CSAPP观看笔记

# Lecture01

观看记录📝

05:23 课程主题Abstraction is Good But Don’t Forget The Reality

07:22 课程目标 1.让你更有效率；2.给你十分有用的工具；

07:35 课程内容和课程意义

课程内容

几个关于系统的例子

09:13 #1 Ints和Floats (gdb)

17:03 #2 Assembly汇编语言

18:37 #3 内存的重要性

25:23 #4 底层优化

28:59 #5 I/O的重要性和网络

课程意义

29:07 课程意义部分

课程观点

30:31 builder角度和程序员角度

34:44 关于北大旁书店的笑话 还蛮好笑的

38:49 学术诚信:高层次可以，低层次不行

46:14 学习策略，每次阅读每章三次，然后做练习题

51:16 课程邮箱，FAQ

1:02:01 六个实验的介绍

1:02:08 数据实验

1:03:24 炸弹实验

1:06:09 攻击实验

1:07:11 缓存实验

1:09:27 Shell实验

1:09:55 malloc实验

1:12:20 proxy实验

# Lecture2

观看记录📝

02:13 bit比特

03:33 二进制的数字表示

04:49 编码一个字节以及16进制表示

06:39 C中数据类型的长度

08:50 布尔代数和操作符

11:10 对向量操作的布尔代数

12:07 使用布尔代数表示和操作集合

14:30 逻辑操作和位操作的对比

18:44 移位操作

21:09 数字表示

26:37 数字表示的范围

有符号数和无符号数

35:58 位表示到两者的转换

37:16 有符号数和无符号数的相互转换

41:46 C中有符号数和无符号数

55:57 有符号数和无符号数的转换

57:53 关于无符号数的两个问题

1:01:03 有符号数的扩展

1:06:04 扩展和截断的总结[加油武汉]

# Lecture3

观看记录📝

位表示

位级别的操作

整数

- 表示：无符号和有符号

- 约定和转换

- 扩展和截断

- 加法，减法，乘法，移位

内存，指针，字符串的表示

总结

02:32 无符号数的加法

06:19 无符号运算加法的可视化

06:42 补码的加法

13:42 补码加法的可视化

15:02 无符号数的乘法

17:20 C中有符号数的乘法

24:33 2的指数幂乘法和左移等价

26:32 2的指数幂除法和右移等价

28:48 逻辑移位和算术移位应用

36:03 不要使用无符号数的原因

39:03 如果使用无符号数的规范

44:22 使用无符号数的原因44:56 内存，指针，字符串的表示

48:00 内存的结构梳理

49:39 机器的字长

53:14 面向字节的内存结构

【这里关于内存，需要推荐一下另一门课《十分钟计算机科学-内存部分》，看完对这部分会有更好的理解】

57:05 小端序和大端序

1:01:19 数据表示的例子

1:08:38 字符串部分「很简单地略过」

1:09:58 整数问题的小测验

# Lecture 4

浮点数 观看记录📝

2:44 二进制分数的表示，浮点数“浮点”的来源

08:07 IEEE floating point

08:25 浮点数的表示

10:56 规格化的值

20:20 解释exp阶码部分，这个地方其实解释有误；其实展示的是阶码整体的范围，包括非规格化时；

23:53 非规格化的值

26:14 特殊值

28:29 浮点数编码的可视化

29:49 浮点数编码的简单例子

39:56 浮点数编码的属性

41:34 浮点数运算的基本思想(运算后舍入)

44:25 舍入的方法

51:33 二进制的舍入

54:56 浮点数的乘法

56:30 浮点数的加法

56:54 浮点数加法和乘法的数学性质

62:59 C中的浮点数

67:20 浮点数的练习题

# Lecture05 机器级编程(基础)

观看记录📝

01:49 课程学习方法

03:54 intel处理器的历史和架构

06:25 RISC和CISC的历史

08:12 英特尔架构的里程碑

12:51 Corei7的架构

14:28 竞争对手AMD的历史

15:35 64位的转变

17:49 ARM处理器

19:35 C,assembly,machine code

22:02 Machine/Assembly code的视角

25:19 将C代码翻译为目标代码的过程

30:12 C代码到汇编代码的例子，-O优化

34:36 汇编代码的特性

34:48 数据类型

35:56 操作

36:54 sumstore的例子

39:14 Disassembling反汇编器：objdump和gdb

47:50 Assembly基础:寄存器，操作数，move指令

48:12 X86-64寄存器

53:38 move指令

56:59 Simple Memory Addressing Modes

58:09 简单寻址的例子

56:59 Complex Memory Addressing Modes

65:12 寻址的例子

66:15 算术和逻辑操作

66:40 lea指令

73:02 总结

# Lecture06 机器级编程(控制)

观看记录📝

04:00 处理器的状态（关于当前运行程序的信息，rax,rsp,rip）

05:51 条件码

07:55 比较指令(cmp,test)

11:46 读状态码(setx,)

20:14 状态分支

20:36 跳转指令(这里还提及了+=的来源)

26:54 C中goto语句(尽量还是不要用，不清晰)

29:05 通用条件表达的转换(使用分支)

31:41 条件移动

35:48 条件移动的例子

36:49 条件移动不适合的情况

38:27 循环

(总的来说就是把各种形式的循环归并成一种形式)

38:29 "Do while"的例子(C语言)

40:44 "Do while"的例子(编译结果)

41:05 通用的"Do while"的Goto版本

41:21 通用的"While"的Goto版本(1)

44:07 "While"的例子

45:34 通用的"While"的Goto版本(2)

46:26 "For Loop"的形式

47:38 "For" -> "While Loop"的转换

50:10 Switch语句

51:01 Switch语句的例子

53:44 Switch的解释(不是if-else的组合！！！超酷的)

1:12:31 总结

# Lecture07 机器级编程(过程)

观看记录📝

Procedures

Stack Structure

Calling Conventions

Passing control

Passing data

Managing local data

Illustration of Recursion

2:45 ABI

3:50 Procedures的机制

7:34 X86-64Stack

ABI中调用的约定

14:38 过程中的控制

15:32 过程中的控制(例子)

21:43 过程中的数据(参数传递)

23:06 过程中的数据(例子)

26:19 过程中的局部变量管理

26:22 栈帧

31:15 一个具体的例子

34:42 X86-64/Linux栈帧的结构

39:06 Calling incr的例子

过程调用中寄存器的使用

46:29 Register保存的约定

49:38 不同寄存器的用途

51:17 被调用者保存寄存器的使用示例及其工作原理

递归部分的解释

54:31 递归的例子

1:03:40 递归部分的总结

# Lecture08 机器级编程(数据)

观看记录📝

数组

04:16 数组声明

05:14 数组访问

14:38 数组的例子

21:49 数组和指针的不同(C语言部分，可看书)

39:47 嵌套数组

45:32 嵌套数组的访问

51:04 多层数组的元素访问

54:58 N\*N的矩阵

56:41 16\*16数组访问的例子

结构

59:24 结构体的表示

1:00:16 指向结构体成员的指针

1:05:34 结构体和对齐的概念

1:07:03 对齐的原则

1:11:15 节省内存

浮点数

1:13:00 浮点数的历史

1:14:44 SSE3编程

1:16:01 Scalar和SIMD(真的好感动，实习时用过)

1:17:13 FP运算的知识

其实这里应该提及多个浮点数一起运算的加速效果

# Lecture09 机器级编程(高级专题)

观看记录📝

内存布局

2:14 最大寻址大小的解释(47位)（limit）

10:30 内存布局的解释(stack,heap,data,text/shared lib)

12:07 内存分配的例子

缓冲区溢出

17:54 缓冲区问题介绍

19:39 String库的缓冲区例子-gets,strcpy,strcat

23:26 缓冲区问题例子

32:15 缓冲区问题(缓存区溢出攻击例子)

33:31 代码注入攻击解释(很有意思)

36:47 当你替换代码是，如何确定注入的代码提供了准确的地址

38:09 Buffer overflows的历史

42:30 蠕虫和病毒的区别

43:09 避免缓冲区攻击的手段

43:24 避免程序中的缓冲区溢出

45:28 系统级别的防御:ASLR,权限，canary(啊啊啊啊啊,canary我遇到过)

1:03:51 关于这三种手段的老师的评价

1:10:59 attack lab讲解

1:12:59 联合体

1:13:40 联合体内存分布

1:15:40 字节顺序的问题

# Lecture10 程序优化

观看记录📝

01:10 整体的程序优化手段介绍

03:08 性能的两条人间真实:常数因子也很重要;必须了解系统知识才能优化性能

04:32 关于Compiler(编译器)的一些介绍

06:32 代码移动(Code Move)，先计算需要计算的值，然后在之后一直使用这个值

08:45 计算量减少(Reduction in Strength),将乘法转化为加法

10:07 共享通用表达(Share Common SubExpressions),通过抽取子表达式，将其提前转化为共享变量的方式减少计算量

13:00 优化中的障碍

13:10 障碍1，过程调用

18:08 为什么编译器无法优化这种过程调用错误,变量在变化，无法确定函数体内容

21:02 障碍2，内存别名导致，重复对内存地址进行读取的问题;

22:57 障碍2是由于内存别名的存在，导致程序无法确定两个内存是否完全无关，所以需要重复读取;

32:11 无序执行(乱序执行)，利用指令级别的并行

32:32 程序加速的例子

35:13 CPE指标的引入

38:30 程序加速，改进例子的极少

40:35 指令级并行的基本思想

44:26 超标量计算机简介

45:03 流水线技术，及其如何将计算分解为一系列不同的阶段

48:40 HasWell，不同阶段对应的时钟周期

51:00 计算机中，顺序依赖概念的解释

51:30 循环展开，对性能提高的效果展示

52:50 循环展开对于不同运算的性能提高效果

53:29 运算顺序改变，对于不同运算的性能提升效果

61:20 Avx2指令加速

66:46 条件预测，提前加载指令进行加速

72:30 获得高性能程序的一些关键点

# Lecture12 Cache Memory

观看记录📝

01:09 内存的层级结构

01:54 Cache的概念

04:42 Cache的组织结构

07:30 Cache读

07:30 Cache的寻址方式

10:38 Direct Mapped Cache(E = 1)

20:58 E-way Set Associative Cache(E = 2)

35:38 Cache写

write hit

36:11 wrtie through and write back(脏位)

write miss

37:47 write-allocate and No-write-allocate

39:40 英特尔i7的cache架构

42:42 Cache性能的度量:MIss Rate, Hit Time, Miss Penalty

44:53 关于Cache数量度量的例子

46:58 写出缓存友好的代码

50:57 Memory Mountain

53:48 Memory Mountain Test代码

56:30 Memory Mountain可视化代码

59:14 Ridges of temporal locality时间局部性的山脊

1:03:53 重新排列循环

1:04:01 Matrix Multiplication例子

1:05:13 Miss Rate分析

1:08:05 Matrix Multiplication的总结

1:10:31 分块技术

1:14:41 分块技术的Cache Miss分析

1:16:14 Blocking的效果分析

# Lecture13 链接

观看记录📝

01:51 静态链接的过程

03:40 为什么要用链接：模块化，效率(编译)

05:20 连接器的作用

05:25 符号解析

07:20 重定位

09:20 三种目标文件的分类

09:25 .o可重定位目标模块

09:50 .out可执行文件

10:15 ,so共享目标文件

10:32 目标模块ELF标准

10:58 目标文件的格式:header,segment header table, .text section, .rodata section,.data section,.bss section, .symtab section, .rel .txt section,.rel .data section,.debug section, section header

15:13 链接器的符号分类，global symbols, external symbols, local symbols

19:30 局部非静态变量 vs 局部静态变量

21:30 链接器如何解决重复符号定义问题？强符号，弱符号;

32:04 避免全局变量的原因

36:00 重定位入口

44:15 加载可执行的目标文件

48:30 打包常用的函数

51:06 静态链接库

55:51 静态库链接的过程

56:13 使用静态库的过程

60:02 共享链接库

62:52 动态链接的加载过程

67:17 运行时的动态链接

杂项

53:13 常用库

示例

24:12 链接错误示例

sum

01:04 C程序示例

18:34 符号解析的过程

32:49 重定位的过程

41:26 重定位后的text部分

案例学习

70:24 库打桩