

《系统能力培养综合实践》大纲

一、课程名称：系统能力培养综合实践之模拟器

二、课程性质：必修、实践课

三、学时与学分：4 周，2 学分

四、课程先导课：计算机组成原理、计算机组成原理实验、计算机组成原理课程设计、操作系统原理、操作系统课程实验、操作系统课程设计、C 语言程序设计、C 语言程序设计实验、C 语言程序设计课程设计、汇编语言程序设计、汇编语言程序设计实验、编译原理、数据结构、数据结构课程设计、离散数学、计算机系统结构、计算机系统结构课程实验、嵌入式系统等。

五、课程介绍

计算机系统能力是指能自觉运用系统观，理解计算机系统的整体性、关联性、层次性、动态性和开放性，并用系统化方法，掌握计算机软硬件协同工作及相互作用机制的能力。系统能力包括系统分析能力、系统设计能力和系统验证及应用能力三个方面，三个方面相辅相成，共同构成计算机专业本科毕业生的基本能力和专业素养。

《系统能力培养综合实践之指令模拟器》就是为了培养计算机专业学生的系统能力而设置的。课程以软件实践为主，内容循序渐进、由浅入深，使学生能够快速入门，且帮助学生通过实践对理论、技术和方法进行巩固和理解，但是又具有高阶性、趣味性和挑战度，不断激发学生的兴趣和创造性；课程强化系统观，能够强化并检验学生的系统化综合能力；课程结合工程应用，能帮助学生理解计算机系统从底层硬件到高层软件的全套技术，使得学生对计算机系统各层次的技术有更加深刻的认知。

六、课程目标

《系统能力培养综合实践之模拟器》的具体目标包括：

目标 1: 安装与配置开发工具链。基于主流版本的虚拟机平台与开源操作系统 Linux，安装基本的代码编辑、编译、调试、版本管理软件与重要插件，如，vim、gcc、gdb、git，并掌握这些软件的基本使用方法，为下一步的系统级开发奠定坚实的基础。

目标 2: 构建基础编译环境。通过 git 工具下载模拟器软件，并根据文档完成基础开发环境的配置。通过此过程充分理解 Linux 系统中环境变量的功能与重要性，及其在 Makefile 中的常见使用方式。

目标 3:打造基础调试工具。开发一些必要的调试基础设施，如，表达式求值功能、监视点、（寄存器）比较器，为解决开发过程中可能遇到的复杂软件缺陷做好准备。

目标 4:理解模拟器的基本原理。掌握使用一段程序模拟一条（类）机器指令的基本方法，并能够在系统给出的基本框架下，结合指令集相关文档，实现 X86、MIPS 或 RISC-V 等众多主流指令集中的一个。

目标 5: 构造基本运行时环境。理解运行时环境的基本概念及其在系统中的重要作用，并通过阅读文档，在系统给出的基本框架下，完成主要库函数的功能开发。

目标 6: 构建基本输入输出环境。理解计算机系统与输入输出设备交互的基本原理与方式方法，并依据文档要求，在系统中添加对于（虚拟）键盘、时钟、图形控制器等常见输入输出设备的支持。在完成上述目标后，系统即能具备较好的可交互性与可展示性。

目标 7: 理解系统调用的基本原理与实现流程。了解计算机系统权限分级的原因与现状，理解自陷指令的功能与基本流程，理解系统调用的作用，并依据文档要求实现基本的系统调用。

目标 8: 理解文件系统的基本工作原理。了解文件系统的主要功能，依据文档要求，实现一个简化的文件系统，支撑对复杂应用所包含的独立数据文件的访问，从而使得模拟器能够支持复杂应用。

目标 9: 理解多任务系统的基本工作原理。了解多进程/任务并行的基本原理，了解虚存的基本原理及其对于多任务并行的重要性，了解分时功能的必要性，按文档要求，实现一个支持上述功能的简易多任务系统。

七、课程目标对毕业要求的支撑关系

支撑的毕业要求二级指标点	课程目标
3.3 能为计算机复杂工程问题解决方案进行系统设计，并能在设计中体现创新意识	目标 1
4.1 能够基于计算机科学的原理，通过文献研究或相关方法，调研和分析计算机复杂工程问题的解决方案	目标 2
4.2 能根据计算机复杂工程问题解决方案的特定对象特征，选择研究路线，设计实验方案、构建实验系统，并进行实验和正确采集实验数据	目标 3

4.3 能对实验结果进行理论分析，对实验现象进行解释，并能通过信息综合得到合理有效的结论	目标 4
5.2 能够根据解决计算机复杂工程问题的实际需要，选择和使用恰当的仪器、开发或选用满足特定需求的现代化工具，对具体对象进行模拟和预测，并能分析优势与局限性	目标 5
9.2 能够在多学科背景下的团队中承担负责人的角色，组织、协调和指挥团队开展工作	目标 6
11.1 理解工程项目中工程管理与经济决策的作用，熟悉计算机工程项目管理的基本方法和技术	目标 7
11.2 能将管理原理、经济方法应用于计算机工程项目的开发、设计和流程优化等过程中	目标 8
12.2 具备自主学习能力，能通过多种途径拓展自己的知识和能力，包括理解能力，归纳总结的能力和提出问题的能力等	目标 9

八、课程内容与要求

本课程按照进阶的思路布置设计开发的内容，由浅入深，不断过渡到创造性和挑战性实践。详细如下表所示。

实验名称	内容要求	实现方式要求	最高得分
1. 安装与配置开发工具链	安装基本的代码编辑、编译、调试、版本管理软件与重要插件，如，vim、gcc、gdb、git，学习这些软件的基本用法	虚拟机或物理机平台	5 分
2. 构建基础编译环境	通过 git 工具下载模拟器软件，配置环境变量，学习 make 和 makefile 的基本使用方法		
3. 打造基础调试工具	开发表达式求值、监视点、比较器等基本调试工具，并在系统开发中熟练使用上述工具。	NEMU 基础平台	20 分
4. 实现寄存器文件与指令模拟功能	用结构与联合模拟寄存器文件，并用程序实现 X86、MIPS 或 RISC-V 指令集的模拟		40 分
5. 构建运行时库	实现 klib 的基本功能函数		55 分
6. 构建基本输入输出环境	实现对于（虚拟）键盘、时钟、图形控制器等常见输入输出设备的支持		70 分

7.实现自陷与系统调用	实现自陷指令与基本的系统调用	NEMU 高级功能	85 分
8.实现简易文件系统	实现一个简易的文件系统，支持带有独立数据文件的应用（仙剑奇侠传）		80 分
9.实现多任务系统	实现虚存、分时调度等高级功能，实现一个简单的抢占式分时多任务系统，支持多个应用并行执行		100 分

九、教与学

主要的教学环节包括讲解课程要求和重、难点，学生实践，过程辅导，验收答辩，实验报告及批阅等阶段。

1. 教学方法

本课程设计的教学方法主要体现在如下几个方面：

1) 通过课堂讲授，使学生明确课程设计要求，但是课堂只着重讲授少量的基础理论和基本知识，针对学生的共性问题进行讲解；

2) 着力培养学生计算机系统能力，由浅入深，使学生能够方便的入门，但是又要具有高阶性、新颖性和挑战度，不断激发学生的兴趣和创造力，在实践过程中发展问题，采用启发式的方法，加强指导和辅导；

3) 执行严格的考勤管理，努力营造良好学习氛围，要求学生在规定地点、时间内集中完成实践；实施精细的进度控制，设置多个检查点，并做好检查记录；

4) 注重团队分工和合作，在实践过程中充分发挥每个团队成员的积极性和创造性；

5) 引入工程项目的管理和工程化方法，结合工程实际进行考核，激发学生学习兴趣，培养学生自主学习的能力。

2. 学习方法

1) 掌握有关的基本原理，注重理论指导下的工程实践；

2) 实践中遇到问题要积极思考，并基于原理和问题现象进行分析，训练发现问题、分析问题、解决问题的能力；

3) 注重团队内各成员作用的发挥，同学之间的讨论和与任课老师的交流；

- 4) 做好设计方案的顶层设计和论证；
- 5) 通过兴趣引导，激发同学们的创造力。

3. 思政元素：

指令集是计算机系统的软硬件界面，是计算机系统的核心要素，其决定了上层软件的基本形态、构建方式与运行效率。类比我国的社会主义制度，只有优秀的政治制度，才能提升社会整体效率、发展潜力与容错能力。“加快经常性事件”是计算机系统多年来高速发展的指导思想，这恰与唯物主义辩证法中对于主要矛盾的观点不谋而合，引导同学们在将来的职业生涯中更多地从哲学层面分析问题、运用哲学思想解决问题。

十. 时间分配

序号	主要内容	时间
1	课堂讲授，布置任务，集中讲解	第 1 次天
2	第一部分实验基础环境配置	第 1 天
3	第一阶段基础调试工具开发	第 1-3 天
4	第二部分基础平台开发	第 4-9 天
5	第二阶段中期进度检查（单人演示）	第 10 天
6	第三部分高级功能开发	第 10-15 天
7	最终结果检查（单人演示）	第 16 天

*每天时间：8:00 – 11:50 , 14:00 -17:30

十一、实验考核与成绩评定

1. 课程成绩构成

课程综合成绩由平时成绩、实验成绩（含验收答辩情况）和实验报告成绩三部分构成，各部分成绩的比例如下：

平时成绩：5%。主要考察出勤、团队及小组交流讨论情况。

实验成绩（含验收答辩）：65%。主要考察各实验完成的质量，其次考察演示的情况，通过指导老师评分的方式进行评价。

实验报告：30%。主要考核报告的质量（对完成内容的理解程度，对背景知识的掌握程度）和格式规范等方面的内容。

各课程目标成绩占课程综合成绩的比例如表 1 所示。

表 1：系统能力培养综合实践之蓝牙小车各课程目标成绩占比

考核项目	成绩占比	考核内容
目标 1+目	5%	考核学生对于常用开发工具的掌握情况，评价标准见表 2

标 2		
目标 3	15%	考核学生对于常见调试方法的掌握情况，评价标准见表 3
目标 4	20%	考核学生对（CPU）模拟器基本工作原理的掌握情况，专业英文技术文档阅读能力，评价标准见表 4
目标 5	15%	考核学生对常用运行时库函数的掌握程度，评价标准见表 5
目标 6	15%	考核学生对常见 IO 设备工作原理及其工作流程的掌握情况，评价标准见表 6
目标 7	10%	考核学生对系统权限分级的理解，以及常见系统调用的认知程度，评价标准见表 7
目标 8	5%	考核学生对文件系统工作原理的理解，评价标准见表 8
目标 9	15%	考核学生对于多任务系统中虚存、分时等重要技术组件的认知与掌握程度，评价标准见表 9

2. 评价标准

1) 课程目标 1 与目标 2 的评价标准如表 2 所示。

表 2 课程目标 1 与目标 2 评价标准

优秀	良好	中等-及格	不及格
能够正确安装与配置开发工具，熟练使用上述工具进行系统开发，并在开发过程中体现出良好的代码管理能力*	能够正确安装与配置开发工具，并熟练使用上述工具进行系统开发	能够正确安装与配置开发工具	不能正确安装与配置开发工具

*通过指导老师评分的方式评价

2) 课程目标 3 评价标准如表 3 所示。

表 3 课程目标 2 评价标准

优秀	良好	中等-及格	不及格
能够正确实现表达式求值功能、监视点与比较器，并能在开发过程中熟练使用这些调试工具解决实际软件缺陷	能够正确实现表达式求值功能、监视点与比较器	能够正确实现表达式求值功能、监视点，不能正确实现比较器	不能正确实现必要的调试工具，包括：基础的表达式求值功能，监视点，以及比较器

3) 课程目标 4 评价标准如表 4 所示。

表 4 课程目标 4 评价标准

优秀	良好	中等-及格	不及格
能够独立实现寄存器文件模拟，且能够模拟较为完整的指令集，所有测试用基础应用（如 CPUtest、Microbench，打字游戏等）与高级应用（马里奥或仙剑）均能正常执行	能够独立实现寄存器文件模拟，且能够模拟较为完整的指令集，所有测试用基础应用均能正常执行	能够独立实现寄存器文件模拟，且能够模拟部分的指令集，部分测试用基础应用（如 CPUtest、Microbench）能够正常执行	不能独立实现寄存器文件模拟，且不能模拟较为完整的指令集，所有测试用应用均无法正常执行

4) 课程目标 5 评价标准

课程目标 5 评价标准如表 5 所示。

表 5 课程目标 4 评价标准

优秀	良好	中等-及格	不及格
能够实现 string.c 和 stdio.c 列出的库函数,并能通过所有基础应用测试 (CPUtest, Microbench) 与高级应用测试 (马里奥或仙剑)	能够实现 string.c 和 stdio.c 列出的库函数,并能通过所有基础应用测试	能够实现 string.c 和 stdio.c 列出的库函数,并能通过自定义用例验证实现的正确性	不能独立实现 string.c 与 stdio.c 内列出的库函数

5) 课程目标 6 评价标准

课程目标 6 评价标准如表 6 所示。

表 6 课程目标 4 评价标准

优秀	良好	中等-及格	不及格
能够正确实现对时钟、键盘、图形控制器这些常用 IO 设备的支持,并通过所有应用测试 (基础与高级)	能够实现对时钟、键盘、图形控制器这些常用 IO 设备的支持,能够通过除仙剑外的应用测试	能够实现对时钟、键盘、图形控制器这些常用 IO 设备的支持,能够通过除马里奥与仙剑外的应用测试	未能实现对时钟、键盘、图形控制器这些常用 IO 设备的支持,未能通过全部基础应用测试

6) 课程目标 7 评价标准

课程目标 7 评价标准如表 7 所示。

表 7 课程目标 7 评价标准

优秀	良好	中等-及格	不及格
能够正确实现自陷指令与课程列出的常用系统调用	能够正确实现自陷指令与部分常用系统调用	能够正确实现自陷指令	未能实现自陷指令与常用系统调用

7) 课程目标 8 评价标准

课程目标 8 评价标准如表 8 所示。

表 8 课程目标 8 评价标准

优秀	良好	中等-及格	不及格
能够实现一个简易的文件系统,在此之上运行仙剑奇侠传 (读取独立的数据文件),并通过游戏存档与读档测试	能够实现一个简易的文件系统,并在此之上运行仙剑奇侠传	能够实现一个简易的文件系统,并通过文件读写测试	未能实现简易文件系统

8) 课程目标 9 评价标准

课程目标 9 评价标准如表 9 所示。

表 9 课程目标 9 评价标准

优秀	良好	中等-及格	不及格
实现了分时与虚存功	实现了虚存功能,可支	可支持简单的多道程	未能实现多任务并行执

能,可支持多个相同(内存地址空间)任务的分时(并行)执行	持多个相同任务并行执行	序并行(程序自动加载功能)	行
------------------------------	-------------	---------------	---

系统能力培养综合实践课程组

2015 年 6 月制定

2019 年 8 月修订

2020 年 1 月修订