

清 华 大 学

# 综 合 论 文 训 练

题目：基于海内外一流典型课程与学  
生需求侧分析的计算机系专业  
课程教学调研

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：计算机系本科生课程学生调研委员会

指导教师：计算机系教学工作委员会

2022 年 6 月 3 日

## 中文摘要

计算机系专门核心课程尤其是系统类课程，是培养计算机系专业人才的重要环节，而且课程本身的知识量及难度、课程作业的工作量及难度都非常具备挑战度，需要学生付出大量的时间精力学习专业能力。本调研围绕计算机系核心课程展开，从学生反馈调研和海外对标调研两个相互联系的主线入手，按照“一纵三横、点面结合”的调研策略，深入课程教学环节展开。调研结果整理归纳了同学们学习专业核心课程在多方面的学习状态和特点，从同学们关心的层面入手，结合学生反馈与海外对标，整理总结了涉及到课堂教学、课件、实验指导书等多方面多层次的教学建议。

**关键词：**计算机系课程；学生反馈；海外调研；学习特点；教学建议

## ABSTRACT

The core courses of the Department of Computer Science, especially the system courses, are important parts of the training of computer science professionals, and the amount and difficulty of the courses themselves, as well as the workload and difficulty of the coursework, are very challenging and require a lot of time and effort from students to learn professional skills. The research is focused on the core courses of the Department of Computer Science, and starts from two interrelated lines of student feedback research and overseas benchmarking research, according to the research strategy of "one vertical and three horizontal, point and surface combination", and goes deep into the teaching process of the course. The results of the survey summarized the learning status and characteristics of the students in various aspects of the core courses, and summarized the teaching suggestions involving classroom teaching, courseware, and lab manuals from the level of students' concern, combined with student feedback and overseas benchmarking.

**Keywords:** Computer Science Courses; Student Feedback; Overseas Research; Learning Characteristics; Teaching Suggestions

# 目 录

第 1 章 引言 .....	1
1.1 研究背景 .....	1
1.2 组织架构 .....	1
第 2 章 文献综述 .....	2
第 3 章 研究方法 .....	3
第 4 章 从三门专业课观察同学们的学习状态 .....	4
4.1 计算机组成原理 .....	4
4.1.1 课堂教学 .....	4
4.1.2 课程实验 .....	4
4.1.3 教学建议 .....	6
4.2 计算机网络原理 .....	8
4.2.1 课前准备 .....	8
4.2.2 课堂教学 .....	9
4.2.3 实验设置 .....	9
4.2.4 教学建议 .....	10
4.3 软件工程 .....	10
4.3.1 大作业 .....	10
4.3.2 教学建议 .....	12
第 5 章 以操作系统为例的教学环节跟进调研 .....	13
5.1 开学前四周 .....	13
5.1.1 概况 .....	13
5.1.2 结果分析 .....	13
5.2 期中考试 .....	17
5.2.1 考前辅导学生需求 .....	17
5.2.2 考后学生反馈 .....	17

第 6 章 以操作系统为例的海外调研 .....	20
6.1 各高校课程具体情况概述 .....	20
6.1.1 MIT 6.S081 概述 .....	20
6.1.2 Stanford CS140 概述 .....	21
6.1.3 CMU 15-410 概述 .....	21
6.1.4 UCB CS162 概述 .....	22
6.1.5 清华大学计算机系《操作系统》概述 .....	22
6.2 总结：各高校课程和实验设计特点 .....	23
6.2.1 MIT 6.S081 .....	23
6.2.2 Stanford CS140 .....	23
6.2.3 CMU 15-410 .....	23
6.2.4 UCB CS162 .....	24
6.2.5 清华大学计算机系《操作系统》 .....	24
6.3 经验总结与提案建议 .....	24
6.3.1 关于实验指导书的编写 .....	24
6.3.2 关于课程设计 .....	25
第 7 章 结论 .....	27
插图索引 .....	28
参考文献 .....	29

# 第 1 章 引言

## 1.1 研究背景

计算机系专业核心课程，是计算机系培养本科生计算机系统思维和能力的重要课组。在实际教学过程中，系统类课程往往是同学们在学习过程中需要投入足够多的时间精力，深入思考理解系统结构原理，努力克服困难完成实践环节的富有挑战性的课程。

本调研项目致力于深入调研同学们在系统类课程学习中的学习规律、学习状态、学习需求，同时针对性地调研海内外一流计算机专业的系统类课程教学现状与可借鉴经验，从而总结出能够提高系统类课程教学效果的经验方法，进而促使清华大学计算机科学与技术系的课程体系与教学效果取得更上一层次的提升，助力信息学科本科生人才培养，其经验方法也可以进一步推广到其他工科院系的课程实践，成为清华大学工科专业课程教学效果提升的经验素材和灵感源泉。

## 1.2 组织架构

本文的其余部分组织如下：第 2 章介绍相关的文献综述，包括学生评教、海内外课程对比、清华计算机系课程特点。第 3 章介绍调研方法，包括学生反馈调研与海外调研双线并行的调研方法与“一纵三横”的课程体系。第 4 章介绍从《计算机组成原理》、《计算机网络原理》、《软件工程》三门课程调研同学们的学习状态。第 5 章介绍《操作系统》跟进教学环节的反馈调研。第 6 章介绍《操作系统》的海外对标调研。第 7 章介绍了调研的结论。

## 第 2 章 文献综述

学生评教发展至今，有利有弊，与教学改进之间关系复杂。<sup>[1]</sup>指出了学生评教自从 20 世纪 20 年代在美国高校首先引入以来的争议，包括学生评教复杂的影响因素、学生评教和学习结果之间的关联、网络与纸笔评教的不同等。<sup>[2]</sup>通过数据分析探究了学生的课程成绩与评教分数之间的因果关系，指出了学生评教可能存在的问题，学生评教需要更加专业科学的标准和流程设计。而深入到教学环节的学生需求侧反馈沟通将会是下一个阶段的重点。“以学生为中心”，符合学生认知、引入学生决策、聚焦学生发展，将会对新的学生评教改良提供借鉴。<sup>[3]</sup>

在海内外课程对比方面，<sup>[4]</sup>对 MIT 和清华的计算机课程体系进行了对比。MIT 在具体课程评价上标准更清晰明确、更强调过程性评价。两者的课程体系虽然在结构上看起来比较类似，但是其内在逻辑和组织方式则存在底层的差异，也透露着中美两国高等教育各自的特殊性。<sup>[4-5]</sup>这也就启示着：海外课程调研可以在教学思路、教学设计、学生指导等多方面借鉴经验，但同时也需要结合具体情况具体分析。清华大学计算机系课程特色，其鲜明的特点包括：统一规划课程实验体系，建设统一的教学实验支撑平台，建设计算机系统综合实验开放式课程。<sup>[6-8]</sup>

### 第 3 章 研究方法

我们的调研框架分为两条相互补充和统一的主线，一条致力于同学需求调研，一条致力于海内外一流计算机学科的教学实践调研。

同学需求调研深入跟随课程教学全流程，分为三个阶段进行调研。我们招募了 11 名听课志愿者，每堂课结束后反馈上课体验并进行讨论，收集整理这些意见；在此之后，我们根据已经收集到的意见进一步制作收集意见的问卷或者文档，通过各班班委分发给每个班至少 30% 的上课同学；最后，针对比较关键的问题，我们在课程群中发放由简短选择题组成的问卷，了解同学们对于关键问题的意见情况。

海内外一流计算机学科典型课程的调研是带着我们在学生意见调研中发现的核心关切展开。我们深入研究系统课优秀的讲授模式、技巧、逻辑，对比不同的课件展开叙述形式对同学们学习效果的影响，查看系统类实验设计的原则和方法，调研高质量的实验指导书是如何把同学们需要的实验信息与知识叙述清楚。总而言之，都是通过典型对标课程的调研来针对性的借鉴能够解决同学们问题的可资提高教学效果的经验方法。

我们选取的课程采取“一纵三横”的策略：“一纵”，就是选定《操作系统》课程，深入跟进教学环节跟踪学生反馈展开调研，并且同时开展对标的海外调研；“三横”，是《计算机组成原理》、《计算机网络原理》、《软件工程》三门课程，在某一个时间点（例如大作业结束或者学期末）通过大规模问卷或者访谈法了解同学们的学习现状。



## 第 4 章 从三门专业课观察同学们的学习状态

本章介绍《计算机组成原理》、《计算机网络原理》、《软件工程》三门课程的调研结果，深入分析同学们在这三门计算机系专业核心课程中的学习状态和课程教学现状。

### 4.1 计算机组成原理

#### 4.1.1 课堂教学

同学们的学习状态特点归纳如下：

关于学习理论知识的主要方式：大部分同学都会选择上课听讲与看回放，而此情况下，大部分同学仍需要读 PPT 自学，或许我们可以考虑下课堂讲授中知识的传达率。

关于课堂学习的收获：有近乎半数的同学认为课堂学习有一定的收获。

关于上课积极性的影响因素：表示“对课程不感兴趣”的同学数目事实上微乎其微，而绝大部分同学们普遍赞同的一点是，不管因为何种原因，“课后总是还得自学”。同时，虽然有三成的同学感觉上课无聊而不想听，但是有五成的同学选择了注意力易丢失，也许可能是上课过快的节奏导致了这一点的发生。此外，还可以考虑其他学业压力等因素的影响。

关于提高学习兴趣：同学们更加关注学习到的课程知识在今后会有怎样的应用。同学们知道自己不是为了考试而复习考试，不是为了造机而造机，不是为了完成培养方案而完成培养方案；而我们的老师则希望在本科教学阶段尽可能地让同学们能装备上更多的知识和技能供未来发展使用。但在某些程度上，我们可能忽略了给同学们指引，这个课程中所学到的知识在将来有哪些可能的应用。也许如何填补这一信息差是未来值得我们思索的方向。

#### 4.1.2 课程实验

这一部分我们调研了同学们对于课程大实验的看法，包括对于实验基本情况的调研，对于实验中遇到问题解决方法的调研。

关于实验工作量：有三成的同学认为大实验的工作量较为合适，而大部分同学对大实验的工作量持较为谨慎的态度，甚至有三成的同学认为大实验的任务量是比较不合适的。

关于实验难度：有三成多的同学认为大实验的难度较为合适，而同样，大部分同学对大实验的实验难度持较为谨慎的态度，超过三成的同学认为实验的难度过大。

关于实验收获：同学们对于这次实验的收益是十分认可的。但是，这并不代表我们可以忽视实验任务量和难度超过了大部分同学的可承受范围的现状。少部分同学指出，目前对大实验的付出远远超出了自己的承受能力，甚至是以更加宝贵的健康为代价。这种付出是不是值得的，可能还有待商榷。

关于实验观念：这个问题的调研结果分歧较为明显。接近半数的同学对这个问题并不持认可的态度，而超过半数同学认可我系的优良传统，认为造计算机是计原课程的必要组成部分。造机实验自然是要保留，但是以什么形式保留，如何保留，才能在考虑到同学们精力有限的大前提下让同学们得到最大的收获，是我们必须要关注的问题。我们课程的目标不是为了无限度的压榨同学们的精力，而是让同学们能够学会必备的知识与技能。而这需要课程组来思考，怎样才能以合理的付出，换取合理的回报。

关于实验花费时长：平均值为 67h，中位数为 60h，极大值为 150h。有大概 25% 的同学的总付出时间超过了 100h，也就是说，仅大实验就相当于是 120 学时的内容，即相当于一门 8.33 学分的课程。而如果将这 100h 均摊到 21 天中，平均每日的投入时间需要 5h 左右。如图 4.1 所示。

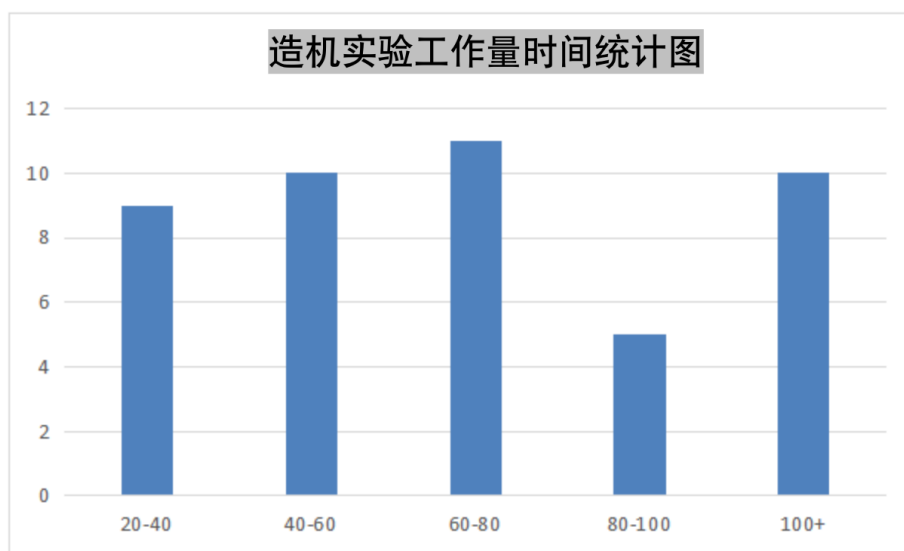


图 4.1 造机实验工作量时间统计图

关于实验实现功能：在我们的调研样本中，有四分之一的组实现了 uCore。与往届相比，同学们完成的实验功能数量再次得到了提升。部分同学认为同辈压力

的增大可能是导致这一现象的诱因，但这有待我们后续调研。

关于扩展实验与分数：有三成的同学选择“心里有一个预期分数（比如 90 分），如果确信自己到了那个分数就不再接着做了”，而现在大作业的评分标准除了 80 分和 100 分之外，这部分同学并不知道自己已经取得了多少分数，为了追求心理的安稳，他们只能继续投入无限的精力，继续加扩展需求。如果没有明确的评分标准，表中前三项总计超过七成的同学会将精力无限度地投入到扩展功能中，而如果有明确的评分标准之后，这个比例会降至三成。

### 4.1.3 教学建议

#### 4.1.3.1 课堂教学

- 比起跳跃的逻辑与思维方式，同学们更加容易理解有逻辑性与组织性的课堂讲述，这也是能让同学们提升上课兴趣的方式之一。希望老师上课时，讲稿内容与 PPT 的联系可以更加紧密，即使发散思维也要保持讲授内容前后的连贯性。
- 对于知识的讲解可以更加细致 PPT 上有时会有不知道想阐释什么的图片，或者只有不知道全称是什么的缩写，而老师在课堂上讲授这部分内容时只是读一遍课件，同学们一般还得自己重头查重头学。对他们来说，这种上课方式反而有“耽搁时间”的感觉。建议老师对于课程的重难点多进行一些讲解，多举几个更有普遍性的例子。
- 有的时候直接听讲会难以理解，但是做了一些实验再听就会觉得内容非常熟悉、能够跟上授课节奏。如果想让同学们知道，听某节课必须要有完成某项作业的背景知识，这件事应该及早通知大家。

#### 4.1.3.2 课程理念

- 创新不是无源之水、无本之木，希望在给学生提创新要求的时候，课程组要讲清什么是创新，并结合业界例子加以分析。部分同学认为，做扩展需求并不一定就代表着“创新”这个理念，反倒是看到别的组加了新功能，自己也得加类似功能的这种做法与“创新”反而背道而驰。
- 要更加注重课程作业的时间和平均学习效率。课程组应当考虑任务量，或者应当在课程开始发布作业的工作时间的期望，这样能够让学生有权衡课程之间精力分配的余地，能够更好的进行成本的衡量。
- 部分同学认为计原课程严重影响了平时学习的节奏，需要牺牲掉其他课程的学习和正常的学习节奏乃至身体健康以换取尽可能多的精力投入到计原之

中。

#### 4.1.3.3 答疑活动

同学们的建议归纳如下：

- 希望助教能多理解普通同学的问题和困难，不要总拿自己的水平去考虑问题和实验；
- 多数答疑活动的内容很基础，这种答疑可以考虑提前一些进行；
- 部分助教对内容不是很熟，答疑效果有待提升；
- 对于实验难点可以发出讲解视频；
- 超过半数同学不去参加助教答疑活动是因为觉得线下找助教太麻烦，于是我们可以考虑开启线上答疑的模式，助教设置每周固定的时间段创建“腾讯会议”，等待同学进入答疑，与线下答疑相比这无疑更加方便。

参加过答疑活动的同学普遍认可答疑的效果，这说明了答疑还是具有相当的水平。可是，参加过答疑活动的同学竟然不到半数，除去那些热爱自学的同学之外，没参加过答疑活动的同学占有了相当一部分比例。或许我们可以思考的方向是，不管是要缩短物理距离还是缩短心理距离，怎样才能让这些同学把他们想问的问题给问出来，进而得到解答。

#### 4.1.3.4 课程总体

同学们的水平是逐年提高的，这一点从大实验的扩展功能完成度便可以体现出来。而课程逐年近乎一致的优秀率限制导致课程成绩只能反应同学们在本届同学中的“相对”水平，而体现不出其相对于往年同学来说的优势或不足。我们的建议是，每年的优秀率不应是定值，而应该根据当届同学们的水平和实验完成情况进行一定的调整。也就是说，同学们所认为的优秀，不是“水平达到年级前 20%”这种相对指标，而是“完成了某项工作，掌握了某些知识，具备了某些技能”这种绝对指标。对于想取得好成绩的同学们，前者会导致他们无休止地投入精力内耗，而后者通过这一“绝对标准”的制定，虽然可能不会减少他们为了达到“优秀”的目标所需要投入的精力，但至少会让他们更加从容不迫地，不是基于同辈压力，而是基于热爱和探索精神，去进行创新。鼓励同学们有自己的学习节奏与发展节奏。

本报告提及到，同学们对于造机实验的工作量和难度认可度较低，而对于造机实验是否应保留这个问题也莫衷一是。

我们系的《离散数学》，春季学期的《计算机网络原理》，以及《数值分析》《操作系统》《编译原理》等等课程，给同学们更多的选择性。在这里我以《数值分析》

为例：

课程的考核环节有以下几部分：平时分数（雨课堂答题：平时作业：编程实验 = 1 : 2 : 3）期中考试期末考试前沿技术调研报告。

最终的考核成绩为：方案 1 =  $0.3 * \text{平时分数} + 0.3 * \text{期中考试} + 0.4 * \text{期末考试}$ ；  
方案 2 =  $0.3 * \text{平时分数} + 0.3 * \text{期中考试} + 0.4 * \text{前沿技术调研报告}$ ；  
方案 3 =  $0.3 * \text{平时分数} + 0.7 * \text{期末考试}$   
最终成绩 = max 方案 1, 方案 2, 方案 3.

作为一门既偏理论分析与理解，又偏前沿调研与应用的数学课，其方案 1 适用于平时一直跟随课堂节奏的同学，方案 3 适用于期中考试失利，或是学期中科研压力过大而导致顾此失彼的同学，方案 2 适用于不喜欢考试这种形式，而喜欢自己做调研的同学。这可以让不同类型的同学们都在这个课堂中学有所得。或许我们的课程考核也可以采用类似的方式进行设计，将基础功能与扩展功能分开，分别作为不同的考核部分，设计不同的考核方案，而将同学们在扩展功能以外的创新作为 Bonus 分数直接加入总评之中。这既能鼓励大家创新，又能满足不同类型的同学按照自己的节奏来学习和生活的需求。

## 4.2 计算机网络原理

### 4.2.1 课前准备

关于对网络课程体系组成与内容的了解：在课前，同学们对于网络相关课程体系并不熟悉，而作为网络方向的第一门课程，《计算机网络原理》应该起到向同学们介绍网络方向有哪些可学习的课程的作用。目前课程已经让超过半数的同学对后续的课程有了一定的了解，而我们希望课程可以继续努力，让剩下四成的同学也有更多的收获。崔勇老师的课堂在本学期（2022 春）已做出了一定的改革，而课程调研委员会也在对后续课程做相关调研并整理报告，以期协力达到更好的效果。

关于课前对网络知识和操作的熟悉程度：大部分同学在选修课程之前认为自己网络知识与相关操作“较为不熟悉”，而认为自己网络知识十分熟悉的同学几乎不存在。

关于修读课程前补充知识的可能途径：超过半数的同学认为，《计算机网络原理》作为同学们接触的第一门网络课程，且作为课程体系中的一门必修课，不应存在前置知识阻碍同学们对现有知识的理解。其应当类似于《程序设计基础》，帮助同学们从头开始建构整个网络领域的知识框架。

### 4.2.2 课堂教学

关于学习网理论知识的主要方式：超过半数的同学选择平时课上听讲，课下读 PPT 自学，而在考前刷往年题并和同学对答案。纵使有接近六成的同学选择了上课听讲或补看回放，但超过六成的同学还是选择需要自己读 PPT 进行自学。或许这与课程知识密度过于庞大，同学们在课上的接受能力有限有关。

关于课堂学习的主要收获：认为上课学到了感兴趣的知识的同学不到半数，认为上课对考试复习有帮助的同学不到四分之一，而认为上课与实验有关的同学更是微乎其微。或许我们课程组应该思考：上课应该给同学们带来什么。

关于上课积极性的影响因素：“对课程不感兴趣”的同学数目事实上并不多，而绝大部分同学们普遍赞同的一点是，不管因为何种原因，“课后总是还得自学”。庞大的知识量与上课过快的节奏可能是导致这一点发生的重要原因。此外，还可以考虑其他学业压力等因素的影响。

关于如何提高学习兴趣：同学们更加关注学习到的课程知识在今后会有怎样的应用。同学们知道自己不是为了考试而复习考试，不是为了造路由器而造路由器，不是为了完成培养方案而完成培养方案；而我们的老师则希望在本科教学阶段尽可能地让同学们能装备上更多的知识和技能供未来发展使用。但在某些程度上，我们可能忽略了给同学们指引，这个课程中所学到的知识在将来有哪些可能的应用。也许如何填补这一信息差是未来值得我们思索的方向。

### 4.2.3 实验设置

关于实验花费时间：平均值为 25 h，极大值为 76 h。有大概 25% 的同学的总付出时间超过了 30 h。

关于实验工作量：三成的同学对实验的工作量持中立的态度，而超过半数的同学对实验的工作量较为赞同。

关于实验难度：三成的同学对实验的难度持中立的态度，而超过半数的同学对实验的难度较为赞同。

关于实验收获：虽然同学们对实验的工作量和难度都较为赞同，但是他们投入精力之后认为收获可能并不是很明显。通过实验，认为自己收获一般的同学超过了三成，认为自己收获较多的同学不到三成。现在同学们只是按照 RFC 协议实现了路由器，而他们并不知道这个项目对于自己的现在以及未来有什么帮助。

关于实验中是否遇到困难：同学们在实验中遇到的困难并不多，接近半数的人可以独立解决，四成左右的人也只是偶尔需要求助。

关于实验中遇到困难的解决方式：绝大多数同学在遇到困难时倾向于向同学请教。

关于答疑活动的参与情况：绝大多数同学并没有参加过助教组织的答疑活动。大部分同学选择“答疑的时候没有问题需要问，而真正在遇到问题时，助教却又不在于”，怎么解决这个问题或许是课程组需要思考的。

#### 4.2.4 教学建议

##### 4.2.4.1 课堂教学

- 同学们希望能够在课前上传讲课时用的 PPT，方便直接进行批注做笔记
- 在授课方式上，希望教学团队在单纯读 PPT 之外，用便于同学们理解的方式进行讲述，多讲一些方法论。
- 在课件内容方面，同学们更希望 PPT 的内容可以言简意赅，避免冗长的大段文字的叙述方式，标出关键词与重点概念即可
- 在课堂讲授时，可以通过更多的穿插重要知识点的练习题来提升大家听讲效率

##### 4.2.4.2 实验

- 部分同学指出，实验指导书中需要着重强调一下查重扣分的事情。虽然在课上、课件中都有过强调，但对于只偷懒抄了几行代码但是忘记写参考来源的同学来说，扣掉所有分数非常亏
- 同学们认为，最后检查打印的日志特别难懂，如果写错了的话，查错极其麻烦，感觉调试无从下手

### 4.3 软件工程

#### 4.3.1 大作业

问题一：你认为目前课程大作业给分方式的设置是否合理？你是否赞同在功能实现上同组同分的方式？

这个问题同学们的回答差异较大。有半数的同学认为同组同分是合理的，而也有半数的同学认为应该按照工作量与实现功能的难易度进行梯度给分。但是，无论是赞同或者不赞同的同学，都表达了对小组合作给分具体情况具体分析的要求：如要考虑到有组员恶意划水的情况，要考虑到有同学基础较差的情况等等。

问题二：你对其他项目的难度系数有没有一定程度的了解？你是否觉得不同

项目前难度系数差异过大？(可以附上项目号具体说明)

本学期选课的同学普遍认为 PRJ 4（过程管理）与 PRJ 6（客户端配置系统）的工作量高于或远高于课程总体水平，而 PRJ 2（新人培训系统）与 PRJ 5（文物修复系统）的工作量相对较少。同学们普遍认为，要想做到公平，就必须将不同项目间的难度系数尽可能地拉平，或者在最终的给分上体现出工作量的差别。

问题三：你认为目前大作业是否有充足的指导(比如 SECODER 平台的使用)? 你是否觉得配置仓库 CI 与 Dockerfile 的过程是对课程最初阶段所讲基本技术的有益实践？

绝大部分同学认为目前大作业的指导较为不足。绝大部分同学反馈 CI/CD 的配置和 Dockerfile 的书写这两方面的指导和文档不够，体验很差，甚至有些组需要花接近两至三周的时间才能完成这部分的配置，极大地挫伤了同学们的积极性。但是，很多同学都赞同的一点是，平台助教很热心，基本上问了问题都能得到详细的答复和解决方案。于是，我们希望教学团队可以及时更新 SECODER 平台的文档与使用手册，将一对一“单点”式解决转换为广播式解决，减轻同学们在配置环境方面的负担。

问题四：你是否认为大作业给平常的学习与生活节奏带来了过大的负担？可以简单就自己对项目的总投入时间，以及预期合理的投入时间谈一谈。

绝大多数同学指出，自己在大作业中投入了每周超过 20 小时的时间，甚至有些同学会达到每周 40 到 50 小时，这相当于在有其他课业压力的情况之下，每天需要投入额外的 6 个小时来进行开发工作。大多数同学指出自己为了大作业的项目而熬夜，少部分同学提出自己为准备每周的组会展示有通宵的经历。而部分小组内工作安排又极为不均，容易产生有些组员投入了极大的时间精力，而有些同学几乎没有任务分配的情况，但最终他们拿到的分数是接近相同的。

同学们普遍认同的每周投入时间，大概是每人每周 6 到 10 小时左右。这样既大致符合一门 3 学分的课程对同学们课下实践的要求，又不会给同学们带来过大的身体与心理负担。

部分计算机金融双学位的同学与信息计算科学专业的同学反应，《软件工程》与目前学期选择的其他课程合计起来产生了巨大的压力。信计的同学指出，这学期需要修习《概率论》《复分析》等十分需要投入精力的课程。针对这一点，课程咨询委员会的建议是，希望教学团队可以提前告知选课同学们整个课程与每个项目所预期的时间投入量，以让同学们更好的规划自己的培养方案完成进度。



## 4.3.2 教学建议

### 4.3.2.1 课堂讲授

- 课堂讲授中可以增添与大作业更相关的内容和在实践过程中更加实用的知识
- 课件结构可以更加明晰，课程内容可以按照一定逻辑进行讲授并对重点进行强调
- 尽量使用同学们已经了解的知识进行举例或增加对相关知识的解释（比如在用网络协议举例时先对相关概念进行解释，因为大多数同学未修或在修网络原理课程）

### 4.3.2.2 课堂展示

- 给出展示的具体要求
- 解决同一项目内小组展示内容重复问题
- 改进互评打分的评分方式，避免设计展示时更注重形式而不是内容
- 改善每个等级限制组数的评分机制

### 4.3.2.3 大作业

总体来说，同学们认为极大的工作量与项目内互卷的氛围是同学们对这门课持有一定负面情绪的原因。此外，课程对优秀率的限制也导致增加了内卷的程度。

从客户代表的层面来说：有部分同学反应，客户代表在项目中的参与度较低，感受不到其存在。

从助教团队的层面来说：有些认为自己基础较为一般的同学反应，在项目过程中清晰地感觉到自己写了很多丑陋的代码，只是完成了功能，但是架构不算特别合理，边界条件也没有考虑全面，性能也很一般。显然有很多地方可以优化，显然有很多地方可以用一些更好的软件/技术，但是自己无从下手。希望可以有一位这样助教，他不是一个上位的考察评价者，不仅仅关注项目实现出来的效果，而能够真正看到我们的代码，从各种细节上帮我挑挑错、提提建议，让代码不只是写完了，而是写得好。比起让我们做一个“大而全”的项目，更希望可以带着我们做一个“小而美”的项目。

从课堂讲授的层面来说：少部分同学反应课堂讲授内容与大作业不匹配。课堂是业界规范，大作业偏重实践，最后感觉课堂讲授对大作业带来的提升效果不大。

## 第 5 章 以操作系统为例的教学环节跟进调研

### 5.1 开学前四周

#### 5.1.1 概况

本调研分为两个阶段。第一阶段，我们收集整理了 11 位课堂反馈志愿者的前四周上课反馈。第二阶段，我们基于第一阶段的初步认知设计问卷并发放。问卷由各班班长协助发放，共收到有效问卷 80 份。

#### 5.1.2 结果分析

##### 5.1.2.1 课程调整为全周授课

22.5% 的同学对此明确支持，支持理由包括：减小每周任务量从而更好消化课程内容、给同学更多缓冲时间。2.5% 的同学明确反对，反对理由包括：挤占期末时间、期末考试压力增大。75% 的同学对此没有什么意见，其中部分同学并不知道操作系统以前不是全周。总体来看，暂时无法确定这一调整在课程深入之后同学们是否会体验良好，但是赞成的同学相对更多。通过结果可以发现同学们主要是关心课程压力是否能够被分散减轻，部分同学会担心期末压力增大。

##### 5.1.2.2 课件

有 31% 的同学明确表示认为课件质量较高，同学们认为优点包括：便于老师修改和学生在线查阅、内容和结构清晰、提纲挈领地展现了讲课思路、绘制的图片描述清晰、模块化程度好等。有 19% 的同学提出了自己认为课件应该进一步改善的建议或问题，包括：PPT 重点内容包含更多信息、重点更突出一些、给出授课内容在教材上的对应或者相关阅读材料、部分知识点讲解缺少具体的例子、实验讲解部分没有 Ucore 示例、对课件上的提问部分提供书面答案、课前及时上传最新版课件方便做笔记、课前传课件补充 PDF 版本方便记笔记等。

##### 5.1.2.3 课堂讲授

关于课堂互动。从课堂反馈志愿者的意见来看，陈渝老师的课上互动效果较好。课前提问可以帮助同学们集中注意力，快速进入思考状态，让同学们带着疑问或者“警觉感”在听相关解释的时候更加专注。部分同学认为开头梳理每节课重点内容对自己帮助比较大。不过另一方面，部分同学提出了预热时间过长的问

觉得上课进度略显拖沓，可见关于课堂预热如何展开的问题存在不同的同学需求。对此，我们在问卷中征求了更多同学的意见，50%左右的同学明确支持课堂互动，10%左右的同学明确不支持课堂互动，其他同学保留意见。关于进一步改善课堂互动效果，同学们也提出了一些建议包括：精选问题控制提问数量、课前总结整节课要讲的内容以及前后联系（例如课前 note）提高学生进度感、每次提问点的同学不要太多。

关于具体讲授。从课堂反馈志愿者的意见来看，课件中加入了细致的名词解释，有助于同学们更好地理解概念。课堂反馈志愿者提到：陈渝老师在讲到新名词的时候及时给出定义，可以帮助同学更好地跟上上课节奏。举例来说，陈老师在讲到“执行环境”的时候马上给出了定义解释，帮助同学能跟住进度一直能听懂；而在讲“裸机程序”的时候隔了几页 PPT 才解释，则让同学短时间内没有反应过来。部分志愿者建议李国良老师讲课举例的时候，对于一些内容可以进行板书演示或者在 PPT 上画画，帮助同学们更好地理解。尤其是讲到相对复杂的结构图或者示意图的时候，希望可以结合一个具体的处理过程举例说明。

关于以上两个意见，我们又通过问卷了解到了更多同学的相关意见：关于讲到新名词的时候及时给出定义，93.75% 的同学明确支持这一做法。关于这个话题，部分同学还提出了更多的建议包括：当堂 Demo 演示、多强调重点有侧重性、通过举例子帮助理解抽象概念与过程（只讲定义有的时候不一定够）、重要的概念思路可以不只讲一次、某一章的新名词可以在整章 PPT 中出现的地方都小字标注（讲可以只讲一次，标注希望整章都标）。关于板书演示或者在 PPT 上画画，86.25% 的同学明确支持这一做法。关于这个话题，部分同学还提出了更多的建议包括：把思路讲清楚、希望图已经在 PPT 上或者提前在 PPT 做好动画（6.25% 的同学提议）、提供课程导图或者大纲等。

#### 5.1.2.4 课上实验讲解

关于课上实验讲解，课堂反馈志愿者们提出了如下困惑或建议：没开始写实验的同学心里连实验框架都没有，没有整体的把握，不明白讲课的逻辑，听完还是一头雾水，课下自己看 PPT 能明白每一页的内容，但是不明白这些内容为什么要放在一起，不懂为什么要讲那些地方，回去还是得从头看实验文档；（如果能够在接触过实验且没有完成实验的时候再听具体代码讲解也许效果更好）听代码解析的问题是大屏幕字小且没有做过实验的同学没有整体概念，容易抓不到重点而放弃听讲；建议讲实验可以讲一讲实验框架全局的理解，讲清楚每个步骤和每个函

数如何组合到一起，最终实验效果展示出来是什么样的。

在问卷调研中，更多同学的意见如下。关于课上的实验讲解，58.75% 的同学认为有需求，41.25% 的同学认为没有需求。认为有需求的理由包括：演示更加直观、避免理解偏差、提高理解效率、节约摸索时间、有助于把握什么是重点、补充指导书没有讲清楚的部分、弥合课程内容和实验的 gap、期待老师把大致框架讲清楚、理解实验思路等。认为没有需求的理由包括：耽误课程进度、还没开始做实验缺少实际操作的场景从而很难理解、部分同学偏好看指导书自学所以不想听等。关于课上的实验讲解，21.25% 的同学认为帮助较大，16.25% 的同学认为帮助不大，52.5% 的同学认为确实学到了东西但是不知道帮助大不大，10% 的同学没有听过。如图 5.1 所示。部分同学认为帮助不大的原因包括：还没写实验、习惯通过实操而不是听讲来学习实验技能、部分同学自学完框架已经理解了、实验讲解的逻辑性不够导致听得云里雾里、希望先把框架讲清楚而不是一上来就讲细节。

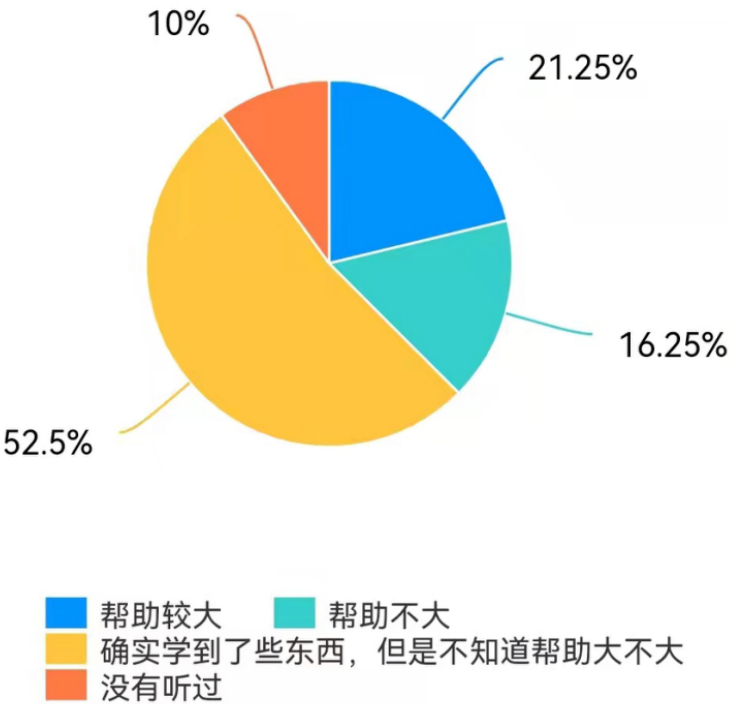


图 5.1 课上的实验讲解是否有帮助统计图

同学们提出了一些可能可以改善实验讲解效果的建议，其中提议同学占据一定比例的建议包括：1. 实验指导书把指导给全；2. 举例讲解、有实际操作和运行的展示或录屏；3. 具体代码讲解，结合代码解释难点；4. 先把框架讲解清楚，大致梳理代码框架，解释一下各个代码文件的作用；5. 只展示每个部分的代码是不够的，

需要解释框架的不同部分如何联动起来，讲解整个操作系统的运行流程；6. 讲解实验流程，提供比较明确的步骤图；7. 讲解之前先让大家做一做实验；

此外，关于改善实验讲解效果，还有一些个别同学提出的 **idea**：由做过实验的助教或者同学开小讲座；设置一个不计分的预实验由老师/助教带着一行行补齐代码；从实验目的和要求（**why**）讲起而不是直接讲怎么做；课下录制视频，方便反复观看，以及随时暂停视频动手实验；可以讲解需要做的任务在框架内的位置和实现思路（包括 **C** 和 **Rust**）；考虑到同学进度不同，完善实验指导书更加迫切；

### 5.1.2.5 实验指导书与代码框架

目前有 66.25% 的同学表示依然没有理解实验框架。对于实验指导书和代码框架，有 43.75% 的同学认为已经足够用来理解实验框架，有 13.75% 的同学认为不足以理解实验框架，另有 42.5% 不确定这件事情。关于实验框架是否需要额外讲解的问题，有 32.5% 的同学认为需要，有 62.5% 的同学认为不需要。如图 5.2 所示。

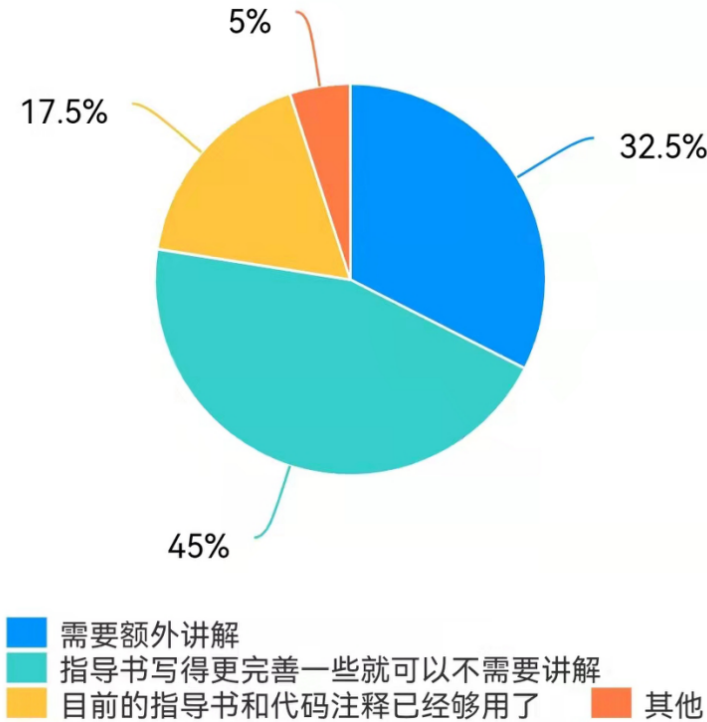


图 5.2 实验框架讲解统计图

关于实验指导书，有 31% 的同学认为帮助较大，有 16% 的同学明确提出了应该进一步改善的建议或问题，包括：详细说明总体框架、详细说明实验内容、增加阅读代码的指导例如阅读代码的推荐顺序、增加每个部分具体代码作用的说明、有一个统计收集解答学生问题的共享文档、把指导书中艰涩难懂的措辞改得清晰

易懂和语言简洁一些、Ucore 指导书内容不够、希望突出重点、增加介绍如何设计操作系统的自顶向下的讲解等。

## 5.2 期中考试

### 5.2.1 考前辅导学生需求

#### 5.2.1.1 概况

问卷由各班班长协助发放，共收到有效问卷 41 份。

#### 5.2.1.2 结果分析

对 2021 年春季和秋季的期中考试难度的感受:51% 的同学认为难度较大，27% 的同学认为难度中等，20% 的同学还没看过往年题，2% 认为考试较容易。

关于实验的分数和与实验相关的考试部分的分数之和超过总分的 50%:

71% 的同学表示认可，理由包括:

- 实验能加深对知识的理解，对能力提升有帮助
- 平时在实验中投入时间最多，应该在成绩中有较多体现
- 实验印象深刻，无需额外大量复习，减少了考试的不稳定性

29% 的同学表示反对，主要理由:

- 实验已经在有专门的部分进行考察，考试应重点考察原理
- 记不住实验框架细节，而试卷上难以给出全部代码
- 可能难以保证考察的实现细节都 uCore 和 rCore 都友好

关于授课之外的辅导：认为较有帮助的已有帮助形式包括助教的实验视频讲解、显著比以往详细的实验代码注释/详细的实践文档、rCore 指导书课后习题和参考答案、课前预习提示、课程录屏。

关于帮助信息，同学们还有如下想法：提供往年题答案、减少课前预习量、提供类似编原的 QA 墙。

关于考前辅导：收集到的想法包括安排往年题选讲，安排习题讲解。

### 5.2.2 考后学生反馈

#### 5.2.2.1 概况

本次调研收集同学们在期中考试中遇到的困难、对考试的想法以及在后半学期希望得到的帮助，由各班班长协助发放，共收到有效问卷 34 份。

### 5.2.2.2 结果分析

下面对同学们的反馈，按分数段进行归纳。

(一) [90, 100], 4 份问卷 -希望能查看考卷，知道自己错在哪里 -认为考察内容有些偏，如 rustSBI、缺页率算法时间差是否取等 -认为课堂内容和实验指导书内容重叠太大，可能导致大家不想上课

(二) [80, 90), 13 份问卷 -希望能提供往年题答案或减少往年题的内容，自己做往年题时不太会，考试时仍然不会 -关于考察的内容，如 Linux 执行 ‘1/0’ 时的输出、程序向 Linux 发出了哪些系统调用、qemu: -有同学认为不能反映 OS 课教学目的、能不能答对靠运气 -有同学表示自己没有复习这部分内容，而着重复习了 OS 实现和课件内容 -认为 lab2 没截止就考，不太合适 -认为一些简答题难以回答，认为自己答对了一些简答题但被扣了分 -认为题目表意不太清楚，没能和知识点产生联系 -感觉老师进度有些慢，lec6 有点赶

(三) [70, 80), 11 份问卷 -认为考察如“程序向 Linux 发出了哪些系统调用”等记忆性内容不太合适，因为这些内容在实际使用时只需要检索即可得到，认为应该多考察算法和实验代码 -认为考察的前两道题没有复习到，平时学习没有得到反馈，“太难了”“被爆杀” -认为 lab2 没截止就考，不太合适，自己还没怎么看过 -认为复习资料比较少 -认为课件一些问题没有讲清楚（是否包含？） -表示自己对实验熟悉不够

(四) [60, 70), 3 份问卷 -表示自己理论部分尚可，但实验丢分严重，因为自己不明确实验的重点 -表示备考中遇到一些不确定的问题，在同学小范围讨论没有有效解决 -表示自己很多概念没有搞清楚，没能从整体角度理解 OS

(五) [0, 60), 3 份问卷 -“助教小哥哥不理我” -准备退课 -希望用答案更好地总结之前错误

综合来看，以下反馈较为集中：①同学们对考察 Linux 和 qemu 的题目较为意外，一些同学认为这不是 OS 课程的重点；②同学们希望提供往年题的答案，这样在考前把没理解的知识点弄懂。此外，有很多同学都希望提供考试答案、讲解和成绩分布。由于这部分内容已经在网络学堂提供，在上面没有列举这部分内容。

### 5.2.2.3 建议

1. 如果可以的话，不妨提供一下往年题的答案或者讲解，大多数同学还是更擅长通过做题来掌握知识点。

2. 可以总结一些期中考试出题的建议：没有截止的实验尽量不要考察

3. 可以把考试出题类似风格的题目体现在平时，例如课后练习、实验的问答题，让同学们在平时就经过一些 **concept check** 或者 **practice** 来检验自己的掌握水平，及时地发现自己没有掌握的漏洞而不是拿到试卷才发现略知一二但是做不出题。（如果不想增加同学负担，可以提供这样的材料，不强制练习不占分数）

4. 考试重实验没有问题，但是建议在平时的实验环节，就可以按照考试的考察角度给同学一些深入实验具体内容的问答题。



## 第 6 章 以操作系统为例的海外调研

### 6.1 各高校课程具体情况概述

本文列出了对照表如图 6.1 所示。

	MIT	UCB	Stanford	CMU	THU
实验安排	70% Lab: 单人完成, 共10次, 每次持续时间1~2周。每次实验包括几个小任务, 为xv6添加新函数或者新功能(例如多线程素数筛法、文件查找等)(无考试)	36% Projects: 大实验, 4人组队, 1次预热+3次正式实验, 正式实验每次持续4~5周 18% Homeworks: 小作业, 进行代码补全后完成自动评测, 回答问答题(有考试)	90% Project: 3人组队, 1次预热+4次正式, 每次2周 10% Short Homework: 随堂布置, 需要在上课后第二天24:00前提交	Exams: 期中 & 期末 Programming Project 2人组队, 1次预热+3次正式实验, 每次1~2周 Homework: 2次左右, 书面题目练习 Book Report: 读一本书, 写1~2页的报告	40%实验, 单人完成, 共5次
实验设计指导	Check-off meetings (20%): 每学期参与至少3次, 需要回答助教关于大实验的问题	实验布置后一周进行 Design Review, 助教检查系统设计、尽早发现问题、评估对项目的理解	提供设计文档的模板; 进行代码安全性和易读性的检查, 如果有问题会扣分 每次实验开始前均配备助教主持的Section	实验介绍由教师完成, 作为Lecture的一部分, 讲解到实验的知识点后随即开始介绍实验要求。	每周助教答疑
实验理念告知	告知学生代码量不大, 但需要对概念的深入理解, 应当先充分阅读文档再开始编程	包含在Design Review中	包含在实验前Section中	包含在教师授课中	写在实验指导书中
预热实验	✓	✓	✓	✓	×

图 6.1 海内外课程对比图

#### 6.1.1 MIT 6.S081 概述

课程内容和评分标准: 70% Lab: 单人完成, 共 10 次, 每次持续时间 1 2 周; 20% Check-off meetings: 每学期参与至少 3 次, 需要回答助教关于大实验的问题; 10% Homework 和 participation: 阅读课堂材料, 课前提出一个问题, 课上老师会解答。

实验设计: 每次实验包括几个小任务, 为 xv6 添加新函数或者新功能(例如多线程素数筛法、文件查找等)。

每个小任务的实验指导包括: 难度分级: easy、moderate、hard; 概述任务内容: 你需要完成 xxx 函数, 为了 xxx。如果完成本任务会显示出 xxx; 完成任务的步骤: step-by-step, 应该在哪个文件写什么函数; 问答题: 完成后需解释出现结果的原因; 需要提交: 代码、问答题答案、完成实验花费的时间。

难度分级: 提供预计的完成时间; 告知学生如果超出完成时间太多, 立即向助教寻求帮助; 告知学生代码量不大, 但需要对概念的深入理解, 应先充分阅读文档再开始编程; 设置扩展题目, 供感兴趣的同学探索, 不计分。

### 6.1.2 Stanford CS140 概述

课程内容和评分标准: 90% Project: 3 人组队, 1 次预热 +4 次正式, 每次 2 周; 10% Short Homework: 随堂布置, 需要在上课后第二天 24:00 前提交。

实验设计: 四次实验内容分别是: 线程、用户程序、虚拟内存、文件系统, 每次均为实现 Pintos 系统的部分功能; 评分: 50% 测例 + 50% 代码和文档; 提供设计文档的模板; 助教会进行代码安全性和易读性的检查, 如果有问题会扣分; 每次实验开始前均配备一次助教主持的 Section, 进行实验相关的知识点讲解、解释实验文档的要求、告知建议的实现顺序和其他实用建议。

文档包括:

1. 实验概述: 实验介绍和需要提前阅读的资料 (链接到附录); 回顾必要的知识点; 提供建议的实现顺序和思路, 具体到函数或文件。
2. 需求: 每个小任务均包含: 任务内容介绍、实现思路、需要实现的函数的格式和功能。
3. FAQ: 包含代码量的展示和常见错误的回答。

### 6.1.3 CMU 15-410 概述

课程内容和评分标准: Exams: 期中考试和期末考试; Programming Project: 2 人组队, 1 次预热 +3 次正式实验, 每次持续 1 2 周; Homework: 2 次左右, 书面题目练习; Book Report: 读一本书, 写 1 2 页的报告; 不同部分的分数占比根据是否有疫情而不同; 无疫情: 55% 实验、35% 考试、10% 作业和读书报告; 有疫情: 75% 实验、12% 考试、6% 作业、7% 读书报告。

实验设计: 实验 1 单人完成, 为基础知识点实验环境 Simics 的熟悉; 实验 2 开始双人合作。实验介绍由教师完成, 作为 Lecture 的一部分, 讲解到实验的知识点后随即开始介绍实验要求。CMU 的大实验资料较为分散, 不存在统一完整的实验文档, 且每次实验的资料可能是由不同的助教维护, 编写风格不同, 故实验文档的参考意义有限。以第一次实验的文档为例, 包括: 实验目的: 既有实验需要完成的内容 (三个硬件设备驱动), 也包含其他软性要求 (代码简洁性、注释完整度、使用伪代码架构、使用 Simics、gdb 等调试工具、和 TA 交流等)。重要日期: 作为检查自己进度的参考。实验相关知识点介绍: 内核态编程、中断异常、硬件规格和接口、Simics 介绍。告知测试方法、文档撰写原则、代码风格原则。其他上手建议。

读书报告: 课程提供书单, 学生确定阅读书目, 根据推荐的进度阅读并最终提交阅读报告。要求着眼于三个问题: 1. 教育意义: 材料中最出乎意料的、最有趣

的部分是什么？2. 建议：你是否会推荐材料中的某个部分给身边的人？为什么？3. 作业的价值：对于读书报告的形式建议？

#### 6.1.4 UCB CS162 概述

课程内容和评分标准：36% Exams：每次 Exam 包含 1/3 学期的内容，与 Projects 强相关；36% Projects：大实验，4 人组队，1 次预热 +3 次正式实验，正式实验每次持续 4~5 周；18% Homework：小作业，进行代码补全后完成自动评测，回答问题；10% Discussion：概念解释、书面题目练习、答疑等。

实验设计：15% Design + 70% Code + 15% Report；实验布置后一周进行 Design Review，助教检查系统设计、尽早发现问题、评估对项目的理解。

文档包括：任务介绍和 Setup；要求实现的功能；问答题；提供 suggested plan（实验完成顺序），给出阶段性 checkpoint（不计分）；结尾有 FAQ，包括助教的代码量、常见错误的原因；详细的文档附在附录，介绍框架的编写和运行模式，供翻阅查找和感兴趣的同学进一步了解；有白盒测试，需要保证代码简洁、易懂、安全；需要提交：代码、介绍系统设计的文档、组内互评（均分 20）。

#### 6.1.5 清华大学计算机系《操作系统》概述

课程内容和评分标准：40% 实验：单人完成，共 5 次，每次持续 2 周。60% 考试：其中 30% 期中、30% 期末。可以选择更具挑战性的大实验，评分标准改为 30% 中期汇报、30% 期末汇报，40% 实验要求在中期汇报前完成。

实验单人完成，实验指导书按照内容模块划分共有 10 章，每一章均包含以下部分。

- 引言
  - 本章知识点的诞生背景、作用
  - Setup 部分：框架代码的获取和运行方式
  - 代码树结构
  - 代码导读：介绍代码中不同文件和数据结构配合的方式
- 知识点介绍
- 课后练习
  - 分为编程题和问答题，题目用星号的数量简单标识难度（一星容易、两星有一定工作量、三星有难度）。
  - 课后练习提供参考答案，用于自查知识点掌握情况和帮助考试复习
- 实验练习

- 包含实践作业和问答作业，为实验中考察的部分。
- 文档分为以下几部分：简易的难度提示（可能会出现）；实验简介；需要实现的函数功能、参数、可能的错误；实验要求和选做内容；实验报告要求

## 6.2 总结：各高校课程和实验设计特点

### 6.2.1 MIT 6.S081

1. 实验为主：xv6 实验为课程的最主要部分（评分占 90%），10 次实验的数量也是最多的。无考试。
2. 独特的课前提问设计：每次课前阅读提供的资料并提出一个问题，老师上课进行解答。
3. 每次实验均拆分为多个小任务，每个任务均有难度分级，提供完成时间的参考。
4. 提供建议的实验完成顺序，精确到文件或函数。
5. 实验问答题均为对实验现象原因的解釋。
6. 除了提交实验代码和文档，还需要提交完成实验的时间供统计分析。

### 6.2.2 Stanford CS140

1. 实验为主、综合考察：大实验占最主要部分（评分占 90%），小组完成，同时考察设计、代码实现、代码风格，设计方案、代码和标准化测试均有一定比重。无考试。
2. 每次实验前有 Section 讲座，其中会进行知识点讲解，和 Lecture 形成互补。Lecture 中讲解理论性较强的知识，Section 中讲解和实验强相关的知识。
3. 每次大实验均提供助教的 Git commit 记录指示代码量。
4. 小作业时效性强：课上布置第二天就需要提交，用于帮助及时复习课上内容。

### 6.2.3 CMU 15-410

1. 实验和考试并重：具备期中期末考试、同时实验任务量也相对较大。评分根据是否有疫情有两套方案，有疫情时考试占比减小、大幅增加实验占比。
2. 实验说明作为授课的一部分、由教师完成：讲到实验涉及到的主要知识点时，顺势布置实验。

3. 提供时间点参考：“到 x 月 x 日，你应该已经完成了 xxxxx”。

4. 独特的读书报告设计：整个学期需要阅读完一本书（并提供了书单范围和参考进度），讲出读书中感兴趣的部分。

#### 6.2.4 UCB CS162

1. 课程组成最为细碎：考试、小实验、大实验、小班讨论。（因此助教团队也最为庞大）

2. 独特的小班讨论设计：概念回顾、概念考核、书面题目练习。

3. 大实验融合了软件工程的要求：要求提交设计文档、实验开始一周后需要对设计进行答辩、最高的代码风格和文档要求、最严格的白盒测试。

4. 每次大实验均提供助教的 Git commit 记录指示代码量。

5. 提供建议的实验完成顺序、给出不计分的 Checkpoint。

6. 除了提交实验设计和代码，还需要进行组内互评。

#### 6.2.5 清华大学计算机系《操作系统》

1. 实验和考试并重，但考试会大量涉及实验的内容，因此实验重要性很高。可以选择做更有挑战性的实验。

2. 实验指导书详尽，达到教材的水平，可以用于课后练习和考前复习。

3. 提供最详尽的函数接口、功能、可能的错误的说明。

### 6.3 经验总结与提案建议

#### 6.3.1 关于实验指导书的编写

1. 每个任务最开始应先用一定的篇幅概述内容，让学生了解任务在操作系统中的位置以及实验的价值，激发同学们的问题意识（前一阶段的实验缺陷）、目的意识（本次任务实现的功能如何工作）。

例如：前一阶段的实验仍存在的缺陷（“上一阶段中，我们完成了 xxx，但无法应对 xxx 的情况”）；本次任务实现的功能如何工作（“你需要实现 xxx 函数 / 功能，它会先 xxx，再 xxx，从而达到 xxx 的效果”）。

2. 对于实验任务的说明应当给出实现思路、及正确完成任务时的现象，让同学们知道比较明确的可操作步骤，避免看完文档不知从何处下手。

例如：实现思路：“先编写 xxx 文件中的 xxx 函数，再添加 xxx 函数…”、“先实现 xxx 功能，再添加 xxx 功能…”；正确完成的现象：“完成此步后能看到 xxx 输

出 / 完成此步后可以通过 xxx 测试点”，发挥 checkpoint 的指导作用。

3. 应当给出任务难度的直观表述（完成时间和/或代码量）。

MIT 通过任务难度分级提供参考完成时间；UCB 和 Stanford 通过助教的 Git commit 信息提供大致的代码量；CMU 提供参考时间节点跟进进度。

有两个目的：便于同学合理安排完成作业的时间；当同学发现自己的投入远大于给出的参考时可以最及时寻求帮助。

4. 实验任务文档应当独立出来，并尽量简洁，便于学生最快了解实验要求，不致于大量的阅读后丧失兴趣和信心。框架的实现方法和运行机制说明，应当与实验任务文档分开。前者是供查阅 / 感兴趣的同学深入研究使用的，可以作为附录；而实验任务文档正文应尽量简洁。

例如：将现有指导书中，每次实验所涉及章节的“引言”部分和“实验练习”部分全部拿出来并组合，形成一个更直观的实验任务文档。

5. 对于 Setup、实验环境搭建等部分，提供具体执行代码，不设门槛。

6. 应当进行适当的进度提示和鼓励，有利于学生了解完成进度、获得正反馈。

例如：每次任务提供多个测试点，每个步骤可以通过部分测试点，直至最后通过全部测试点即完成实验；给出正确完成任务后的现象（参见第 2 条）；直接告知进度（“你已经完成了一半了！”、“截至此，实验全部完成，你应当获得了全部的代码分数”）。

### 6.3.2 关于课程设计

1. 实验的难点应当设置在对于实验现象的观察和背后的原理分析，而非实现思路 and 操作的执行。

MIT 的 Check-off meetings、Stanford 的设计文档、UCB 的 Design Review 均要求回答实验的设计思路，所以要求学生不能仅仅让系统跑起来，还需要了解背后的原理。海外高校的实验指导书中均会提供大致的实现思路，甚至会精确到文件和函数。

具体操作例如：在实验指导中提供大致的实现思路。（参见上一节第 2 条）；将问答题和实验有机结合，着重于对实验现象的观察和探索实验现象产生的原因。回答作为实验报告的一部分参与计分。

2. 实验内容应当与课程讲授内容区分开，课堂讲授内容应当设置预习或课后练习辅助理解，课程讲授内容应当是更好进行原理分析的基础。

MIT 的材料阅读和课前提问、UCB 的 Discussions、Stanford 的小作业均聚焦

于概念理解（Concept Check）或书面题目演练，其目的都是辅助课堂教学，二者均计入最终成绩。

实验均布置在对应知识点的授课结束后，实验的分析需要先透彻了解概念、考试需要充分了解实验，形成学习链条。

3. 采用“迟交额度”的政策，提供实验 ddl 一定程度的宽松度，以应对突发情况。

调研的海外高校均有类似的机制：整个学期有一定的迟交份额（MIT 为 72 小时；UCB 为大实验 4 天、小实验 3 天；CMU 为 3 天），学生可以使用份额而分数不受任何影响，但需要自行合理分配。列举的部分使用场景：生病、实习面试、文艺体育等学生活动、其他课程的 ddl 紧迫、提交失误。如果迟交份额使用完，将会按照规则严格扣除分数。

## 第 7 章 结论

学生反馈调研方面，我们发现大部分同学都会选择上课听讲或者看课堂回放，而且半数以上的同学认为上课是有收获的，说明课堂教学是对同学们来说是学习知识不可或缺的环节，课堂听课本身的质量也就显得非常重要。我们也发现大部分同学在听课之后仍然需要读 PPT 或者看书自学才能学懂知识，这种“课后还得自学”的情况成为了影响大多数同学上课积极性的原因。关于上课积极性，除了“课后需要自学”这个主流因素外，有一半的同学都认为“上课注意力易丢失”，而“对课程不感兴趣”的同学并不多。在具体讲授方面，讲到新名词的时候及时给出定义和板书演示/PPT 写画是对于同学们来说非常重要的两个做法。在课上讲解实验时，实际操作和运行（展示或录屏）、大致梳理代码框架、提前让大家接触实验，都是非常有效的教学方法。在实验环节，专业核心课程实验的工作量和挑战度决定了同学们势必需要付出大量的时间精力。与之相关的实验指导以及实验评分等环节，不同的课程则体现出了不同的特点，值得在本调研的基础上继续深入。

在海外调研方面，我们选取的海外课程包括 MIT 6.S081、Stanford CS140、CMU 15-410、UCB CS162，在大体了解课程内容和评分组成的前提下，关注实验设计和实验指导书的编写方法。整理了不同学校的课程和实验设计特点，总结出关于实验指导书编写和实验设计的若干建议。我们对实验指导书编写的建议包括：每个任务最开始需要帮助同学了解清楚任务在操作系统全局中的位置和任务本身的问题和目的、给出明确的实现思路和步骤以及 checkpoint、给出任务所需的时间或者代码量参考、实验指导书的任务文档和知识查阅分离等。我们对实验设计的建议包括：重视实验现象的观察和背后的原理分析、课堂讲授为实验原理分析提供基础、借鉴“迟交制度”等。



## 插图索引

图 4.1	造机实验工作量时间统计图 .....	5
图 5.1	课上的实验讲解是否有帮助统计图 .....	15
图 5.2	实验框架讲解统计图 .....	16
图 6.1	海内外课程对比图 .....	20

## 参考文献

- [1] 胡乐乐. 美国高校学生评教研究 90 年: 历史, 争议与反思[J]. 高等教育研究, 2017(9): 91-109.
- [2] 哈巍, 赵颖. 教学相“涨”: 高校学生成绩和评教分数双重膨胀研究[J]. 社会学研究, 2019, 1: 84-105.
- [3] 李贞刚, 陈强, 孙婷婷. “以学生为中心”改进学生评教的思考与实践[J]. 现代教育管理, 2019(1): 62-66.
- [4] 文雯, 周璐, 芮振华, 等. 形似与神异: 中美研究型大学课程体系比较——以两所顶尖研究型大学计算机本科专业为例[J]. 高等工程教育研究, 2022(1): 175-181.
- [5] 杨欢, 周竞文, 周海芳. 若干世界一流大学计算机基础课程调研[J]. 计算机教育, 2020.
- [6] 刘卫东, 张悠慧, 向勇, 等. 面向系统能力培养的计算机专业课程体系建设实践[J]. 中国大学教学, 2014(008): 48-52.
- [7] 李山山, 刘卫东. 面向系统能力的计算机组成原理实验实施[J]. 计算机教育, 2014(15): 107-110.
- [8] 赵赞. 中国大学计算机专业课程设置的历史沿革及改革前瞻[D]. 辽宁师范大学, 2007.