

CS13104: 计算机系统 课程概述

郑贵滨

计算机学院，语音处理研究室

要点

- 课程主题
- 五个事实/现实
- 可执行程序的生成与执行
- 计算机系统层次模型
- 本课程在CS/CE课程体系中的地位

课程目标: 抽象很好, 但别忘记现实!

- **多数计算机科学与计算机工程的课程强调抽象**
 - 抽象数据 (类) 型
 - 渐进分析Asymptotic analysis
- **抽象是有限制的**
 - 特别是在bug (程序缺陷-故障/错误) 面前
 - 需要理解底层实现的细节
- **学完本课程的有用的收获**
 - 成为更加有效地程序员
 - 能够发现并有效地排除bug
 - 能理解并调整程序性能
 - 为CS/SE的后续系统课程打基础
 - 编译、操作系统、计算机网络、计算机体系结构、嵌入式系统、存储系统等。

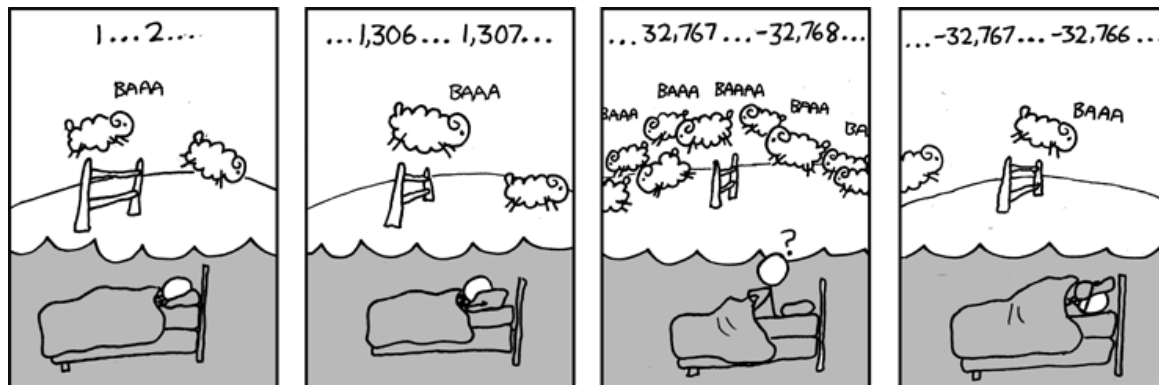
现实1: int不是整数, float不是实数

■ 例 1: $x^2 \geq 0$?

■ Float's: Yes!

■ Int's:

- $40000 * 40000 \rightarrow 1600000000$
- $50000 * 50000 \rightarrow ??$



■ 例 2: $(x + y) + z = x + (y + z)$?

■ 无符号/有符号 Int: Yes!

■ 浮点数Float:

- $(1e20 + -1e20) + 3.14 \rightarrow 3.14$
- $1e20 + (-1e20 + 3.14) \rightarrow ??$

计算机的算法/算术运算

■ 不生成随机值

- 算术运算有重要的数学特性

■ 不要假设所有的“通常”数学特性都成立

- 因为数据表示的有限性
- 整数操作满足“环”(ring)特性
 - 交换律, 结合律, 分配律
- 浮点操作满足“排序”(ordering)特性
 - 单调性, 符号值

■ 观察

- 要理解哪一种抽象应用在哪些上下文中
- 对于编译器开发人员和认真的应用程序员, 这些都是重要事项

现实2: 你不得不懂汇编

- 有可能是, 你永远不用汇编语言写程序
 - 编译器比你更好更耐心
- 但是: 要理解汇编是机器级执行模型的关键
 - 程序有Bug时的行为
 - 高级语言模型会失败
 - 调优程序性能
 - 理解由/不由编译器实现的优化
 - 理解程序低效的根源
 - 实现系统软件
 - 编译器把机器代码作为目标
 - 操作系统要管理进程状态
 - 创造/对抗恶意软件 (malware)
 - x86 汇编是很好的语言选择

现实3: 存储事宜

RAM随机存储器是一个非物理抽象

■ 存储器不是无限的

- 存储器需要分配与管理
- 很多应用是存储支配/控制的

■ 存储引用错特别要命(有害)

- 在时间和空间方面影响深远

■ 存储器性能并非一致

- Cache与虚拟存储器的效应能大大影响程序性能
- 针对存储系统的特点, 调整程序, 能大幅提升速度

例：存储引用Bug

```
typedef struct {  
    int a[2];  
    double d;  
} struct_t;  
  
double fun(int i) {  
    volatile struct_t s;  
    s.d = 3.14;  
    s.a[i] = 1073741824; /* Possibly out of bounds */  
    return s.d;  
}
```

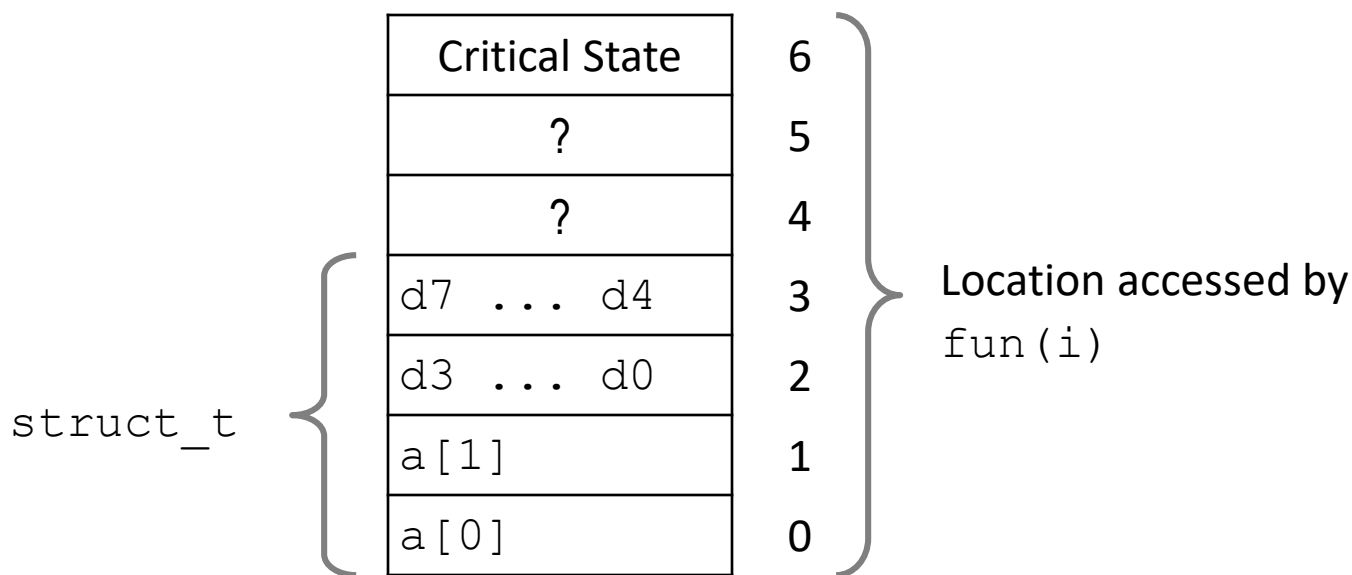
fun(0)	->	3.14
fun(1)	->	3.14
fun(2)	->	3.1399998664856
fun(3)	->	2.00000061035156
fun(4)	->	3.14
fun(6)	->	Segmentation fault

例：存储引用Bug

```
typedef struct {
    int a[2];
    double d;
} struct_t;
```

fun(0)	->	3.14
fun(1)	->	3.14
fun(2)	->	3.1399998664856
fun(3)	->	2.00000061035156
fun(4)	->	3.14
fun(6)	->	Segmentation fault

注释:



存储引用错误

- C and C++ 不提供任何存储保护
- 数组访问的越界
 - 无效指针值
 - 滥用 malloc/free
- 导致险恶/恶意的bug
 - Bug是否产生效果，依赖于系统或编译器
 - 远距离的行为(Action at a distance)
 - 崩溃的对象逻辑上与你正访问的不相干
 - 可能在bug生成很久才被第一次观察到Bug的影响

存储引用错误

■ 我该如何应对?

- 用 Java, Ruby, Python, ML, ...编程
- 理解将出现的可能交互(interactions)
- 使用或自己开发一个工具来发现引用错 (e.g. Valgrind)

现实4: 性能比渐进复杂性更重要

- 常数因子也有关系!
- 即使是精确的操作数量, 也无法预测性能
 - 很容易看到, 代码编写不同, 会引起10:1 性能变化
 - 要多层次优化: 算法、数据表示、过程、循环
- 优化性能一定要理解系统
 - 程序是怎么编译和执行的
 - 怎样测量系统性能和定位瓶颈
 - 如何在不破坏代码模块性和通用性的前提下提高性能

例：内存系统性能

- 存储器的层次化组织
- 性能 依赖于访问模式
 - 包括怎样遍历多维数组

```
void copyij(int src[2048][2048],  
            int dst[2048][2048])  
{  
    int i,j;  
    for (i = 0; i < 2048; i++)  
        for (j = 0; j < 2048; j++)  
            dst[i][j] = src[i][j];  
}
```

4.3ms

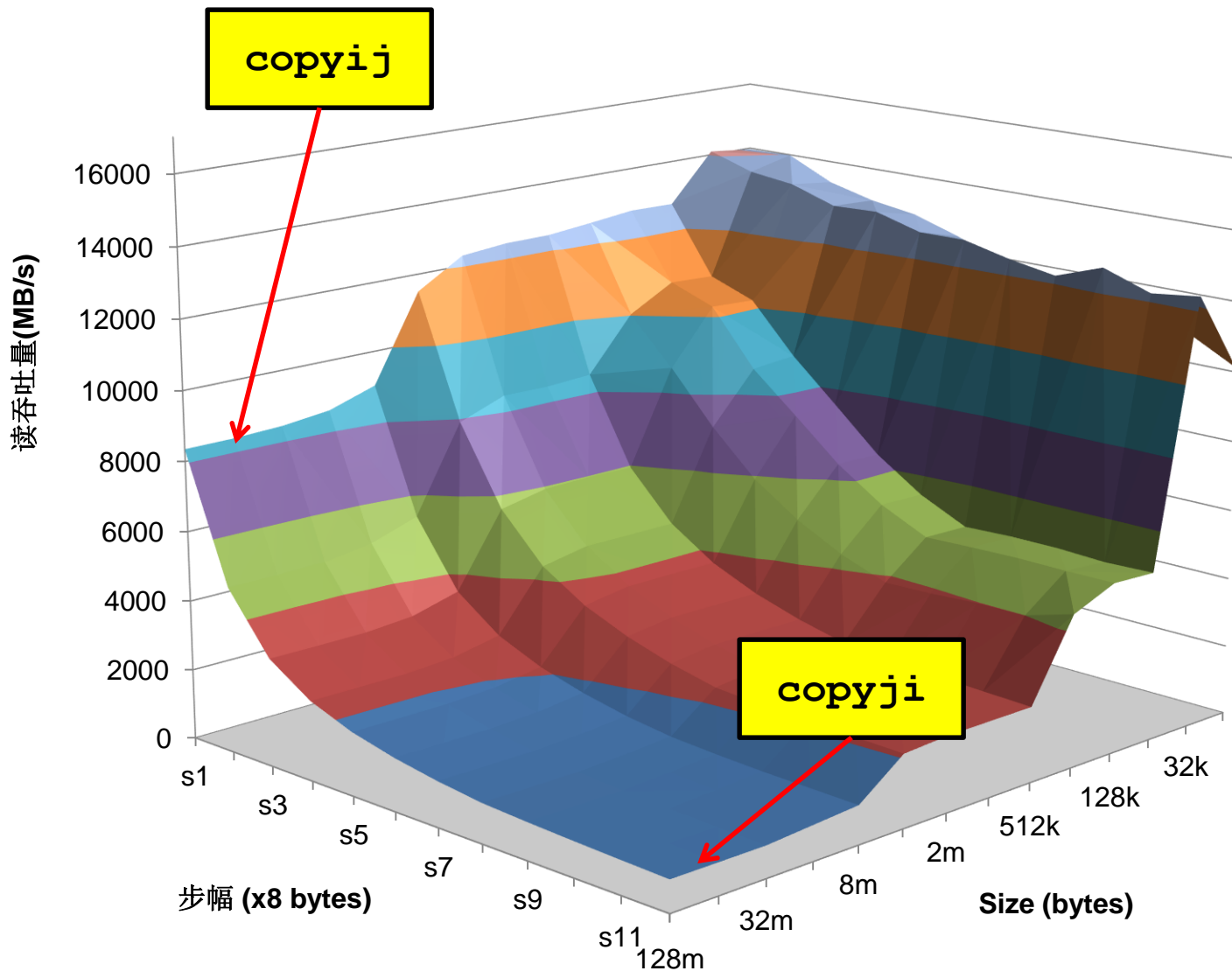
2.0 GHz Intel Core i7 Haswell

```
void copyji(int src[2048][2048],  
            int dst[2048][2048])  
{  
    int i,j;  
    for (j = 0; j < 2048; j++)  
        for (i = 0; i < 2048; i++)  
            dst[i][j] = src[i][j];  
}
```

81.8ms

- 存储器的层次化组织
- 性能依赖于访问模式
 - 包括怎样遍历多维数组

为什么性能不同



现实5:计算机做的事情比执行程序多得多

■ 要进行数据的输入输出

- I/O系统对程序可靠性与性能很关键

■ 要通过网络互相通讯

- 网络环境下出现了很多系统级问题
 - 自主进程的并发操作
 - 拷贝不可靠的媒体
 - 跨平台的兼容性
 - 复杂的性能问题

课程愿景

- 多数系统课程以"建设"为中心
 - 计算机体系结构
 - 用Verilog设计流水线处理器
 - OS
 - 实现OS的示例部分
 - 编译器
 - 编写简单语言的编译器
 - 网络
 - 实现并模拟网络协议

课程愿景

- **我们的课程以程序员为核心—程序员的视野**
 - 目标：通过更多地理解底层系统，成为更高效的程序员
 - 使你能
 - 编写更加可靠、有效的程序
 - 将需要钩子的特性合并到操作系统中
 - 如, 并发, 信号句柄
 - 这门课包括你们不会在其他地方看到的内容
 - 不是仅仅针对专门黑客的课程
 - 我们把隐藏的黑客带到每个人的面前!

二、可执行程序是怎么生成的？

经典的 “hello.c” C-源程序

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

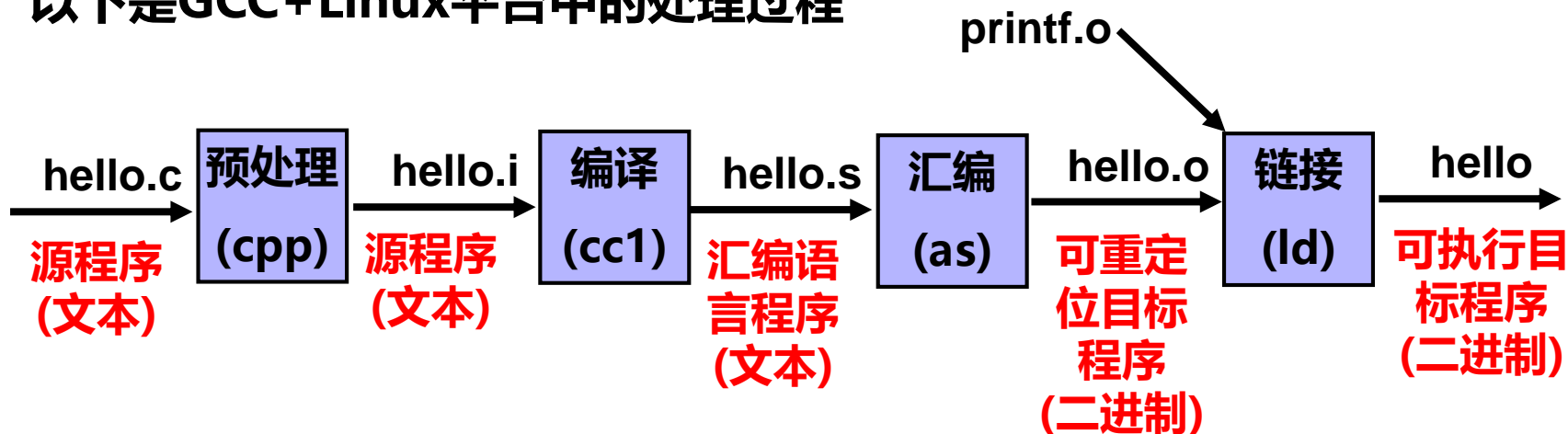
hello.c的ASCII文本表示

```
# i n c l u d e < s p > < s t d i o .
35 105 110 99 108 117 100 101 32 60 115 116 100 105 111 46
h > \n \n i n t < s p > m a i n ( ) \n {
104 62 10 10 105 110 116 32 109 97 105 110 40 41 10 123
\n < s p > < s p > < s p > < s p > p r i n t f ( " h e l
10 32 32 32 32 112 114 105 110 116 102 40 34 104 101 108
l o , < s p > w o r l d \ n " ) ; \n }
108 111 44 32 119 111 114 108 100 92 110 34 41 59 10 125
```

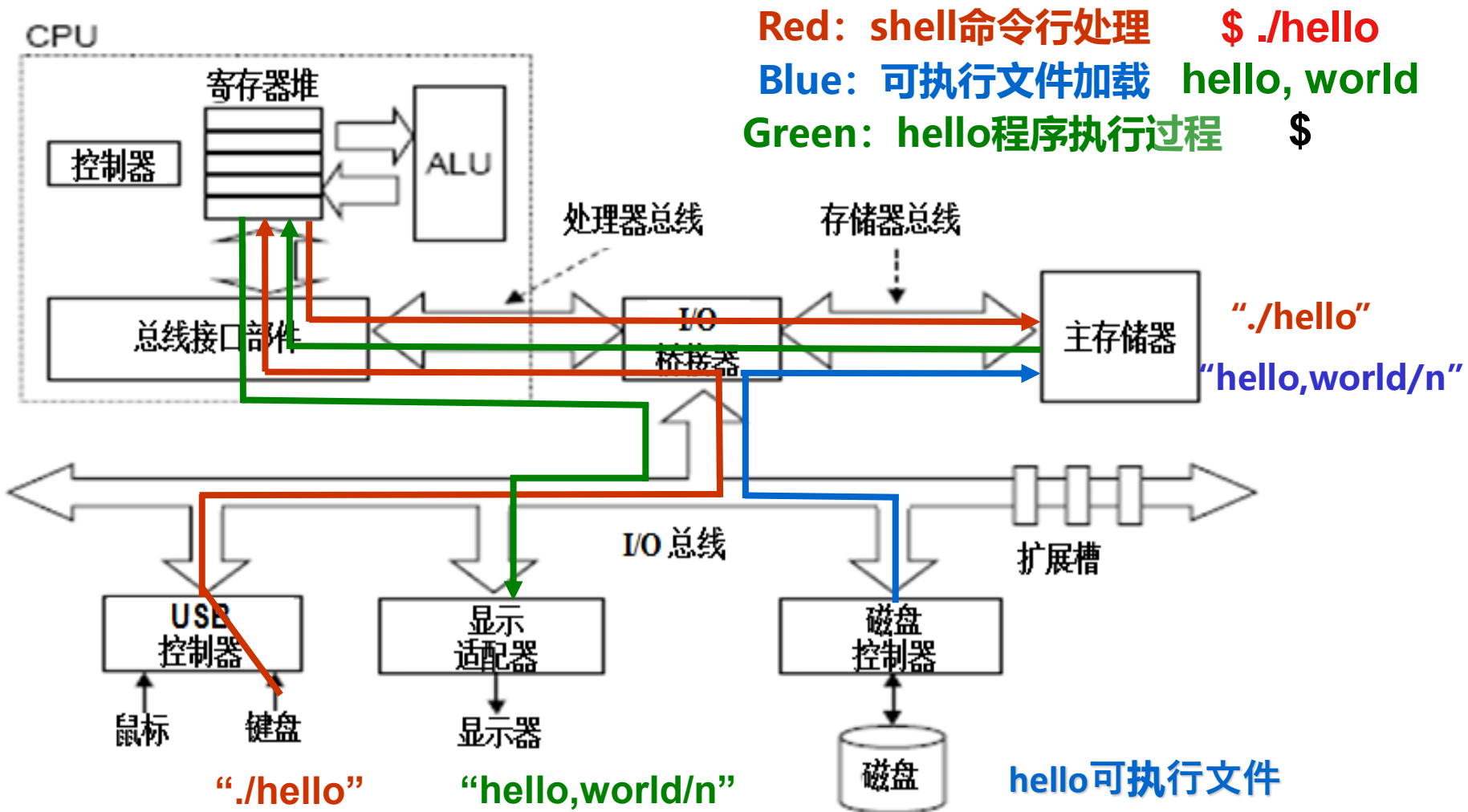
功能：输出 “hello,world”

计算机不能直接执行hello.c!

以下是GCC+Linux平台中的处理过程



三、可执行程序是怎么执行的？



数据经常在各存储部件间传送。故现代计算机大多采用“缓存”技术！
 所有过程都是在CPU执行指令所产生的控制信号的作用下进行的。

四、怎么优化程序？

1. 更快（本课程重点！）
2. 更省（存储空间、运行空间）
3. 更美（UI 交互）
4. 更正确（本课程重点！ 各种条件下）
5. 更可靠
6. 可移植
7. 更强大（功能）
8. 更方便（使用）
9. 更规范（格式符合编程规范、接口规范）
10. 更易懂（能读明白、有注释、模块化）

五、计算机系统层次模型

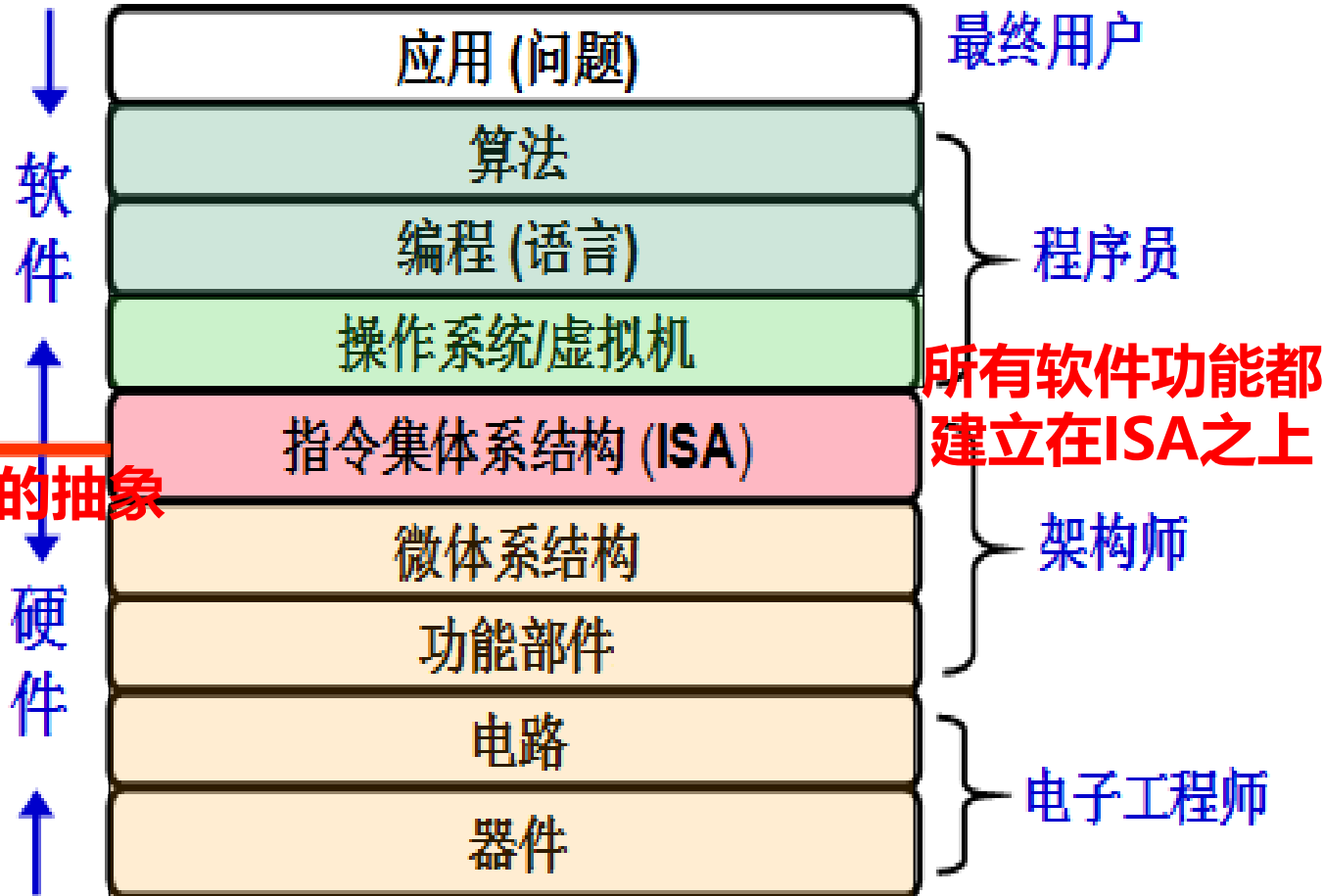
程序执行结果
不仅取决于
算法、程序编写
而且取决于
语言处理系统
操作系统
ISA-机器语言
微体系结构

ISA是对硬件的抽象

不同计算机课程
处于不同层次

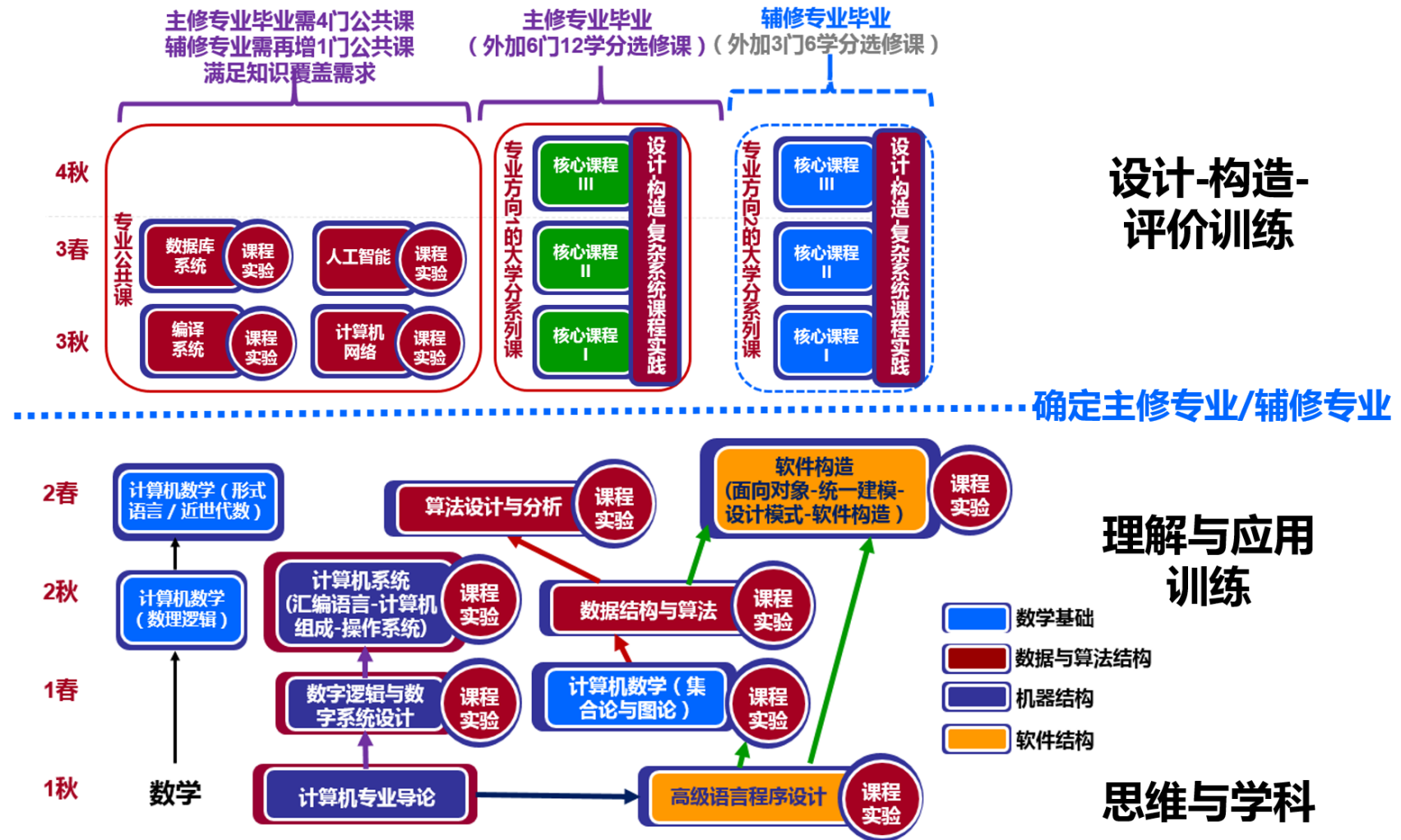
必须将各层次关
联起来解决问题

功能转换：上层是下层的抽象，下层是上层的实现
底层为上层提供支撑环境！



最高层抽象就是点点鼠标、拖拖图标、敲敲键盘，但这背后有多少层转化啊！

本课程在CS/CE 课程体系中的角色



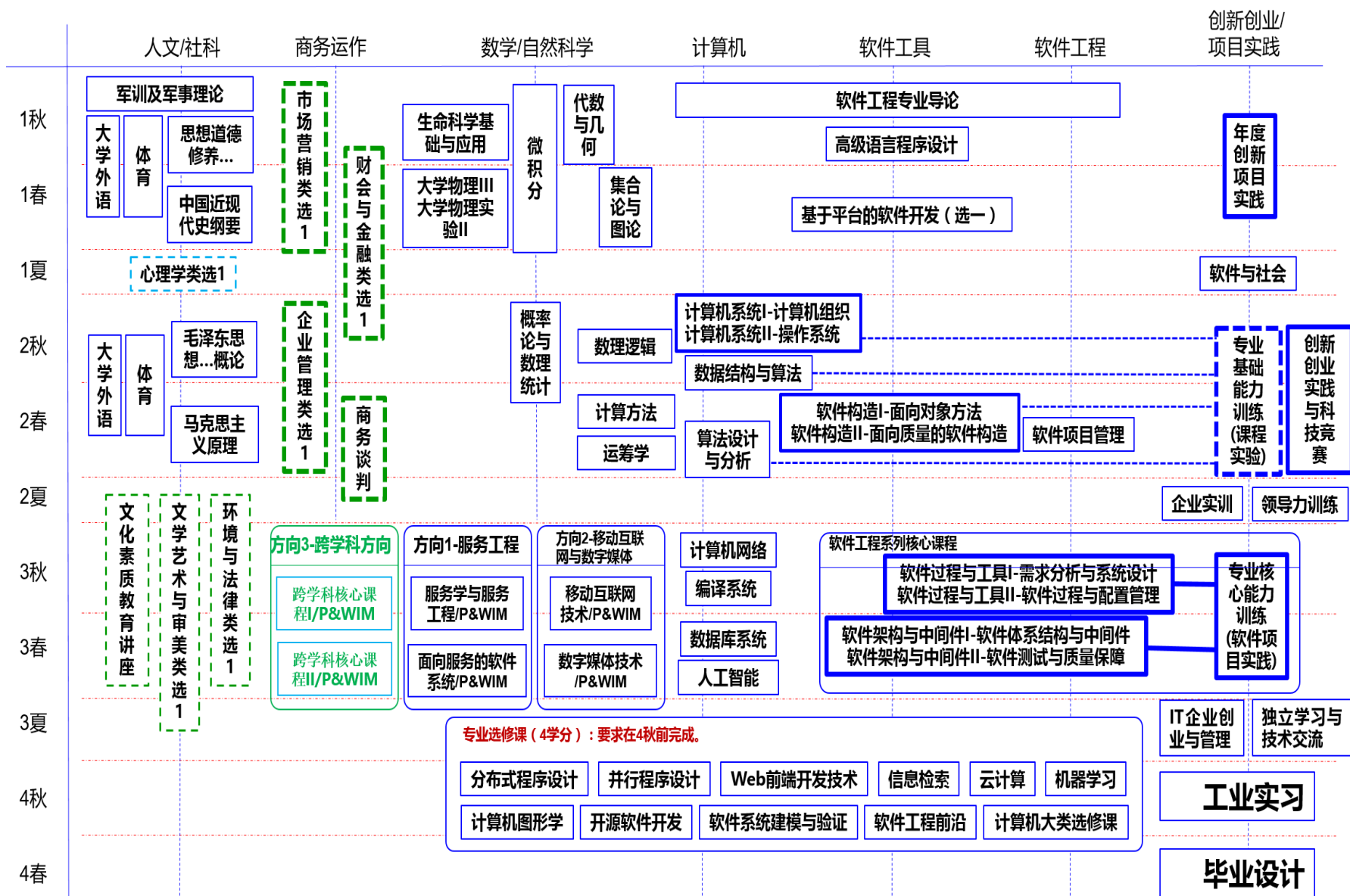
计算机大类专业课程选择框架(总学分：主修 159.5+辅修 21.0)

学校要求				学院-大类专业要求				学院-专业要求			
类别	课程	学期	学分	类别	课程	学期	学分	主修专业	(学生选择专业)		
公共基础课			23.0	数学与自然科学基础课			8.0	类别	课程	学期	学分
	思想道德修养...	1 秋	2.0		集合论与图论	1 春	4.0	专业限选课：四类课程中分别选一			12.0
	中国近现代史纲要	1 春	2.0		数理逻辑	2 秋	2.0	限选 1	计算机网络 (课程+实验)	3 秋	3.0
	毛泽东思想...概论	2 秋	4.0		近世代数	2 春	2.0	限选 2	编译系统 (课程+实验)	3 秋	3.0
	马克思主义基本原理	2 春	3.0	专业基础课			30.0	限选 3	数据库系统 (课程+实验)	3 春	3.0
	军训及军事理论	1 秋	3.0		计算机专业导论	1 秋	2.0	限选 4	人工智能 (课程+实验)	3 春	3.0
	大学外语	1-2 学年	6.0		高级语言程序设计	1 秋	3.0	专业核心课-方向系列：九个方向选一 (参见另表)			12.0
	体育	1-2 学年	3.0		数字逻辑与数字系统设计	1 春	3.5	系列 1/I	(课程 + P&WIM)	3 秋	4.5
数学与自然科学基础课			24.5		计算机系统 I-计算机组织	2 秋	2.5	系列 1/II	(课程 + P&WIM)	3 春	4.5
	微积分	1 秋 1 春	11.0		计算机系统 II-操作系统	2 秋	2.5	系列 1/III	(课程：按选修处理)	3 春/4 秋	3.0
	代数与几何	1 秋	4.0		数据结构与算法	2 秋	3.5	注：P&WIM-Project & Writing Intensive Module.			
	概率论与数理统计	2 秋	3.0		算法设计与分析(含数值算法)	2 春	4.0	专业选修课：满足专业方向选修和总学分要求			12.0
	生命科学基础与应用	1 秋	1.0		软件构造 I-面向对象方法	2 春	2.5	专业选修 1	(学生选择)	3 秋-4 秋	2.0
	大学物理 III	1 春	4.5		软件构造 II-面向质量的软件构造	2 春	2.5	专业选修 2	(学生选择)	3 秋-4 秋	2.0
	大学物理实验 II	1 春	1.0		形式语言与自动机	2 春	2.0	专业选修 3	(学生选择)	3 秋-4 秋	2.0
人文与社会科学基础课			10.0		信息安全概论	2 春	2.0	专业选修 4	(国际课程中选 1)	3 秋-4 秋	2.0
	(经管类选 1)	4 秋前	1.5	跨学科课程			6.0	专业选修 5	(视野拓展型课程选 1)	3 秋-4 秋	2.0
	(环境与法律类选 1)	4 秋前	1.5		(学生选择课程)	2-3 学年	3.0	专业选修 6	(视野拓展型课程选 1)	3 秋-4 秋	2.0
	(工程伦理类选 1)	4 秋前	1.5		(学生选择课程)	2-3 学年	3.0	辅修专业-专业核心课-方向系列			15.0
	(心理学类选 1)	4 秋前	1.5	其他课程 (计学分)			4.0	限选	(学生选择-辅修用)	3-学年	3.0
	(文学艺术与审美类选 1)	4 秋前	1.5		企业短期实训	2 夏	2.0	系列 2/I	(课程+ P&WIM)	3 秋	4.5
	(文学艺术与审美类选 1)	4 秋前	1.5		独立学习与技术交流	3 夏	1.0	系列 2/II	(课程+ P&WIM)	3 春	4.5
讲座	文化素质教育讲座	4 秋前	1.0		领导力训练	2-3 学年	1.0	系列 2/III	(课程：按选修处理)	3 春/4 秋	3.0
创新创业课			4.0	国际课程 (不少于 1.0 学分)				辅修专业-专业选修课			6.0
	年度创新项目实践	1 春-2 秋	1.0		(可与其他类别课程共享)		1.0	辅修-选修 1	(学生选择)	3 秋/夏	2.0
	(学生选择课程与实践)	4 秋前	3.0	毕业设计			14.0	辅修-选修 2	(学生选择)	3 春/夏	2.0
					毕业设计	4 春	14.0	辅修-选修 3	(视野拓展型课程选 1)	3 秋/4 秋	2.0

各专业方向-分学期-专业核心课程，暨大学分系列课程

系列课程 专业方向	课程 I(3 秋)	课程 II(3 春)	课程 III(3 春/4 秋)	课程 I-II-III 联合实现的实验-复杂工程 问题求解能力训练(3 秋 3 春或 4 秋)
专业公共课				
计算机大类/软件工程大类	计算机网络	数据库系统		仅存在对应各课程的实验,各课程间无联系。
	编译系统	人工智能		
专业方向				
A1-计算机工程	计算机组织与体系结构	操作系统设计与实现	嵌入式系统设计与实现	典型(嵌入式)计算机的设计、实现与分析
A2-计算机科学	随机计算	随机算法	计算理论	典型随机数据处理系统的设计、实现与分析
A3-并行与分布	并行与分布系统	并行与分布算法	云计算	典型并行/分布算法的设计、实现与分析
B1-自然语言处理	机器学习	自然语言处理	信息检索	典型机器学习系统设计、实现与分析
B2-视听觉信息处理	视听觉信号处理	模式识别与深度学习	视听觉信息理解	典型视听觉信息系统设计、实现与分析
B3-数据科学与大数据技术	大数据计算基础	大数据分析	数据挖掘	典型大数据系统的设计、实现与分析
C1/1-信息安全-网络安全	密码学原理与应用	网络安全	软件安全	典型内容安全系统的设计、实现与分析
C1/2-信息安全-内容安全	系统安全	信息内容安全	逆向分析	典型网络安全系统的设计、实现与分析
D1-生物信息学	生物信息学	基因组信息学	系统生物学	生物信息学算法设计、实现与分析
E1-物联网工程	信息物理系统-理论与建模	信息物理系统-技术与系统	信息物理系统-分析与验证	典型信息物理系统的设计、实现与分析

SE 课程体系的角色



软件工程大类专业课程选择框架(总学分：主修 159.5)

+

学校要求				学院-大类专业要求				学院-专业要求			
类别	课程	学期	学分	类别	课程	学期	学分	主修专业	(学生选择专业)		
公共基础课			23.0	数学与自然科学基础课			8.0	类别	课程	学期	学分
	思想道德修养...	1 秋	2.0		集合论与图论	1 春	2.0	专业限选课：			12.0
	中国近现代史纲要	1 春	2.0		数理逻辑	2 秋	2.0	限选 1	计算机网络 (课程+实验)	3 秋	3.0
	毛泽东思想...概论	2 秋	4.0		计算方法	2 春	2.0	限选 2	编译系统 (课程+实验)	3 秋	3.0
	马克思主义基本原理	2 春	3.0		运筹学	2 春	2.0	限选 3	数据库系统 (课程+实验)	3 春	3.0
	军训及军事理论	1 秋	3.0	专业基础课			26.0	限选 4	人工智能 (课程+实验)	3 春	3.0
	大学外语	1-2 学年	6.0		软件工程专业导论	1 秋	3.0				
	体育	1-2 学年	3.0		高级语言程序设计	1 秋	3.0	专业核心课-必修系列			9.0(或 6.0)
数学与自然科学基础课			24.5		基于平台的软件开发(选一)	1 春	2.5	系列 1/Ⅰ	(课程 + P&WIM)	3 秋	4.5(或 3.0)
	微积分	1 秋 1 春	11.0		计算机系统 I-计算机组织	2 秋	2.5	系列 1/Ⅱ	(课程 + P&WIM)	3 春	4.5(或 3.0)
	代数与几何	1 秋	4.0		计算机系统 II-操作系统	2 秋	2.5	注：P&WIM-Project & Writing Intensive Module			
	概率论与数理统计	2 秋	3.0		数据结构与算法	2 秋	2.5	专业核心课-方向系列(本专业或跨专业)			6.0(或 9.0)
	生命科学基础与应用	1 秋	1.0		算法设计与分析	2 春	3.0	系列 2/Ⅰ	(课程 + P&WIM)	3 秋	3.0(或 4.5)
	大学物理 III	1 春	4.5		软件构造 I-面向对象方法	2 春	2.5	系列 2/Ⅱ	(课程 + P&WIM)	3 春	3.0(或 4.5)
	大学物理实验 II	1 春	1.0		软件构造 II-面向质量的软件构造	2 春	2.5	注：两个核心课系列必修，但只需完成 1 个系列的 P&WIM。			
人文与社会科学基础课			8.5		软件项目管理	2 春	2.0	专业选修课：满足专业方向选修和总学分要求			4.0
	(文学艺术与审美类选 1)	1-2 学年	1.5	商务类课程			7.5	专业选修 1	(国际课程中选 1)	3 秋-4 秋	2.0
	(文学艺术与审美类选 1)	1-2 学年	1.5		(财会与金融类课程选 1)	1-3 学年	2.0	专业选修 2	(视野拓展型课程选 1)	3 秋-4 秋	2.0
	(环境与法律类选 1)	1-2 学年	1.5		(企业管理类课程选 1)	1-3 学年	2.0				
	(心理学类选 1)	1-2 学年	1.5		(市场营销选 1)	1-2 学年	2.0				
	软件与社会	1-2 学年	1.5		商务谈判	1-2 学年	1.5				
				其他课程(计学分)			4.0				
讲座	文化素质教育讲座	4 秋前	1.0		企业实训	2 夏	2.0	国际课程(不少于 2.0 学分)			
创新创业课			4.0		独立学习与技术交流	3 夏	1.0		(可与其他类别课程共享)		2.0
	年度创新项目实践	1 春-2 秋	1.0		领导力训练	2-3 学年	1.0				
	IT 企业创业与管理	4 秋前	1.0	毕业设计			14.0	工业实习			9.0
	(学生选择课程与实践)	4 秋前	2.0		毕业设计	4 春	14.0		工业实习(在企业)	4 秋	9.0

各专业方向-分学期-专业核心课程，暨大学分系列课程

系列课程 专业方向	课程 I(3 秋)	课程 II(3 春)	课程 I-II 联合实现的实验-复杂工程问题求解能力训练(3 秋 3 春)
专业公共课			
软件工程大类	计算机网络	数据库系统	仅存在对应各课程的实验,各课程间无联系。
	编译系统	人工智能	
专业方向			
F1-软件工程 (必修)	软件过程与工具 I-需求分析与系统设计 软件过程与工具 II-软件过程与配置管理	软件架构与中间件 I-软件体系结构与中间件 软件架构与中间件 II-软件测试与质量保障	典型软件系统的设计、实现与分析
		移动开发	
N1-服务工程 (可选一)	面向服务的软件系统	服务工程与应用	典型服务系统的设计、实现与分析
N2-移动互联网与数字媒体 (可选一)	移动互联网技术	数字媒体技术	面向数字媒体的典型移动系统设计、实现与分析
S-其他学科方向 (可选一)	跨学科核心课程 I+P&WIM (要求 I 和 II 必须同一学科)	跨学科核心课程 II+P&WIM (要求 I 和 II 必须同一学科)	典型 XX 学科计算系统的设计、实现与分析

教师



刘宏伟



史先俊



郑贵滨



吴锐

欺骗: 描述

- **请注意，特别是你在哈工大是第一学期**
- **什么是欺骗？**
 - 共享代码: 通过拷贝, 重敲, 看看, 或提供文件
 - 描述: 一个人向其他人口头描述代码.
 - 辅导: 一行一行地帮你的朋友写实验
 - 为答案进行Web搜索
 - 从前面的课程或在线解决方案中拷贝代码
 - 你只允许使用我们提供的代码, 或 CS:APP website
- **什么不是欺骗？**
 - 解释怎么使用系统和工具
 - 帮助其他人进行高层次的设计问题
- **详情请参阅课程大纲.**
 - 无知不是借口

欺骗: 后果

■ 对欺骗的处罚:

- 从课程里剔除: 不及格 (没有例外!)
- 教师的蔑视

■ 欺骗的检查:

- 有专业的代码剽窃检查工具
- 发现记为0分
- 情节严重可能从学校开除

■ 不要做!

- 太早开始
- 当卡住时问助教等

教材

■ 深入理解计算机系统 3-机械工业出版社

- Randal E. Bryant and David R. O'Hallaron,

《*Computer Systems: A Programmer's Perspective, Third Edition (CS:APP3e)*》, Pearson, 2016

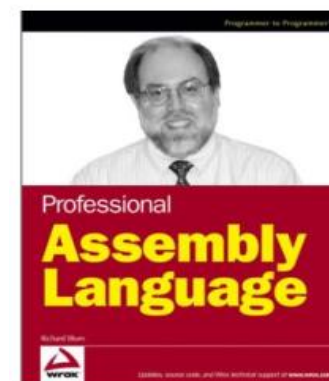
- <http://csapp.cs.cmu.edu> (教材网站)
- 这本书对这门课很重要!
 - 如何解决实验
 - 练习题中有典型的考试题目

■ 计算机系统基础

- 南京大学 袁春风

■ Professional Assembly Language

- Richard Blum



CMU课程 网站

- CMU课程 网站: <http://www.cs.cmu.edu/~213>
 - 完整的课程计划安排表, 考试, 作业
 - 讲授、作业、测验、答案的拷贝
 - 作业的澄清

课程组成

■ 大班讲授

- 高层次的概念

■ 复习-练习-习题

- 应用概念、重要的工具和实验技巧，澄清讲授，考试覆盖相关内容

■ 实验 (7)

- 课程的关键
- 每次1-2 周
- 提供对系统的某方面的深入理解
- 编程和测试

■ 考试

- 测试对概念和数学原理的理解

政策: 实验和检查

■ 团队工作?

- 你必须独立完成所有的实验作业

■ 递交

- 截止时间按实验指导教师规定
- 使用cms.hit.edu.cn乐学网电子提交 (没有例外!)

考核

考核环节	分值比例	考核/评价细则
实验	30%	1. linux下C工具应用； 3学时； 2分 2. 数据表示 ； 3学时； 5分 3. 破解：二进制炸弹； 3学时； 5分 4. 漏洞攻击 ； 2学时； 3分 5. Cache高速缓冲器； 3学时； 5分 6. 微壳TinyShell ； 3学时； 5分 7. 存储器分配 ； 3学时； 5分
作业	10%	平时作业5次：汇编2次，组原1次，编译连接1次，OS 1次
大作业	10%	大作业3次：汇编3分、组原4分、OS3分，格式为毕设论文的正文。
期末考试	50%	一纸开卷试卷

程序与数据

■ 主题

- 位操作,算术预算, 汇编语言程序
- C控制与数据结构的表示
- 包括体系结构与编译的方面

■ 实验

- L1 (datalab): 位操作
- L2 (bomblab): 拆除一个二进制炸弹
- L3 (attacklab): 代码注入攻击的基础知识

存储器层次

■ 主题

- 存储技术,存储层次, 高速缓冲器, 磁盘, 局部性
- 包括体系结构与编译的方面

■ 实验

- L4 (cachelab): 建立一个 cache 模拟器, 并为局部性进行优化.
 - 学习如何在你的程序中利用局部性.

异常控制流

■ 主题

- 硬件异常，进程，进程控制，Unix信号，非局部跳转
- 包括体系结构、OS与编译的方面

■ 实验

- L5 (tshlab): 编写自己的 Unix 外壳.
 - 第一次引入并发

虚拟存储器

■ 主题

- 虚拟存储器, 地址翻译, 动态存储器分配
- 包括体系结构、OS的方面

■ 实验

- L6 (malloclab): 编写你自己的存储器分配程序包
 - 真实感受系统底层的编程

寻找组织？

■ QQ群

HIT-CS-2018-郑贵滨

群 号： **873473488**

加群密码： hit-cs-zgb



■ 乐学网：计算机系统（2017 郑贵滨）

密码： hit-cs-zgb

■ 答疑：

周五, 14:00-15:00 综合楼603

**Welcome
and
Enjoy!**