



南京大學  
NANJING UNIVERSITY



# 本课程的主要学习内容

南京大学

计算机科学与技术系

袁春风

email: [cfyuan@nju.edu.cn](mailto:cfyuan@nju.edu.cn)

2015.6

# PostPC Era: Late 2000s - ??



iPhone  
iPod  
iPad

**Personal Mobile Devices (PMD):**  
Relying on wireless networking  
Apple, Nokia, ... build \$500 smartphone and tablet computers for individuals  
=> Objective C, Android OS

**Cloud Computing:**  
Using Local Area Networks,  
Amazon, Google, ... build \$200M  
**Warehouse Scale Computers**  
with 100,000 servers for  
Internet Services for PMDs  
=> MapReduce, Ruby on Rails



# 后PC时代计算机教学面临的挑战

---

## 后PC时代的几个特征

- 计算资源多样化，I/O设备无处不在，数据中心、PMD与PC等共存
- 软件和硬件协同设计（硬件、OS和编译器之间的关联更加密切）
- 对应用程序员的要求更高

- 编写高效程序必需了解计算机底层结构
- 必需掌握并行程序设计技术和工具
- 应用问题更复杂，领域更广

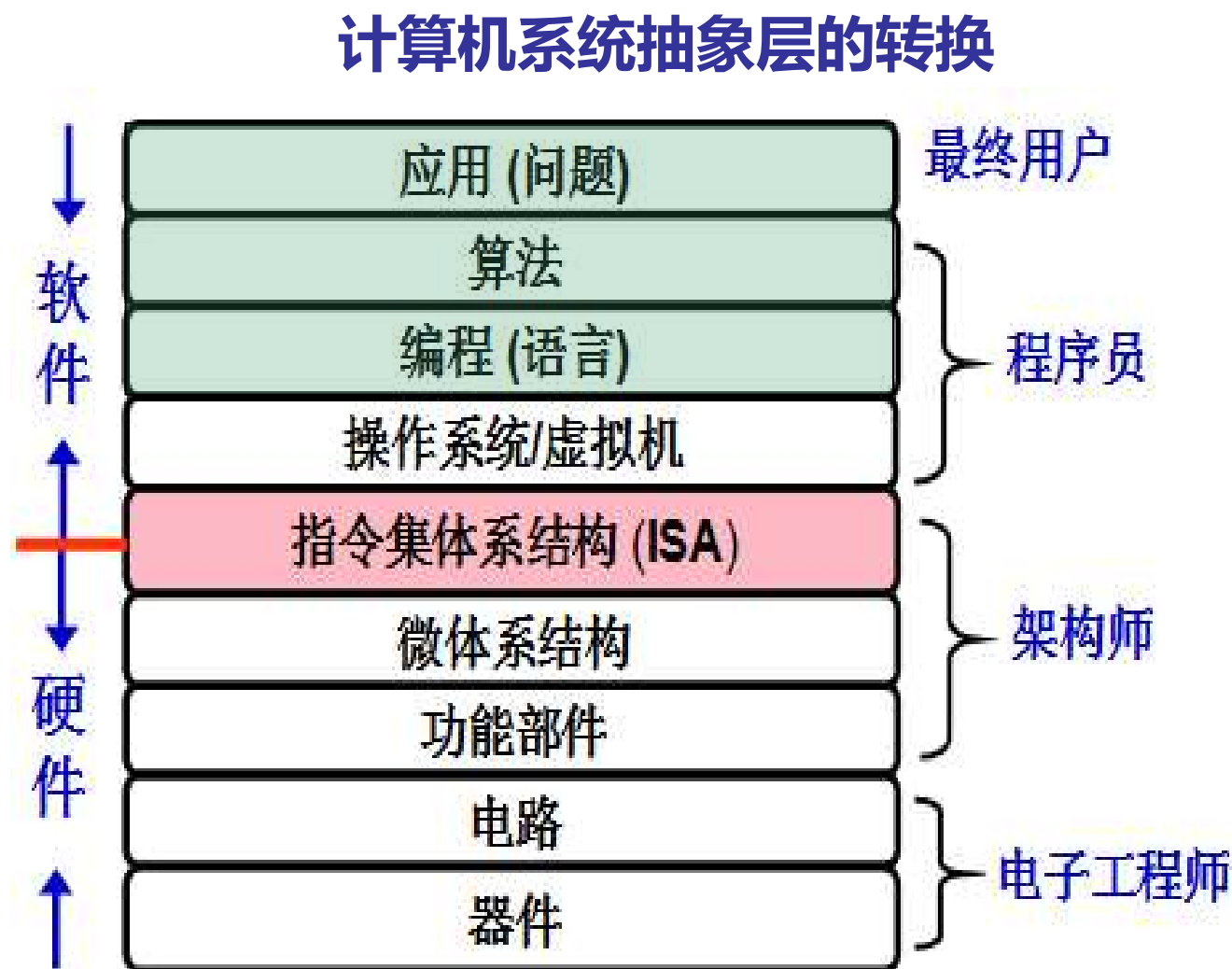
- 气象、生物、医药、地质、天文等领域的高性能计算
- Google、百度等互联网应用领域海量“大数据”处理
- 物联网（移动设备、信息家电等）嵌入式开发
- 银行、保险、证券等大型数据库系统开发和维护
- 游戏、多媒体等实时处理软件开发，。。。。。

大规模  
分布式  
多粒度并行

“并行”成为重要主题、培养具有系统观的软/硬件贯通人才是关键！

# 计算机系统抽象层的转换

- 计算机学科主要研究的是**计算机系统各个不同抽象层的实现及其相互转换的机制**
- 计算机学科培养的应该主要是在**计算机系统或在系统某些层次上从事相关工作的人才**



所有计算机专业课程基本上围绕其中一个或多个不同抽象层开展教学！

本课程为**基础课程**，试图通过讲解程序开发和执行来构建计算机系统总体框架

# “计算机系统基础” 课程

---

## 从程序员角度出发来认识计算机系统

- 教学目标：

培养学生的**系统能力**，使其成为一个“**高效**”程序员，在程序调试、性能提升、程序移植和健壮性等方面成为高手；建立完整的计算机系统概念，为后续的OS、编译、体系结构等课程打下坚实基础

- 以 **IA-32+Linux+C+gcc** 为平台（开源项目平台）

- 与以下**MOOC课程**的想法类似

<https://www.coursera.org/course/hwswinterface>

**主要内容：描述程序执行的底层机制**

- 思路：

在程序与程序的执行机制之间**建立关联**，**强化理解**而不是记忆

# “计算机系统基础” 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  sum;
```

```
    sum=sum(a,0);
```

```
    printf(“%d”,sum);
```

```
}
```

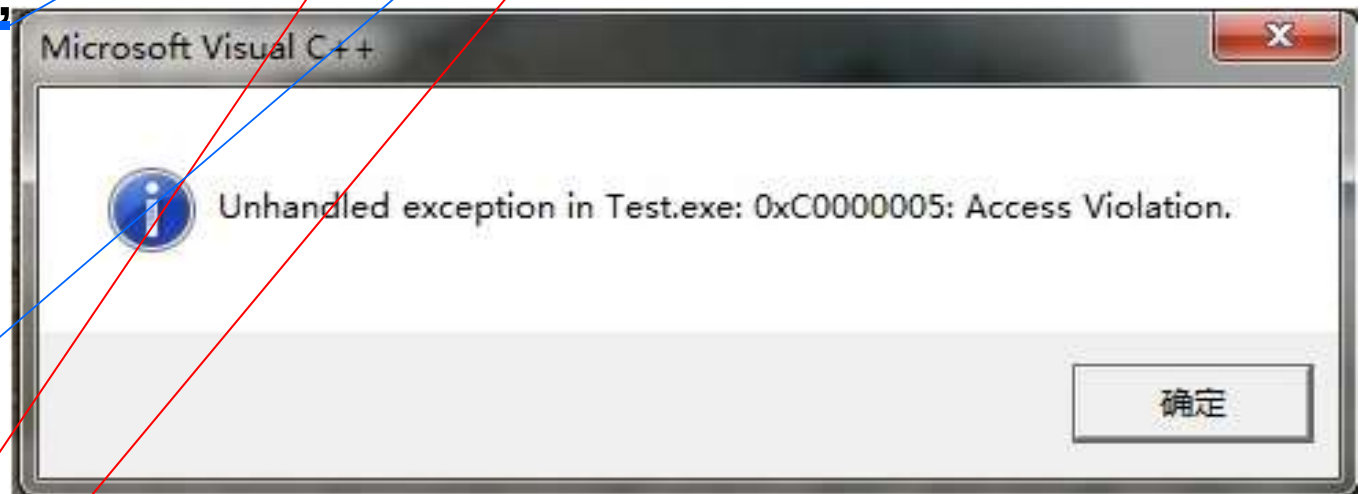
数据的表示

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)



# “计算机系统基础” 课程内容概要

- 使学生清楚理解：

**计算机是如何生成和运行可执行文件的！**

“问题求解” 解决  
应用→算法（数据  
结构）→编程层

- 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

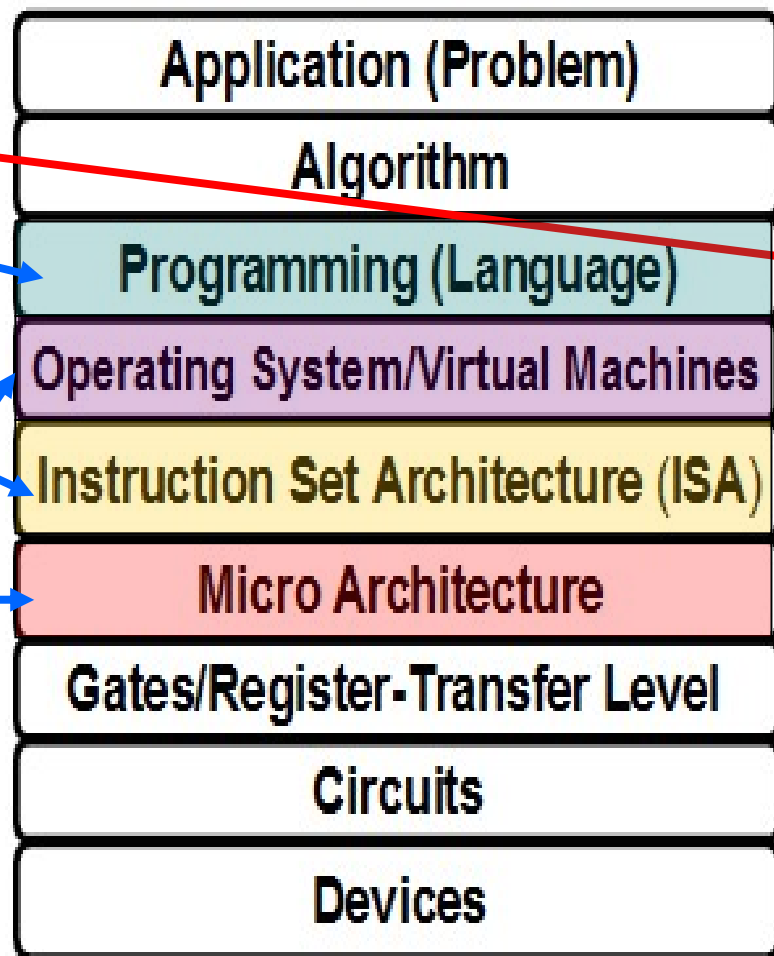
- 指令系统、机器代码、汇编语言

- 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



# “计算机系统基础” 课程内容概要

---

## 三个主题：

- 表示 ( Representation )
  - 不同数据类型 ( 包括带符号整数、无符号整数、浮点数、数组、结构等 ) 在寄存器或存储器中如何表示和存储？
  - 指令如何表示和编码 ( 译码 ) ？
  - 存储地址 ( 指针 ) 如何表示？如何生成复杂数据结构中数据元素的地址？
- 转换 ( Translation )
  - 高级语言程序对应的机器级代码是怎样的？如何转换并链接生成可执行文件？
- 执行控制流 ( Control flow )
  - 计算机能理解的“程序”是如何组织和控制的？
  - 如何在计算机中组织多个程序的并发执行？
  - 逻辑控制流中的异常事件及其处理
  - I/O操作的执行控制流 ( 用户态→内核态 )



# “计算机系统基础” 课程内容概要

---

**先导知识：**C语言程序设计

**内容组织：**分两门短课程

- **第一部分 可执行文件的生成（表示和转换）** 本阶段将学习  
第一门短课程
  - 计算机系统概述
  - 数据的机器级表示与处理
  - 程序的转换及机器级表示
  - 程序的链接

**计算机系统基础(一) ----程  
序的表示、转换与链接**
- **第二部分 可执行文件的运行（执行控制流）**
  - 程序的执行
  - 层次结构存储系统
  - 异常控制流
  - I/O操作的实现

**计算机系统基础(二) ----程  
序的执行、异常及IO处理**