相对发生位置也必须的，例如，给定一个应用程序其中包含两个任务，任务A和任务B。其中任务A的任务入口函数是task1，任务B的入口函数是task2，其具体执行代码如图 14所示。任务A被操作系统调度到核心编号为1的CPU上运行，而任务B被调度到核心编号为2的CPU上运行。图 15和图 16分别给出了任务A和任务B可能的执行(假设Semaphore的初始值是0)。任务A和任务B并行竞争同一个信号量，如果两个任务同时访问信号量，哪个任务先获取信号量的操作权，操作系统是无法不保证的。从图 15和图 16中可以看出任务A和任务B并行竞争信号量会影响任务的执行流程，图 15中，由于任务A和任务B同时访问信号量，而此时任务A成功获取到了信号量操作权，那么任务A在执行中会由于获取不到有效的信号量而阻塞，导致操作系统产生了一次任务切换，此时任务A被换出，任务A释放了对信号量的操作权，位于另一个核上运行的任务B稍后获取到了信号量操作权执行了信号量释放操作，此时信号量进入有效状态，任务A重新就绪，进而后续任务A又会被切换进CPU核1继续运行，直到此次时间片用完或者因获取不到后续有效的信号量而被切换。图 16中，在竞争信号量过程中，位于核0上的任务B成功竞争到信号量操作权，对信号量进行释放操作，稍后，任务A才获取信号量的操作权，进行信号量操作，由于之前任务B释放了信号量，任务A可获取到信号量，而不会因为信号量无效而被阻塞触发切换而被系统换出。

图 15和图 16给出的任务A和任务B执行都是可能出现的，但两者结果却存在差异，其根本在于多核环境下应用程序中任务之间同步竞争造成任务对同步事件操作顺序不确定，为了重现应用程序在多核上的执行结果，其任务在多核上重放执行时产生的同步事件的顺序是必须保证与记录执行阶段是完全一致的。

|  |  |
| --- | --- |
| 1 | ACoreOs\_task task1(*void*\* param){ |
| 2 | while (true){ |
| 3 | ACoreOs\_semaphore\_obtain(semaphore); |
| 4 | //do some other things |
| 5 | } |
| 6 | } |
| 7 |  |
| 8 | ACoreOs\_task task2(*void*\* param){ |
| 9 | while (true){ |
| 10 | ACoreOs\_semaphore\_release(semaphore); |
| 11 | //do some other things |
| 12 | } |
| 13 | } |

图 14 任务相应执行代码



图 15 并行访问信号量时任务A先获取到操作权



图 16 并行访问信号量时任务B先获取到操作权

同步事件和异步事件之间、异步事件和异步事件之间发生顺序也具有并行性，例如一个CPU正在进行同步操作，而另外一个CPU产生了中断，或者两个CPU并行进入中断处理函数等，这些产生的事件也会存在类似于上面提到的因为事件发生的顺序不确定性对任务的执行造成影响。不同的是，这两类事件的并行性，操作系统本身并没有引入机制来对事件顺序进行串行化，或者标记出哪一个事件先于另外一个发生。如何正确解决这类事件发生的并行性对程序执行造成的不确定性也是本文研究的一大难点。

### 系统特殊性

目前已有的执行重放方案都是面向于普通桌面操作系统（Linux以及Windows）以及服务器系统（类Unix系统）的。本文的目标系统是ACoreOs，与一般的操作系统不同的是，ACoreOs的应用程序的开发环境是一个基于宿主机—目标机的开发架构。应用程序的开发、编译在宿主机端完成，目标机端负责运行操作系统以及系统之上开发的应用程序。由于程序的执行发生在目标机端，动态产生的信息也发生在目标机端，然而，在重放执行阶段，宿主机端需要读取记录执行阶段产生的日志信息并控制程序的二次执行，因而，产生的日志信息必须被上传到宿主机端。同时，在任务重放执行阶段，运行在宿主机上的重放模块必须能够动态获取目标端任务执行的相关信息和控制目标端任务的执行过程。

## 本文研究限制

在多核处理器系统中，另外一种的内存竞争是数据竞争：多个任务之间通过共享内存来进行数据的交互。正如本文前面提到，由于数据竞争记录开销过大，而本文也没有给出有效的软件方式减小这种开销，使其可使用于实时系统，故本文对数据竞争导致的不确定性重放不作深入研究。

## 本章小结

本章介绍了相关技术及其原理。首先介绍了代码插桩技术的特点，然后阐述了调试技术以及嵌入式开发中广泛使用的交叉调试技术，接着详细介绍了本文的目标操作系统—嵌入式多核实时操作系统ACoreOs，介绍了ACoreOs的任务状态、任务调度、任务间通信机制以及中断处理等基本机制，为后面实现和设计ACoreOs下应用程序执行重放方案奠定了基础，本章最后，指出了本文在提出执行重放方案时面临的挑战和相关问题以及本文研究限制。