

面向系统能力的 操作系统实验教学

北京航空航天大学

王雷 龙翔 沃天宇 孙海龙 姜博

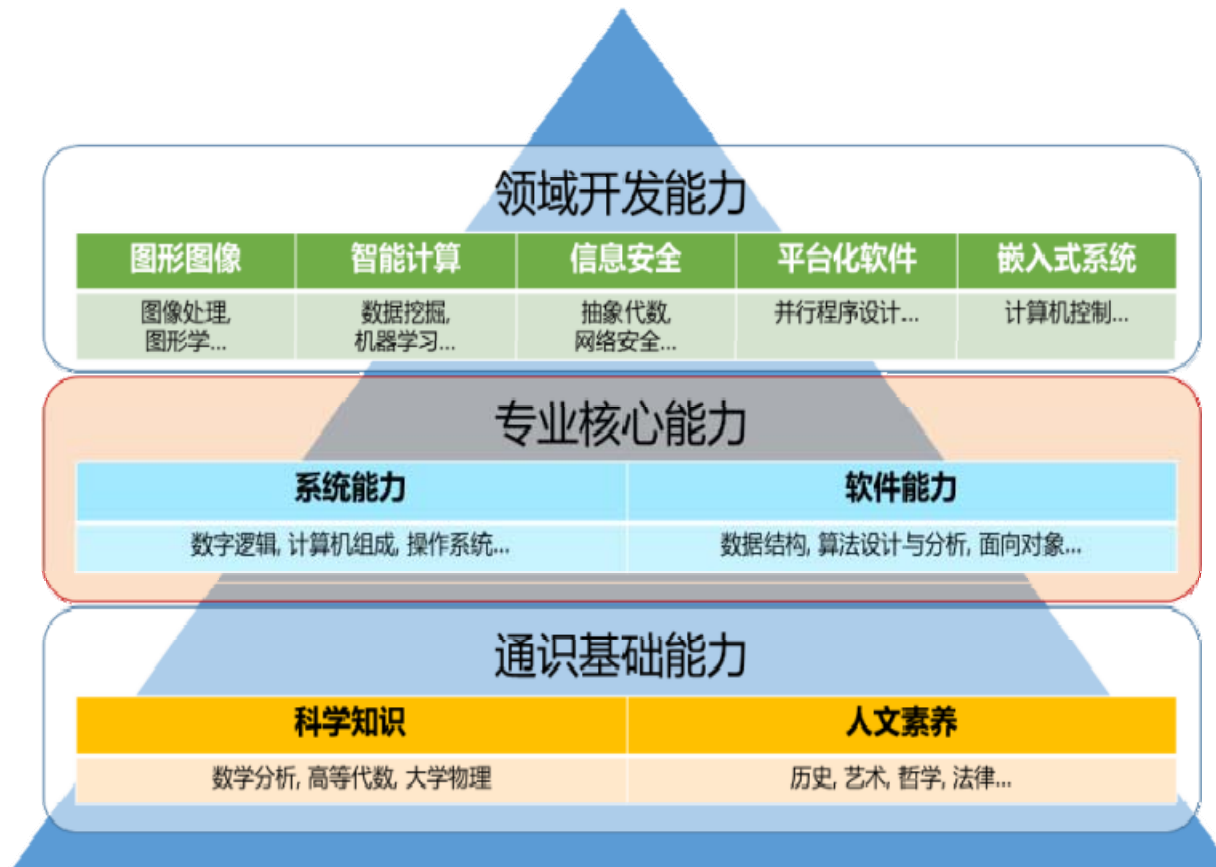
2018年11月23日

计算机专业核心能力

- 系统能力是依据确定的系统功能，设计与开发系统结构，实现工程目标的能力。
- 系统能力包括系统知识和工程实践。**系统知识**是掌握计算机核心系统的工作原理、构造方法、软硬件协同、层次间逻辑关联，**工程实践**是用工程方法开发计算机应用系统。

计算机专业核心能力

- 2006年北航计算机学院将“系统能力”和“软件能力”明确定义为本科生的专业核心能力，并始终为本科教学建设的重心所在。



OS实验的设计目标

总体目标：

- 提供工程化方法指导学生设计操作系统，使学生能在一学期内（部分）实现一个小型操作系统。

目的：

- 实践操作系统课程原理，填补OS理论与实践的鸿沟，增强学生实现具有复杂软件的能力，
- 尽量建立核心课程之间联系，操作系统课程设计与硬件课程（数字逻辑，组成原理）等衔接。

OS实验的设计目标

- 如何用工程化的方法指导学生完成OS设计?
 - 分阶段逐级制定实验任务
 - 构建OS框架引导
- 如何建立核心课程之间联系?
 - 和组成原理和编译原理一致，基于MIPS体系结构
 - 逐步完善计组的平台，与OS对接
 - 让有兴趣的学生开展探索实验
 - 例如，在自己实现的CPU平台，利用中断实现进程切换

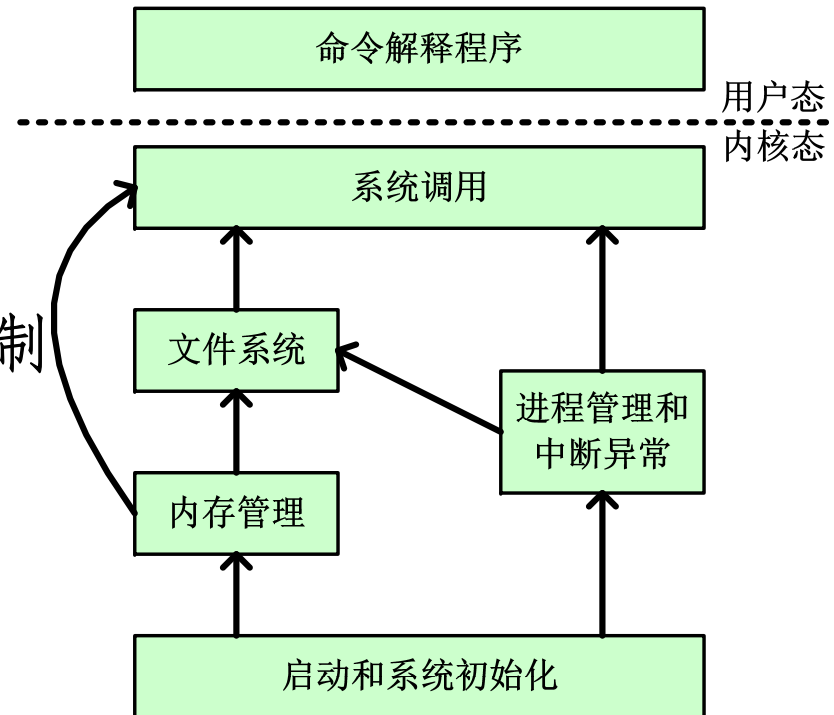
操作系统实验实施

- 2007年引入MIT的操作系统实验
- 2008年Prof. Frans Kaashoek 清华讲课
- 2009~2010年MIPS移植
- 2008~2012年分组与小规模实施
- 2013~2018年独立完成与大规模实施

实验的各个部分及相互关系

■ 实验设计（在MIPS仿真器）

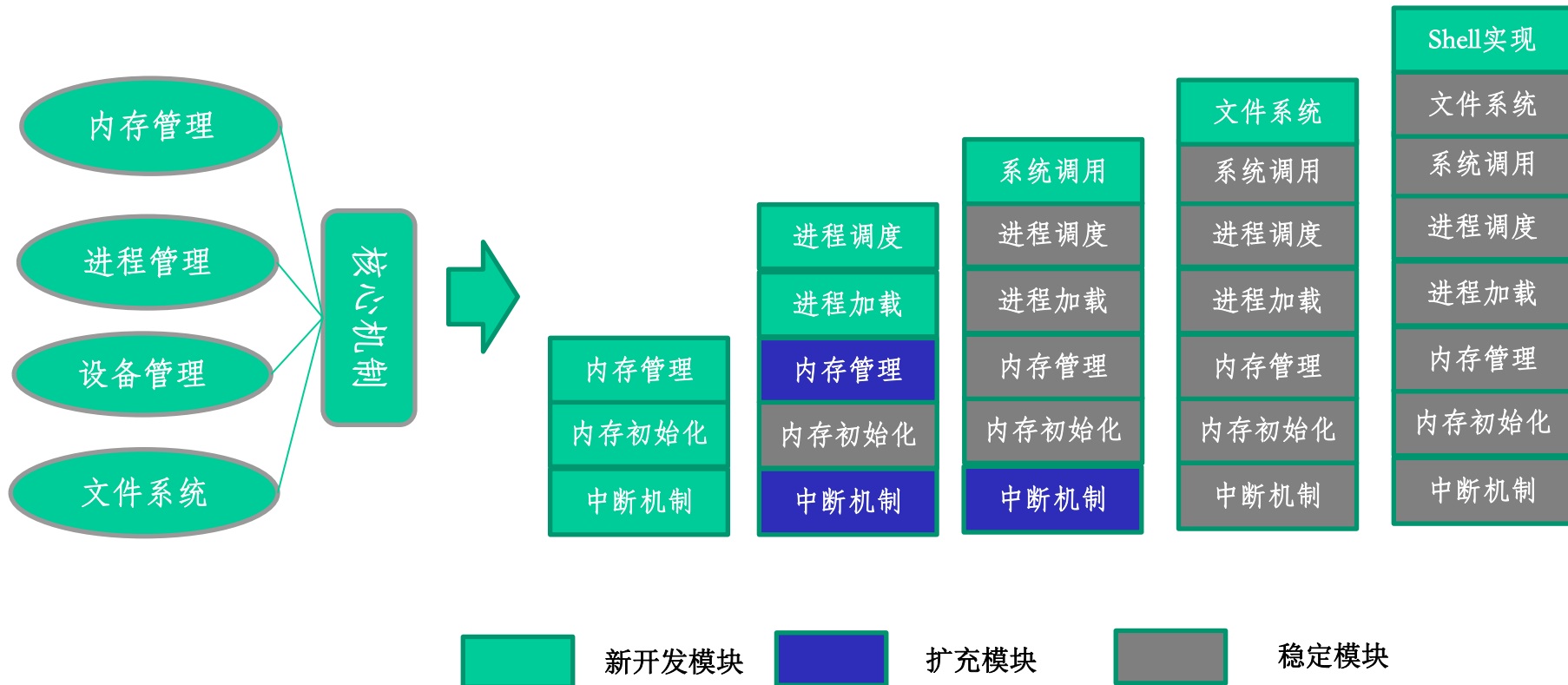
- lab0: 基础知识
- lab1: 启动和系统初始化
- lab2: 内存管理
- lab3: 进程管理和中断异常机制
- lab4: 系统调用
- lab5: 文件系统
- lab6: 命令解释程序



基于工程化方法的实验设计

原理

lab 1-6



实验报告和兴趣小组

- 课下任务+课上测试
- 实验报告
 - 代码分析与问题回答
- 兴趣小组（可选）
 - 面向自己CPU的移植（计算机组成原理课所搭建的CPU）
 - 面向ARM结构的移植
 - 面向QEMU的移植

实验实施

- 2013年（实验班20人）
 - 100%完成实验4
 - 70%完成实验5和6
- 2014年（255人）
 - 38%完成实验4
 - 8%完成实验5和6
- 2015年 156
- 2016年 147
- 2017年 282
- 2018年 397

教学存在的问题

- 如何给出较有层次的指导？
 - 如何让学生能够自主的一步一步实现OS系统的构建？
 - 如何让大量学生在开发中及时得到帮助？

教学存在的问题

- 如何给出较有针对性的指导？
 - 如何让大量学生在开发中及时得到帮助？
 - 让学生能够自主的一步步逐步实现OS系统的构建？

实验任务的层次化设计

管理学习内容

课件与实验指导书发布

在线预习（操作过程视频）

在线答疑

通知与公告

作业（实验报告）提交

教学存在的问题

- 实验教学如何扩展?
 - 25人（2013年）~ 397人（2018）
 - 如何有效管理实验的过程？
 - 如何快速给出评测结果？
 - 学生来自不同学院，能力和态度存在差异。

教学存在的问题

■ 实验教学如何扩展？

- 25人（2013年）~397人（2018）
- 如何有效管理实验的过程？
- 如何快速给出评测结果？
- 学生来自不同学院，能力和态度存在差异。

管理学习过程

构建在线实验操作环境

包括完整工具链：虚拟机、模拟器、git、编译器

管理学习成绩

自动化在线评测（基于git hook）

自动在线统计与发布成绩（for学生，for教师）

针对前几次实验成绩不理想的同学，调整激励政策

教学存在的问题

- 如何对教学进行反馈，完善教学
 - 仅从有限的作业、测验和考试结果等数据反馈，不够及时也不够完善。
 - 如何及时反馈代码提交的结果？
 - 如何发现学生知识储备的短板？
 - 如何及时了解实验设置的问题？

教学存在的问题

- 如何对教学进行反馈，完善教学
 - 仅从有限的作业、测验和考试结果等数据反馈。不够及时也不够完善。
 - 如何发现知识储备的短板?

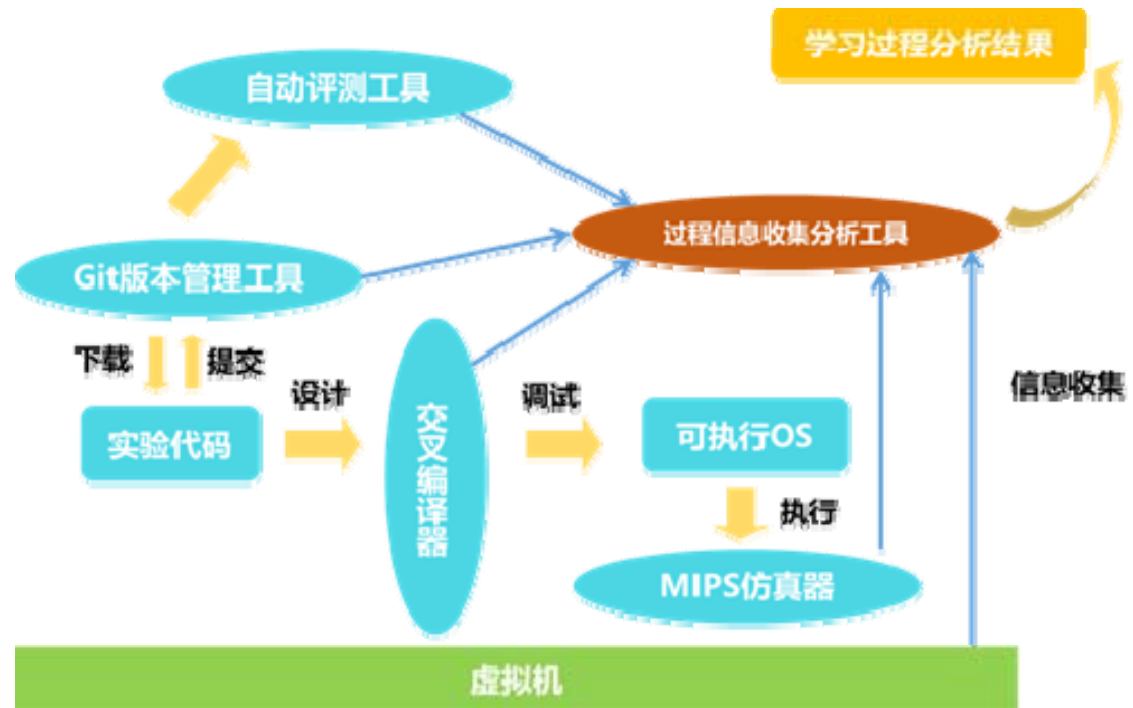
持续监控学习过程，改进教学效果：

1. 全程操作过程记录:学习时长、提交频次、按键、命令、打开文件等大量细粒度的开发行为数据。以便在微观层次分析和发现能力形成的规律；
2. 数据驱动的实验效果评估

操作系统实验集成环境结构

系统分为以下几个部件：

- **虚拟机**：包含实验需要的开发环境，例如Linux环境、交叉编译器、MISP仿真器等。
- **git版本管理工具**：包括学生各实验的代码以及相关信息。
- **自动评测工具**：这部分集成在git服务器中，自动完成学生提交的实验代码的测试。
- **过程信息收集分析工具**：集成在虚拟机中收集学生学习行为数据。

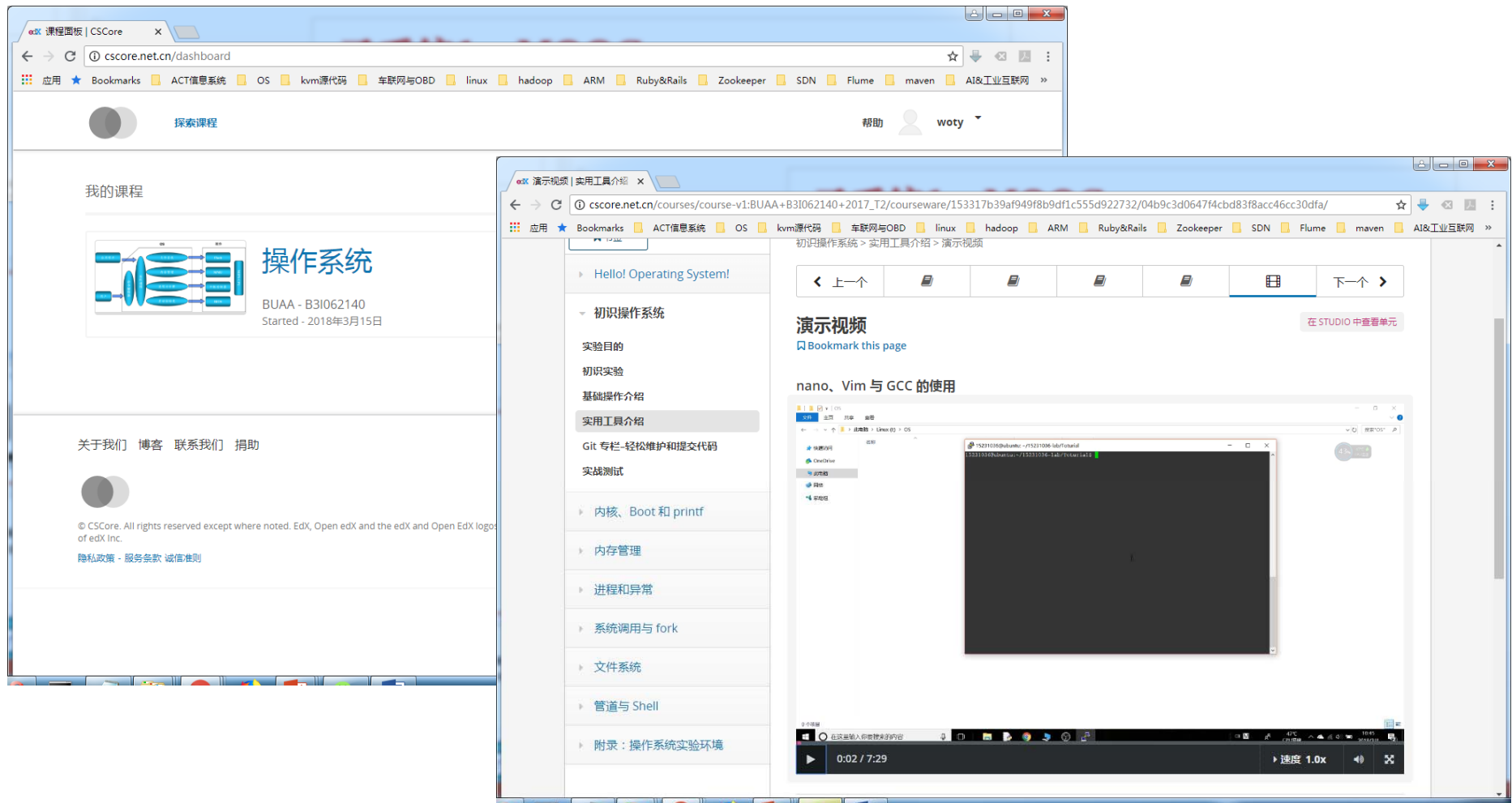


操作系统实验环境总体结构

- 子系统1：学习资料管理
 - SPOC系统+课程中心
- 子系统2：学生在线实验环境
 - 统一、受控、在线访问实验环境（命令行）
 - 规模可扩展、安全可靠
 - 全过程记录
- 子系统3：代码管理与自动评测系统（Git）
 - 学生代码管理（分支管理）
 - 实验任务自动测试（按照分枝名自动触发相应评测）
 - 增量学习过程管理（根据评测成绩自动下发新实验代码）
 - 成绩收集与发布
- 子系统4：离线数据分析（建设中）

子系统1： 学习资料管理

- SPOC系统——基于OpenEdx的MOOC系统



SPOC平台简介

- 基于开源教育应用 Open edX
- 提供实验步骤指导、视频演示、在线讨论、离线指导书发布
- 网址：<http://cscore.net.cn>
- 支持外网访问

操作系统

Help liukangxu

Discussion Instructor

to BUAA's B3I062140!

Course Updates and News

March 14, 2018

欢迎来到操作系统实验课程网站!

Hide

Course Handouts

课程资料

- 由于实验不断更新迭代，请以CSCore网站发布的内容为准，PDF版本仅供离线环境下参考。
- [最新版本实验指导书 \(PDF版本\)](#)

Hi, 大家好，我们是操作系统实验课程团队，欢迎大家在本学期中与我们一道探索操作系统背后的奥秘。相信大家开始这门课程前已经从师兄师姐口中了解到了一些“江湖传言”，不论大家听到的版本是怎样的，我们可以肯定地告诉大家：操作系统实验这门课程始终在不断地演进，我们在一次次的迭代过程中，对实验代码、指导书乃至课程组织结构、评价体系等不断进行优化，只为了能够帮助大家更好地完成实验并从中感悟操作系统设计与实现所蕴含的思想和技巧。在整个实验流程中，除了基于Linux的虚拟实验环境外，最重要的一环便是本课程网站，它是大家学习操作系统实验的利器。课程网站除了为大家提供详尽的实验指导（其中还包含关键操作的视频教程），同时也为同学间相互交流或与助教、老师直接沟通提供了平台。

平台功能

进入课程后，课程主页包含以下内容：

- 课程通知（请随时关注重要通知）
- 资料下载（由于迭代较快，离线版指导书仅供参考，强烈**不建议整本打印**）

以及课程内容和讨论区的跳转链接

- 课程内容（逐周发布实验过程指导）
- 线上讨论（探讨实验中的疑问）

课程主页
课程内容
线上讨论

The screenshot shows a web page for 'BUAA's B3I062140! 操作系统'. At the top, there are three navigation buttons: 'Home', 'Course', and 'Discussion'. The 'Course' button is highlighted with a red box and labeled '课程主页'. Below the navigation is a 'Welcome to BUAA's B3I062140! 操作系统' message. The main content area is titled 'Course Updates and News' and contains a post dated 'March 14, 2018' with the text '欢迎来到操作系统实验课程网站!'. The post body contains a welcome message from the course team. To the right of the post, there is a 'Course Handouts' section titled '课程资料' with a list of links, including '最新版本实验指导书 (PDF版本)'. This section is highlighted with a red box and labeled '资料下载'. The 'Course Updates and News' section is also highlighted with a red box and labeled '课程通知'.



线上讨论

除顶部导航栏入口外，还可通过每个课程页面底部的入口访问，如右图所示。

发帖时请根据实际情况选择“提问”或“讨论”类型，若他人的回答成功解决问题，请点击“标记为答案”按钮。

“举手之劳，造福他人”

Discussion

Topic: 初识操作系统: 初识实验 / ssh—远程访问我们的环境

Show Discussion

< Previous

Next >

Discussion

Topic: 初识操作系统: 初识实验 / ssh—远程访问我们的环境

Hide Discussion

发帖

Add a Post

? SSH登录出错 (connection timeout) 怎么解决?

如题, 在使用 putty 登录实验环境时, 出现如下错误:

★ Following

1

< Previous

Next >

当前页面所有相关讨论

Add a Post

Post type

Questions raise issues that need answers. Discussions share ideas and start conversations.

? Question

Discussion



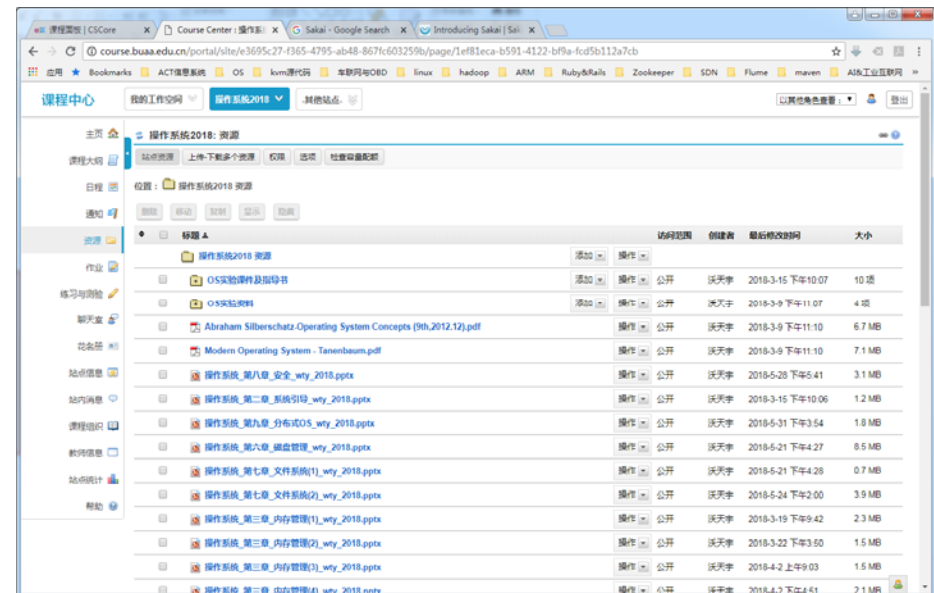
子系统1：学习资料管理

■ 北航课程中心

- 学校基础设施
- 基于Sakai (<https://sakaiproject.org/>)
- Since 2005, University of Michigan, Indiana University, MIT and Stanford University, University of California, Berkeley and Foothill Community College



北京航空航天大学



计算机学院

王雷



子系统2：学生在线实验环境

■ 对学生的操作要求

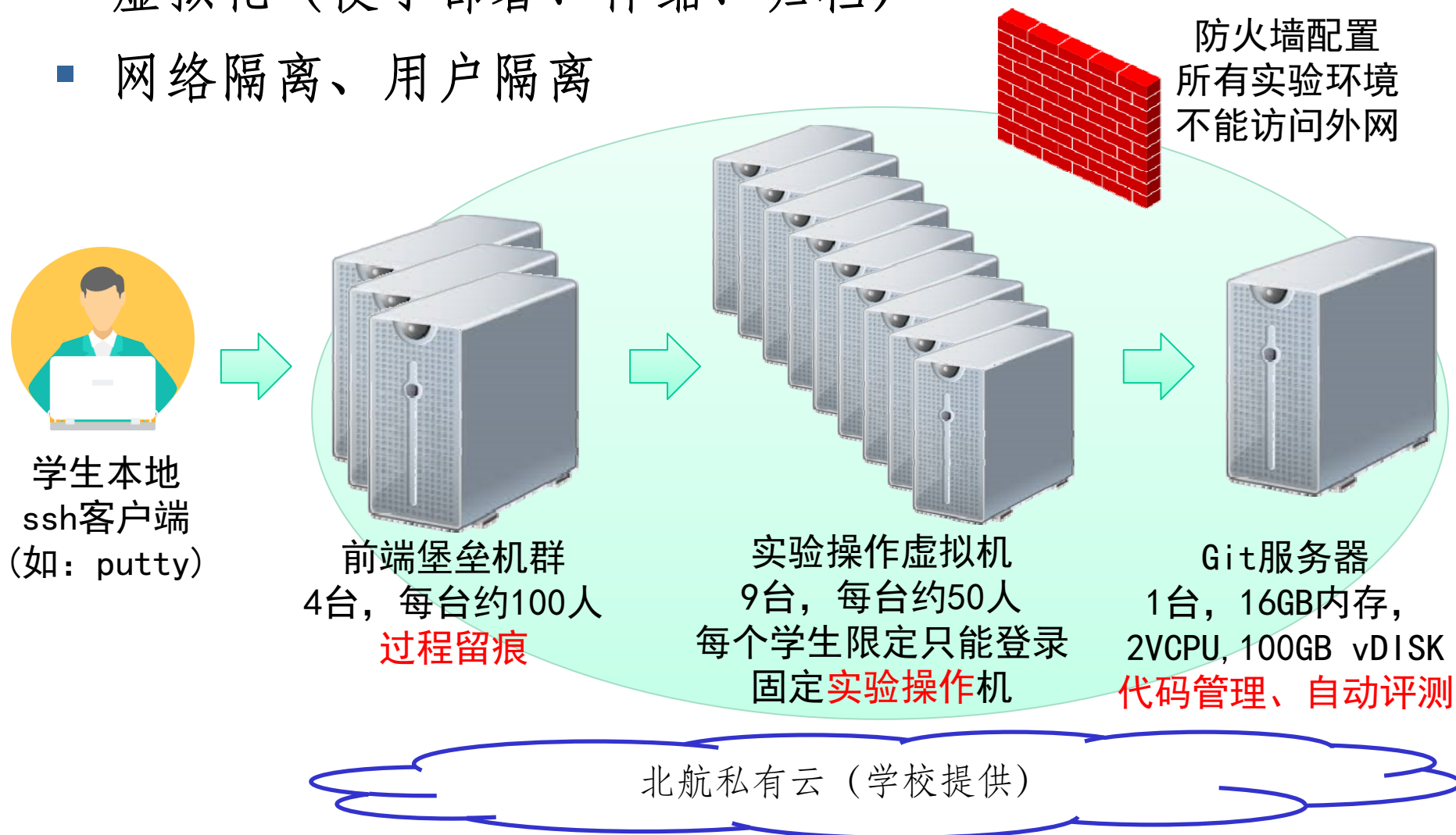
- 学生要能独立完成一个微型OS的编写（填写代码块），包括C语言、汇编语言、Makefile、link script等
- 使用Linux命令行编辑工具，如vim、nano、emacs进行文本编辑
- 使用make、gcc等（交叉）编译、调试工具链生成MIPS目标代码（内核文件）
- 使用模拟器gxemul装载并执行所生成的内核文件，观察实验现象，调试实验代码
- 使用git管理代码（分支），提交课程代码库进行自动评测

子系统3：代码管理与自动评测系统

- 对学校过程的自动控制
 - Git代码库：管理所有实验相关代码
 - 每个学生一个代码库
 - 每个学生只能访问自己的代码库（与登录账号绑定）
 - » TODO：校园网统一认证
 - 每个实验作为一个代码分支（lab0~lab6，还有各次课上测试分支，如lab1-exam1）
 - 自动化评测
 - 根据所提交分支自动运行git hook脚本，在受控环境下编译、执行学生代码，自动比对输出结果，自动评分
 - 根据评分结果，自动下发下一实验的基础代码（增量学习）
- 过程留痕（堡垒机）
 - 采用前端机收集操作过程，包括命令、键盘敲击等
 - 用于事后分析

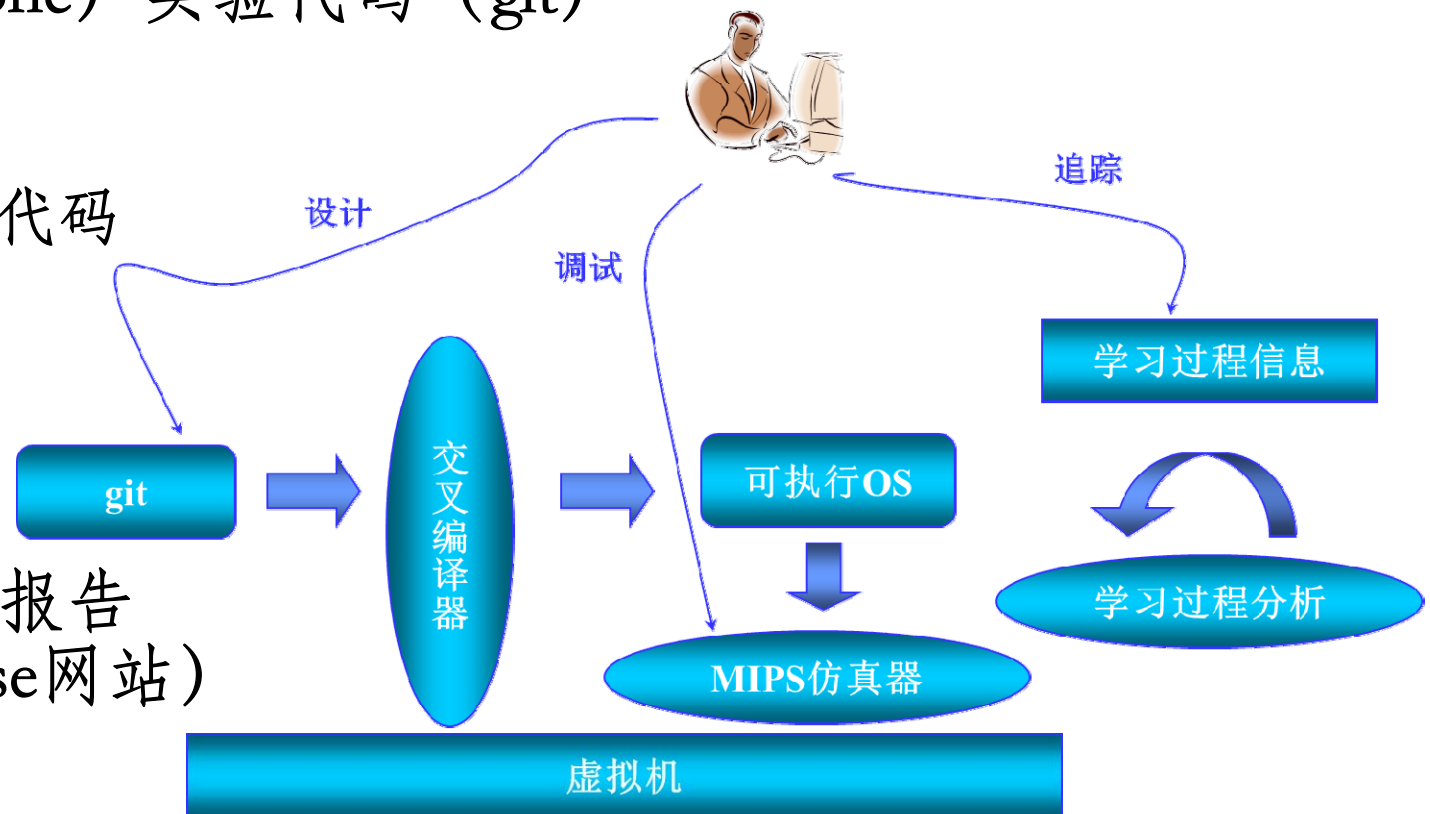
操作系统实验环境部署结构

- 虚拟化（便于部署、伸缩、归档）
- 网络隔离、用户隔离

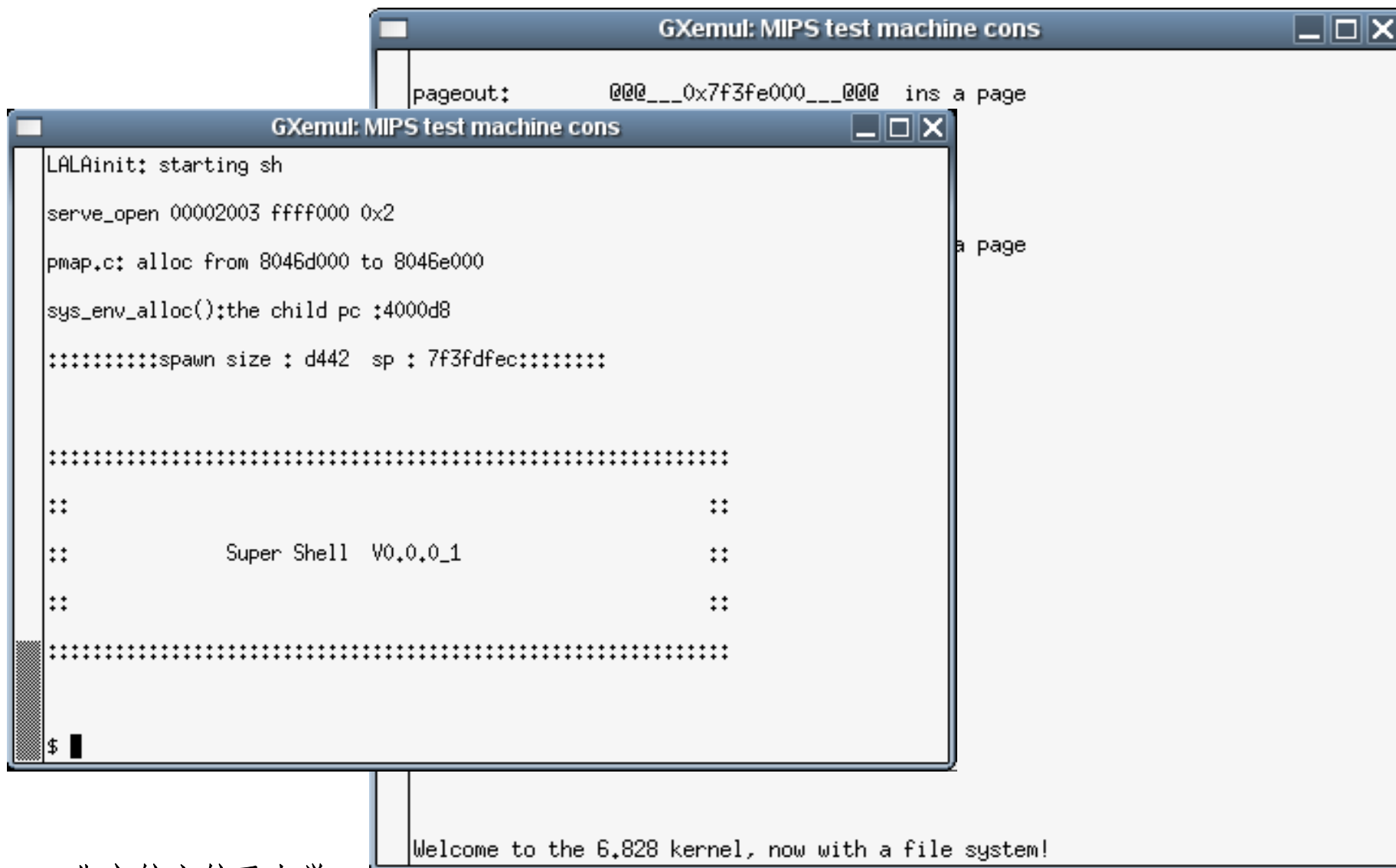


操作系统实验基本流程（学生）

- 1. 上MOOC网站预习实验知识
- 2. 登录实验系统（通过堡垒机）
- 3. 下载（clone）实验代码（git）
- 4. 完成实验
 - 填写空缺代码
 - 编译
 - 仿真运行
- 5. 提交评测
- 6. 撰写实验报告（提交course网站）



GXEMUL仿真器上运行的完整系统



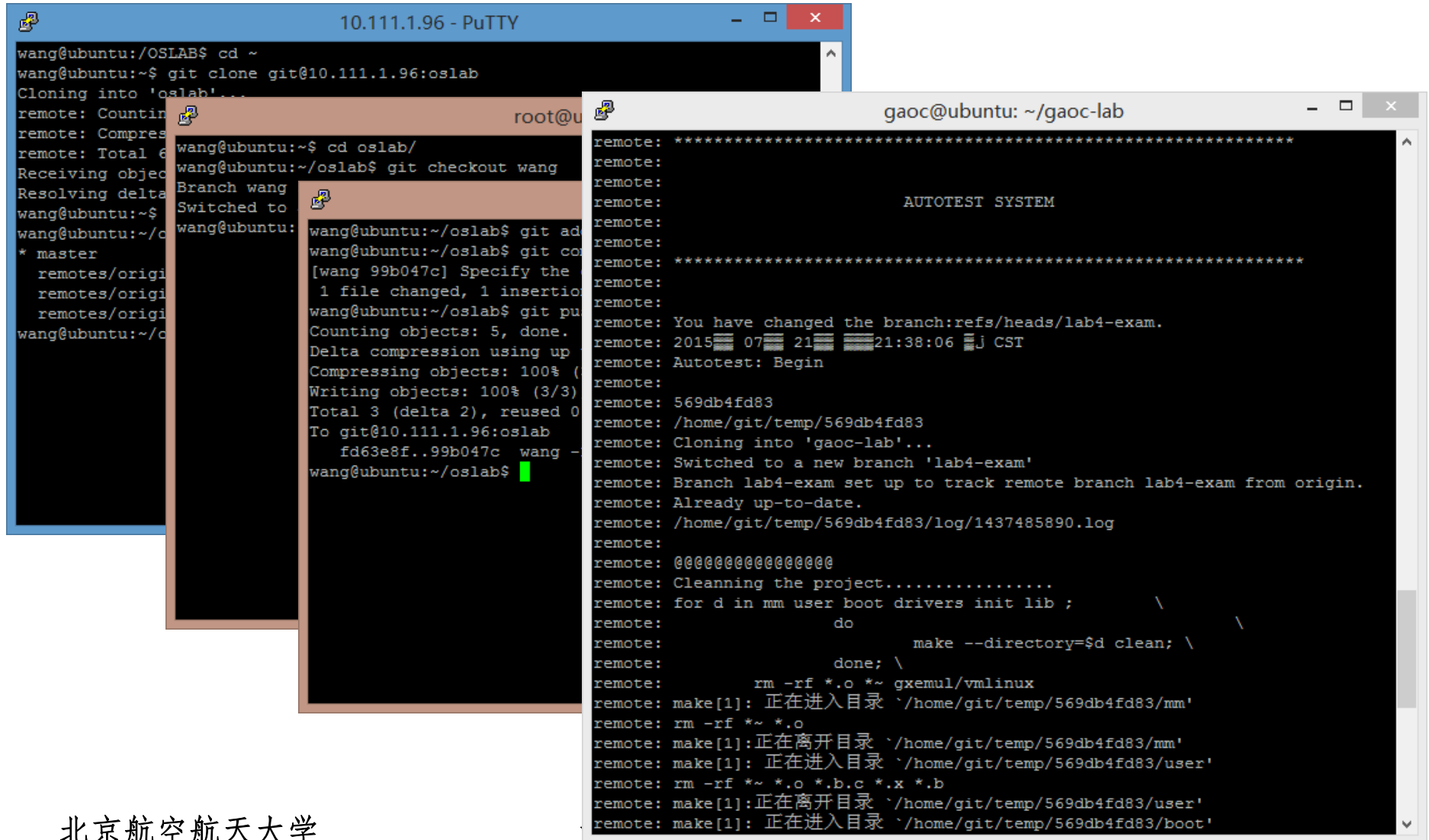
```
GXEmul: MIPS test machine cons
pageout:      @@@__0x7f3fe000__@@@ ins a page

GXEmul: MIPS test machine cons
LALInit: starting sh
serve_open 00002003 ffff000 0x2
pmap.c: alloc from 8046d000 to 8046e000
sys_env_alloc():the child pc :4000d8
:::::::::spawn size : d442  sp : 7f3fdfec:::::::::

:::::::::
::
::          Super Shell  V0.0.0_1
::
::
:::::::::
$ █

GXEmul: MIPS test machine cons
Welcome to the 6.828 kernel, now with a file system!
```

自动评测系统



```
wang@ubuntu:/OSLAB$ cd ~
wang@ubuntu:~$ git clone git@10.111.1.96:oslab
Cloning into 'oslab'...
remote: Counting objects 1/1
remote: Compressing objects 100% (1/1)
remote: Total 6 (delta 0), reused 0 (delta 0)
Receiving objects 100% (6/6)
Resolving deltas 100% (0/0)
wang@ubuntu:~$ cd oslab/
wang@ubuntu:~/oslab$ git checkout wang
Branch wang
Switched to branch 'wang'
wang@ubuntu:~/oslab$ git add .
wang@ubuntu:~/oslab$ git commit -m "wang"
[master] wang
1 file changed, 1 insertion(+)
wang@ubuntu:~/oslab$ git push
Counting objects: 5, done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3)
Writing objects: 100% (3/3)
Total 3 (delta 2), reused 0 (delta 0)
To git@10.111.1.96:oslab
fd63e8f..99b047c wang -> wang
wang@ubuntu:~/oslab$
```

```
gaoc@ubuntu: ~/gaoc-lab
remote: *****
remote:
remote: AUTOTEST SYSTEM
remote: *****
remote: You have changed the branch:refs/heads/lab4-exam.
remote: 2015-07-21 21:38:06 CST
remote: Autotest: Begin
remote: 569db4fd83
remote: /home/git/temp/569db4fd83
remote: Cloning into 'gaoc-lab'...
remote: Switched to a new branch 'lab4-exam'
remote: Branch lab4-exam set up to track remote branch lab4-exam from origin.
remote: Already up-to-date.
remote: /home/git/temp/569db4fd83/log/1437485890.log
remote: @@@@@@@@@@@@@@@@@@
remote: Clanning the project.....
remote: for d in mm user boot drivers init lib ; \
remote: do
remote:     make --directory=$d clean; \
remote: done; \
remote: rm -rf *.o *~ gxemul/vmlinux
remote: make[1]: 正在进入目录 `/home/git/temp/569db4fd83/mm'
remote: rm -rf *~ *.o
remote: make[1]:正在离开目录 `/home/git/temp/569db4fd83/mm'
remote: make[1]: 正在进入目录 `/home/git/temp/569db4fd83/user'
remote: rm -rf *~ *.o *.b.c *.x *.b
remote: make[1]:正在离开目录 `/home/git/temp/569db4fd83/user'
remote: make[1]: 正在进入目录 `/home/git/temp/569db4fd83/boot'
```

学习行为数据

■ 重点统计内容

- 登录时间及时长;
- 登录次数;
- 输入命令;
- 打开文件数;
- 代码提交次数;
- 日志文件大小;
-

发现的问题

- 预备知识问题
- 代码复制问题
- 实验代码阅读问题

预备知识问题——问卷调查

预先掌握程度	Linux	GCC	make	vi	gxemul	lds	shell	git
熟悉	1	0	0	5	0	0	3	3
使用过	16	16	13	17	8	9	14	18
没使用，但知道	9	8	11	3	10	5	6	4
不知道	3	5	5	4	11	15	6	4
有效填写人次	29							

预备知识问题——问卷统计

<i>Tools</i>	<i>familiarity</i>	<i>used</i>	<i>unused but known</i>	<i>unknown</i>
Linux	3.45%	55.17%	31.03%	10.35%
GCC	0%	55.17%	27.59%	17.24%
make	0%	44.83%	37.93%	17.24%
vi	17.24%	58.82%	10.35%	13.79%
Gxemul	0%	27.59%	34.48%	37.93%
lds	0%	31.03%	17.24%	51.73%
shell	10.35%	48.27%	20.69%	20.69%
git	10.35%	62.07%	13.79%	13.79%

发现的问题和解决方案

发现问题：

- 学生操作系统实验课的预备知识普遍不足。

解决方案：

- 增加本次实验的时间（马上进行）
- 与其他课程协商，在其他课程中训练（事后）
- 增加一个星期培训(下一级的实验课调整)

代码复制问题

- 通过分析学生复制代码行为（**control C** 和 **control V** 操作）、日志总量和学生调试中出现的错误，我们发现有一些学生参考了同届或往届的代码。这个问题在**2015**年就存在，**2016**年随着实验次数的增加，变得更加严重。

代码复制问题

- 2016年的改进：
 - a) 人工核对发现代码复制严重学生的实验日志；
 - b) 在评优答辩环节中增加了代码分析的答辩。
- 2017年后的改进：
 - a) 将实验测试分为课上实验和课下实验自动评测两部分组成，涵盖了前5个实验（第6个实验需学生现场演示）
 - b) 课下实验收集了学生的访问次数、文件阅读学习情况等行为数据。
 - c) 课上实验增加了课上在线测试内容，真实地考察了学生对各个不同实验内容的理解程度和完成质量，减少了代码复制的影响。

实验代码阅读问题

- 为了降低操作系统实验的难度，我们将一些比较难的、体系结构相关的代码都提供给了学生。希望同学们能自己去学习和阅读这些代码，以加强对操作系统的了解。
- 但是通过分析学生经常打开次的文件可以看出，学生经常打开的文件中只有少量.s结尾的汇编文件。也就是，大多数同学并没有去阅读体系结构相关代码。

实验代码阅读问题

- 2015年的改进：
 - 在实验内容介绍过程中着重分析一些关键的汇编文件。
- 2016年后的持续改进：
 - 增加一些关于体系结构相关代码的问题，去引导学生阅读相关代码，并提交实验报告。实验报告中包含代码分析、思考题、手册中存在问题等内容。
 - 增加课堂教学中与体系结构相关部分的原理内容并详细讲解。

学习行为统计及与成绩的相关性

- 2015~2017学生实验课成绩
- 成绩相关统计分析结果
 - 重视程度
 - 成绩与日志文件数量的相关性

三年的实验课成绩

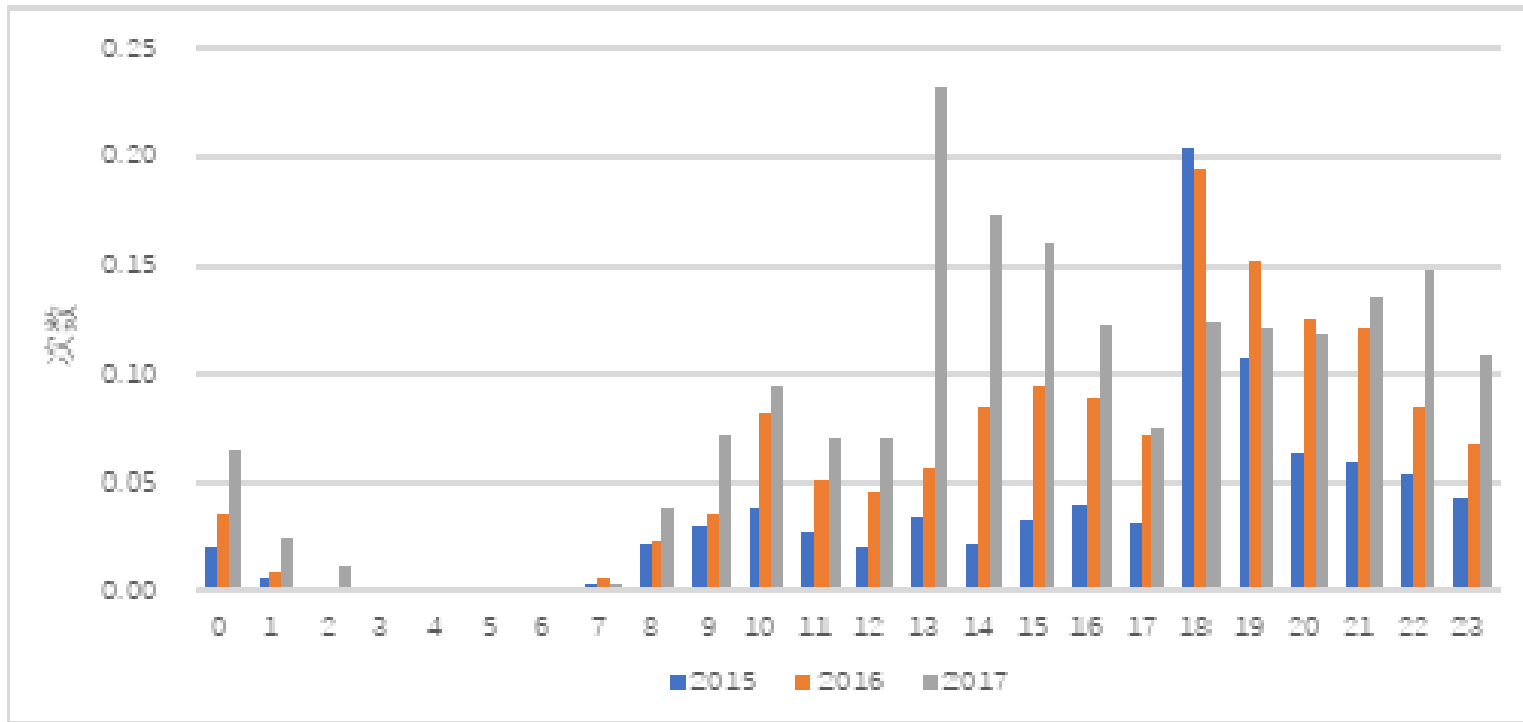
在实验难度逐步增大，考核点增加的情况下

	2015	2016	2017
90-100	20	25	54
60-89	125	116	191
0-59	11	6	37
总计	156	147	282

登录时长

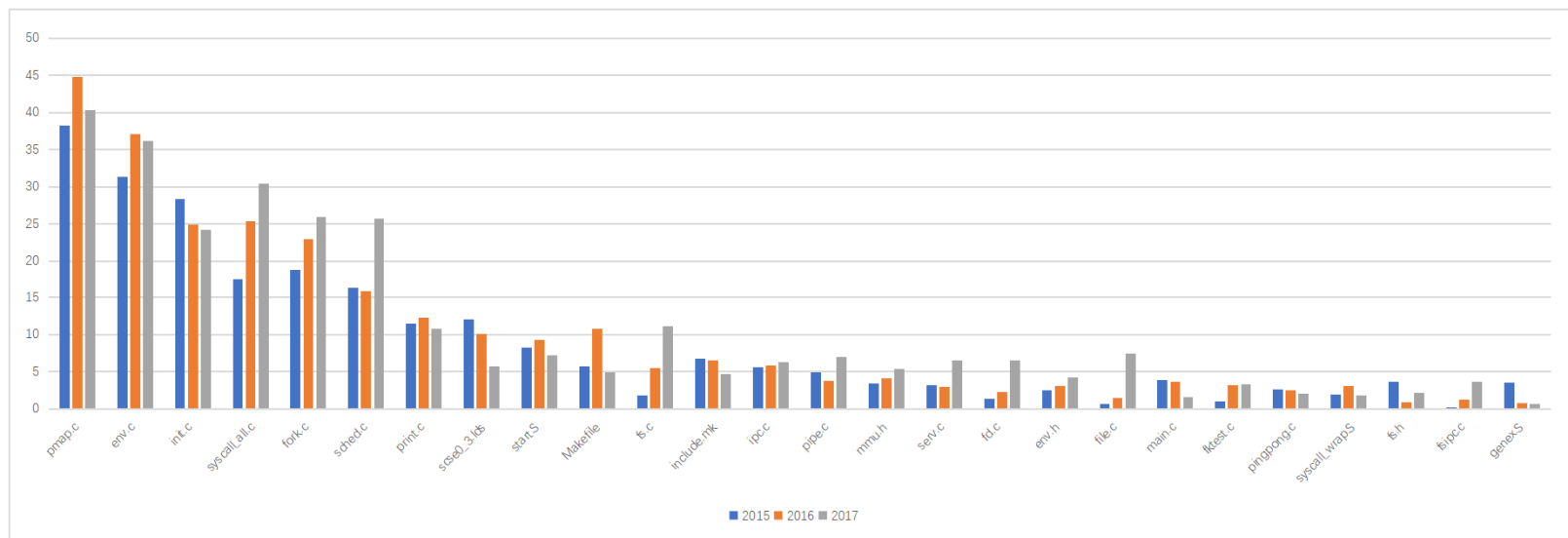
		登录次数	登录时长(小时)
2015	Lab1	26	14.7
	Lab2	11	7.8
	Lab3	14	9.2
	Lab4	12	8.5
2016	Lab1	29	12.7
	Lab2	25	10.4
	Lab3	32	10.6
	Lab4	67	20.3
2017	Lab1	30	27.7
	Lab2	20	20.7
	Lab3	17	18.7
	Lab4	21	22.6

登录时间分布



从图中可以看出，15和16年大部分学生会选择18点登录系统开始实验，而2017年在下午13点就达到了高峰。

打开文件的次数

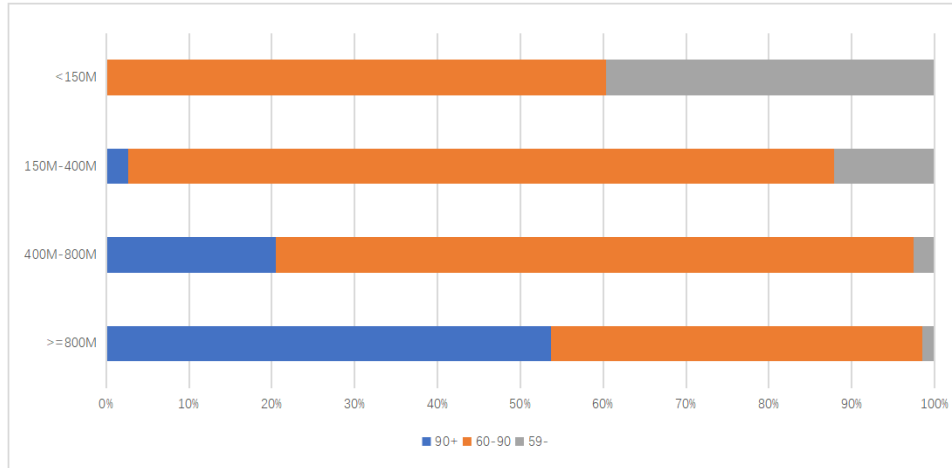


1. 横坐标是文件名，纵坐标是打开次数。
2. 从文件名可以看出汇编文件只有三个，说明学生对体系结构相关代码阅读的兴趣依然没有提高。还需要我们加强引导；
3. 2017年学生阅读的文件中我们发现syscall_all.c、fork.c、shed.c、file.c和fs.c文件的阅读人数大幅增加。而这些文件都是lab4、lab5和lab6中代码，说明在17年完成了后面实验的人数大幅增加了。

对实验的重视程度

- 从统计看出：
 - 学生实验上机时间逐年增加；
 - 学生对实验的重视程度逐年增加。
- 原因：
 - 经过预备知识的培训后学生增加了对操作系统实验的信心。
 - 2017年有更多学生选择完成后面更有难度实验，因此花的时间更多，理论课与实践课结合的效果有所体现。

成绩与日志文件数量的相关性



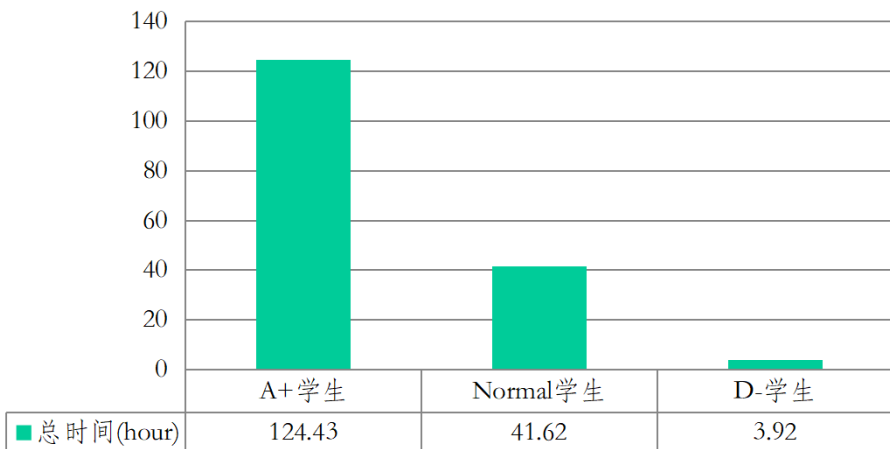
	2015	2016	2017
90+学生	880 MB	532 MB	1259 MB
通过的学生	316 MB	238 MB	505 MB
不通过学生	25 MB	32 MB	138 MB

2017年的统计:

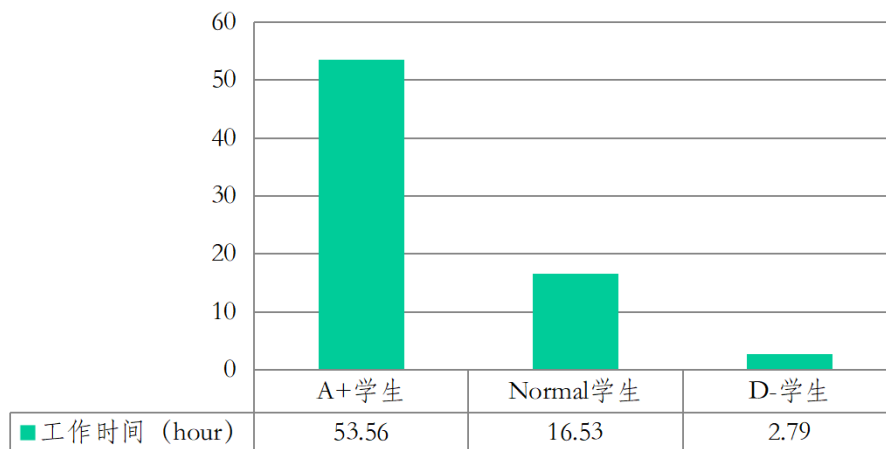
- 大于等于800MB的学生中，约53.7%的学生拿到了90分以上，而不及格的学生只有约1.5%。
- 400~800MB的学生中，约20.5%的学生得分在90以上，76.9%的学生在60到90之间，不及格的学生占约2.6%。
- 150MB~400MB的学生中，2.7%的学生是在90分以上，85.1%的学生是60~90分，不及格的学生达到了12.2%。
- 150MB以下的学生没有大于90分的，60.3%是60~90分的学生，而其余39.7%没能通过课程。

细粒度的行为分析

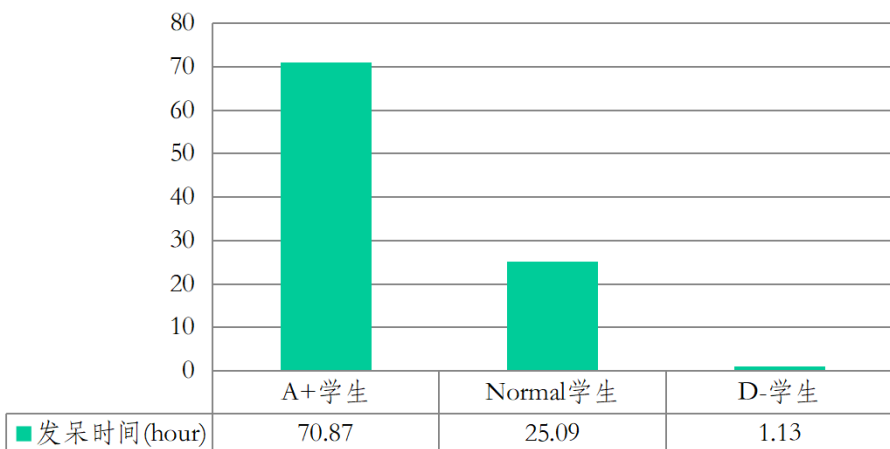
总时间(hour)



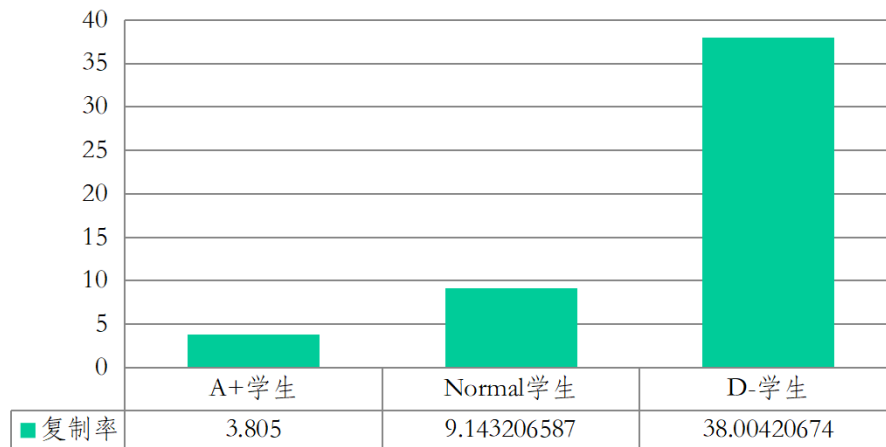
工作时间 (hour)



发呆 (思考?) 时间(hour)



复制率



总结和下一步工作

■ 总结:

- 通过工程化的方法引导学生实现了小操作系统
- 实现了对教学过程中学习行为的分析
- 根据发现的问题，对教学过程进行持续改进
- 提高了学生实验完成率

■ 下一步工作

- 完善实验平台，进一步增加数据挖掘和分析功能
- 有针对性的向学生反馈，更细粒度的指导

谢谢！