

概述

计算机系统基础

任课教师：

龚奕利

yiligong@whu.edu.cn

提纲

- 课程主题
- 一些重要的事实
- 本课程在CS课程体系中的位置
- 学术诚信

课程主题：

抽象很好，但别忘记现实的模样

- 大多数CS课程强调抽象 (Abstraction)
 - 抽象数据类型
 - 渐近的性能分析
- 这样的抽象有其局限性
 - 特别是在出现bug的时候
 - 需要理解底层实现的细节
- 上完这门课可能的收获
 - 成为更有效率的程序员
 - 能够更有效率地找到和消除bug
 - 能够理解程序性能并调优
 - 为后续CS中的系统课程做好准备
 - 编译、操作系统、网络、计算机体系结构、嵌入式系统、存储系统等等

往届同学课程学习心得

可以让人对计算机体系有了一个总体的了解，这门课程不仅让后续课程的学习变得轻松了许多(其他班的同学学体系结构时看见汇编都被吓了大跳)，也让人学习了很多实际中有用的东西。比如说实验就要求安装Linux来使用。

感觉对具体实现有个初步概念，然后一步一步很有成就感，解开谜团的感觉。对将来的电脑学习充满了兴趣。

计算机系统基础是让我开始觉得计算机真的有意思的课程。做labs虽然耗费精力但是好玩。对计算机组成原理的过渡效果非常好，计组学习基本没有太大压力。尤其是优化程序性能的那一部分，对编程帮助也蛮大的。

别整虚的（比如相信书光看就能看懂，大佬除外），多动手做课后题和lab

计算机系统基础是CMU的镇校神课当然要上



事实#1:

Int不是整数, Float不是实数

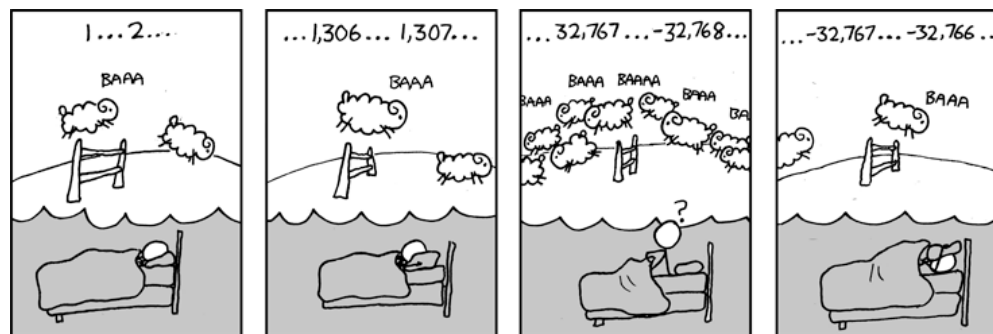
■ 例子1: $x^2 \geq 0$?

- Float: Yes!

- Int:

- $40000 * 40000 \rightarrow 1600000000$

- $50000 * 50000 \rightarrow ??$



■ 例子2: $(x + y) + z = x + (y + z)$?

- Unsigned & Signed Int: Yes!

- Float:

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

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

计算机中的运算

■ 没有随机数这回事

- 算术运算有重要的数学属性

■ 不能假设所有的“常见”的数学属性都是成立的

- 由于表达的有限性
- 整数运算满足“环”属性
 - 交换律、结合律、分配律
- 浮点运算满足“有序”属性
 - 单调性

■ 观察

- 要能理解哪种抽象适用于哪种上下文
- 编译器编写者和靠谱的应用程序员面临的重要的问题

事实#2:

你得要会汇编

- 你可能永远都不会用汇编写程序
 - 编译器比你写得好得多&比你更有耐心
- 但是：理解汇编是打开机器级执行模型的钥匙
 - 在有bug的情况下，程序的行为
 - 高级语言模型失效
 - 程序性能调优
 - 理解编译器做/不做哪些优化
 - 理解哪些原因会导致程序低效
 - 实现系统软件
 - 编译器的目标是生成机器代码
 - 操作系统必须能管理进程的状态
 - 与恶意软件做斗争
 - x86汇编是不二之选！

事实#3：内存很重要

随机访问的内存模型是一种不现实的抽象

- 内存并非是无穷大的
 - 要分配和管理内存
 - 许多应用的内存需求是占主导的
- 内存访问的bug尤其可恶
 - 效果有可能在时间和空间上都很遥远
- 内存性能不是一致的
 - Cache和虚拟内存会极大的影响程序性能
 - 使程序适应内存系统的特性能带来极大的速度提升

内存引用bug示例

```
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

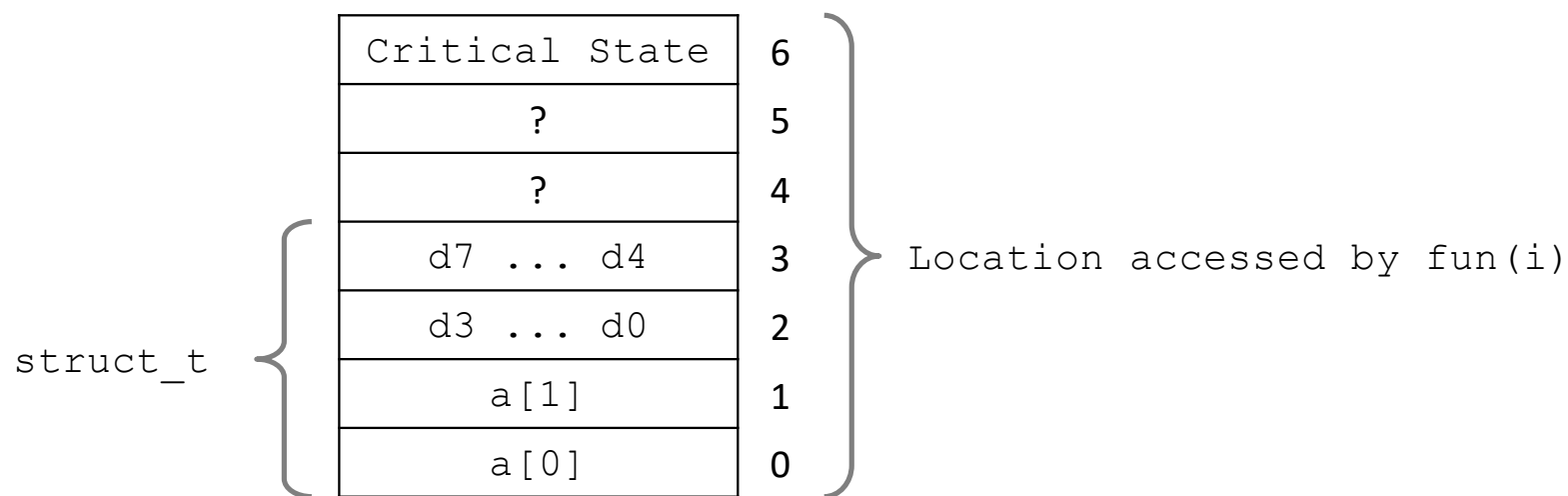
- 结果是与系统相关的

内存引用bug示例

```
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

解释:



内存引用错误

■ C/C++没有提供任何内存保护

- 数组引用越界
- 非法指针值
- malloc/free的滥用

■ 可能导致非常严重的bug

- bug是否会表现出效果依赖于系统和编译器
- 行为可能在很远的地方才显现出来
 - 被破坏的对象逻辑上与正在访问的对象无关
 - bug的效果第一次显现出来可能距离产生已经很久了

■ 我该怎么办呢？

- 用Java, Python, Matlab, ...写程序
- 理解可能会发生什么事情
- 利用或开发一些工具来发现引用错误 (e.g. Valgrind)

事实#4: 性能远不止渐近复杂度

- 常数因子也很重要!
- 即使知道准确的运算数量也不一定能预测程序的性能
 - 代码的书写方式很容易就引发10倍的性能差异
 - 必须在多个层面上进行优化: 算法、数据表示、过程和循环
- 要想优化性能必须要理解系统
 - 如何编译和执行程序
 - 如何测量程序的性能, 确认瓶颈
 - 如何在不破坏代码模块性和一般性的情况下提高性能

内存系统性能示例

```
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

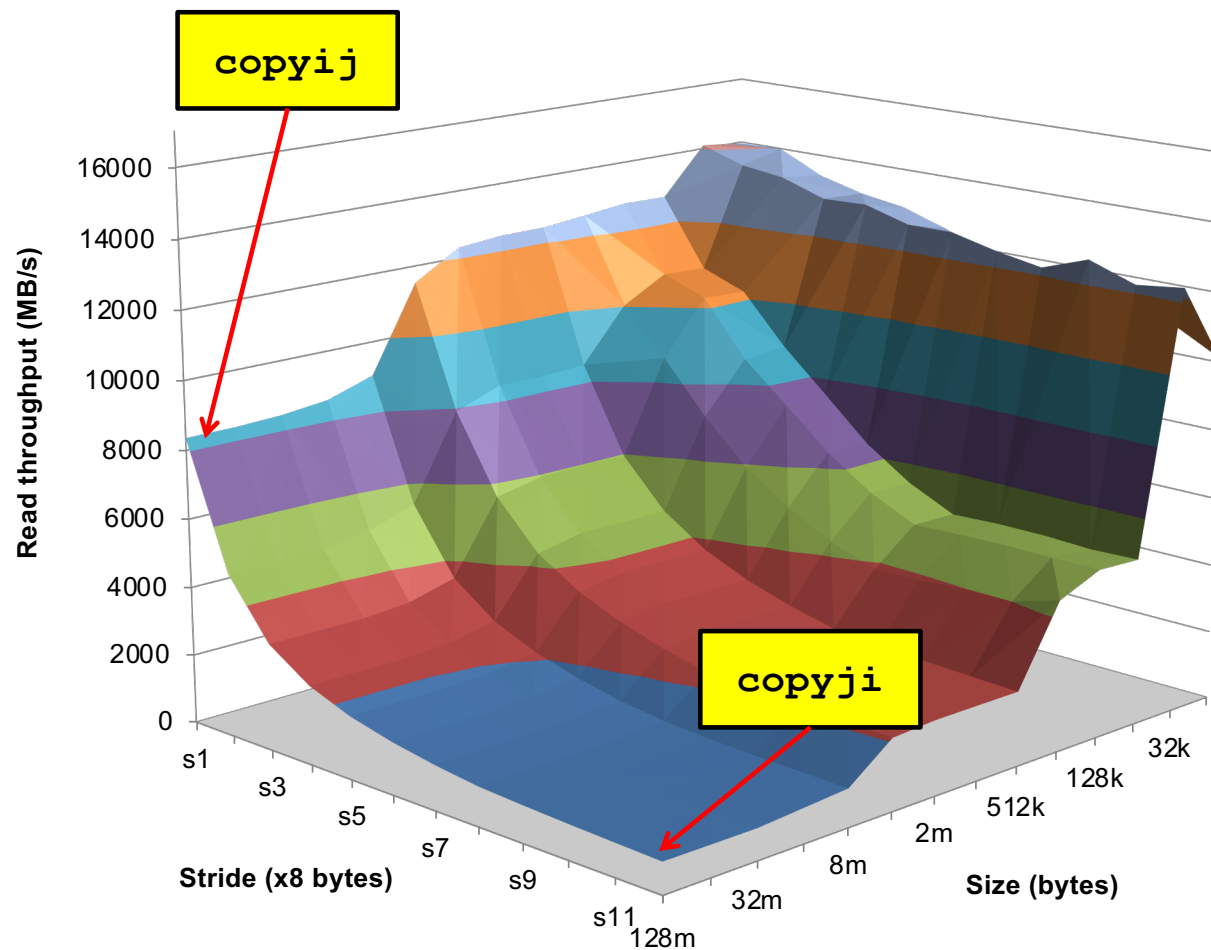
```
void copyji(int src[2048][2048],
            int dst[2048][2048])
{
    int i,j;
    for (j = 0; j < 2048; j++)
        for (i = 0; i < 2048; i++)
            dst[i][j] = src[i][j];
}
```

81.8ms

2.0 GHz Intel Core i7 Haswell

- 层次的内存结构
- 性能依赖于访问模式
 - 包括以何种步长遍历多维数组

为什么性能会不同？



本课程的视角

- 大多数系统课程是以建造者为中心的
 - 计算机组成与体系结构
 - 用Verilog设计流水线化的处理器
 - 操作系统
 - 实现操作系统的某些部分 system
 - 编译器
 - 编写某种简单语言的编译器
 - 网络
 - 实现和模拟网络协议

本课程的视角（续）

- 我们的课程是以程序员为中心的
 - 目的是向学生展示：了解更多的底层系统，你能够成为一个更有效率的程序员
 - 本门课程涵盖的内容在其他课程里都不会找到
 - 不是只有专门的黑客才该学习的课程Not just a course for dedicated hackers
 - 我们让每个人在自己身上找到隐藏的黑客！

课程诚信

■ 什么是作弊？

- 共享代码：拷贝、重新输入一遍、查看或者提供文件
- 描述：一个人向另一个人语言描述代码
- 指导：一行一行地帮助你的朋友写实验
- 在网上搜索答案
- 抄袭往届或网上其他答案的代码
 - 你只应该根据提供给你的代码完成实验

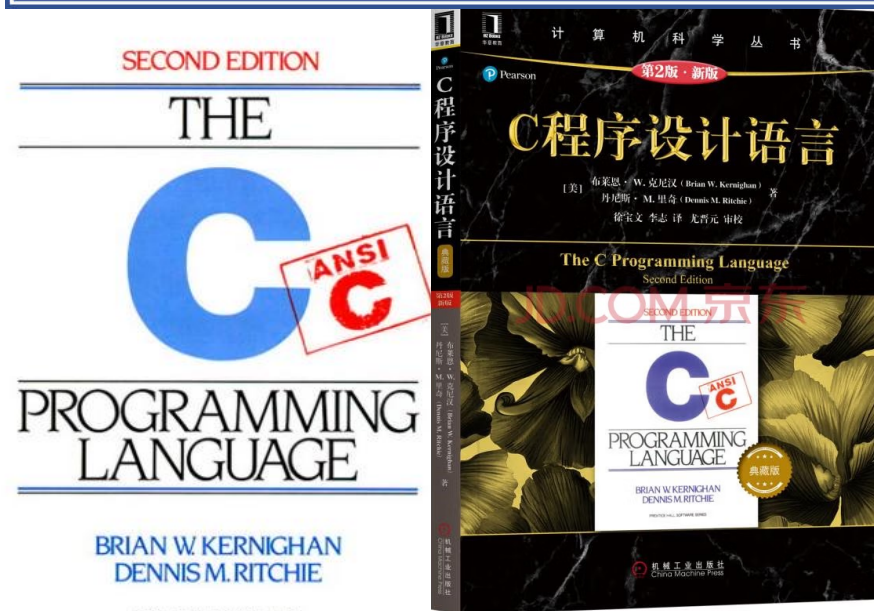
■ 什么不是作弊？

- 解释如何使用系统或工具
- 向其他人讲解一些抽象的设计问题

教材

- Randal E. Bryant and David R. O'Hallaron
 - 英文版: *Computer Systems: A Programmer's Perspective, 3rd Edition*, Pearson, 2016年1月。
 - 中文版: 深入理解计算机系统 (第三版), 机械工业出版社, 龚奕利, 贺莲译, 2016年12月。
- Brian Kernighan and Dennis Ritchie
 - 英文版: *The C Programming Language*, Second Edition, Prentice Hall, 1988年3月。
 - 中文版: *C程序设计语言*, 机械工业出版社, 徐宝文, 李志译2018年4月。

计算机系统基础-深入理解计算机系统 (第三版)



课程组织

- 课堂讲授

- 抽象概念

- 作业

- 基于希冀平台
- 对课堂讲授理论的实践与练习
- 期末考试题目的原型和蓝本

- 实验 (3个)

- 基于希冀平台
- 本课程的核心
- 让你深度理解系统的某个方面

- 翻转课堂/讨论课

- 期末考试

教学辅助

- 签到、课堂练习：微助教
 - 扫码关注公众号：微助教服务号
 - 点击公众号下方“学生”-“全部(A)”-“加入课堂”---“输入课堂编号”手动加入课堂
 - 课堂编号：MH163；课堂名称：计算机系统基础-计科13班
- 通知、答疑：QQ群/私信
- 课程视频：学堂在线（即将上线）
- 作业与实验平台：计算机学院希冀平台
 - <https://cslabcg.whu.edu.cn/>
 - 账号已导入，务必登陆后补充email，方便找回密码
- 教辅TA：QQ群，批改作业，答疑
- 线下：教师办公室时间
 - 每周四中午12点~2点

成绩构成

- 期末考试 (60%)
- 实验 (20%): 根据难度比例有差异
- 课堂表现 (10%)
- 作业 (10%)

实验

- L1 (datalab): 处理位 (bits)
- L2 (bomblab): 拆除二进制炸弹
- L3 (attacklab): 简单的代码注入攻击

Welcome
&
Enjoy!