CNCC

南京大学
计算机系统能力培养探索与实践

袁春风
南京大学计算机系
2019.10

主要内容

- 为何需要进行系统能力培养
- 目前存在的问题
- 南京大学的探索与实践
  - 计算机系统能力培养总体思路
  - 三门新建纵向系统综合实验课
  - 四条横向关联融合的实验链

举例：如何考察程序的执行时间

阿里笔试中有这样一道题目：

在一台主流配置的PC上，调用f(35)所需要的时间大概是（　）。

```
int f(int x) {
   int s = 0;
   while(x++ >0) s+= f(x);
   return max(s,1);
}
```

A．几毫秒　B．几秒　C．几分钟　D．几小时

显然，考的不仅仅是程序设计！

题目解析

学生的答案PPT

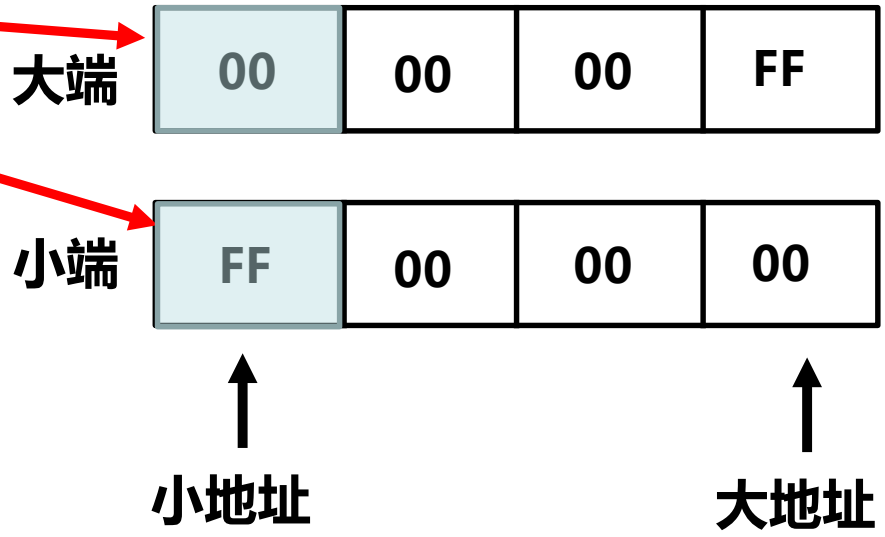
计算printf函数的时间PPT

举例：大端小端问题

中兴笔试题：写程序判断当前CPU是大端CPU还是小端CPU，并作简要说明。

有学生告诉我，他的同学写了一下程序，判断出他的PC是大端！

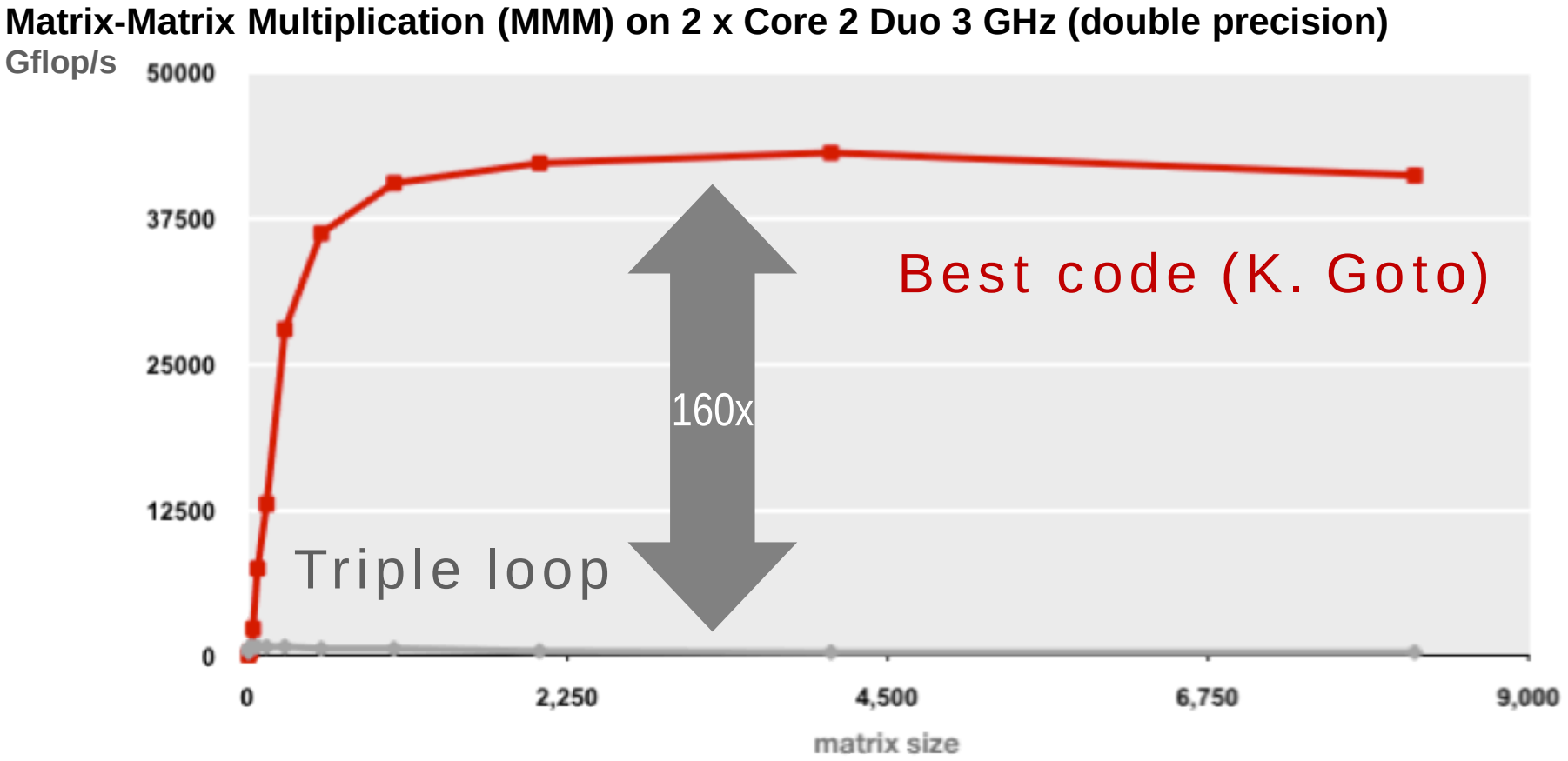
```
union  test {
        int     a;
        char  b;
}
main( ) {
       test.a=0xff;
       if (test.b==0xff)
              printf( “Little endian);
       else
              printf( “Big endian);
}
```



该同学的程序错在哪里呢?

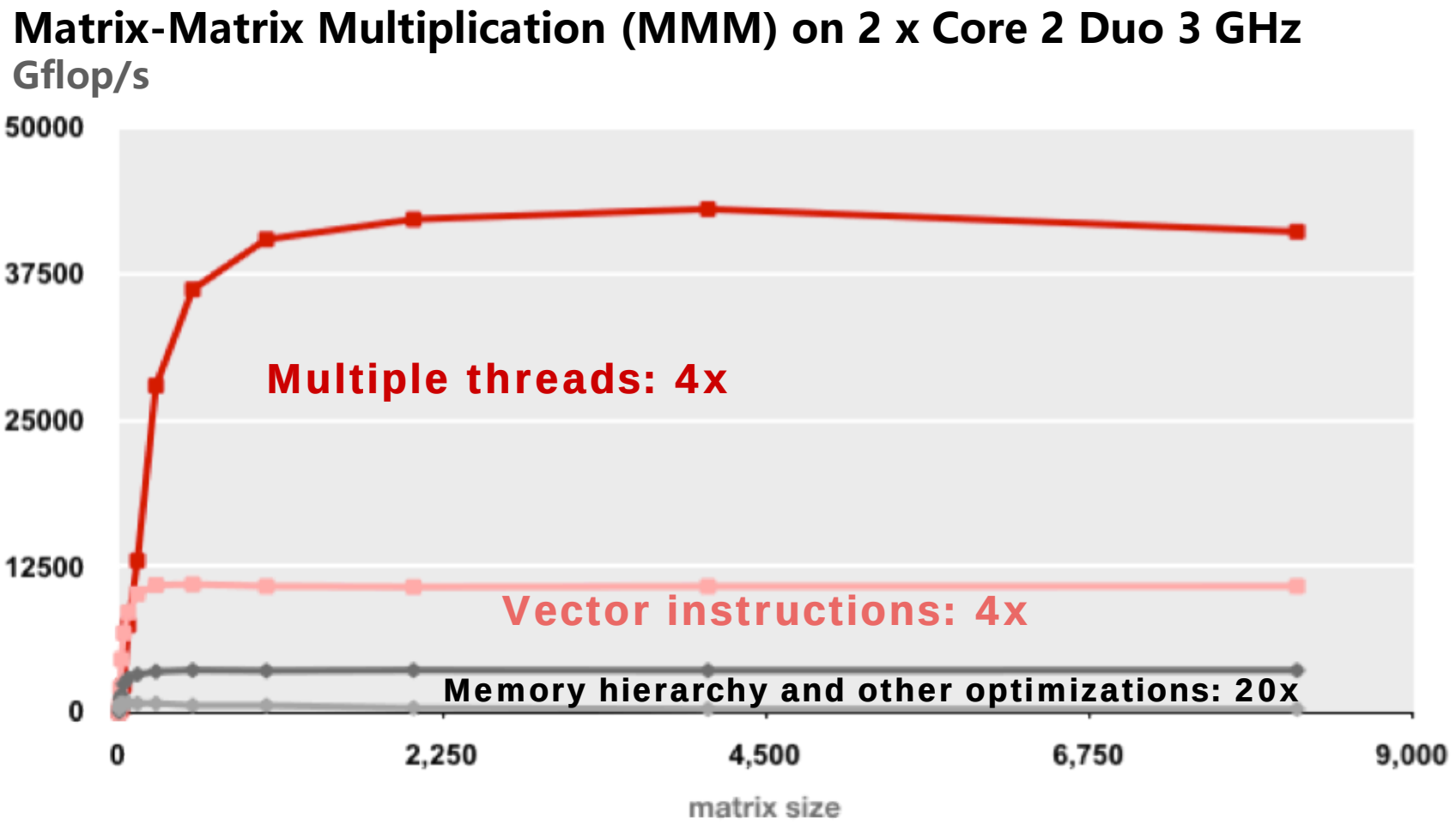
显然，考的也不仅仅是程序设计！

用“系统思维”分析问题



- **Standard desktop computer, vendor compiler, using optimization flags**
- **Both implementations have exactly the same operations count ($2n^3$)**
- **What is going on?**

用“系统思维”分析问题



- Reason for 20x: Blocking or tiling, loop unrolling, array scalarization, instruction scheduling, search to find best choice
- Effect: fewer register spills, L1/L2 cache misses, and TLB misses

系统研究对系统能力的需求

- 某项目开发案例
- 现象： wget下载文件(大于20KB)中途网卡无响应
  - 有IP地址, 但ping不通
- 系统全栈调试
  - printk, tcpdump, wireshark, 断点, verilog波形
  - 以太网驱动, 中断处理函数, DMA描述符, ring buffer大小, AXI crossbar连接, AXI与AXI Lite协议转换
- 问题原因： 以太网中断实际上为level-trigger, 但在PIC(可编程中断控制器)端默认配置为edge-trigger
  - 当中断频繁到来时, PIC可能会因屏蔽而错过一些上升沿
  - 以太网一直在发中断(高电平), 但PIC以为没有(只认上升沿)

微结构实现问题
影响上层软件的运行结果

系统研究对系统能力的需求

- 某项目开发案例
- 任务： 写一个L2 cache命中率尽可能低的差性能程序
- 踩过的坑
  - 库函数 -> 有访存, 贡献了命中率
  - 编译器 -> 循环变量命中率高
  - 软件维护TLB -> TLB缺失需要内核重填, 贡献了命中率
  - 网卡中断 -> 中断处理贡献了命中率
  - Linux虚存管理 -> buddy分配算法不保证物理地址连续
  - 包含式 -> 脏块回写L2必定命中

原文 ->

微结构性能指标
受到上层软件的影响

理解程序的行为须有“系统思维”

程序执行结果

不仅取决于

算法、程序的编写

而且取决于

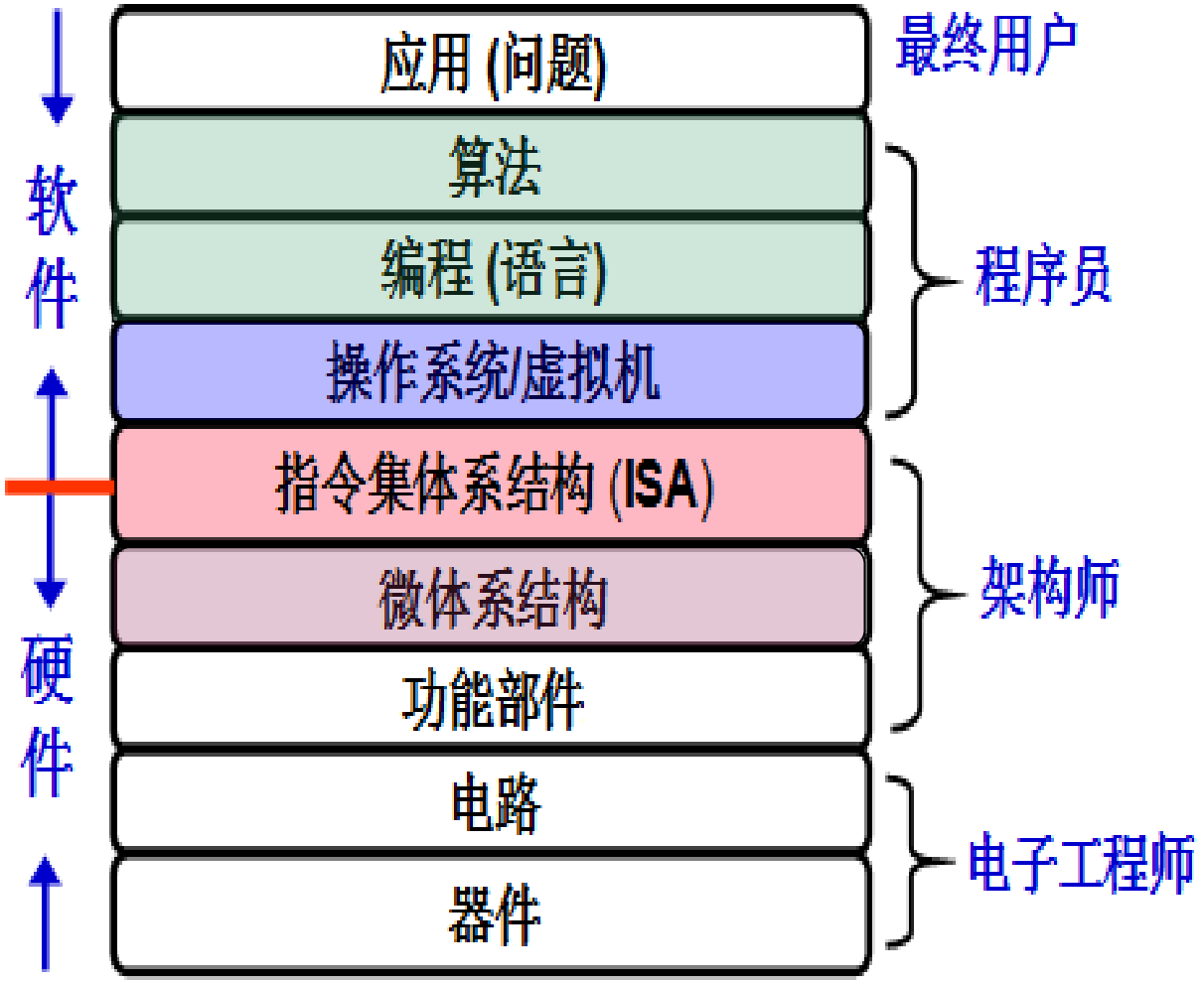
语言处理系统

操作系统

ISA

微体系结构

计算机系统抽象层的转换



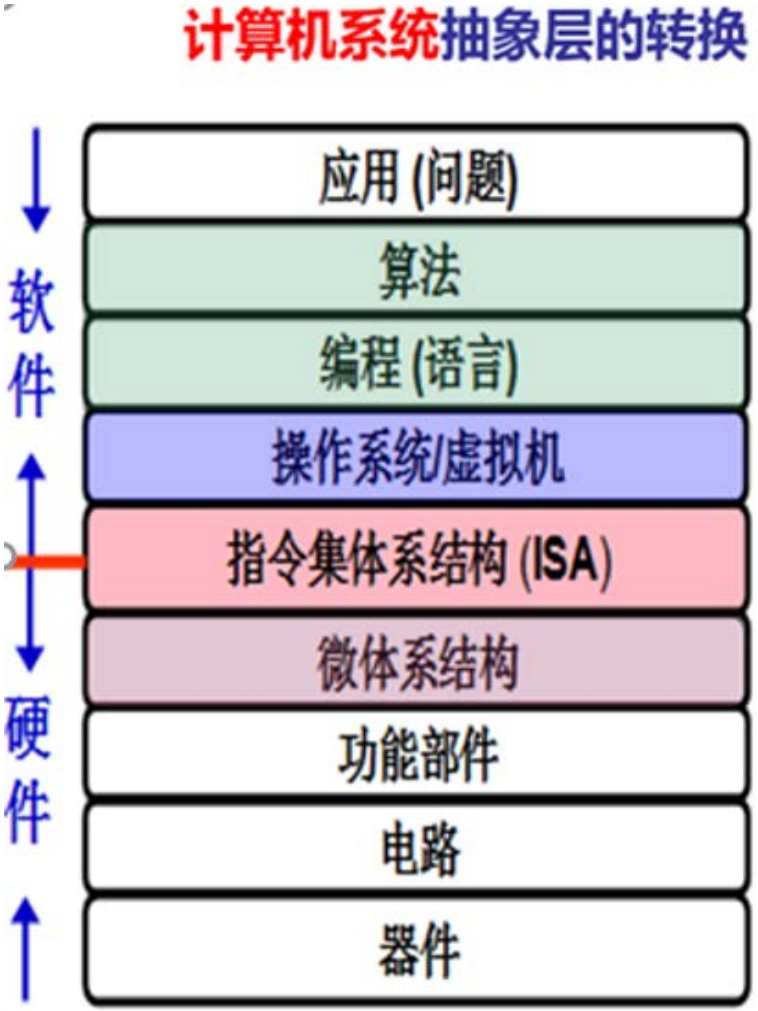
系统能力对系统开发研究人员很重要！

对应用程序员也很重要！

系统能力指什么？

能在系统层面进行分析、设计、调优、检错，站在系统高度解决应用问题。

- 深刻理解系统各层之间的关系
- 对软、硬件功能进行合理划分
- 对系统不同层次进行抽象和封装
- 对系统整体性能进行分析和调优
- 对系统各层面的错误进行调试和修正
- 对系统进行准确的性能评估和优化
- 根据不同应用要求合理构建系统框架
- ……



计算机专业不仅要培养程序员，更需要培养能够构建计算机系统的专门人才！

系统能力是计算机专业重要能力之一

教育部计算机类教指委研究报告：

细化成有不同程度要求的82个能力点

4大专业基本能力		能力点数
计算思维能力		9
算法设计与分析能力		8
程序设计与实现能力		3
系统能力	系统认知	6
	系统设计	19
	系统开发	23
	系统应用	14

系统能力的能力点占4大专业基本能力总能力点的75%。

摘自：教育部计算机类教指委副主任王志英教授的PPT

主要内容

- 为何需要进行系统能力培养
- 目前存在的问题
- 南京大学的做法及成效
  - 计算机系统能力培养总体思路
  - 三门新建纵向系统综合实验课
  - 四条横向关联融合的实验链

美国大学的情况

- Yale Patt 在密西根大学时教组原课时，也按传统的内容，就硬件讲硬件，发现学生不喜欢
- 深入调研后认识到：要改变教学内容，要把硬件和软件结合起来讲解，使学生具有计算机系统整体概念
- 于是，编写并出版了：

  《Introduction to Computing Systems from Bits and Gates to C and Beyond》

  描述了一个“小而完整”的系统
- 书中描述了计算机系统各个抽象层 →
- 他在UTAustin一直开设该课程（报告）

  “Java is nothing!”

SKIP

美国大学的情况

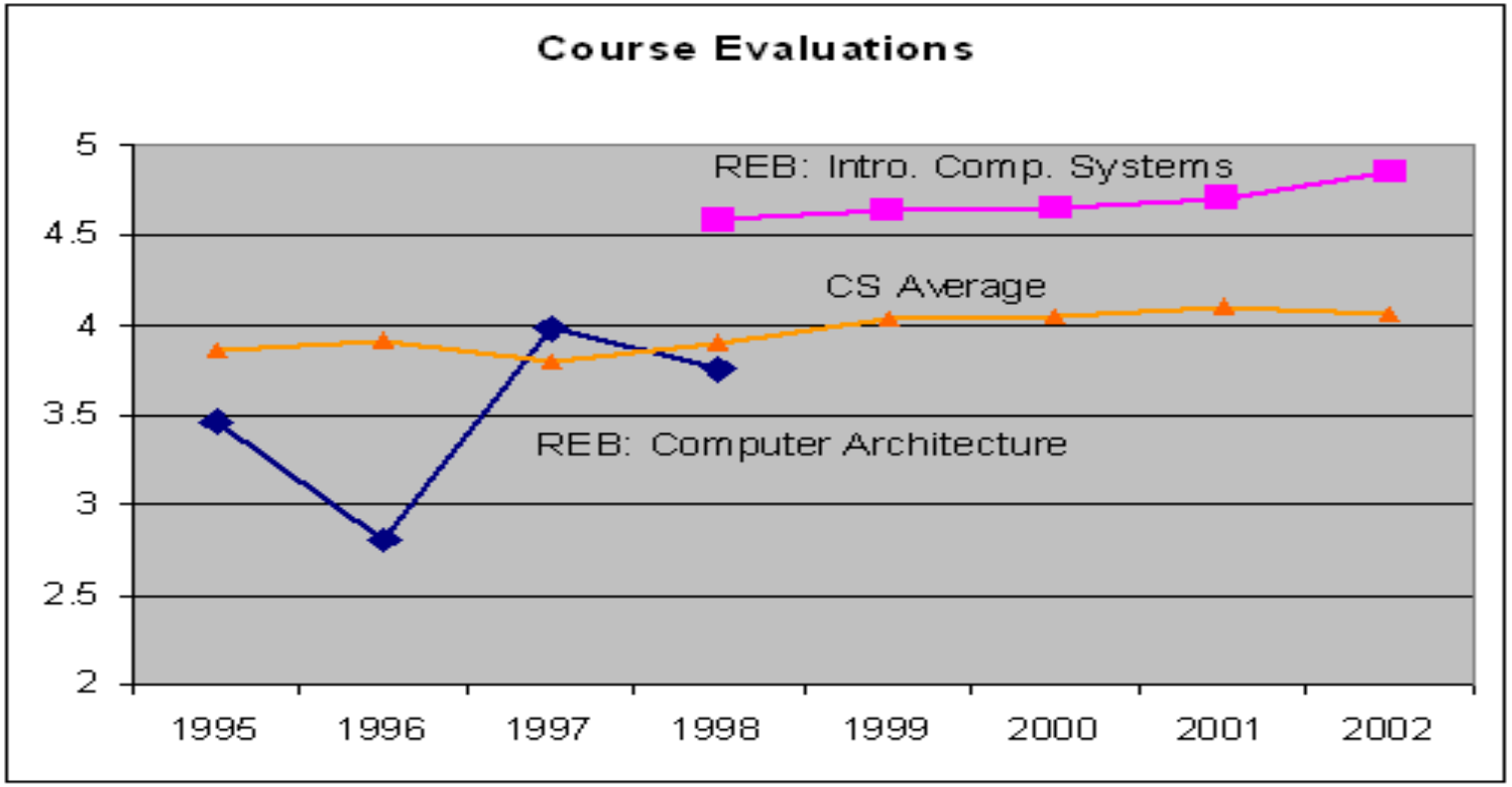
- Yale Patt：曾任美国密歇根大学计算机体系结构实验室主任多年，鉴于他在计算机发展历程中的贡献及对计算机科学教育的深刻理解和倾心投入，被IEEE Spectrum评为美国计算机界的卓越泰斗（与《计算机程序设计艺术》的作者，图灵奖获得者Donald Knuth齐名，全球只有他们俩人享此殊荣），在美国乃至世界计算机体系结构领域有着广泛的影响力。
- 多年来，从计算机科学和计算机工程院系的教学实践中，他们认识到传统的计算机课程体系中缺少帮助本科生建立软件与硬件联系的课程，使得他们对计算机系统中一些非常重要的基本概念缺乏深入理解。例如，无法清楚地解释指针变量的硬件实现；而栈、递归概念更像是在“变魔术”，难以理解。

美国大学的情况

Background (来自CMU SCS的R. E. Bryant院长)

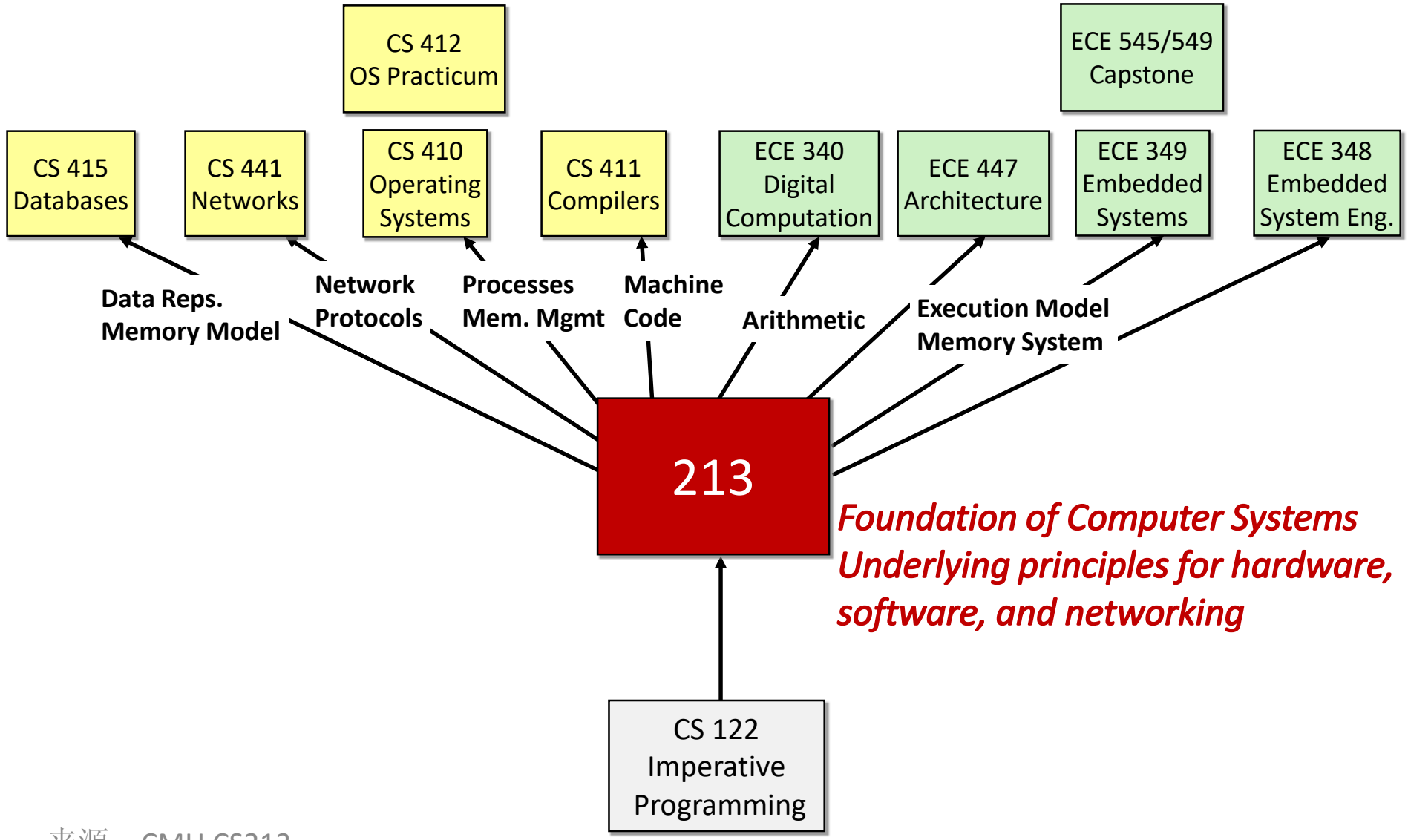
1995-1997: REB/DROH teaching computer architecture course at CMU.

- Good material, dedicated teachers, but students hate it
- Don't see how it will affect there lives as programmers



深有感触

Role within CS/ECE Curriculum



Foundation of Computer Systems
Underlying principles for hardware, software, and networking

来源：CMU CS213

美国大学的情况

15-213: Intro to Computer Systems

Goals

- Teach students to be sophisticated application programmers
- Prepare students for upper-level systems courses

Taught every semester to 400+ students

- All CS undergrads (core course)
- All ECE undergrads (core course)
- Many masters students
  - To prepare them for upper-level systems courses
- Variety of others from math, physics, statistics, …

Preparation

- Assume know C programming

美国大学的情况

偏硬 → 偏软

	MIT 6.004	USB 61C	Stanford CS107	CMU 15-213
课程名称	Computation Structures	Great Ideas in Computer Architecture （Machine Structures ）	Computer Organization and Systems（COS）	Introduction to Computer System（ICS）
开设专业*	EE、CS	CS	CS	CS
课程描述	门电路→功能部件→单周期和流水线 CPU；C 语言→汇编→指令→过程→进程；并行、性能评价	C 语言→汇编→指令；应用级并行→数据级并行→线程级并行→指令级并行→寄存器传送级硬件描述	C 语言→汇编→指令→微体系结构；编译→链接→装入→执行；程序性能优化、存储器结构与管理、并发和多线程、网络编程	C 语言→汇编→指令→微体系结构；编译→链接→装入→执行；程序性能优化、存储器结构与管理、并发和多线程、网络编程
教材**				
模型机				
助教人数				
先行课程或要求				
实验内容	共 8 个实验，涉及门电路特性、ALU、图灵机、汇编、处理器设计、鼠标中断方式 I/O 等	共 12 个实验，4 个大作业，涉及到 Debug、EC2、MapReduce，MIPS 汇编、数据级并行、线程级并行、Cache、虚拟存储管理等，大作业和实验成绩占 60%	涉及数据的表示、堆区分配、过程调用和栈的构成及使用、溢出、编译工具和编译优化等，大作业和实验成绩占 60%	示、Cache、缓冲区溢出、过程调用及栈的构成与使用、堆的分配、代理设置等，实验成绩占 50%
实验手段	各类模拟器	编程、云计算平台、模拟器	编程	编程
相关后继课程或教学内容	“数字系统设计”和“计算机体系结构及系统”等，涉及 CPU 设计实验和体系结构方面的模拟实验	“数字系统设计”和“计算机体系结构及工程”等，涉及 CPU 设计实验和体系结构方面的模拟实验，前者是同时为 CS 和 EE 的学生开设的课程	后继课程为“Digital Systems II”，教材为 COD"P&H"，主要是流水线 CPU 和存储系统设计，实验为用 DHL 设计 CPU（CS 学生不需要）	CS 学生可选 ECE 开设的课程“Introduction to Computer Architecture”，教材为 COD"P&H"，主要是流水线 CPU 和存储系统设计，实验为用 Verilog 语言设计 CPU

Bottom Up
以汇编为中心
强调底层设计
强调硬件与OS的接口

TOP DOWN
以不同粒度的并行为线索
涉及各个层次
实验跨度大

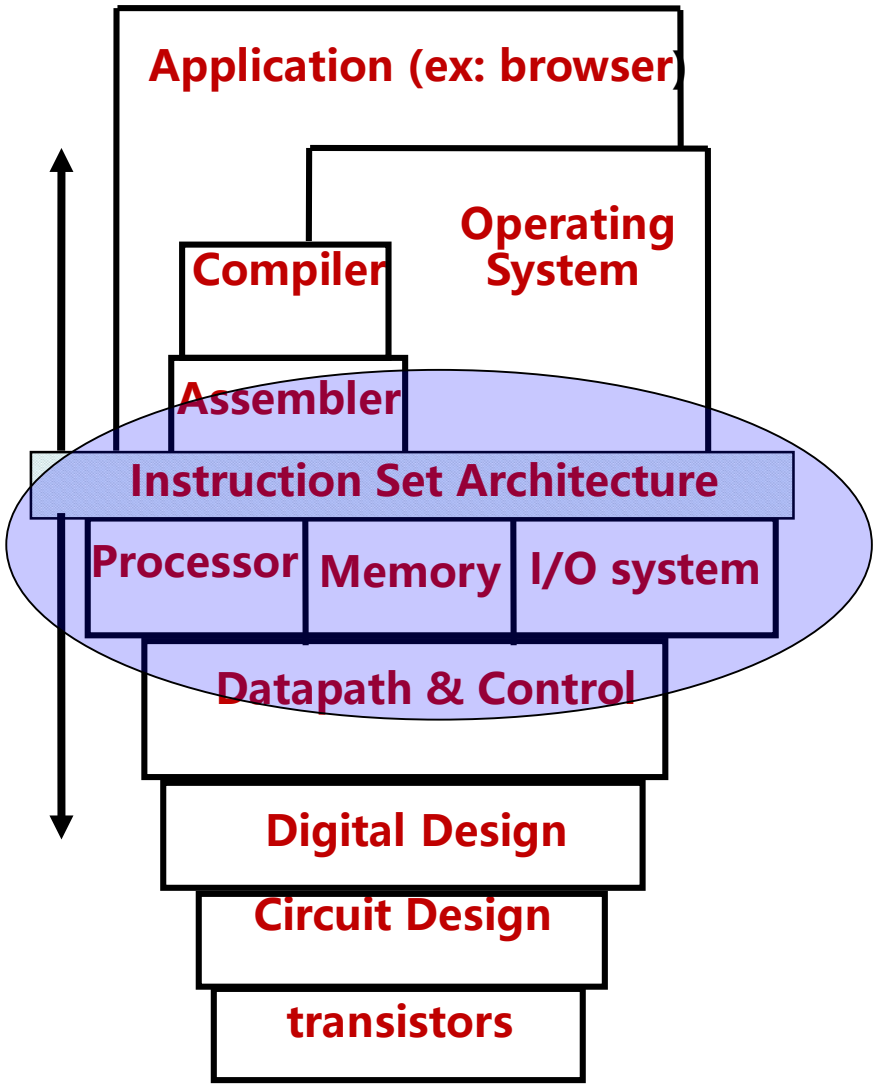
MIXED
强调从程序员角度看到的底层内容
重点为C语言的底层支持、程序优化、存储器分配管理，而不介绍底层数据通路等的具体实现

美国大学的情况

- 都是EE和CS分别开课，互选共享
  - EECS：MIT、UC Berkeley EE(ECE)/CS：Stanford、CMU
- 都采用分流培养模式
  - MIT：EE、EECS、CS三种学位（不同学位还分若干方向）
  - UCB：ECE（E、CNS、CSys）、CSE（CSci）、Mix、Honor
  - Stanford：AI、BioCo、CE、Graph、HCI、Info、CSys、Theory、Unspecialized、Individually Designed
  - CMU SCS：AI、Cognitive Modeling、CSys、Graph、Theory

    偏硬件的方向有：EE、ECE、CE、Csys
- 都有一门所有专业必修的介绍计算机系统的入门课（相当于组原？）
  - MIT 6.004、UCB CS61C、Stanford CS107、CMU CS213
  - 教材有P&P、P&H、B&O，配合K&R
  - 强调C语言数组和指针、过程调用的底层实现、堆的分配、中断、异常、…
- 计算机硬件设计的课程（相当于组原？）都在EE开设，偏硬件方向学生必选，Csys以上方向可以不选（教材有P&H、H&H）
- 关于计算机系统更高级的内容在EECS或Csys及其以上方向开设
  - 例如，MIT 6.033、Stanford CS110 （相当于CS2013的SF）
  - 教材：Principles of Computer System Design: An Introduction by Jerome H. Saltzer and M. Frans Kaashoek（MIT）

中、美情况对比



1. U：EE和CS共享

   C： EE和CS各自为阵

2. U：先建立系统概念后分流培养

   C：没有贯穿系统的前导课程，没有体现分流

3. U：课程内容纵向关联

   C：课程内容横切、关联弱

4. U：先系统框架后细节实现

   C：先数字电路后组成原理

5. U：4-5门/学期，学得深入

   C：7-8门/学期，精力不足

国内部分高校存在的问题

1. 重理论 （部分系统课程基本靠“死记硬背”）
2. 轻实践（学生不能设计CPU、不会写内核程序）
3. 缺关联（不知程序、编译器、ISA、OS等之间的关系）
4. 少综合（利用多个知识点分析复杂问题的能力弱）
5. 弱（无）系统观（没有计算机系统整机概念）

计算机专业全国研究生统考结果证明了上述结论

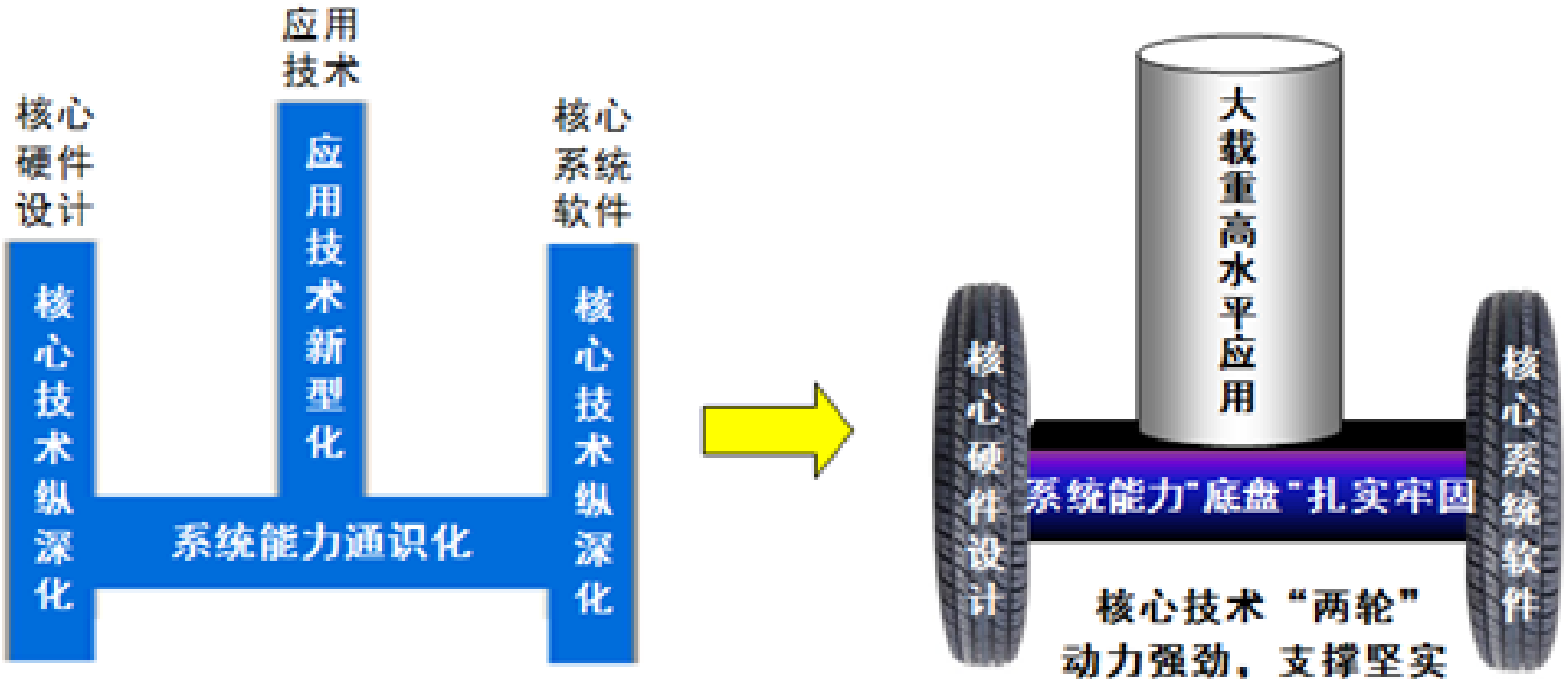
2012年计算机类教指委开始成立计算机类专业系统能力培养研究小组，在全国推广相关研究成果

计算机类专业系统能力培养的改革与实践项目成员单位

主要内容

- 为何需要进行系统能力培养
- 目前存在的问题
- 南京大学的做法及成效
  - 计算机系统能力培养总体思路
  - 三门新建纵向系统综合实验课
  - 四条横向关联融合的实验链

计算机系统能力培养总体思路

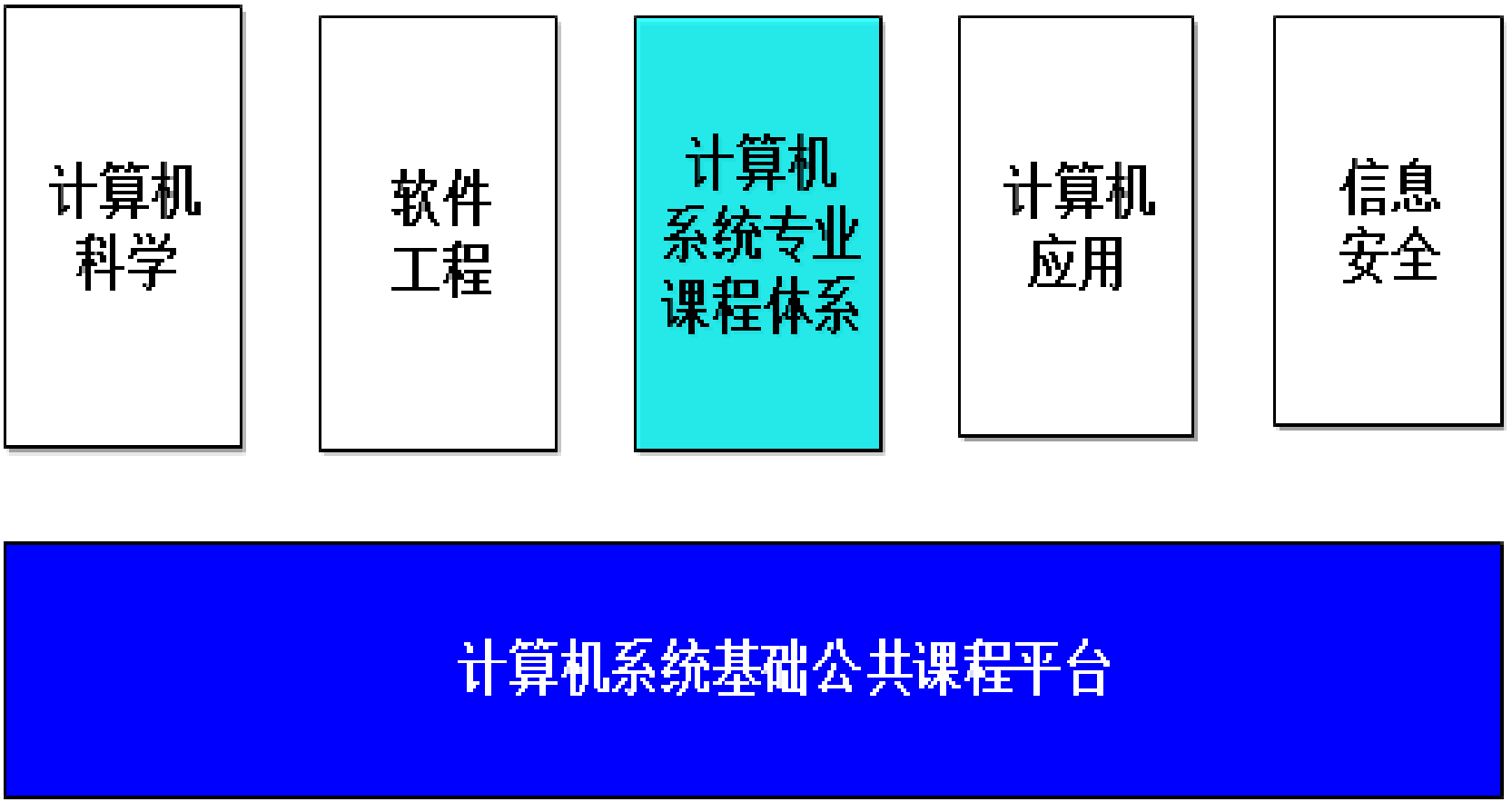


一横三纵的"山"字型教学理念与专业技术人才培养模式

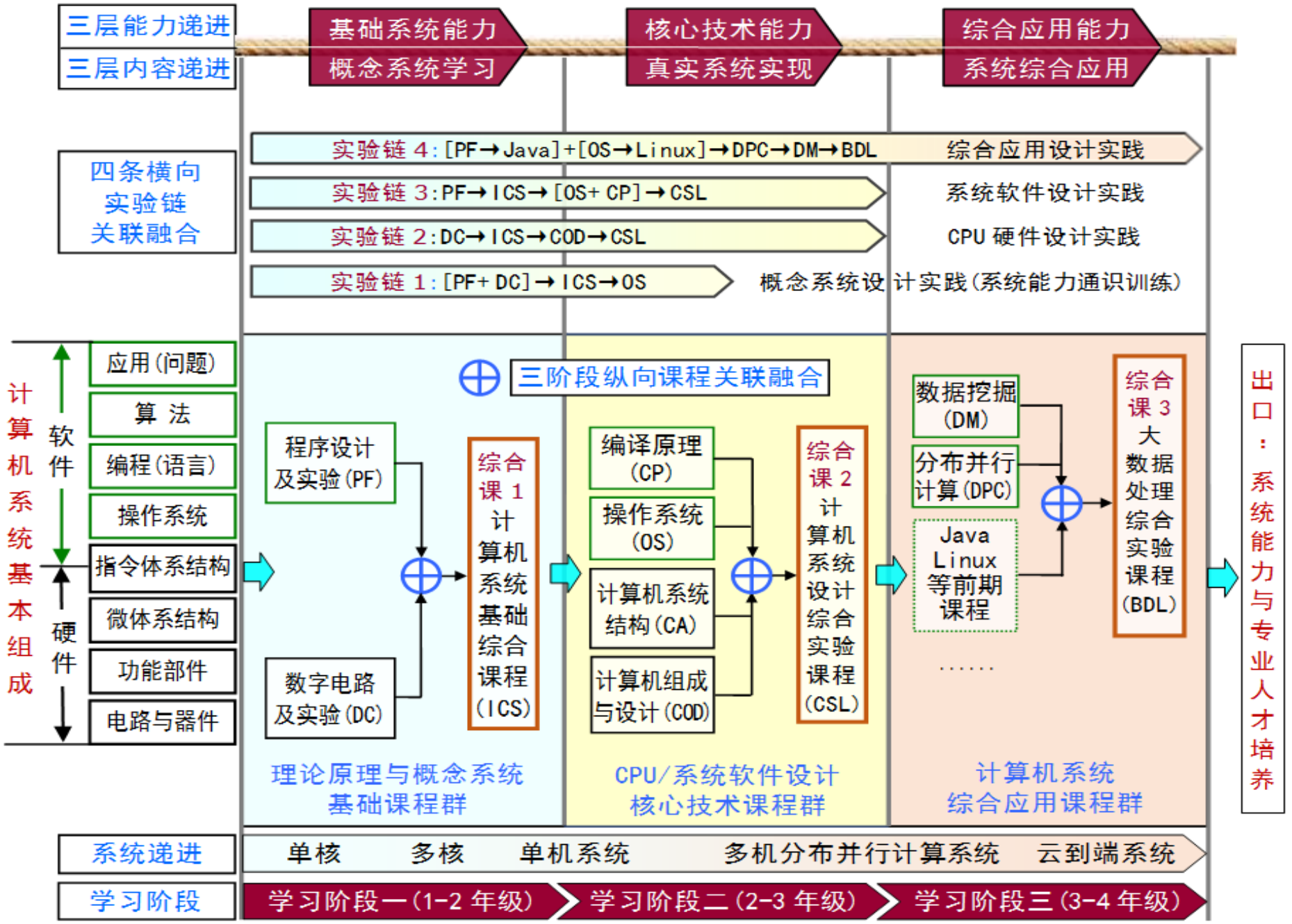
培养所有学生扎实的系统能力基础、系统方向学生强劲的系统核心能力，以及各有侧重的系统应用开发能力，以适应社会和国家战略对专业人才结构和队伍的需求

2013版新教学计划框架

分流培养：5个方向



系统能力培养的两个重要建设点



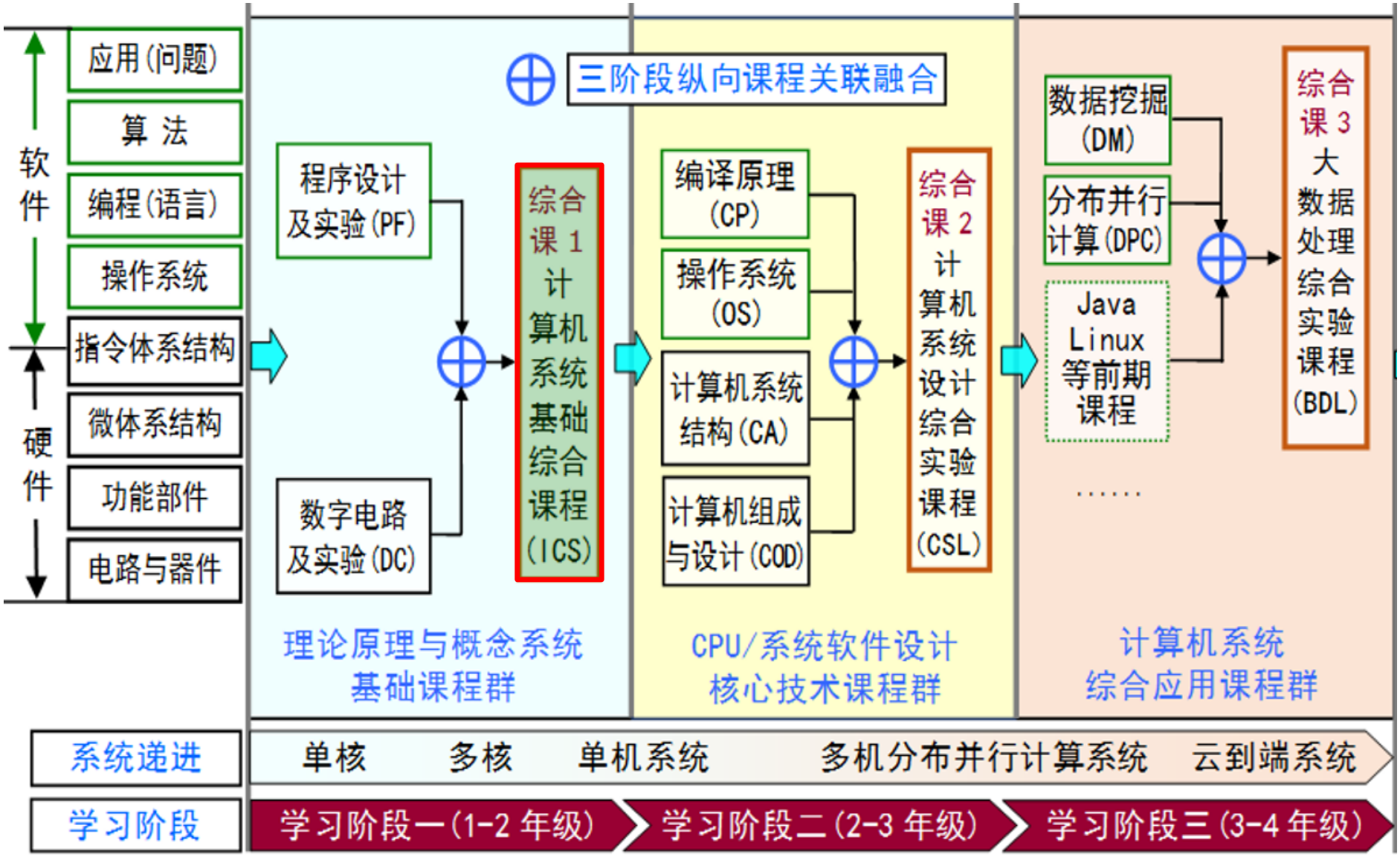
三纵四横、双重关联、经纬交织的网状课程体系


21世纪大学本科
计算机专业系列教材
蔡晓燕 编著
FPGA数字逻辑设计
http://www.tup.com.cn
"十二五"普通高等教育本科国家级规划教材 配套用书
21世纪大学本科
计算机专业系列教材
张泽生 编著
计算机组成与设计实验
http://www.tup.com.cn
COMPILERS: PRINCIPLES AND GUIDED PRACTICE
编译原理实践
与指导教程
普通高等教育"十一五"国家级规划教材
教育部普通高等教育精品教材
"十二五"江苏省高等学校重点教材
面向CS2013计算机专业规划教材
程序设计教程
用C++语言编程
第3版
"十二五"普通高等教育本科国家级规划教材
普通高等教育"十一五"国家级规划教材
21世纪大学本科
计算机专业系列教材
袁春风 编著
计算机组成与系统结构习题解答和教学指导(第2版)
http://www.tup.com.cn
INTRODUCTION TO COMPUTER SYSTEMS
Problem Solving and Teaching Guide
计算机系统基础
习题解答与教学指导
重点大学计算机教材
嵌入式系统
基础教程
"十二五"普通高等教育本科国家级规划教材
操作系统教程
(第5版)
计算机组成与系统结构(第2版)
袁春风 主编
杨若瑜 王帅 唐杰 编著
http://www.tup.com.cn
INTRODUCTION TO COMPUTER SYSTEMS
计算机系统基础
袁春风 编著
机械工业出版社
China Machine Press
深入理解大数据
大数据处理与编程实践
UNDERSTANDING BIG DATA
BIG DATA PROCESSING AND PROGRAMMING
主编：黄宜华（南京大学） 副主编：苗凯翔（英特尔公司）
BIG DATA
机械工业出版社
China Machine Press

主要内容

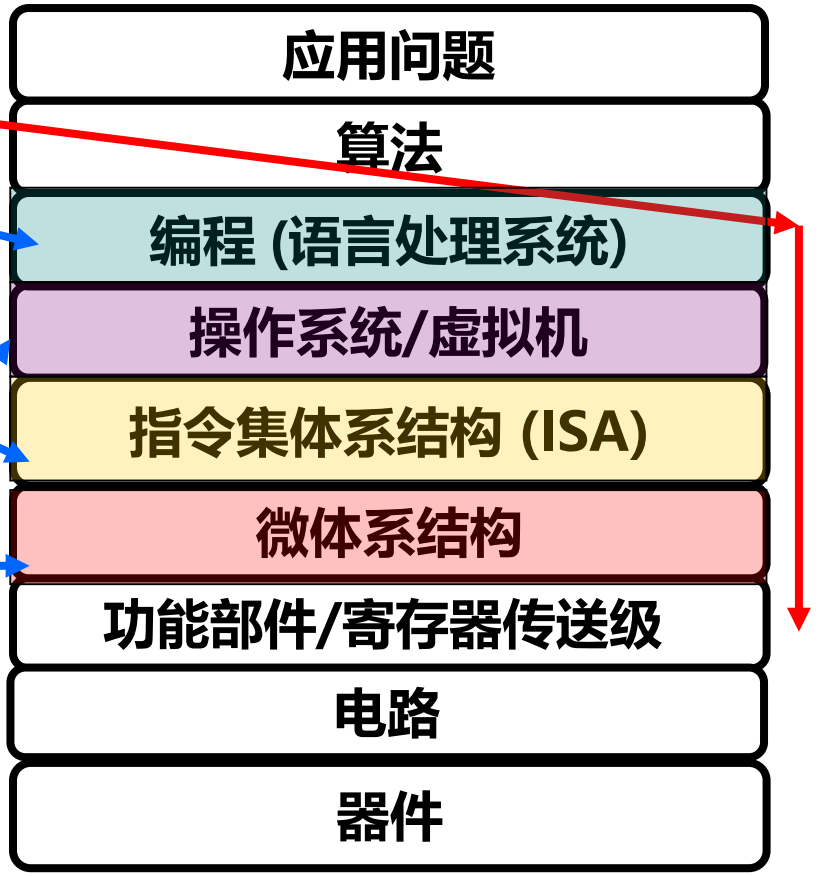
- 为何需要进行系统能力培养
- 目前存在的问题
- 南京大学的做法及成效
  - 计算机系统能力培养总体思路
  - 三门新建纵向系统综合实验课
  - 四条横向关联融合的实验链

三门新建系统综合课程之一：ICS



计算机系统基础(ICS)概要

- 思路：将编程和电路间各层串联，使学生清楚理解计算机如何生成和运行可执行文件?
- 重点在高级语言以下各抽象层
  - C语言程序设计层
    - 数据的机器级表示、运算
    - 语句和过程调用的机器级表示
  - 指令集体系结构（ISA）和汇编层
    - 指令系统、机器代码、汇编语言
  - 微体系结构及硬件层
    - CPU的通用结构
    - 层次结构存储系统
  - 操作系统、编译和链接的部分内容



定位：系统“通识课”，着重系统概念框架，而非系统具体实现

课程内容概要

```
/*---sum.c---*/
int sum(int a[ ], unsigned len)
{
    int   i，sum = 0;
    for   (i = 0; i <= len–1; i++)
          sum += a[i];
    return sum;
}

/*---main.c---*/
int main()
{
    int   a[1]={100};
    int   s;
    s=sum(a,0);
    printf(“%d”,s);
}
```

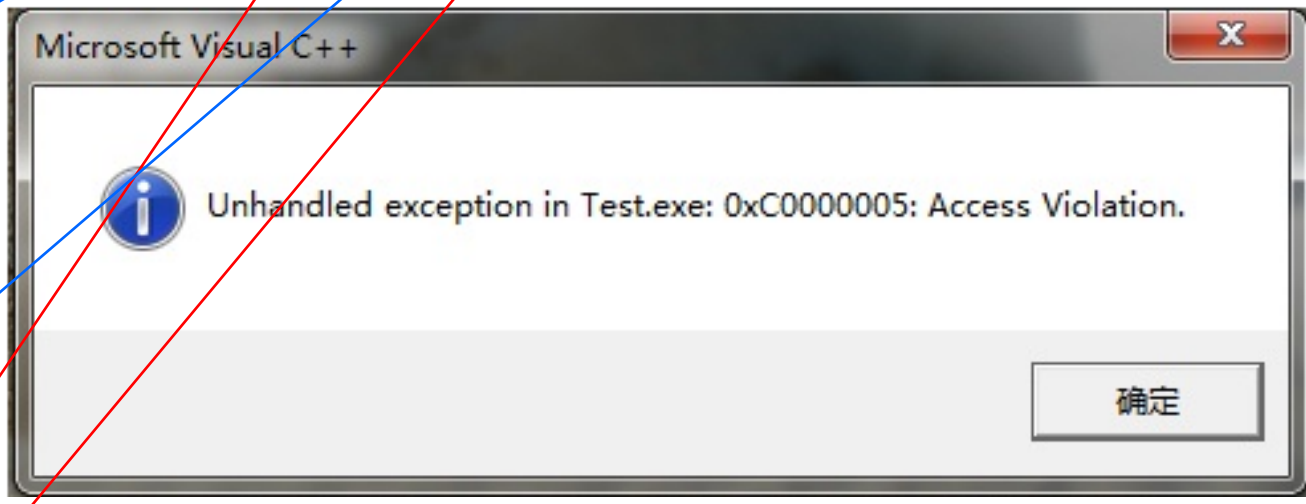
数据的表示

数据的运算

各类语句的转换与表示(指令)

各类复杂数据类型的转换表示

过程（函数）调用的转换表示



链接（linker）和加载

程序执行（存储器访问）

异常和中断处理

输入输出(I/O)

实验内容及实施情况

- **实验类型**
  - **Homework**

    小规模编程练习、运行结果分析
  - **Lab**

    数据表示（位操作）、二进制炸弹、缓冲区溢出、逆向工程
  - **Project**

    一个小型项目(Programming Assignment, PA)

    实现功能完备但简化的x86模拟器NEMU及一个简易调试器

    涵盖教材中约95%的内容（动态链接在Lab中）

    觉得自己上课听懂了？做做PA就知道了！

实验平台与工具

IA-32 + GNU/Linux + gcc + C

其它工具: gdb, make, git

PA最新发布网址：

- https://www.gitbook.com/book/nju-ics/ics2019-programming-assignment

Project（PA）的主要内容

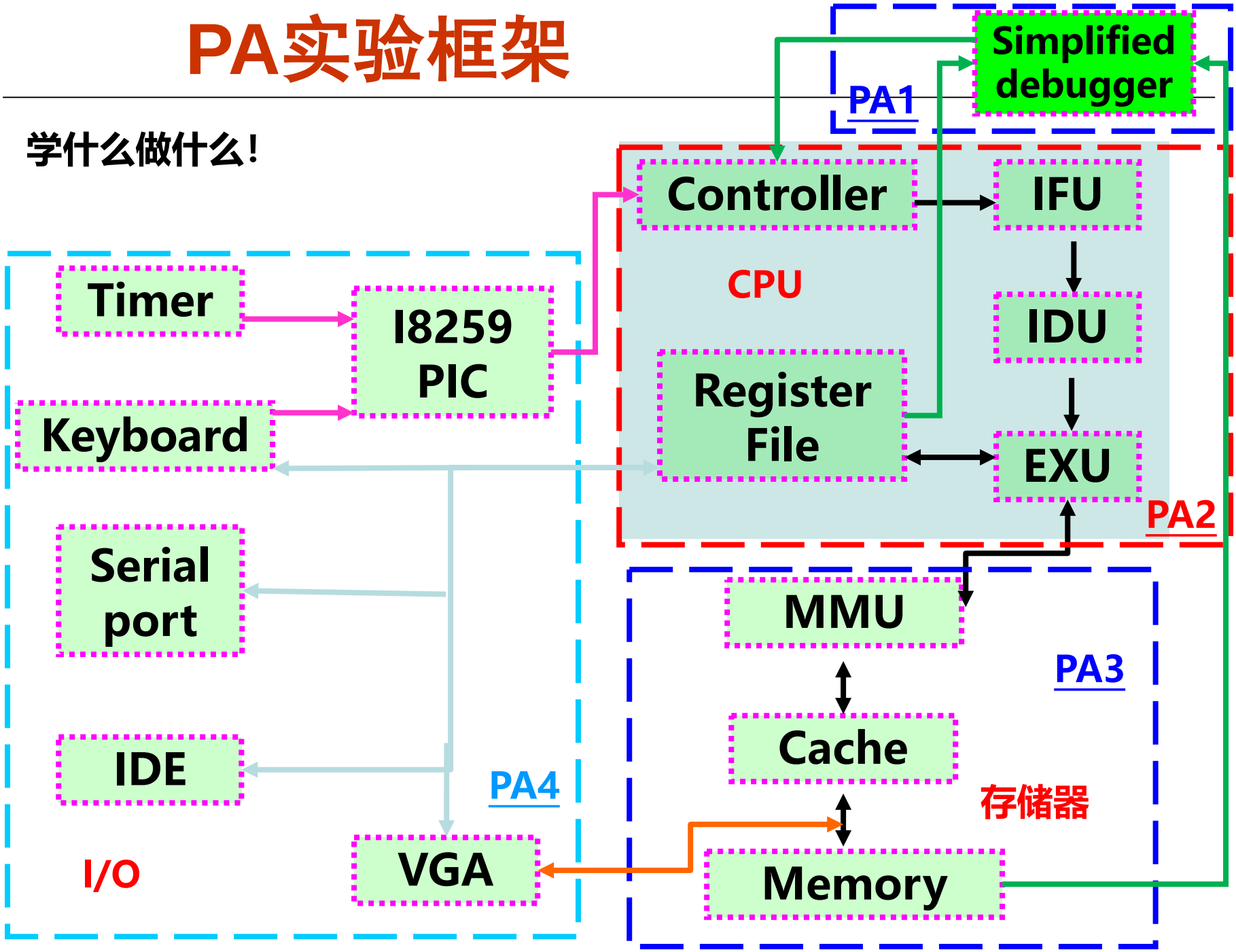
- 简易调试器（PA1）
  - 单步执行, 打印寄存器/内存, 断点, 表达式求值, 监视点
- CPU核心、简易调试器高级功能（PA2）
  - 支持x86保护模式下大部分常用指令（不支持x87指令）
  - 符号表, 调用函数链、ELF格式和加载
- 存储管理（PA3）
  - MMU：分段(GDT/LDT), 分页, TLB（不支持保护机制）
  - 两级联合cache（L1、L2）
  - DRAM（包含 row buffer 和 burst 的物理特性）
- 中断/异常/I/O（PA4）
  - IA-32中断机制(IDT)（不支持保护机制）
  - 时钟, 键盘, VGA, 串口, IDE, i8259 PIC的简单模拟
  - 独立编址I/O, 内存映射I/O（VGA）
  - 文件操作、系统调用（write）、键盘中断

例如，定点加减指令需考虑所有情况，并生成各标志，因而需理解教材中关于整数加减运算公式的含义。

模拟器实现的功能几乎涵盖“计算机系统基础”教材中的所有内容

PA实验框架

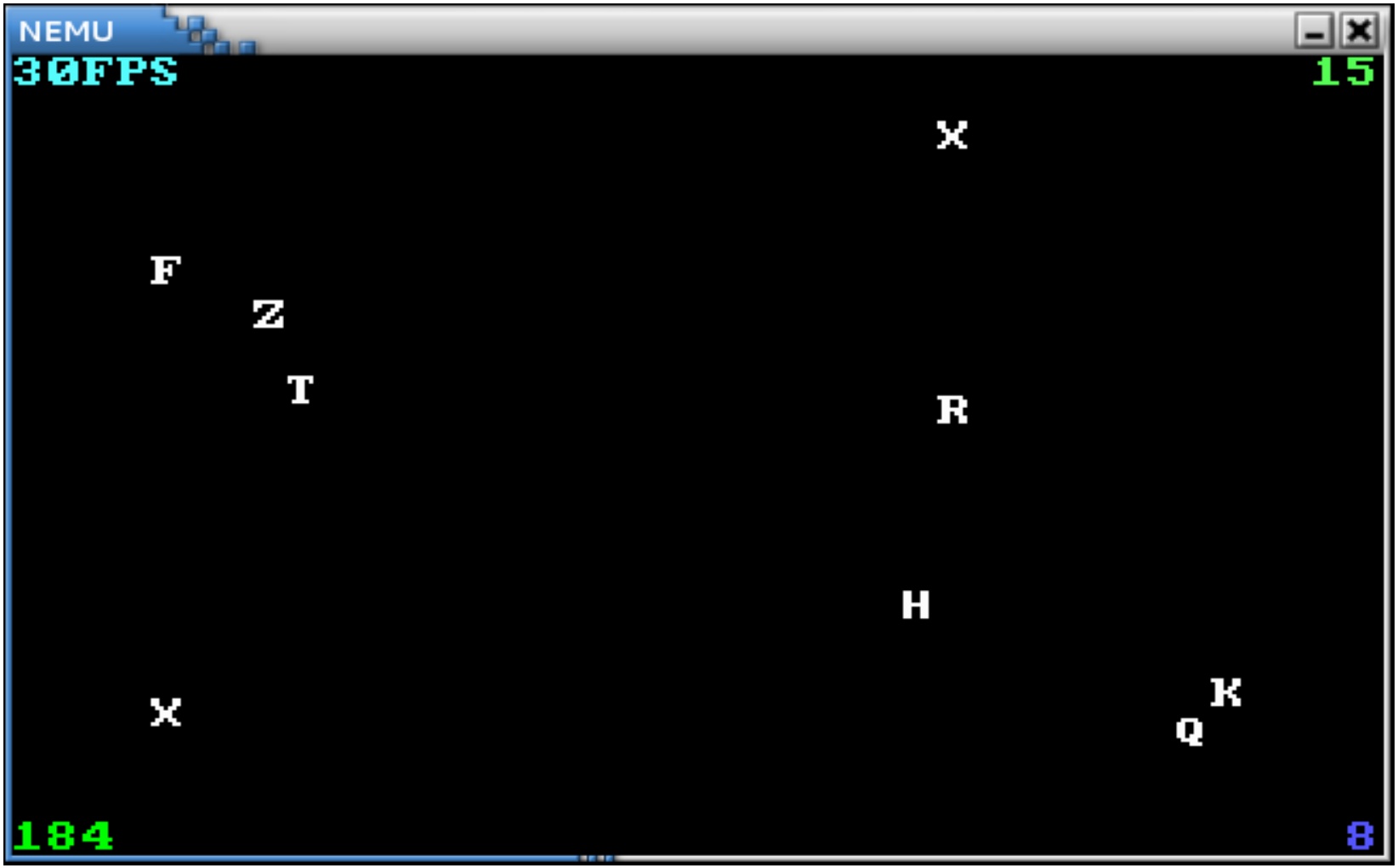
学什么做什么!



学生提交的实验结果

- 中断/异常/I/O（PA4）

移植打字小游戏



学生提交的实验结果

- 中断/异常/I/O（PA4)

移植仙剑奇侠传

体会、困惑和反思

- 比较有效的做法
  - 理论结合实例
  - 前后内容关联
  - 多用图解释概念
  - 多用汇编讲高级语言程序
  - 小班化教学，多互动
  - 先问问题，再给出答案并讨论
  - 多进行随堂小测验
  - 引导学生查资料、多动手
  - 不要学生死记硬背，强调理解
  - 开卷考试
- 困惑
  - 学生受应试教育毒害之深，远超想象
  - 学生的两级分化，越来越严重
- 反思：一门课解决不了所有问题，需要相关课程一起协作

用大量例子，不断让学生体会高级语言、编译器、操作系统、ISA以及微架构之间的关系。

“ICS（计算机系统课程）为目前来看最不水的专业课，没有之一”

“真的学到了非常多的东西，比如Linux系统/Vim/Git的基本使用方法、Makefile文件的编写、基本的汇编语言、计算机原理、基本的操作系统知识，以及如何RTFM和RTFSC等等等等。程序设计的本领也得到了充足的锻炼。”

PA实验（第一版）规模

	预计耗时/小时	代码量/行
PA0 – 开发环境配置	20	无
PA1 – 简易调试器	50	500+
PA2 – 指令系统	60	4000+
PA3 – 存储管理	50	500+
PA4 – 中断与I/O	30	300+

加基础框架
达7000多行

PA1实验的反馈

课程教学的反馈

第一届PA成绩

学生实际所用时间可能比预计耗时还要多得多。

困难来自：

1. 对Linux+GCC+gdb编程环境不熟悉
2. 大规模程序设计和调试等能力不足
3. 二（上）课业太重，没时间消化课程内容
4. ICS课程对PA实验的指导不够

PA实验（第二版）规模

	持续时间/周	预计耗时/小时	代码量/行
PA0 – 开发环境配置	2	20	无
PA1 – 简易调试器	3	30	400
PA2 – 指令系统	5	50	800
PA3 – 存储管理	4	50	500
PA4 – 中断与I/O	3	30	300
总计	17	180	2000

- 预计耗时以中下水平的学生来估算
  - 保守估计200小时

用时反馈

第二届PA成绩

学生反馈1

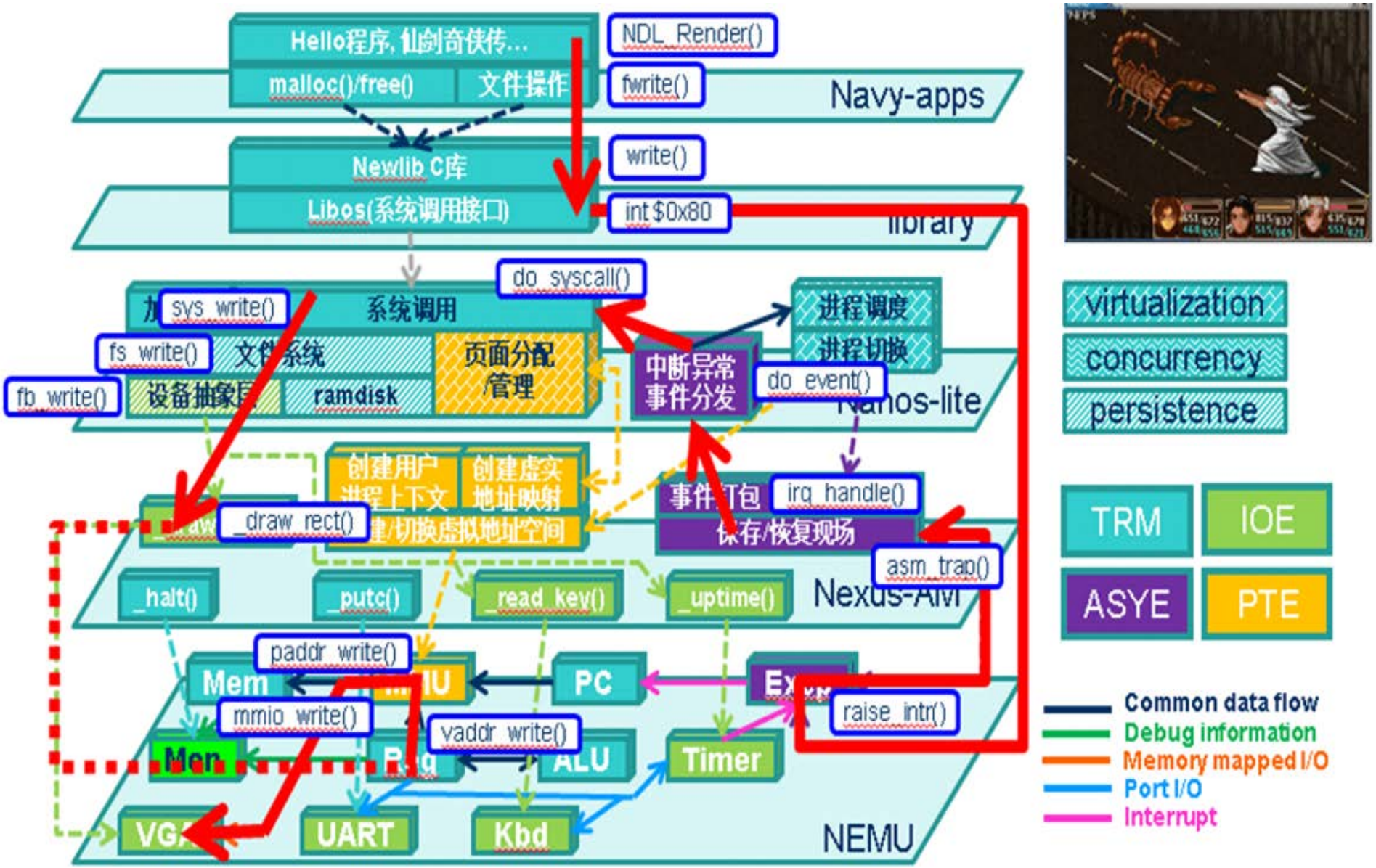
学生反馈2

在一（下）增加了“程序设计实验”课程（2013级学生没开设）

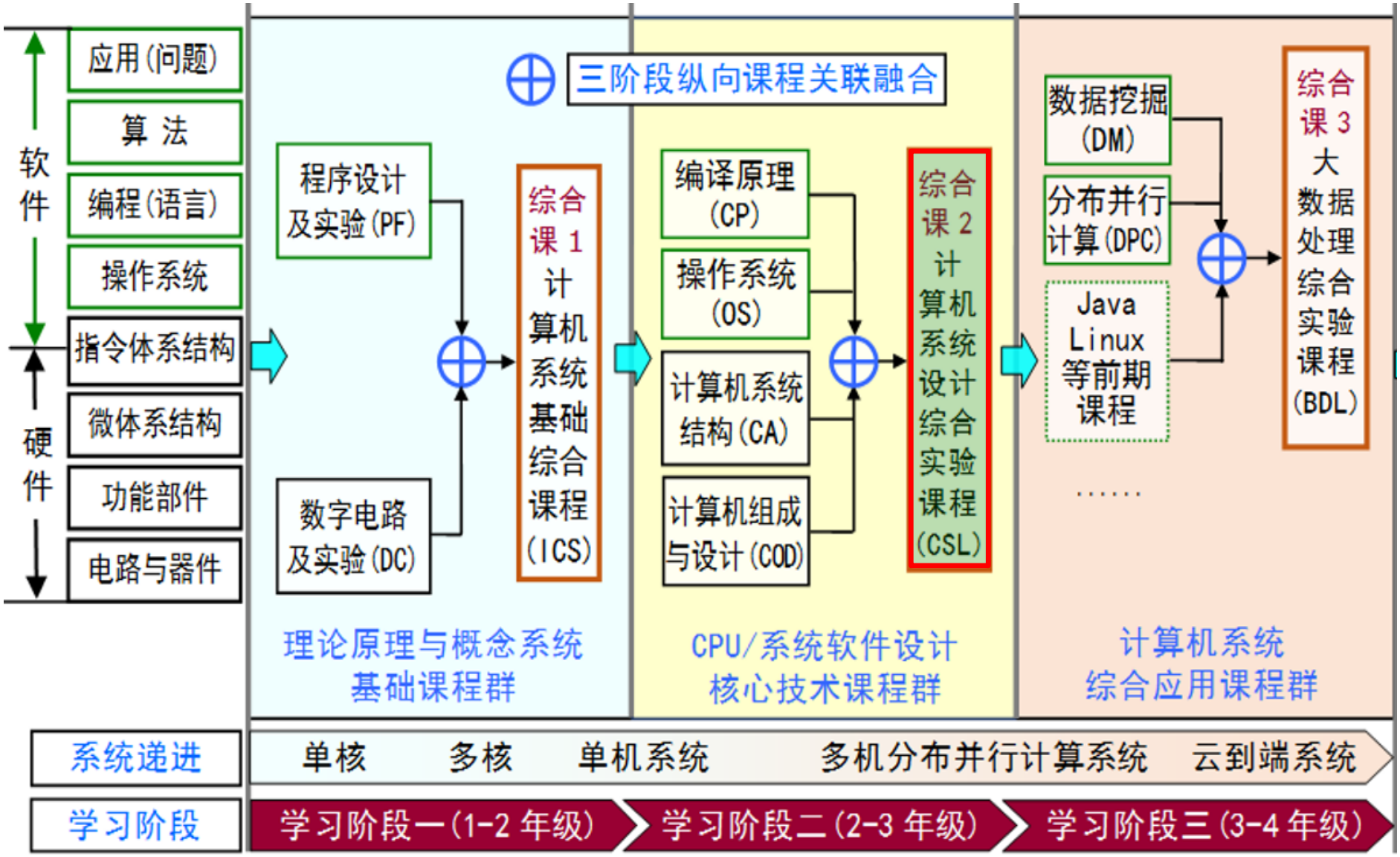
- 第一届（2013级）
  - 平均分: 23.07/40
  - 完成PA3的仅有29人
- 第二届（不包括PA4）
  - 平均分: 27.60/34.6
  - 完成PA3的有96人

PA实验（第三版）改进

仙剑奇侠传屏幕更新过程



三门新建系统综合课程之二：CSL

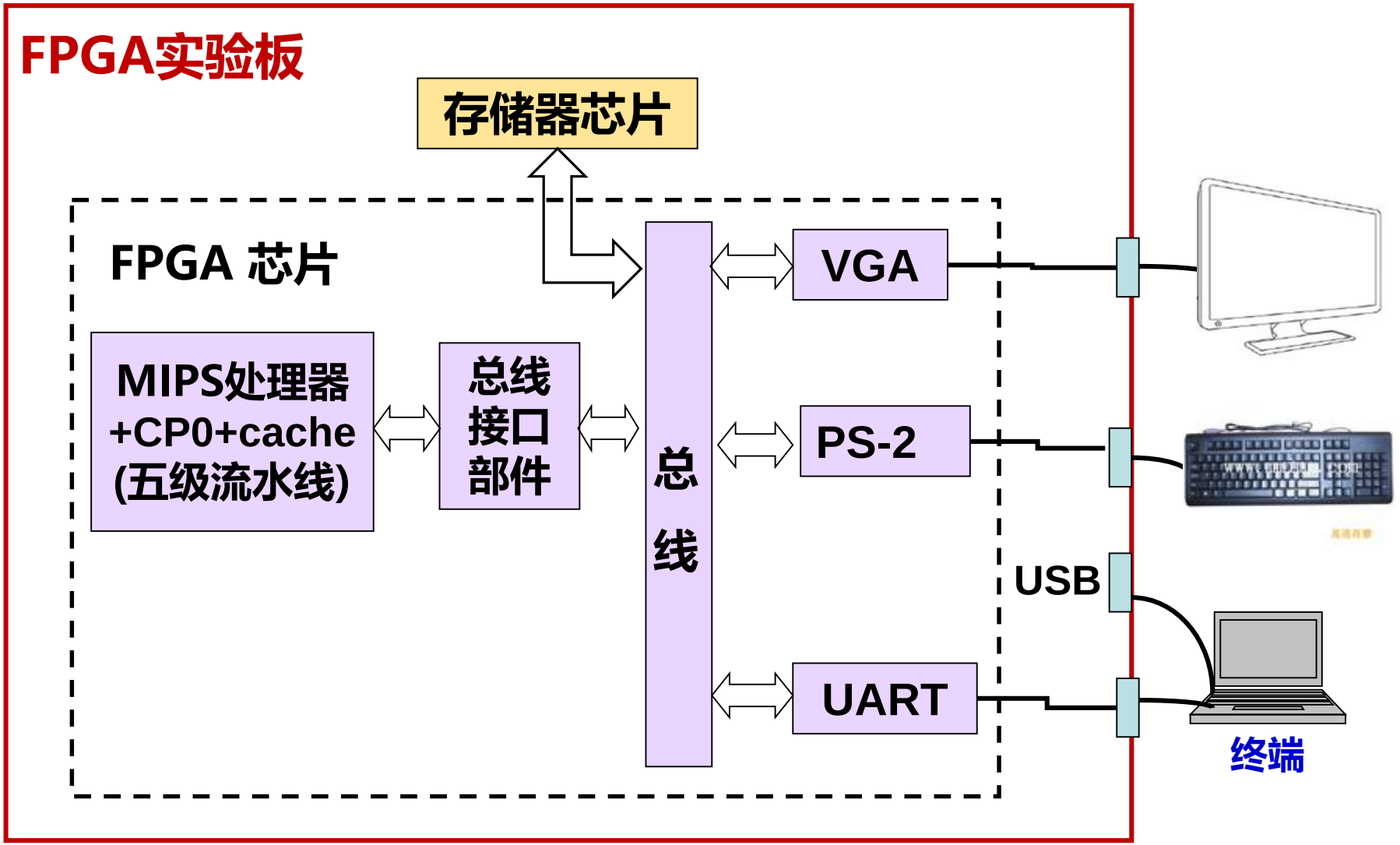


计算机组成与设计（COD）课程内容

- 指令系统设计
  - 指令集体系结构与微体系结构，指令系统设计需要考虑的各个方面
  - RISC与CISC，MIPS指令系统， SIMD指令（数据级并行技术）
- 指令执行过程与微结构
  - 运算、访存、I/O、流控制类指令执行过程、带异常或中断处理的指令执行过程
- 非流水线CPU设计
  - 功能部件设计（ALU、桶形移位器、乘法器、除法器、浮点部件、MMU等）
  - 单周期数据通路设计、控制器设计
  - 多周期数据通路设计、多周期控制器设计、微程序设计
- 流水线CPU设计
  - 流水线CPU设计、冒险处理（控制冒险、数据冒险、控制冒险）
  - 带中断和Cache缺失等处理的流水线设计
  - 超流水线、超标量、动态调度、乱序执行
- 系统互连与I/O组织（弱化传统总线概念，强调现代总线的点对点、串行、异步特点）
  - 外设、外设接口、系统互连、三种 I/O方式（查询、中断和DMA）
- 并行计算系统（细节在“计算机系统结构”和“并行处理技术”中介绍）
  - 多核处理器、众核处理器、多处理器系统、多计算机系统简介
  - 并行程序设计简介（共享存储变量、消息传递、Map-Reduce、CUDA）

计算机组成与设计（COD）实验

NPC（NJU Processing Core）：基于MIPS架构



计算机组成与设计（COD）实验

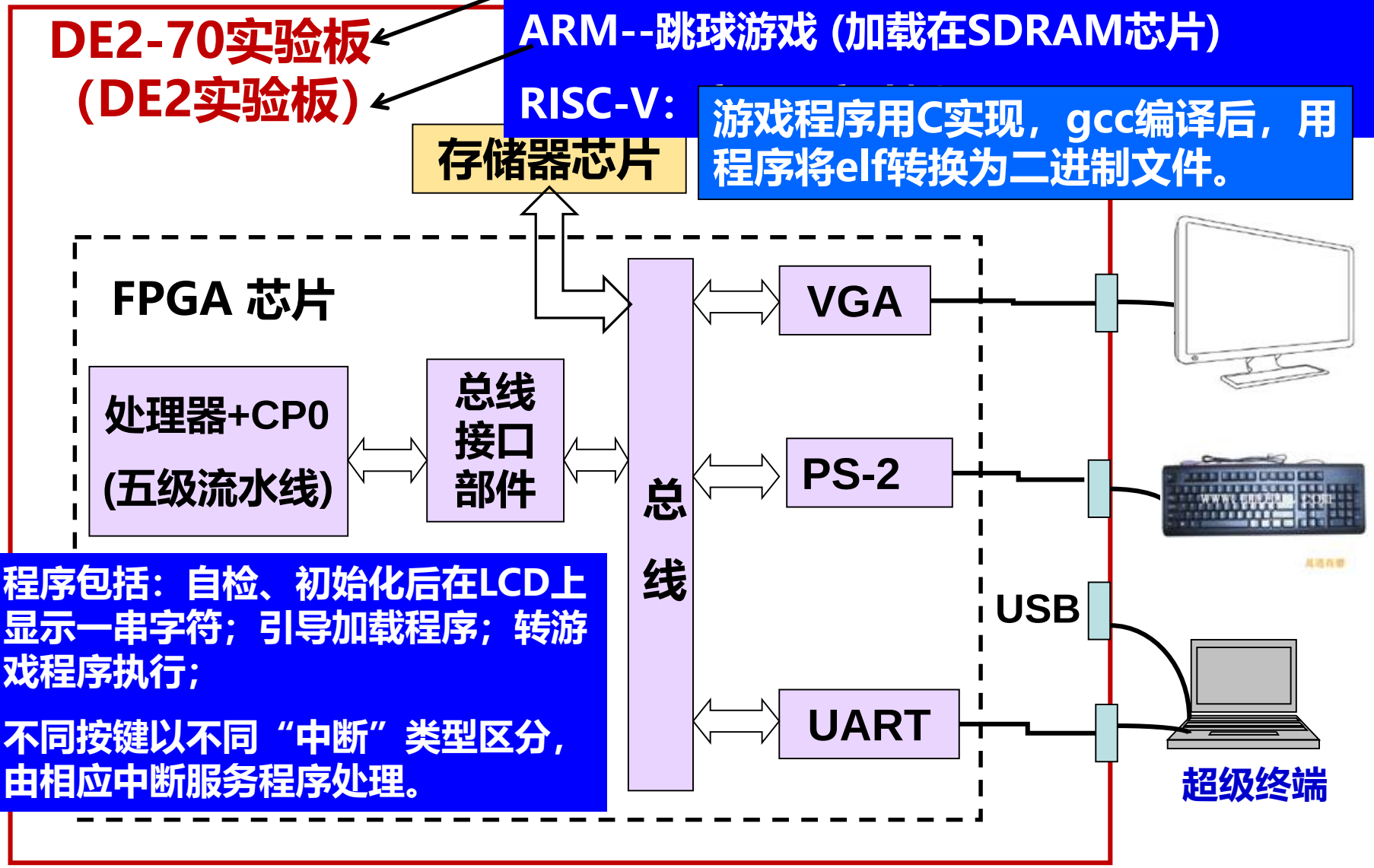
MIPS--俄罗斯方块游戏 (加载在SSRAM芯片)

ARM--跳球游戏 (加载在SDRAM芯片)

RISC-V：

游戏程序用C实现，gcc编译后，用程序将elf转换为二进制文件。

DE2-70实验板
（DE2实验板）



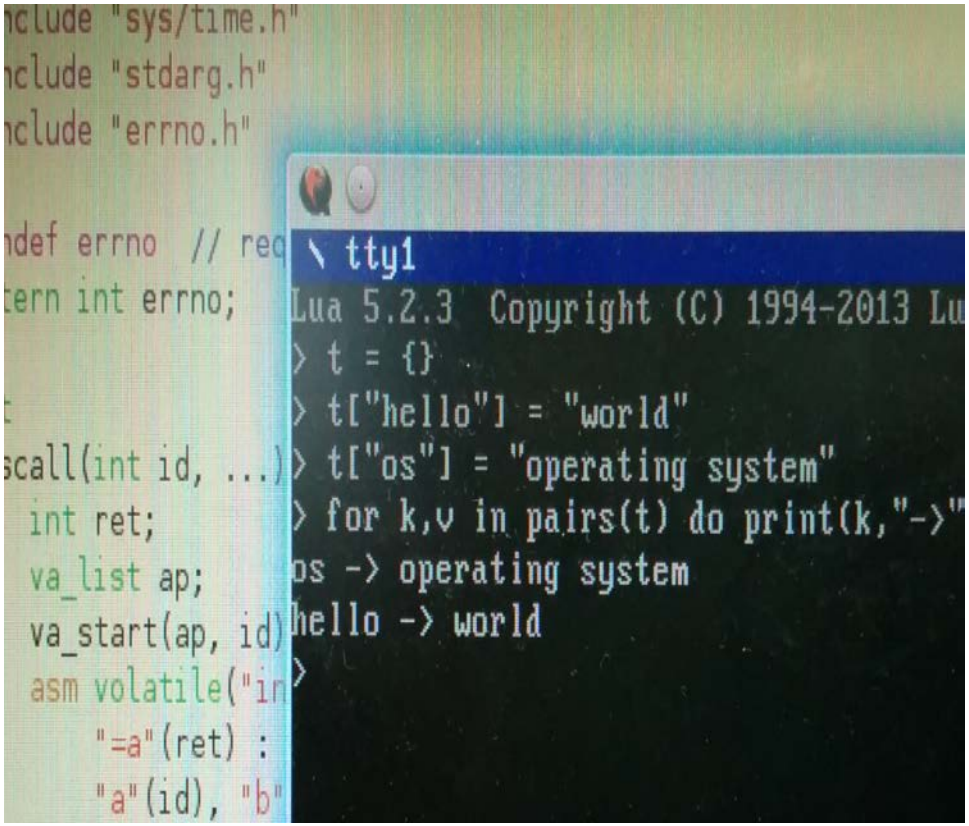
程序包括：自检、初始化后在LCD上显示一串字符；引导加载程序；转游戏程序执行；

不同按键以不同“中断”类型区分，由相应中断服务程序处理。

操作系统（OS）实验

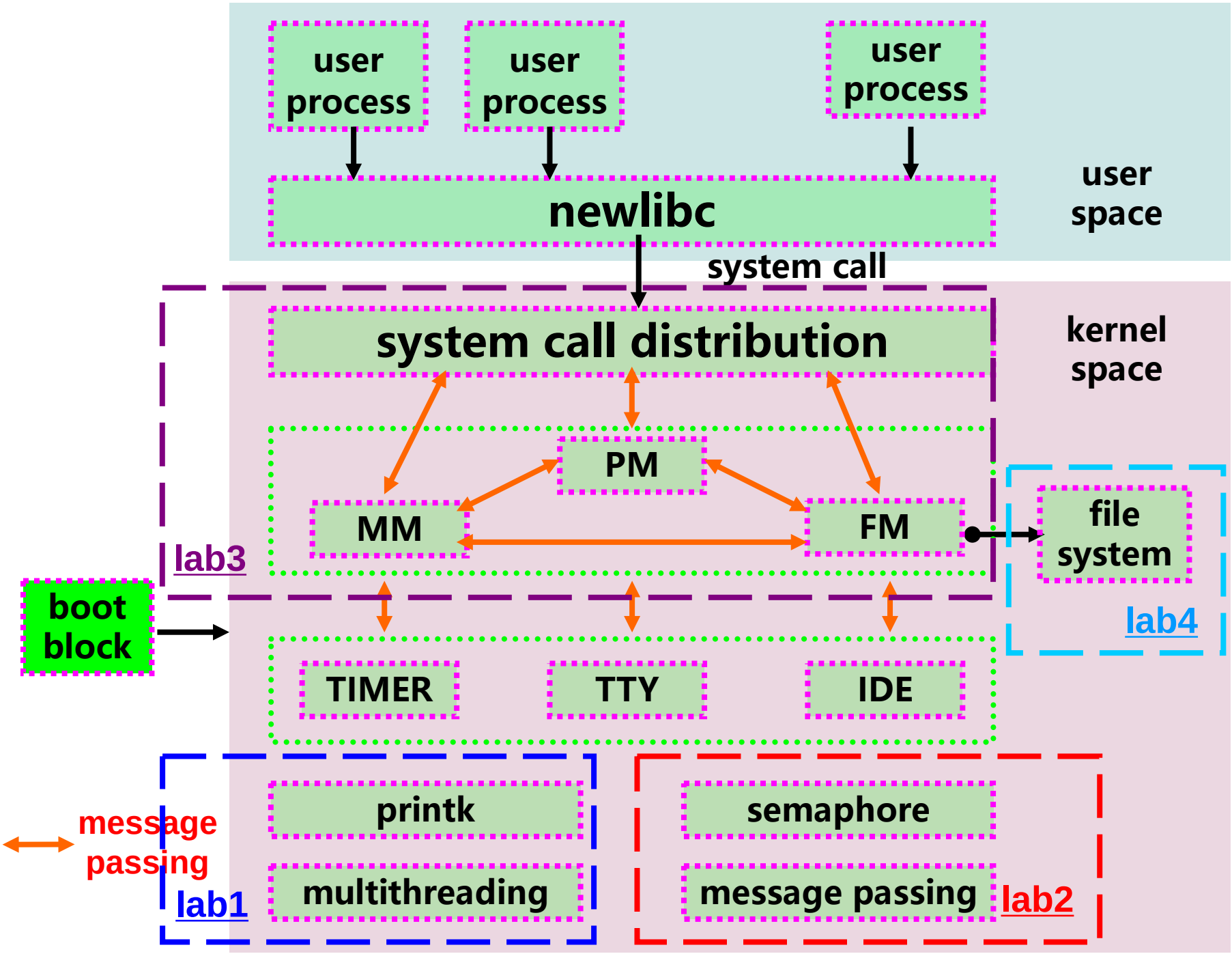
Nanos = Nan(jing U) OS

- 微型教学用操作系统
  - 概念基于Remzi《Operating Systems: Three Easy Pieces》
    - 虚拟化、并发、持久化
- 若干个版本的参考实现
  - 单内核/消息内核/事件内核
- 足够的完整性
  - 18个POSIX.1系统调用
  - Newlibc-Nanos
  - 兼容ANSI C程序



操作系统（OS）实验

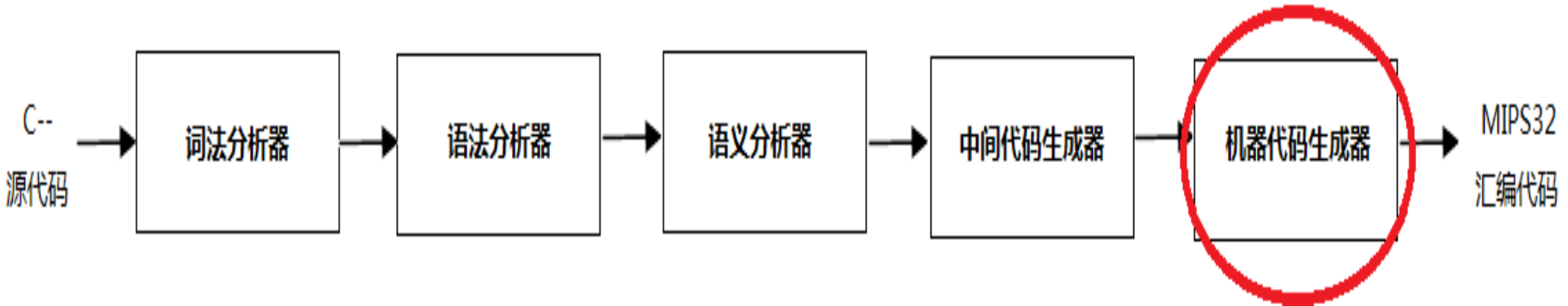
- 实验平台
  - GNU/Linux
  - GCC, gdb, qemu, git
- 实验方案(框架代码量/学生作业量)
  - Lab0：游戏与系统启动 (300/500)
  - Lab1：内核线程调度 (400/300)
  - Lab2：内核同步与通信 (700/500)
  - Lab3：用户进程与存储管理 (-/1000)
  - Lab4：文件系统 (-/2000)
  - 未设置设备管理实验


user process
user process
user process
user space
newlibc
system call
kernel space
system call distribution
PM
MM
FM
file system
lab3
lab4
boot block
TIMER
TTY
IDE
printk
multithreading
lab1
semaphore
message passing
lab2
message passing

编译原理（CP）实验

NCC = NJU C Compiler

编译原理课程的实习内容是为一个小型的类 C 语言(C--)实现一个编译器。如果你顺利完成了本课程规定的实习任务，那么不仅你的编程能力将会得到大幅提高，而且你最终会得到一个比较完整的、能将 C--源代码转换成 MIPS 汇编代码的编译器，所得到的汇编代码可以在 SPIM Simulator 上运行。实习总共分为四个阶段：词法和语法分析、语义分析、中间代码生成和目标代码生成。每个阶段的输出是下一个阶段的输入，后一个阶段总是在前一个阶段的基础上完成，如下图所示：



计算机系统设计综合实验（CSL）概要

- 思路：基于前期实验从逻辑门开始构造一个完整的计算机系统，使学生清楚理解完整的计算机系统软/硬件协同设计过程

- 前期各实验(Project N)
  - NCC (NJU C Compiler)
    - 编译原理 (CP)实验
  - Nanos (Nanjing U OS)
    - 操作系统 (OS)实验
  - NPC (NJU Processing Core)
    - 组成与设计 (COD)实验
  - NEMU (NJU EMUlator)
    - 系统基础(ICS) PA实验

定位：系统方向“收官课”，完成实际计算机系统的设计和实现，弄清细节、踩够“坑”、成就梦想

Project N 综合设计实验的挑战

- Project N实验根据课程内容选择易于实施教学的平台，在相关课程中分别设计
  - NEMU、NCC基于功能相对完整的Linux API，依赖大量运行库
  - NCC生成的是MIPS目标代码
  - Nanos硬编码了大量x86相关代码
  - NPC只能提供MIPS32（或RISC-V）的部分功能

- 拼接成一个完整的计算机系统是个“不可能完成的任务”

原因之一：
OS实验没有对ISA做足够的抽象，从x86到POSIX API有巨大的跨度

Project N 综合设计实验框架

- 抽象计算机（AM）：引入ISA无关的硬件抽象层（C API）
- 显式化“系统软件对硬件功能的需求”
  - 基础版本：图灵机 (TRM)
    - 内存 + 状态机(寄存器)
  - IO扩展 (IOE)
    - 常用设备(键盘、屏幕、tty)的异步非阻塞I/O、总线接口
  - 异步扩展 (ASYE)
    - 中断、异常、寄存器现场管理
  - 保护扩展 (PTE)
    - 地址空间映射和保护
  - 多核扩展 (MPE)
    - 多处理器的启动、识别、同步

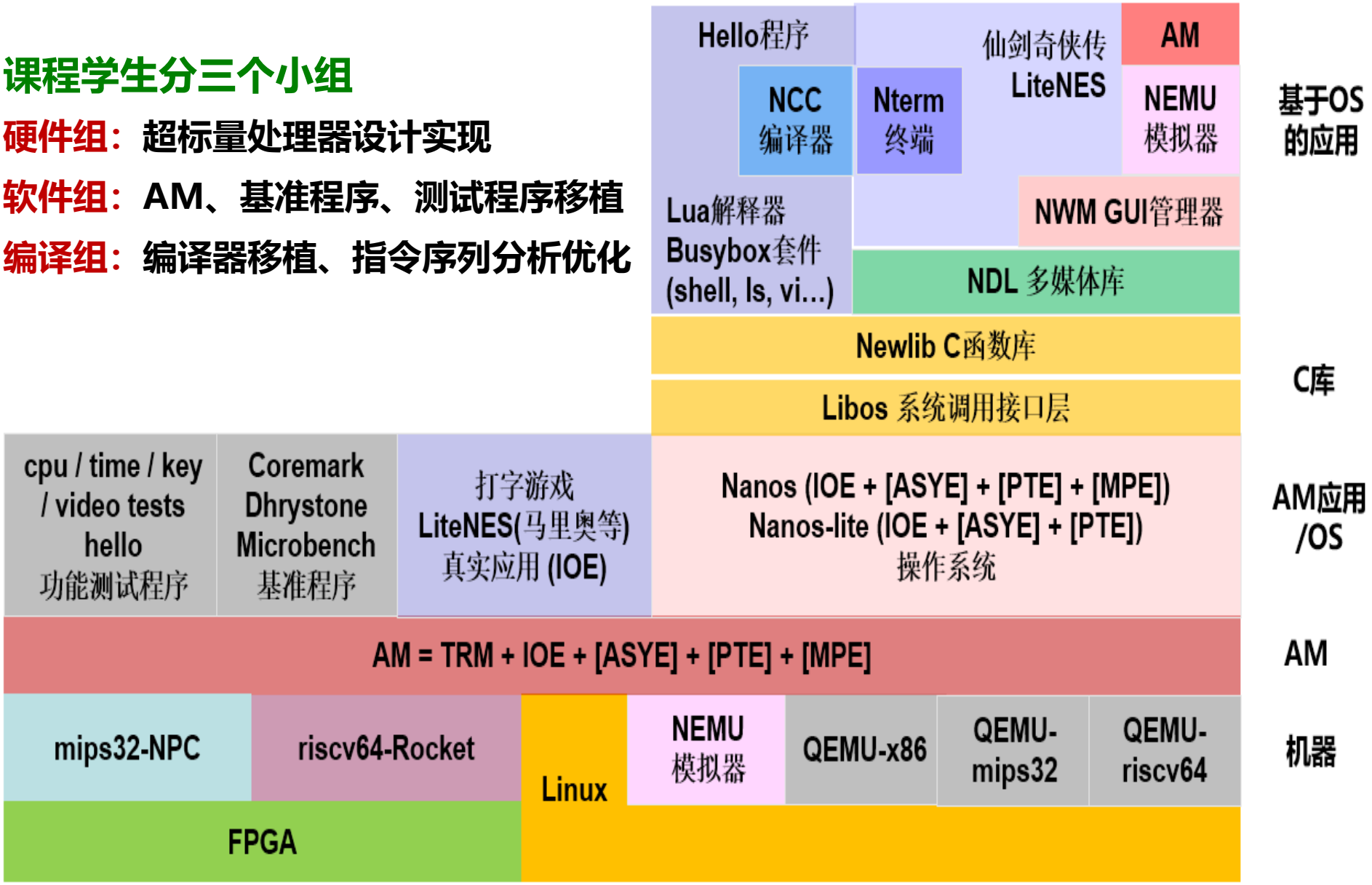
Project N 综合实验体系

课程学生分三个小组

硬件组：超标量处理器设计实现

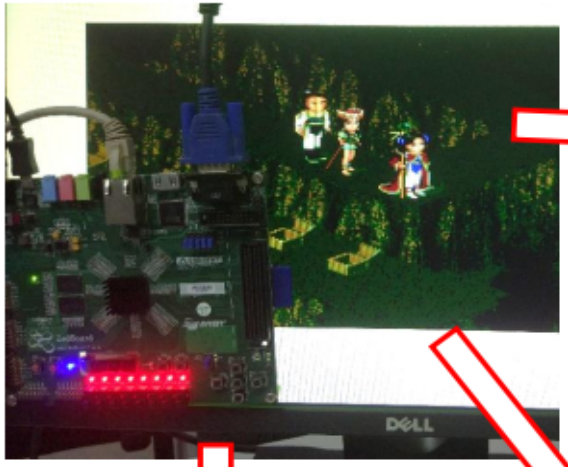
软件组：AM、基准程序、测试程序移植

编译组：编译器移植、指令序列分析优化



2017年首届“龙芯杯”赛完成的系统

我们构建了完整的Project-N生态系统



AM

hello | 仙剑奇侠传 | NES模拟 | NEMU模拟器

编译器 | 窗口管理器

应用

运行库：libos, libc, libndl (图形)

抽象层

功能测试 | 基准程序 | 真实应用 | Nanos 操作系统

操作系统

AM (抽象计算机) = TRM + IOE + [ASYE] + [PTE]

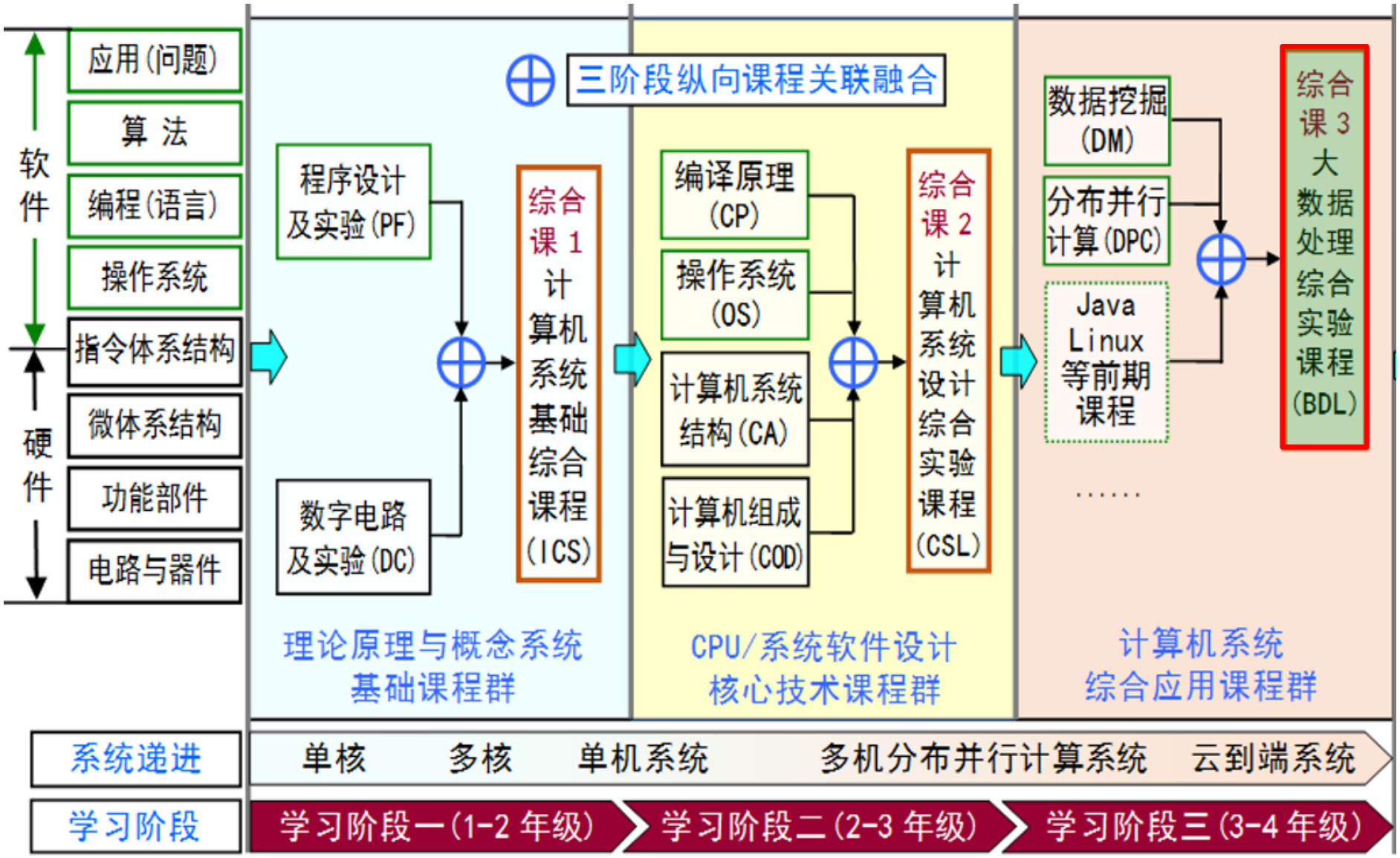
抽象层

mips32 (NOOP-SoC) | NEMU模拟器 | X86 (qemu) | Mips32 (qemu)

ISA

FPGA | Linux

三门新建系统综合课程之三：BDL



大数据处理综合实验课程（BDL）概要

- 思路：将系统能力螺旋式上升到实际综合应用系统，培养学生大数据系统开发和综合应用能力
- 主要内容
  - 并行编程模型和计算框架
    - Hadoop MR、SPARK等
  - 大数据存储管理和索引查询
    - HDFS、HBase、Hive、Alluxio等
  - 业界主流大数据应用
    - 排序、文档倒排索引、文档共现算法、专利文献数据分析应用等
  - 机器学习和数据挖掘基础算法的并行化
    - 聚类、分类、频繁项集挖掘等算法的并行化实现
  - 简单回顾：计算机系统结构、并行架构与平台

定位：系统能力综合应用“收官课”

大数据处理综合实验课程（BDL）

Lab1：Hadoop系统安装与WordCount词频统计编程实验

Lab2：搜索引擎文档倒排索引编程实验

Lab3：Hadoop HBase和Hive编程实验

Lab4：Wikipedia网页PageRank实验

Project（暑假期间）：

自选或由老师指定具有一定难度和工作量的题目，完成一个综合性大数据课程设计。例如：社会网络分析、电影数据聚类、新浪微博推荐系统、网络通信中最短路径、数据库查询、基于内容的图像搜索等。

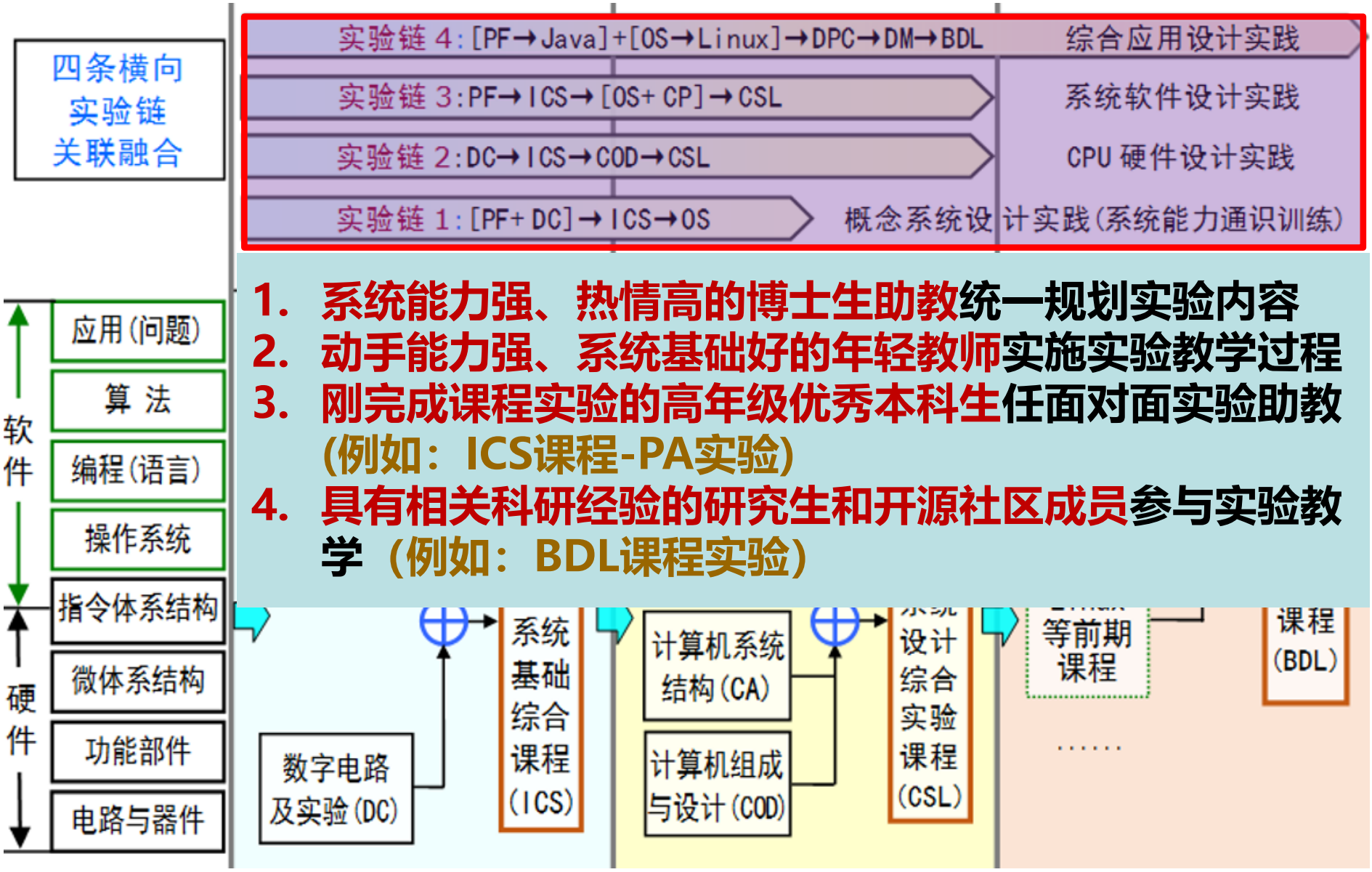
扩充实验：基于Alluxio开源社区的开源项目开发实践：从ICS课程的PA实验使用Git进行个人项目提交，到真正加入大型开源社区进行代码提交、检查、评价等全过程实践

（邀请开源社区成员进行指导）

主要内容

- 为何需要进行系统能力培养
- 目前存在的问题
- 南京大学的做法及成效
  - 计算机系统能力培养总体思路
  - 三门新建纵向系统综合实验课
  - 四条横向关联融合的实验链

四条横向关联融合实验链



1. **系统能力强、热情高的博士生助教统一规划实验内容**
2. **动手能力强、系统基础好的年轻教师实施实验教学过程**
3. **刚完成课程实验的高年级优秀本科生任面对面实验助教(例如：ICS课程-PA实验)**
4. **具有相关科研经验的研究生和开源社区成员参与实验教学（例如：BDL课程实验）**

总　　结

- 南京大学相关课程的教改思路
  - 以“提高对计算机系统全面认识和系统设计能力”为目标
  - 强调软/硬件的关联、课程间的关联、理论与实验的关联
  - 在统一的指导思想和培养目标下，全方位系统地构建相关课程的课堂教学和实验教学方案
  - 传统课程：强化实验
  - 新增课程：系统基础、系统综合设计、大数据系统及其应用

Q/A

敬请批评指正！

谢　谢！