

# The Missing Course of Your Computer Science Education

王慧妍

[why@nju.edu.cn](mailto:why@nju.edu.cn)

南京大学



软件学院



计算机软件研究所



# 本讲概述

---

- 关于习题选修课
  - 为什么开?
  - 为什么选?
- 关于传说中的PA实验
- 学术诚信
- 选讲PA0

# 关于《计算机系统基础实验》选修课

课程网站: [http://www.why.ink:8080/ICS/2025/Main\\_Page](http://www.why.ink:8080/ICS/2025/Main_Page)

实验网站: <https://nju-projectn.github.io/ics-pa-gitbook/ics2025/>

# 课程计划内容

围绕一个PA大实验和数个小实验为载体，通过实验指导及相关知识点课堂深入案例演示，让学生深入了解计算机系统的执行。理解“程序如何在计算机上运行”的根本途径是从“零”开始实现一个完整的计算机系统

## • 课堂授课

- 在课堂通过实际案例及NEMU代码演示计算机系统相关知识点，清晰了解现代系统内部实际的实现及运行细节，重点围绕
  - 前序课程已介绍过的相关知识的实际系统案例代码演示
  - NEMU相关设计细节介绍及实验完成所需的编程指导

- |   |                         |
|---|-------------------------|
| 1. The Missing Course of Your CS Education [slides] | 9. I/O设备选讲 [slides]     |
| 2. C 语言拾遗1: 机制 [slides]                             | 10. 链接与加载选讲 [slides]    |
| 3. C 语言拾遗2: 编程实践 [slides]                           | 11. 系统编程与基础设施 [slides]  |
| 4. Git, Github和经验分享 [slides][分享1][分享2]              | 12. 程序优化选讲[slides]      |
| 5. NEMU 框架选讲 [slides]                               | 13. 中断与分时多任务[slides]    |
| 6. 数据的机器级表示 [slides]                                | 14. 虚拟存储选讲[slides]      |
| 7. x86-64与内联汇编 [slides]                             | 15. “造轮子”的方法和乐趣[slides] |
| 8. 调试: 理论与实践[slides]                                | 16. 复习&答疑[slides]       |

# 课程计划内容

围绕一个PA大实验和数个小实验为载体，通过实验指导及相关知识点课堂深入案例演示，让学生深入了解计算机系统的执行，理解“程序如何在计算机上运行”的根本途径是从“零”开始实现一个完整的计算机系统

## • 课后实验

- 一个PA大作业（NEMU简易全系统模拟器）
  - 通过模拟实现核心“取指-译码-执行”逻辑，基于主流riscv32指令集的核心指令集译码实现，搭建自己的计算机系统模拟器，能够运行“仙剑奇侠传”这样的游戏
- 两次小作业
  - PA1+：自动化测试
  - PA2+：效率优化

# 与前后课程联系

- 新增选修于大二上学期
  - 前：C程序设计基础（大一上），计算系统基础（大一上/下），及软工I（大一下）
    - 已掌握C语言基本编程能力
    - 已了解基本计算机系统概念及基本数字逻辑电路相关知识
  - 后：操作系统（大二下）
    - 衔接后续真正操作系统课程，更容易接受并掌握现代操作系统的新特性及复杂机制
- 紧密联系：和计算系统基础
  - 从DLX指令到riscv32指令集，从指令模拟到全系统模拟
  - 编程量加大（很多），大项目把控能力增强，对系统运行理解加深

# 新增选修原因考虑

## • 过去经验

- 作为实验课主讲老师，在计算机系及人工智能学院授课三年（本人授课每学年上课人数超过200），PA大作业收到比较好的反响，坚持完成后学生表示对于代码能力提升非常明显
  - 40%学生能跟下所有实验
  - 15%学生早早放弃

- 缺点是**两级分化异常严重!**



## • 改为选修形式

- 减轻必修课的外在压力，希望引导学生内驱力完成此课程
- 选修课程可以减轻部分实验内容，留作学生后续完成课程后感兴趣自行二刷（框架代码同时支持x86，mips32和riscv32/64）
- **自行评估自身上课压力，合理劝退**

# 关于传说中地狱难度PA的一些真相

- 适合谁？
  - C/C++编程的训练基础较好
  - 愿意花时间在一门看起来收益不高的选修课
  - 希望更了解系统方向，更好修电脑？
- 它很难，的确很难
  - 确切地说，对最优异的同学来说依然有一些挑战性
    - 如果你感到异常困难，你更需要的其实是C/C++编程的训练
  - （并且我们知道这一点！）

请简短评价一下南京大学ICS-PA项目



南京大学ICS-PA项目是国内计算机系统课程中**最具挑战性也最有价值的实践项目之一**。

- **优点：硬核**，通过“自底向上”亲手实现计算机系统核心部件（CPU、内存管理、操作系统内核），能让人们对计算机系统的理解达到**前所未有的深度**，极大锻炼编码和Debug能力。
- **挑战：极其耗时**，工作量巨大，难度曲线陡峭，需要出色的抗压能力和时间管理。

一句话总结：**过程痛苦，但收获巨大**，是南大计算机教育的王牌课程。

# 如果你已经 选课?

- 自行评估完成压力
- 此处为劝退贴

实验细节	难度
PA0-环境预备	★☆☆☆☆
PA1-简易调试器	★★★★☆
PA2-冯诺依曼计算机系统	★★★★☆
PA3-批处理系统	★★★★★
PA4-分时多任务（鼓励）	★★★★★

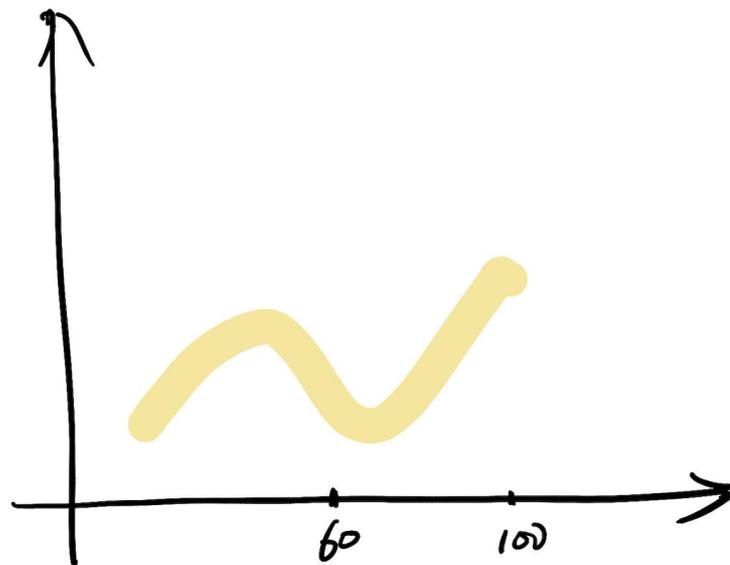
# 如果你已经 选课?

- 再想想

- 还想选? 决心+毅力!

And 恭喜你!

过往真实的分布



# 老师/助教的使用

- 联系/求助：使用邮件

- 能使你更好地整理问题。也许在整理问题的过程中你就发现答案了

- 老师：

- 王慧妍：[why@nju.edu.cn](mailto:why@nju.edu.cn);

- 助教：

- 李晗：<hanli@smail.nju.edu.cn>

- 陈哲敏：<231250123@smail.nju.edu.cn>

- 何棋：<231250083@smail.nju.edu.cn>

- 杨承岳：<231250186@smail.nju.edu.cn>

- Ask

- 对上课内容没有理解的地方、对课程/学习的疑惑、.....

- Do not ask

- 安装XXX错了怎么办？ Segmentation Fault怎么办？



# 相关课程资料

- 课程网站: [http://www.why.ink:8080/ICS/2025/Main\\_Page](http://www.why.ink:8080/ICS/2025/Main_Page)

## 课程资料

1. The Missing Course of Your CS Education [slides]
2. C 语言拾遗1: 机制 [slides]
3. C 语言拾遗2: 编程实践 [slides]
4. Git, Github和经验分享 [slides][分享1][分享2]
5. NEMU 框架选讲 [slides]
6. 数据的机器
7. x86-64与内
8. 调试: 理论
9. I/O设备选讲
10. 链接与加载
11. 系统编程与
12. 程序优化选
13. 中断与分时
14. 虚拟存储选
15. “造轮子”的
16. 复习&答疑

## 课程ppt

### 2. 实验提交

在实验的目录中执行 `make submit` 将会自动完成提交。

#### 设置正确的 TOKEN

我们为每个同学分配了唯一的 TOKEN (8 个字符), 已发送至你的 `smail.nju.edu.cn` 邮箱, 请查收。你需要确保这个字符串在提交时处于 `TOKEN` 环境变量中。如何设置环境变量请 `STFW`。前期请自行开始实验所需环境安装 (详情请参考 PA0 实验)。

在 9 月 7 日 10 点前选课的同学, 请查看 `mail` 邮箱妥善保存 `TOKEN`, 在 9 月 7 日 11 点后选课的同学, 请在此表格中登记你的 `TOKEN` 获取需求, 我们后续会统一发送。

### 2.1 PA 提交方法

脚本会自动根据你的 `git` 分支提交到对应的实验上, 例如 `pa0` 分支会自动提交到 PA0。提交成功后会返回提交的文件 ID:

```
[SUCC ✓] Received ... ics2022.tar.bz2 at 13:51:36
```

提交到系统的代码会进一步经过 Online Judge 的检查, 包括:

- Git 记录检查
- 代码相似度检查
- 编译、运行测试用例。特别注意: Online Judge 编译运行的是 Git repo HEAD 中的代码。因此请确保你的代码最后的修改被编译且被追踪。未提交到 Git repo 中的代码将不会被评测。

### 2.2 Lab 提交方法

在 Lab 对应的目录中 `make submit`。如 Lab1 的工作目录为 `multimod/`, 则在 `multimod` 目录中执行 `make submit`。需要设置 `STUID` (学号) 和 `STUNAME` (中文姓名) 环境变量。

### 2.3 PA与Lab的OJ查询

在相应的oj页面查看 Online Judge 的返回结果, 页面地址为: <http://why.ink:8080/oj/ICS2023/拼接实验项目与邮件给每人自己发的TOKEN>。例如: 对 PA1 实验, `TOKEN` 为 `AABB` 的同学可于 <http://why.ink:8080/oj/ICS2023/PA1/AABB> 页面查看该实验的实时提交 OJ 结果。对 Lab1 实验来说, 该页面地址为: <http://why.ink:8080/oj/ICS2023/Lab1/AABB>。

## OJ提交指南

Type to search

### Introduction

PA0 - 世界诞生的前夜: 开发环境配置

Installing GNU/Linux

First Exploration with GNU/Linux

Installing Tools

Configuring vim

More Exploration

Getting Source Code for PAs

### 实验前阅读

#### 最新消息(请每天至少关注一次)

- 学术诚信(什么事情能做, 什么不能)
- 常见问题(对PA的各种困惑)
- 如何正确求助: 提问的智慧和别像弱智一样提问
- 外部资源
  - PA习题课 (2023 秋季学期, 王慧妍老师): 课程资料
  - 中国大学MOOC的ICS理论课 (袁春风老师): 上, 中, 下
  - 南大SICP编程课

如果你觉得自己的编程基础不过关, 墙裂推荐自学这门课

语言的语法是次要的, 对编程思维的锻炼才是最重要的  
岩老师)

老师: 课程资料

老师: 课程资料

老师: 课程资料, B站录播

从2023年秋季学期开始前, 将被视为往届讲义材料。如果你是修读  
—请勿使用本讲义代替2023年秋季HGS课程的PA实验讲义—提交本实

但其网络可能不稳定, 你可以把这个仓库克隆到本地, 然后通过浏览

本地阅读到最新版本的内容。你需要在仓库路径中手动执行 `bash`  
脚本本地, 再次强调, 如果你选择了离线阅读方式, 将由你来负责获取最

为什么要学习计算机系统基础,

并打算向我们寻求帮助, 请先阅读提问的智慧和别像弱智一样提问

错误或者对实验内容有疑问或建议, 请通过邮件的方式联系余子濠

的cser么? 回复节选

不由自主地教导用最小的付出获得最大的得到, 经常会忘记我们究竟  
但我想每个人心中都有那一份求知渴望和对真理的向往, “大学”的  
的收获和喜悦, 我们才愿意去付出, 去追求, 去探索, 去创造。

## 实验指导手册

# 习题课的内容

---

- 《计算系统基础》部分知识的遗失实践细节
- 对你们从Lab/PA中生存下来是至关重要的
- 对你们未来XX年作为“码农”的生涯都是至关重要的
- 本讲标题来自于[The Missing Semester of Your CS Education](#)
  - jyy墙裂推荐!
  - 我们miss的比这门课多
- 习题课（和课程网站）会布置所有作业

# 你们可能比较关心的

- 分数占比 (期末只剩下10%)
  - PA: 70%、PA+: 10%、期末: 20%

PA0~4

## • 评分标准

### • DDL

- 100%分数

### • Hard DDL

- 80%分数

### 3.2.1 PA: 几乎完全客观评分

- Rejected, 编译错误或没有通过任何测试用例: 10% (诚信分)
- Partial Accepted, 部分 easy 测试通过 (此时不运行 hard 用例): 10%~60%, 根据比例加权换算
- Partial Accepted, 全部 easy 测试通过, 部分 hard 测试通过: 60%~80%, 根据比例加权换算
- Accepted, 通过全部 easy/hard 测试:  $\geq 80\%$ , 剩余部分由人工评价给出
- 没有通过全部 easy 测试用例的作业将没有人工评分的机会 (即意味着实验报告不得分。但我们会阅读你的反馈)。

## 以课程网站为准



# 关于传说中的PA实验

PA是什么东西?

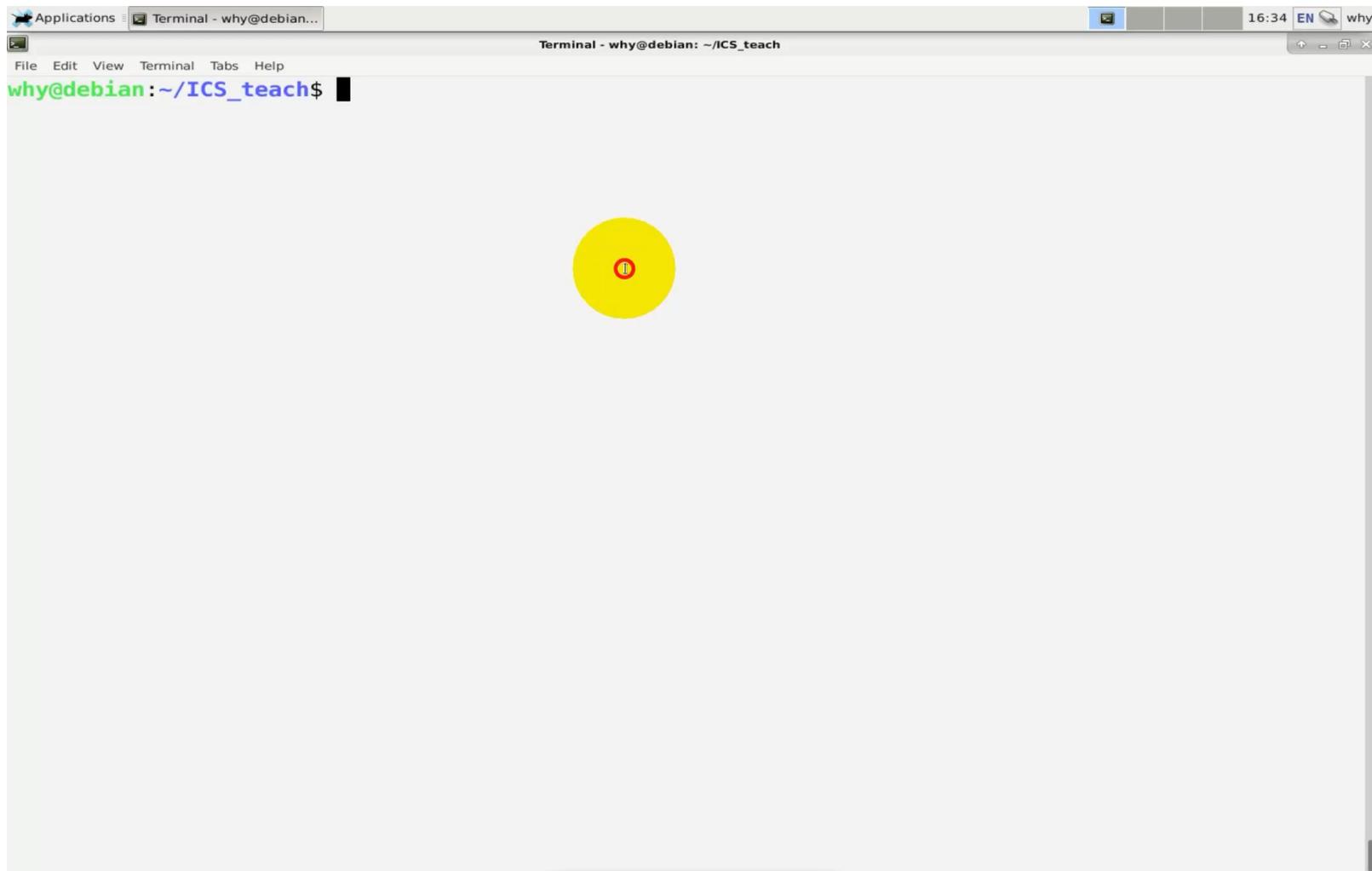
已选课学生将默认TOKEN发送至学校邮箱：学号@smail.nju.edu.cn  
(无需登记)

其他登记TOKEN需求 (表格共享密码: software)

<https://table.nju.edu.cn/dtable/links/bbda8ba45ece40439310>

# 回答终极拷问

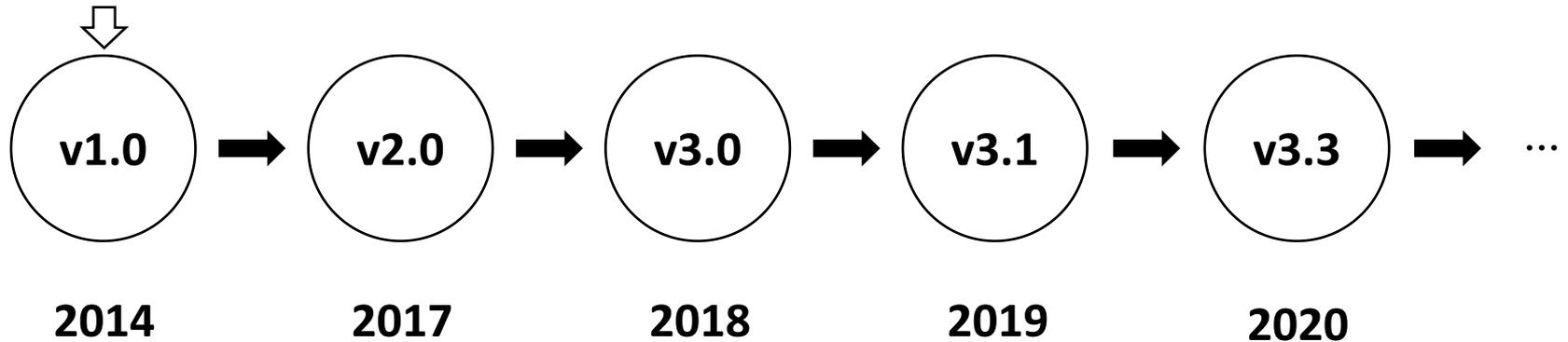
当你运行一个Hello World程序的时候，计算机究竟做了什么？



# PA实验

- 功能完备但简化的模拟器NEMU(NJU EMUlator)的实
  - 实验环境配置 (PA0)
  - 四个连贯的实验内容 (PA1~PA4) , 即: 简易调试器、程序执行、cache与存储管理、异常与I/O

余子濠 开发完成



PA的简要历史

# NEMU, 一个简化的全系统模拟器

实验细节	难度	内容	目标
PA0-环境预备	★☆☆☆☆	准备linux开发环境及NEMU框架运行所需工具链	安装并正确运行NEMU框架
PA1-简易调试器	★★★★☆	实现表达式求值及监视器功能, 为NEMU上debug做准备	能够成功计算表达式并实现断点管理
PA2-冯诺依曼计算机系统	★★★★☆	实现riscv32核心指令集翻译, 实现基本库函数并封装klib	运行纯计算程序, 类hello world程序, 及字符版本超级玛丽
PA3-批处理系统	★★★★★	实现批处理系统, 支持自动加载程序, 支持异常处理及上下文切换恢复	运行Flappy Bird和PAL仙剑奇侠传
PA4-分时多任务 (鼓励)	★★★★★	选作分时多任务功能	鼓励完成, 占分少

## • 说明

- 提供**NEMU框架代码**, 框架代码核心过**5k行**, 能够有效锻炼学生对于**较大项目的整体把握能力**
- 提供**相关工具链及diffTest支撑**, 能够支持引导学生采用常见测试方法**自测并定位修复程序bug**
- 提供相关**文档资料**的链接, 保证学生有需要可以**自行查阅**
- **选修课程, 难度调整**, 控制PA4占比, 能够**降低选修课程的压力**

# NEMU

- NEMU是一个用来执行其它程序的程序（怎么理解？）
  - 支付宝：用软件模拟硬件ATM
    - 取款、存款、转账、汇款等等



- NEMU: 用软件模拟出 “计算机”



✓模拟PC机



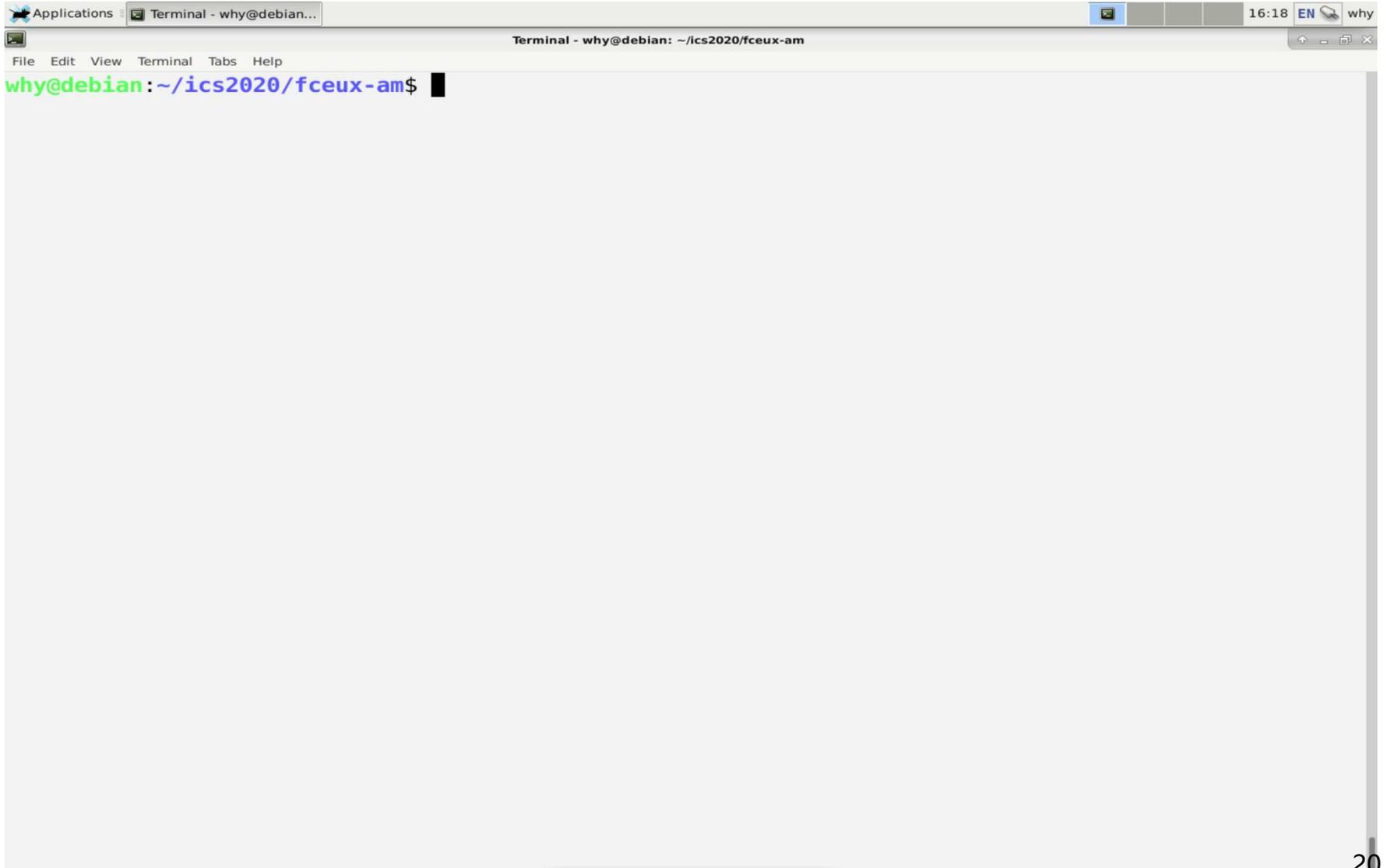
ANDROID

✓模拟手机

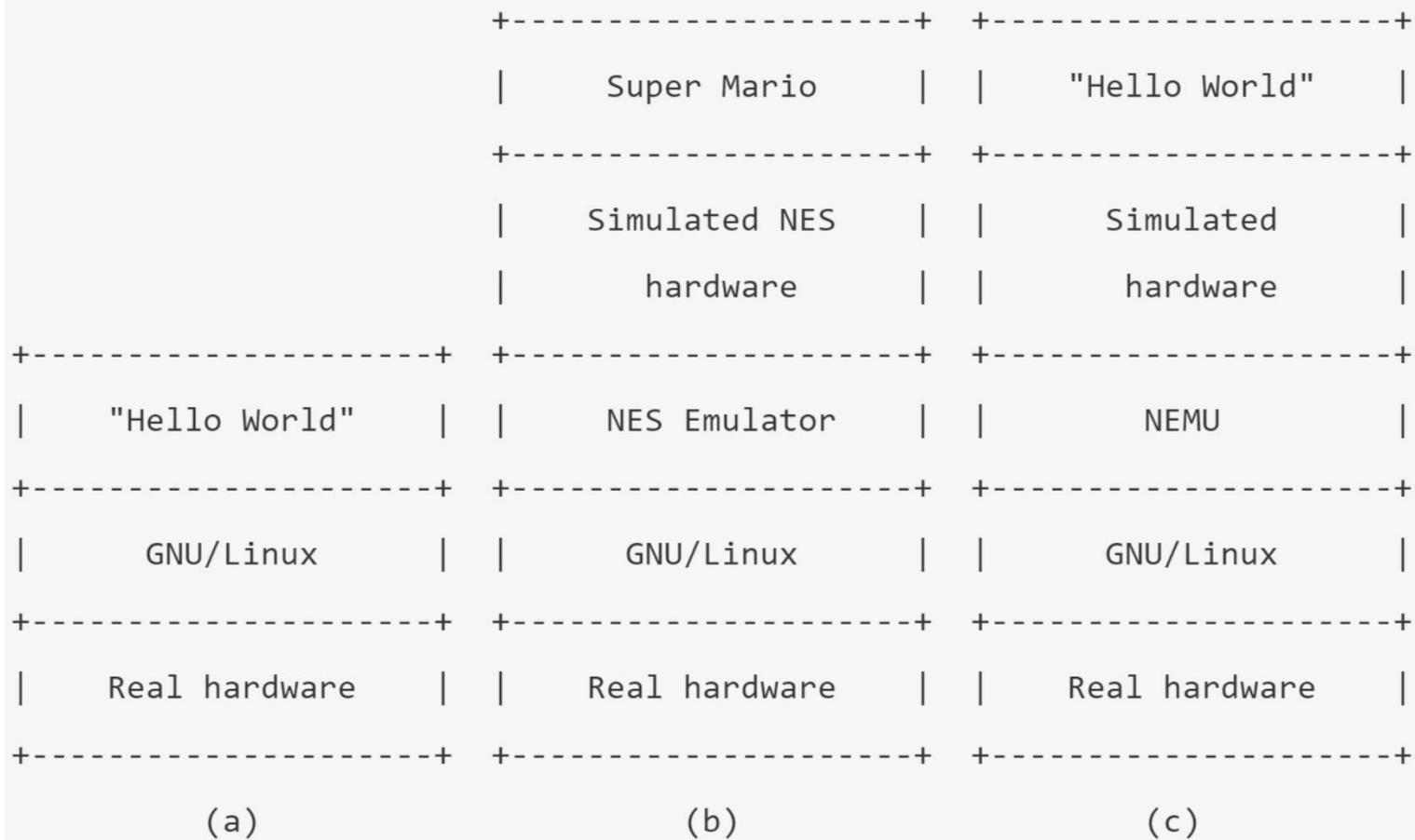


✓模拟游戏机

# 用红白机模拟器NES Emulator玩Mario



# NEMU模拟器



# 学术诚信 (Academic Integrity)

# 学生手册：不能抄作业

- What is academic integrity?

- [What is Academic Integrity? | Academic Integrity at MIT](#)

Plagiarism	Cheating		
<p><b>Do:</b></p> <p>Trust the instructor.</p> <p>Undertake your own work.</p>	<p><b>Do:</b></p> <p>Demonstrate your own achievement.</p> <p>Accept corrections from the instructor as part of the learning process.</p> <p>Do original work for each class.</p>	<p><b>Don't:</b></p> <p>Don't copy answers from another student; don't ask another student to do your work for you. Don't fabricate results. Don't use electronic or other devices during exams.</p> <p>Don't alter graded exams and submit them for re-grading.</p> <p>Don't submit projects or papers that have been done for a previous class.</p>	<p>Don't copy a paper</p> <p>Don't copy without citing</p>
Unauthorized Assistance	Facilitating Academic Dishonesty		
<p><b>Do:</b></p> <p>Trust the instructor.</p>	<p><b>Do:</b></p> <p>Showcase your own abilities.</p>	<p><b>Don't:</b></p> <p>Don't allow another student to copy your answers on assignments or exams. Don't take an exam or complete an assignment for another student.</p>	<p>Don't let others copy and the</p>

# 学生手册：不能抄作业

---

- What is academic integrity?
  - 简单概况：独立完成
- 针对作业的独立完成
  - 自己完成作业
  - 对使用的已有资料作出明确的标示
  - [ACM Policy on Plagiarism, Misrepresentation, and Falsification](#)
- 针对代码的独立完成
  - 自己完成代码的编写
  - 自己完成测试用例
  - 在允许的范围内使用他人的成果

# 具体案例：MIT 6.005 Elements of Software Construction

---

- Individual work
  - Problem sets in this class are intended to be primarily individual efforts. You are encouraged to discuss approaches with other students but *your code and your write-up must be your own.*
  - You *may not use materials produced as course work by other students*, whether in this term or previous terms, nor may you provide work for other students to use.
  - It's good to help other students. But as a general rule, during the time that you are helping another student, *your own solution should not be visible*, either to you or to them. Make a habit of closing your laptop while you're helping.

# 具体案例：MIT 6.005 Elements of Software Construction (cont'd)

---

- Using external resources
  - It's fine to use material from external sources like [StackOverflow](#), but only with proper attribution, and only if the assignment allows it. In particular, if the assignment says “implement X,” then you must create your own X, not reuse one from an external source.
  - It's also fine to use any code provided by this semester's 6.031 staff (in class, readings, or problem sets), without need for attribution. Staff-provided code may not be publicly shared without permission, however, as discussed later in this document.

# 具体的案例 (1)

---

- Alyssa and Ben sit next to each other with their laptops while working on a problem set. They talk in general terms about different approaches to doing the problem set. They draw diagrams on the whiteboard. When Alyssa **discovers a useful class in the Java library**, she mentions it to Ben. When Ben finds a StackOverflow answer that helps, he sends the URL to Alyssa. **OK**.
- As they type lines of code, they speak the code aloud to the other person, to make sure both people have the right code. **INAPPROPRIATE**.
- In a tricky part of the problem set, Alyssa and Ben look at each other's screens and compare them so that they can get their code right. **INAPPROPRIATE**.

# 具体的案例 (2)

---

- Jerry already finished the problem set, but his friend Ben is now struggling with a nasty bug. Jerry sits next to Ben, looks at his code, and helps him **debug**. **OK**.
- Jerry opens his own laptop, finds his solution to the problem set, and refers to it while he's helping Ben correct his code. **INAPPROPRIATE**.

# 具体的案例 (3)

---

- Louis had three problem sets and two quizzes this week, was away from campus for several days for a track meet, and then got sick. Ben feels sorry for Louis and wants to help, so he sits down with Louis and talks with him about how to do the problem set while Louis is working on it. Ben already handed in his own solution, but he doesn't open his own laptop to look at it while he's helping Louis. **OK.**
- Ben opens his laptop and reads his own code while he's helping Louis. **INAPPROPRIATE.**
- Ben has by now spent a couple hours with Louis, and Louis still needs help, but Ben really needs to get back to his own work. He puts his code in a Dropbox and shares it with Louis, after Louis promises only to look at it when he really has to. **INAPPROPRIATE.**

# 具体的案例 (4)

---

- John and Ellen both worked on their problem sets separately. They exchange their test cases with each other to check their work. **INAPPROPRIATE**. **Test cases** are part of the material for the problem set, and part of the learning experience of the course. You *are copying if you use somebody else's test cases, even if temporarily*.

# 具体的案例：PA0

PA0: 安装 Linux 系统，并提交空的文件。允许向互联网/同学求助。

- 遇到问题（如安装错误）找同学询问/解决 **OK**
  - 但你可能就失去了这门课原本的训练
    - 尽可能先自己解决
  - 帮其他同学解决问题的人
    - 一起还原解决问题的过程
- 请别人安装系统，或使用他人的虚拟机镜像 **INAPPROPRIATE**
  - Arm架构?

# Academic integrity

- 感到三观尽毁?
  - 原来拿个测试用例也违反academic integrity?
  - 拿个大腿的作业来改改不香吗? 我还读懂了呢!
  - 老师压根就没精力管, 对他来说吃力不讨好
    - 有些事情是“天然”被禁止的
    - 但是我们的教育里缺失了“这是不对的”
- 那些痛苦是对你的训练 (training)
  - “看懂”和“自己设计测试用例、自己做出来”天差地别
    - PA难度无形剧增←过去没有academic integrity欠的债
  - 不要看不起美国人
    - (大部分) 学生真的明白并自发地执行这个标准

# 知乎贴：作业抄袭中的人生百态



我们不生产~~水~~,我们只是~~大自然~~的搬运工

代码

互联网



代码抄袭：那些让985学

```
1 cur->lineno = temp->lineno;
2 strcpy(cur->type, type);
3 cur->isLexical = 0;
4 cur->children = temp;
```

```
1 head->number_signal = 0;
2 head->line = temp->line;
3 strcpy(head->type, type);
4 head->child_left = temp;
```

1  
2  
3  
4

```
1 $$->is_root=1;
2 $$->no_leaves=1;
3 $$->leaves[0]=(Node*)$1;
4 if(exit_error==0)
5   {print_tree($$,0);}
```

```
1 $$->final=0;
2 $$->num_children=1;
3 $$->children=(Node**)malloc
4   (sizeof(Node*)*$$->
5     num_children);
6 $$->children[0]=(Node*)$1;
7 if(!wrong)
   printNode($$,0);
```

1  
2  
3  
4  
5  
6  
7

```
1 temp->line = a->line;
2 temp->lChild = a;
3 while(num > 1){
4   a->rChilds = va_arg(list,
5     node*);
6   a = a->rChilds;
7   num--;
```

```
1 p_node->left_child = temp;
2 p_node->line = temp->line;
3 for(int i=0;i < num-1;++i){
4   temp->right_child =
5     va_arg(valist, struct
6     Node*);
7   temp = temp->right_child;
```

1  
2  
3  
4  
5  
6  
7

```
1 void P1() {
2   ...
3   puts("Game_Over");
4   ...
5 }
6 void P2() {
7   ...
8   puts("G"); puts("a")
9   puts("e"); puts("_")
10  puts("v"); puts("e")
11  ...
12 }
```



Aims and Objectives of RPg Education

Strategic Framework for TPg Education

HKUST Fok Ying Tung Graduate School

News

抄袭

学术

论文

ICCV

科研热点

关注者  
3,449

被浏览  
6,642,782

# HKUST attaches great importance to academic integrity

The Hong Kong University of Science and Technology (HKUST) has initiated an investigation in accordance with procedures into an alleged plagiarism incident involving HKUST members.

HKUST attaches great importance to academic integrity. The University has established guidelines on academic integrity and expects all members to abide by the relevant regulations. In case of violation, the

## 如何看待 ICCV21 接收的某港科大学生为一作的论文被指抄袭 ICML21 发表的论文?

ICCV21接收论文m-RevNet: Deep Reversible Neural Networks with Momentum被指出与 ICML21接收论文Momentum residual neural networks在核心思路、实验和图表上有多处雷同，疑似抄袭

ICML21论文作者的声明和详细的抄袭证据:

<https://michaelsdr.github.io/momentumnet/plagiarism/>  
🔗 [michaelsdr.github.io/momentumnet/plagiarism/](https://michaelsdr.github.io/momentumnet/plagiarism/)

被指抄袭的论文的一作个人主页网页快照:

# ChatGPT is fun, but not an author

H. HOLDEN THORP [Authors Info & Affiliations](#)

SCIENCE · 26 Jan 2023 · Vol 379, Issue 6630 · p. 313 · DOI: 10.1126/science.adg7879

45,004



PHOTO: CAMERON DAVIDSON

In less than 2 months, the artificial intelligence (AI) program ChatGPT has become a cultural sensation. It is freely accessible through a web portal created by the tool’s developer, OpenAI. The program automatically creates text based on written prompts so popular that it’s likely to be “at capacity” when you attempt to use it. When you do get ChatGPT provides endless entertainment. I recently rewrote the first scene of the classic American play *Death of a Salesman*, but to feature Prince

the animated movie *Frozen* as the main character instead of Willy Lom; I had an amusing conversation in which Elsa—who has come home from selling—ice powers and you’re a queen. You’re unstoppable.” Mash-ups like this are fun, but there are serious implications for generative AI programs like ChatGPT in science and academia.

← r/PublishOrPerish · 2个月前  
Peer-review-Pro · 前 1% 最受欢迎的发帖人

## Researchers hide “positive review only” prompts in papers. Yes, really.

Peer Review

elucidate the inherent complexities of clustering in MMC that currently relies on recent advances in concentration techniques (Wolfer and Kontorovich) and estimation techniques (Appendix D).

asia.nikkei.com

打开

Installing GNU/Linux

First Exploration with GNU/Linux

Installing Tools

Configuring vim

[More Exploration](#)

Getting Source Code for PAs

# PA0 选讲：进入Linux世界

学会用Linux工作吧

# 让时间回到1980s

- 就算回到1980s, 该做的事情还得干啊
  - 管理文件; 编代码; 写作业; 排版杂志; 上网.....

- 例子

- `vi a.txt`
- `ip addr / ping baidu.com`
- `df -h /`
- `find . -name "*.cpp"`
- `fdisk /dev/sdb`
- `shutdown -h 0`
- `apt install qemu-system`
- `pdfjoin a.pdf b.pdf`
- `iconv -f gbk -t utf-8 file.txt`



# 现在都2025年了，还整这些玩意干嘛？

上面这些事情不是点点鼠标就能搞定的吗？

- 被迫接受

- ~~不学你就挂子~~

- 主动接受

- 对系统更强的控制力

- 应用程序通常无法满足 power user 的全部需求

- 生产系统编程的事实标准

- Linux, macOS, Windows, ...运维基本都靠命令行工具

- 来自开源社区的一份礼物

- 非常丰富、可定制、看得见源码的软件栈

- [流传自远古时代的OS实验课程网站中的Linux入门教程](#)

# Linux命令行概述

Linux

Windows + WSL

# 第一课

- 这是个啥？让我用这个度过余生？

\$ █

Unix is user friendly. It's just selective about who its friends are.

Read the manual.

Search the web.

# 为什么大家在一开始都感到困难？

- 1980s: 以 MHz 为单位的主频；80 x 24 的字符终端
  - 无法提供丰富的交互界面
  - 但依然要完成各类任务
  - 你会如何设计？

不可避免，需要用户查阅手册记住一些系统里的约定（下面是最重要的一些）。

- 目录结构、文件命名规律、访问权限
- 命令执行的约定
- 常用命令行工具
- Shell 编程语言

# 常见的命令行工具

```
missing:~$ date  
Fri 10 Jan 2020 11:49:31 AM EST  
missing:~$
```

```
missing:~$ ls  
missing:~$ cd ..  
missing:/home$ ls  
missing  
missing:/home$ cd ..  
missing:/$ ls  
bin  
boot
```

```
missing:~$ ls -l /home  
drwxr-xr-x 1 missing users 4096 Jun 15 2019 missing
```

```
...
```

# 常见的命令行工具

```
missing:~$ pwd
/home/missing
missing:~$ cd /home
missing:/home$ pwd
/home
missing:/home$ cd ..
missing:/$ pwd
/
missing:/$ cd ./home
missing:/home$ pwd
/home
missing:/home$ cd missing
missing:~$ pwd
/home/missing
missing:~$ ../../bin/echo hello
hello
```

# 常见的命令行工具

```
missing:~$ echo hello  
hello
```

```
missing:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  
missing:~$ which echo  
/bin/echo  
missing:~$ /bin/echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
missing:~$ echo hello > hello.txt  
missing:~$ cat hello.txt  
hello  
missing:~$ cat < hello.txt  
hello  
missing:~$ cat < hello.txt > hello2.txt  
missing:~$ cat hello2.txt  
hello
```

# 常见的命令行工具

- 查看文件
  - ls (list), cd (change directory), find, tree, ...
- 阅读手册
  - man (man man); apropos; info
- 文本处理
  - cat (concatenate), wc (word count), grep (g/re/p)
  - tr (translate), cut, awk, sed (stream editor)

命令行工具多到什么程度呢.....

- unzip, strip, touch, finger, grep, mount, fsck, more, yes, unmount, sleep.....

# UNIX哲学

- *Keep it simple, stupid.* (KISS)
  - *Everything* is a file and *pipeline* programs to work together
- 一个命令只做 “一件事”
  - 从stdin输入 (scanf)
  - 向stdout输出 (printf)
  - 使用参数控制行为 (int main (int argc, char \* argv[]);)
- 命令的输入和输出都是人类 + 机器均可读的文本
  - find .
  - wc -l a.txt b.txt
- 把命令的输入/输出连接起来 (管道) 协作完成任务
  - find . | grep '\.cpp\$' | xargs cat | wc -l

# The Shell Scripting Language

祝贺！刚才其实是Shell执行了一段Shell语言编写的程序。

- Shell 是一门基于字符串和命令的编程语言
  - `a=hello` - 赋值 (注意 = 左右没有空格)
  - `$a` - 将变量的值 “粘贴”
  - `$(cmd)` - 将 `cmd` 运行的 stdout “粘贴”
  - `if cmd; then; ... ; fi` - 根据 `cmd` 运行结果执行分支
  - `cmd > file` - 把 `cmd` 的 stdout 重定向到 `file`
  - `cmd1 | cmd2` - 把 `cmd1` 的 stdout 作为 `cmd2` 的 stdin
- 有趣的小问题
  - 如何用 `if` 比较存储了整数字符串的大小?
  - (`if 1 > 2` 会发生什么? )

# 开始编程吧！

- 输出当前用户是不是root
  - `[ $UID -eq 0 ] && echo "is root!"`
- 查看磁盘引导扇区 (Master Boot Record)
  - `cat /dev/sdb | head -c 512 | ndisasm -b 16 -o 0x7c00 -`
- 统计所有cpp文件的行数
  - `find . | grep '\.cpp$' | xargs cat | wc -l`
- 统计命令行命令的频率
  - `history | tr -s ' ' | cut -d ' ' -f3 | sort | uniq -c | sort -nr`
- 以上都是命令行里的命令，但同时也是 bash script 的程序
  - (这就是程序员的宿命吧)

# 自动化工具：程序员福利系列

---

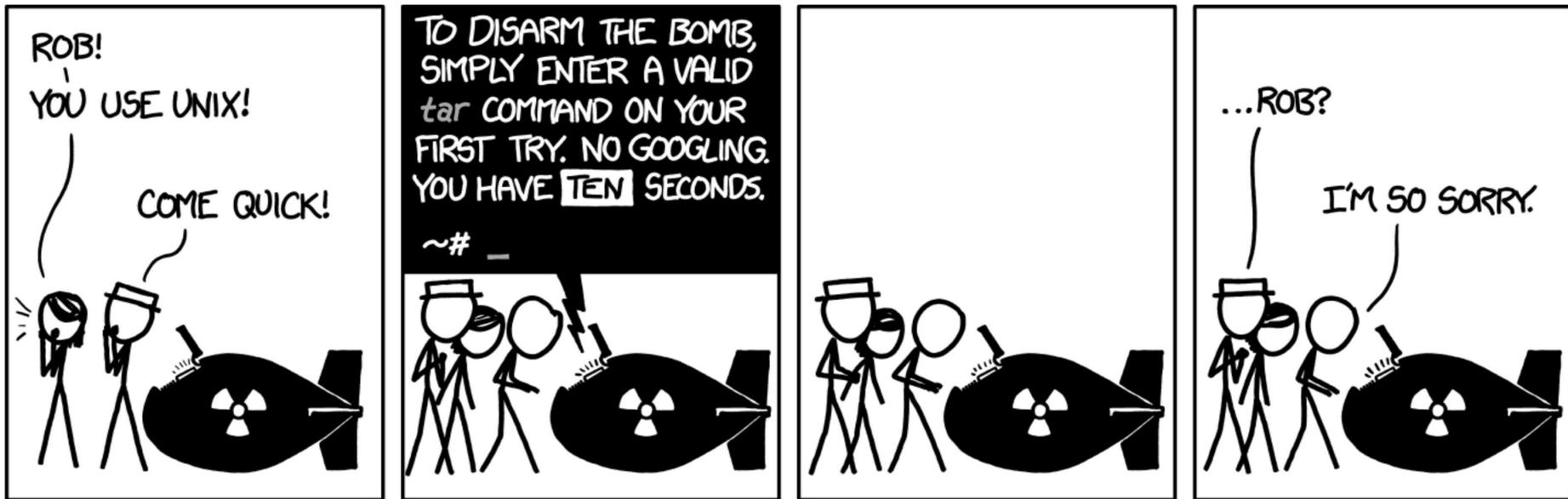
- 重构福利
  - (ex) 对一个目录里的所有.cpp文件执行同样的vim动作
- LaTeX用户福利
  - (pdftocrop) 把LibreOffice的演示文稿导出为PDF，然后将每一页的白边部分裁去，分别命名为fig-1.pdf, fig-2.pdf, ...
- 视频制作福利
  - (ffmpeg) 为视频添加水印和字幕

# 生存指南

# 拥抱变化

- 很烦躁：一下就碰壁？
  - 胡乱尝试一通？对了就对了，不对就抓瞎？
  - 有时候不知道该怎么STFW，RTFW？
- 静下心来，从头开始
  - [The Missing Semester of Your CS Education](#)
  - [流传自远古时代的OS实验课程网站中的Linux入门教程](#)
  - RTFM (slides), STFW, RTFSC
    - 不要觉得Makefile、提交脚本都是和你无关的
- 试图理解一切事情是如何发生的
  - 从读得懂开始，逐渐过渡到会写

# 现代化你的命令行工具



- 例子:

- tldr (替代man)
- zsh-z (替代pushd/popd)
- fzf
- vim/vscode

# 拥抱开源社区

- 用好 Github的 “[Awesome](#)” 系列
  - 例如[The art of command line](#)
- 用好 Stack Overflow / Stack Exchange
- 禁用百度和中文关键字（强烈不推荐大家使用中文系统）
  - Linux/macOS
    - /etc/hosts
  - Windows
    - C:\Windows\System32\drivers\etc\hosts
  - 增加一行127.0.0.1 www.baidu.com

# 选讲PA0

# 总结

# 本次课程最重要的内容

---

- 静下心来，从头开始
  - [The Missing Semester of Your CS Education](#)
  - RTFM (slides), STFW, RTFSC
- 用好 Github的 “[Awesome](#)” 系列
  - 例如[The art of command line](#)
- 用好 Stack Overflow / Stack Exchange
- 禁用百度和中文关键字

# 总结

- 你不会感到学习这门课很舒适
  - 不要用“我们学得比较理论……”来骗自己
    - 就是不扎实
  - Academic integrity 可能让你感到三观尽毁
    - 你会理解到南大还不是“世界一流大学”
- 但请不要放弃/躺倒
- 你们未来是要承担大事业的

[PA0](#)已发布，暂时未上线OJ，请关注群通知

# 一些福利建议

---

- 用好Git
- 关注DDL

# 机器永远是对的 (and RTFM)

RTFM: Read The Friendly Manual

STFW: Search The Friendly Web

# 自动化工具：程序员福利系列 (cont'd)

```
#自动登陆p.nju.edu.cn不香吗？  
curl -d "username=学号&password=密码" \  
http://p.nju.edu.cn/portal_io/login
```

- 这有什么用？
  - 校园网内某台机器的长久连接 (brasd)
  - 作为你的校内代理服务器
  - 实现内网穿透.....
- 一个有趣的问题
  - 密码是明文，被舍友偷窥了怎么办？？？
  - 文件系统有权管理：chmod -r
    - 但是shell script必须读权限？