



中山大學
SUN YAT-SEN UNIVERSITY



计算机组成原理

第一章：计算机概要与技术

中山大学计算机学院
陈刚

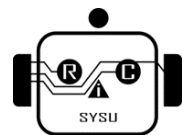
2022年秋季

本讲内容

- 计算机系统概述
 - 技术发展历程
 - 国内计算机系统
- 计算机的基本组成
- 计算机拆解

基本要求

- 掌握计算机的基本概念
- 理解计算机中程序的执行过程



本讲内容

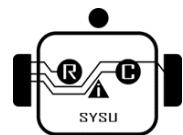
- 计算机系统概述
 - 技术发展历程
 - 国内计算机系统
- 计算机的基本组成
- 计算机拆解

看上去我们已经到达了利用计算机技术可能获得的极限了，但是下这样的结论得小心，因为不出五年这听起来就会相当愚蠢。”

—— 约翰·冯·诺伊曼

基本要求

- 掌握计算机的基本概念
- 理解计算机中程序的执行过程



从技术发展历程看电子计算机

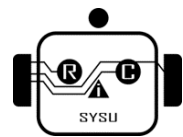
□ 从机械计算装置到电子计算装置

□ 算盘到电子计算机：数学/物理基础、结构、性能增长方式、管理等都发生了翻天覆地的变化



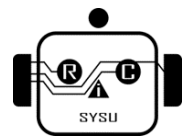
从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的通用基础
- 操作系统：电子计算机的共享基础
- 互联网：电子计算机的泛在基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础

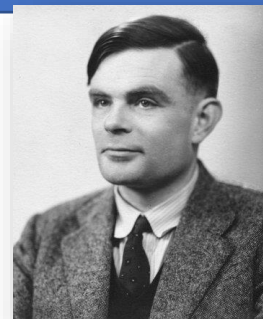


计算模型

□ 什么是(通用)计算?

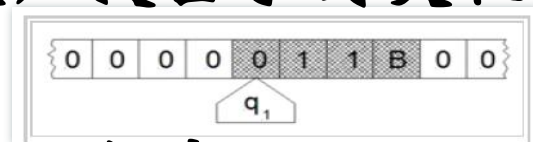
□ 机器能思考吗?

□ 计算机怎样才能开启思维?



阿兰·图灵

□ 1936年，英国科学家阿兰·图灵发表了《论可计算数及其在判定问题上的应用》的论文，提出了图灵机模型

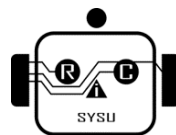


■ 图灵机模型是思想实验，定义了理想中的机器

■ 将计算定义为机械的过程(算法)

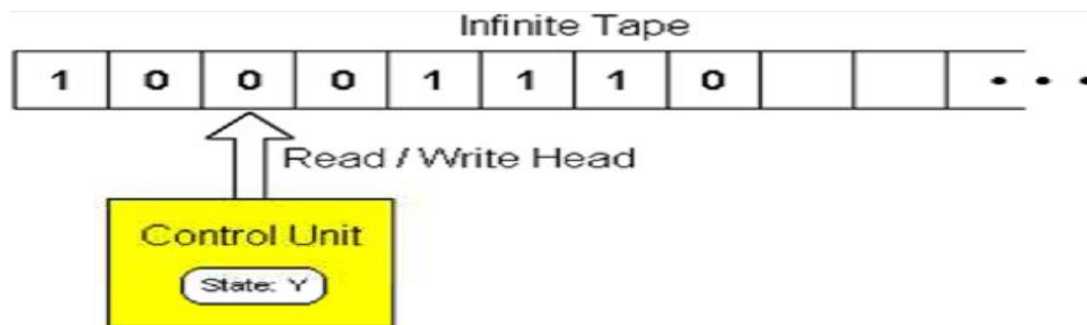
■ 图灵被誉为“**计算机科学之父**”

■ 回答：计算的通用性——理论上



计算模型

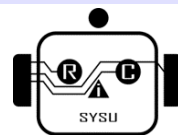
□ 计算 $f(x)=2^x$ 的图灵机程序如下



- 解决了如何自动地实现状态变化的方法
- 图灵的指令系统单一不够完善，没能将指令存储起来重复使用

◇ 计算 $f(x)=2^x$ 的图灵机程序如下

当前状态	扫描到 B 时的写、移动、状态转移	扫描到 0 时的写、移动、状态转移	扫描到 1 时的写、移动、状态转移
q_1	1, L, q_7	0, R, q_1	1, R, q_2
q_2	B, R, q_3	0, R, q_2	1, R, q_2
q_3	0, L, q_4	0, R, q_3	Error
q_4	B, L, q_5	0, L, q_4	Error
q_5	Error	1, L, q_5	0, L, q_6
q_6	B, R, q_1	0, L, q_6	1, L, q_6
q_7	Halt	B, L, q_7	Error



计算装置结构模型

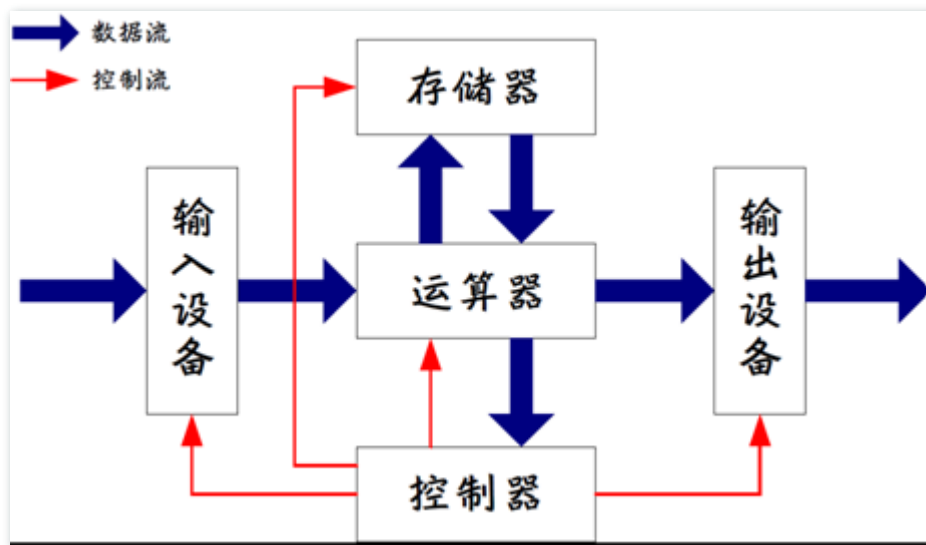
□ 什么是(通用)计算装置模型?

■ 计算机怎样才能实现复制?

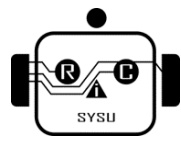
□ 1945年，美籍的匈牙利科学家冯·诺伊曼发表了《First Draft of a Report on the EDVAC》的技术报告，提出了计算装置的体系结构模型

□ 存储程序原理

□ 后人称其为“冯·诺伊曼”体系结构

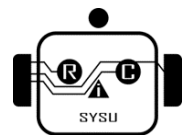


冯·诺依曼



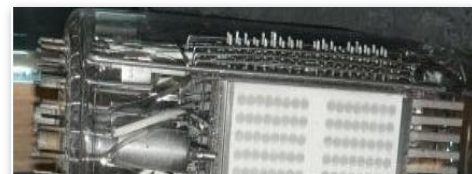
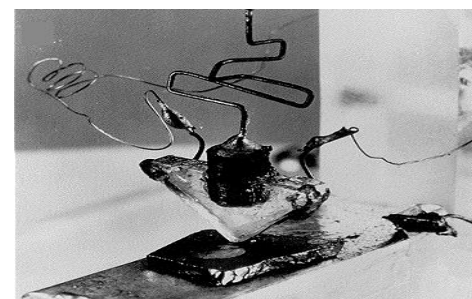
从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



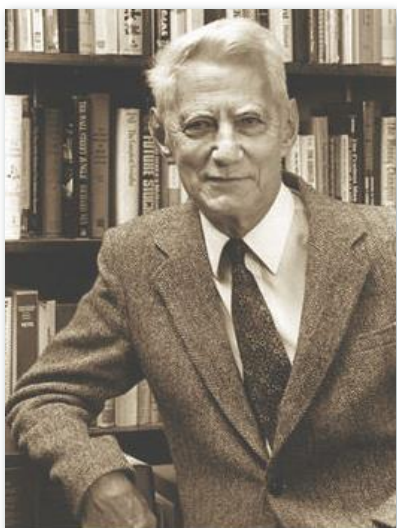
物理器件

- 1904年，弗莱明 (Fleming) 发明电子管二极管
- 1907年，德福雷斯特 (De Forest) 发明真空电子管三极管
- 1947年，布莱顿 (Brattain)、巴丁 (Bardeen) 发明点接触晶体管
- 1949年，肖克利 (Shockley) 发明比点接触晶体管性能更优越的结型晶体管
 - 1956年，肖克利、布莱顿和巴丁三人共同获得诺贝尔物理学奖
- 1946年，Rajchmand 等发明选数管，一个真空管建立的数字存储设备，可存储4096位

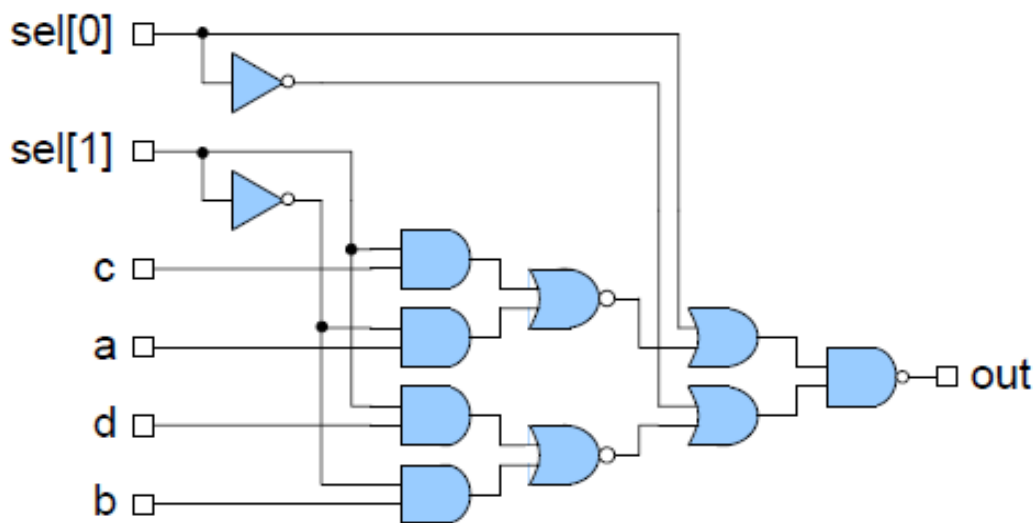


物理器件

- 1938年，美国数学家、信息论创始人香农的硕士论文《继电器与开关电路的符号分析》中引入二值数字电路，将数学世界的布尔代数与物理世界的二值器件完美地联系在一起
- 二值电子器件模拟布尔逻辑运算，奠定了数字电路的理论基石

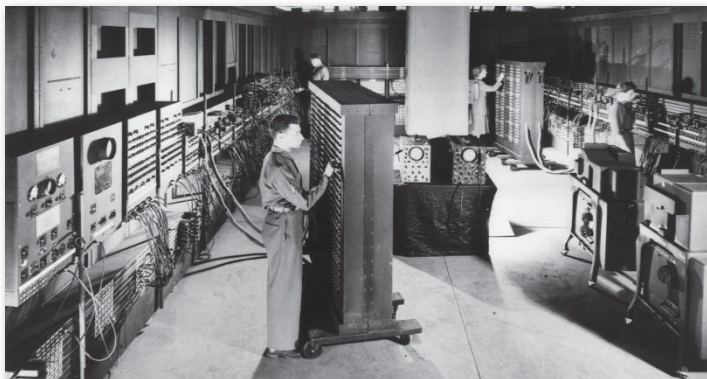


克劳德·艾尔伍德·香农



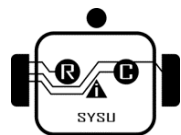
物理器件

- 1946年，第一台通用电子计算机ENIAC诞生，标志电子管构造出电子装置的最高水平
 - 17468个电子管、7200个晶体二极管、1500个继电器、10000个电容器
 - 重量达28吨，体积大约 $2.4\text{m} \times 0.9\text{m} \times 30\text{m}$ 占地167平方米，耗电150千瓦，造价48.7万美金
 - 20个寄存器，每个10b十进制数(100个电子管)，存储1000bit信息，6000个开关(通过设置开关、连接插头和插座来编程)
 - 运算速度每秒可执行5000次加法或者400次乘法运算
- 1950年，EDVAC实现了冯·诺伊曼的设想：采用二进制和存储程序



从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



大规模集成

□ ENIAC等代表了当时的数字电路制造最高水平，继续扩展面临可靠性、功耗、复杂度等一系列挑战

□ 1958年，Jack Kilby发明了集成电路

■ 2000年，获诺贝尔物理学奖

■ Robert Noyce独立发明了集成电路，是Fairchild半导体公司和Intel公司的共同创始人

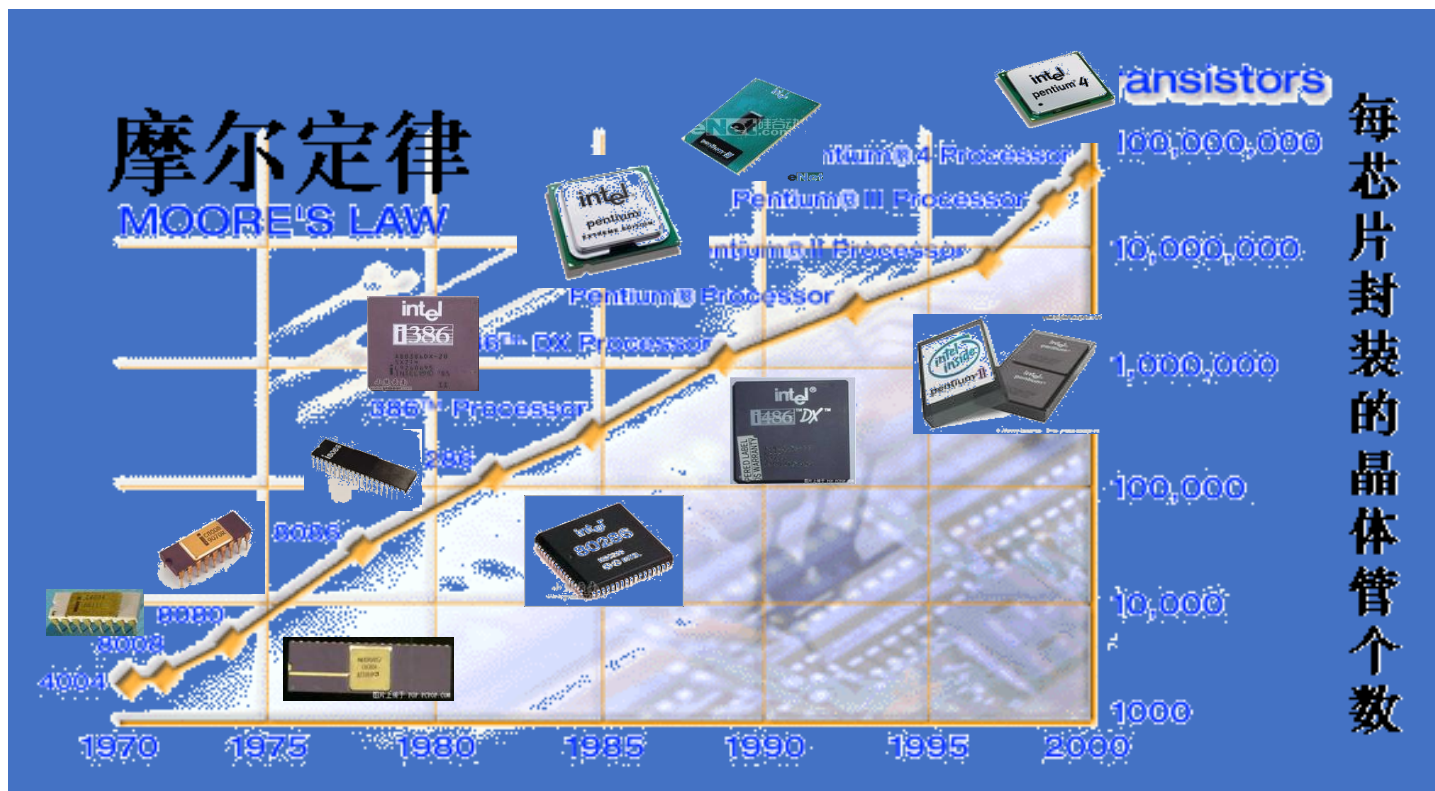
□ 1971年，Intel公司发布第一款微处理器4004

■ 2300个晶体管， $10\mu\text{m}$ 工艺

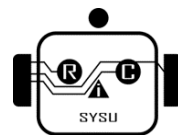


大规模集成

- 1965年，Intel 公司缔造者之一戈登·摩尔发表论文《让集成电路填满更多的组件》，提出**摩尔定律**
- 在价格不变的情况下，每18个月芯片上晶体管数目翻倍，性能也提升一倍

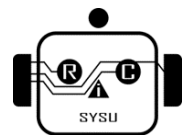


戈登·摩尔



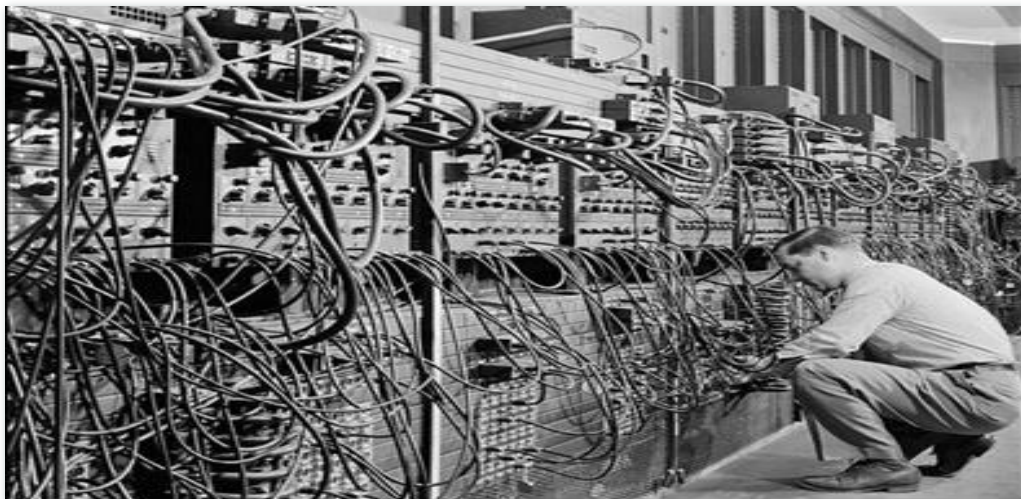
从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



存储程序

- 怎样实现“通用”和“自动”的计算装置？
- ENIAC直接以电路连线来设计程序，需要新程序时，则重新配接线路
- 将问题映射到ENIAC上求解是一个复杂的任务，通常要用几个星期的时间：首先操作各种开关和电缆把“问题”输入ENIAC需要几天；然后再进行计算和测试



存储程序



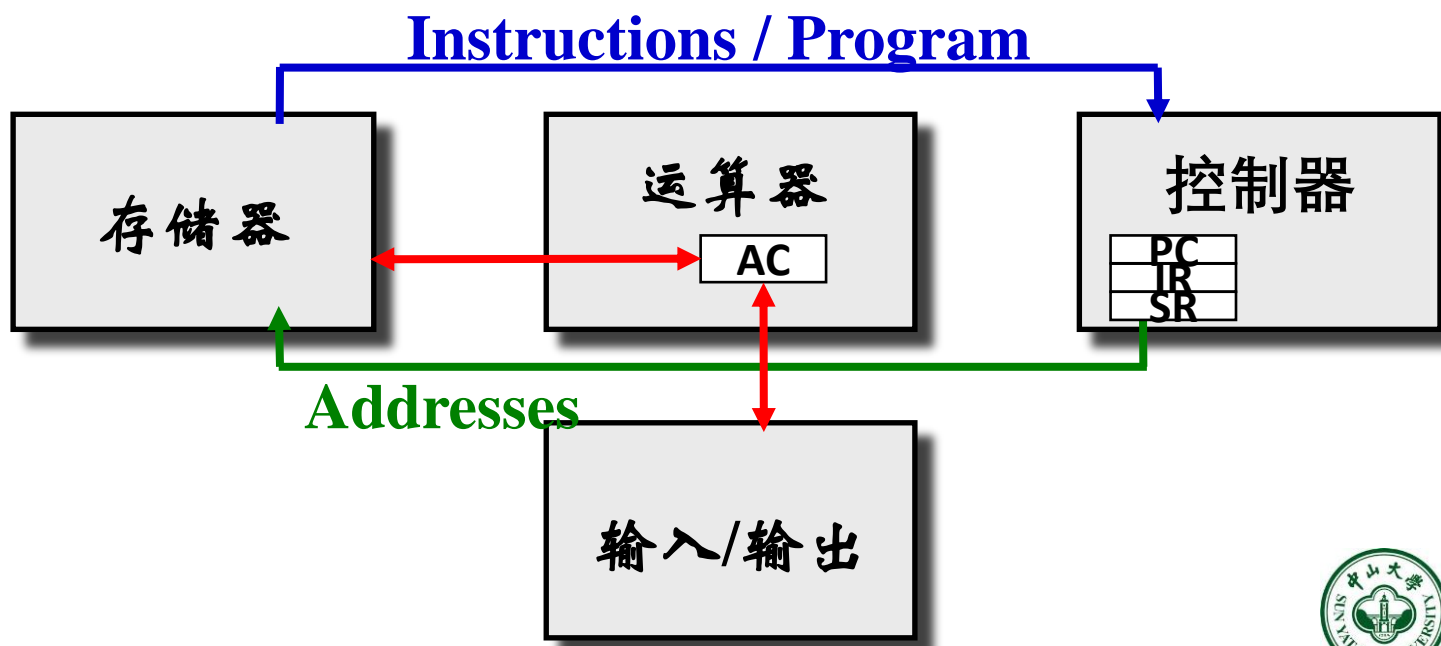
□ 1945年，冯·诺伊曼提出**存储程序原理**

□ 程序指令和数据都以二进制形式表示

□ 程序指令和数据预先存放在存储器中

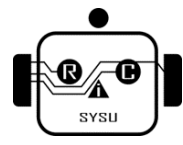
□ 自动化和序列化地执行程序中的指令

□ 存储程序带来了计算机世界的“**组合进步**”



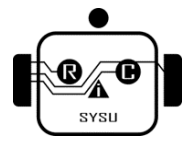
从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



操作系统

- 在实现“通用计算”的基础上，怎么实现多用户的“资源共享”？
- 20世纪40年代，计算机没有操作系统
- 20世纪50年代，出现了具有一定操作系统特征的管理程序(Monitor)
 - 能够自动加载不同的程序以加速计算机的处理
- 1964年，IBM推出系列IBM 360大型机，共用代号为OS/360操作系统。首次出现虚拟机、中断、分时、进程等概念
 - 多道程序——使多个用户能够共享使用计算机



操作系统

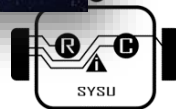
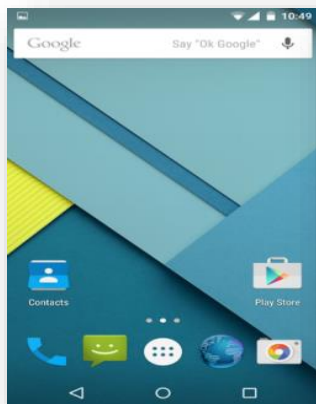
□ 操作系统：通过**虚拟化**实现资源的共享

□ 虚拟化计算机：进程

□ 虚拟化内存：虚存

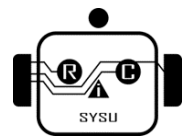
□ 虚拟化外存：文件

□ 虚拟化：对一组类似资源提供一个通用抽象接口，隐藏属性和操作之间的差异，并允许通过一种通用的方式使用、维护和管理资源



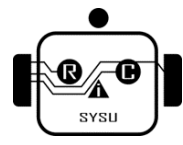
从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



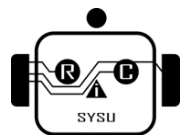
互联网

- 怎样才能更大范围的“共享资源”，实现“无处不在的计算”？
- 1969年，美国国防部ARPA(高级研究计划署)建立世界上第一个网络ARPANET，成为Internet的前生
 - Lincoln实验室的Paul Baran、Donald Davies和Lawrence Roberts发明ARPANET的包交换(Packet switching)技术
 - 包交换技术将通信链路虚拟化，从而共享通信链路，不再需要专用的电路交换
- 20世纪70年代末，Vinton Cerf和Robert Kahn发明TCP/IP协议
- 互联网已经是全球性的网络



从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



多媒体 (Multimedia)

丰富着每个人的生活

□ 人和计算机能够交流吗？

□ 媒体是承载信息的载体

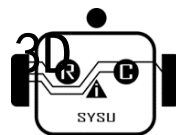
- 存储信息的媒体：磁盘、光盘、U盘等
- 传播信息的媒体：电磁波、电缆、光缆等
- 表示信息的媒体：数值、文字、声音、图像、视频等

□ 怎样才能实现人和计算机的交流呢？

□ 多媒体技术：用计算机和相关设备交互处理多种媒体信息的方法和手段

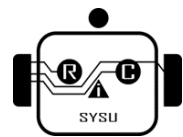
□ 多媒体的信息分类

- 文本：数字、字母、符号等
- 声音：语音、歌曲、音乐等
- 图形：由点、线、面、体组合而成的几何图形
- 图像：特指静态图像。如照片、画片等
- 视频：电视、录像等
- 动画：由多幅静态画片组合而成，产生动态效果。



从技术发展历程看电子计算机

- 计算模型：电子计算机的理论基础
- 物理器件：电子计算机的物质基础
- 大规模集成：电子计算机的工业基础
- 存储程序：电子计算机的“通用”基础
- 操作系统：电子计算机的“共享”基础
- 互联网：电子计算机的“泛在”基础
- 多媒体：电子计算机的“交互”基础
- 人工智能：电子计算机的“智能”基础



人工智能

□ 计算机能够像人一样思考吗？

□ 1950年，阿兰·图灵在《思想》杂志发表论文《计算机器与智能》，提出机器学习、遗传算法、强化学习等，以及著名的“图灵测试”概念

■ 如果机器能够通过图灵测试，它就是一个完全意义上的智能机，和人没有区别

■ 形象地描绘了计算机智能和人类智能的模拟关系

■ 图灵被誉为“**人工智能之父**”

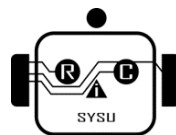
■ 回答：什么是智能计算

□ 1996年，IBM公司的超级计算机

“深蓝”战胜国际象棋世界冠军卡斯帕罗夫；2011年，“沃森”在智力竞猜电视节目《危险边缘》中击败两位人类冠军选手

□ 2016年，谷歌旗下的 AlphaGo 以

4:1 战胜世界围棋冠军、韩国职业九段选手李世石；2017年，AlphaGo 3:0 战胜柯洁

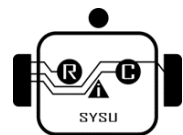


本讲内容

- 计算机系统概述
 - 技术发展历程
 - 国内计算机系统
- 计算机的基本组成
- 计算机拆解

基本要求

- 掌握计算机的基本概念
- 理解计算机中程序的执行过程



计算机简单分类

计算机应用的分类及其特征

桌面电脑(Desktop computers)

- 为个人用户设计
- 以较小的代价为单一用户提供较高的性能
- 可用来执行第三方软件



联想扬天A4600K台式机

服务器(Servers)

- 运行大规模程序的计算机，为多用户提供服务
- 通过互连网络访问，是网络的节点
- 具有可扩展性、高性能的计算能力和I/O能力



IBM X3400塔式服务器

超级计算机(Supercomputers)

- 成千上万个处理器组成
- TB级内存，PB级外存
- 用于高端的科学和工程计算



“天河一号”超级计算机

嵌入式计算机(Embedded computers)

- 集成在其他设备中的计算机，执行一个特制的任务或一组固定的软件
- 软件集成在硬件中，作为一个整体系统使用
- 控制成本和功耗，实现最低限度的性能要求



“MOXA”数据采集计算机

103型计算机(即DJS-1型, 1958)

□ 中国制造的第一台计算机

- ✓ 运行速度250次/秒
- ✓ 内存容量1KB

□ 1960年第一台通用计算机-107机

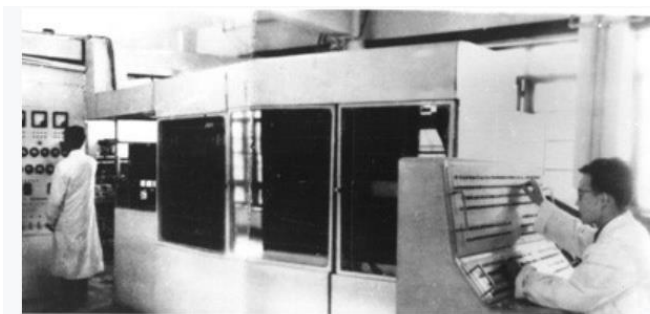
107计算机共有六个机柜，其中中央处理机、磁芯存储器 and 电源各占用两个，另外还有作为输入输出设备的五单位发报机一台、电传打印机一台和控制台一个。全机共使用电子管1280余只，功耗6KW（不包括通风），机房占地面积约60M²。107采用串行运算方式，机器主频62.5千赫，平均每秒运算250次。



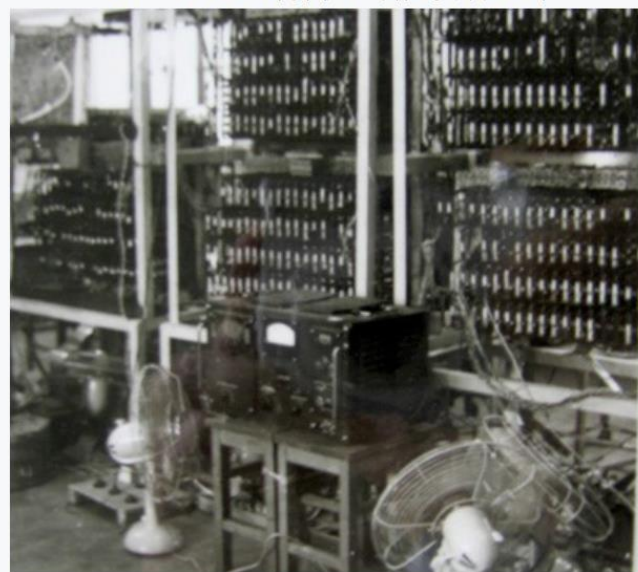
夏培肃院士



夏培肃主编的中国科大第一套《计算机原理》教材



107 (KD-1) 计算机在科大安装调试



107计算机在运行算题

2010年11月全球超级计算机TOP5

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C , NUDT National Supercomputing Center in Tianjin China	186,368	2,566.0	4,701.0	4,040
2	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz , Cray/HPE DOE/SC/Oak Ridge National Laboratory United States	224,162	1,759.0	2,331.0	6,950
3	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU , Sugon National Supercomputing Centre in Shenzhen (NSCS) China	120,640	1,271.0	2,984.3	2,580
4	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows , NEC/HPE GSIC Center, Tokyo Institute of Technology Japan	73,278	1,192.0	2,287.6	1,399
5	Hopper - Cray XE6 12-core 2.1 GHz , Cray/HPE DOE/SC/LBNL/NERSC United States	153,408	1,054.0	1,288.6	2,910

2015年11月全球超级计算机TOP5

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
2	Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray/HPE DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
3	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
4	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	11,280.4	12,660
5	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom , IBM DOE/SC/Argonne National Laboratory United States	786,432	8,586.6	10,066.3	3,945

2016年6月全球超级计算机TOP5

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray/HPE DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
4	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
5	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	11,280.4	12,660

2019年11月全球超级计算机TOP5

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
2	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
3	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
4	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	Frontera - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR , Dell EMC Texas Advanced Computing Center/Univ. of Texas United States	448,448	23,516.4	38,745.9	

2019年11月全球超级计算机TOP5

富岳

Rank	System	Cores	[TFlop/s]	[TFlop/s]	[kW]
1	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	Selene - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646
6	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482

统实验室

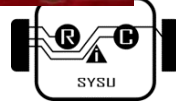
IM LABORATORY
NEED STUDY OF USTC

天河一号A（2010年）

- ❑ CPU+GPU异构体系：实测运算能力倍增至2507万亿次
- ❑ “自主可控”原则：
 - ✓ 2048颗我国自主研发的飞腾FT-1000八核处理器
 - 基于SUN公司UltraSparc T2处理器
 - ✓ 14336颗Intel Xeon（至强）X5670 2.93GHz六核处理器
 - ✓ 7168块NVIDIA Tesla M2050高性能计算卡（\$2500/块）

中国研制成功每秒运算逾千万亿次超级计算机

「天河一号」	■ 全系统峰值性能	1206万亿次/秒
	■ Linpack实测性能	563.1万亿次/秒
	■ 共享存储总容量	1PB
	■ 全系统包含通用处理器(CPU)	6144个
	■ 全系统包含加速处理器(GPU)	5120个
	■ 互联通信网络的单根线传输速率	10Gbps
	■ 目前投资	6亿人民币
	■ 使用寿命预计	10年
	■ 全系统运行情况下耗电	1280度/小时



神威·太湖之光（2016）

□ 2016年6月，Sunway Taihu Light 超算服务器在Lin基准测试中，以93 petaflop/s（注：petaflop：每秒执行1千万亿次浮点运算）的性能指标成为TOP500超算计算机列表的第一名。

Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway

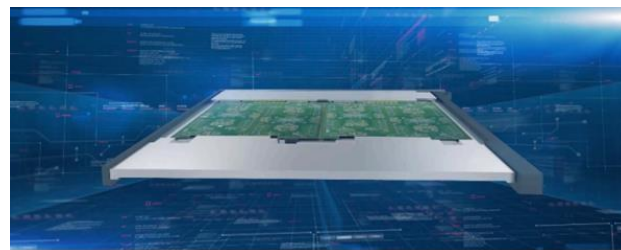
Site:	National Supercomputing Center in Wuxi
Manufacturer:	NRCPCC
Cores:	10,649,600
Linpack Performance (Rmax)	93,014.6 TFlop/s
Theoretical Peak (Rpeak)	125,436 TFlop/s
Nmax	12,288,000
Power:	15,371.00 kW
Memory:	1,310,720 GB
Processor:	Sunway SW26010 260C 1.45GHz
Interconnect:	Sunway
Operating System:	Sunway RaiseOS 2.0.5



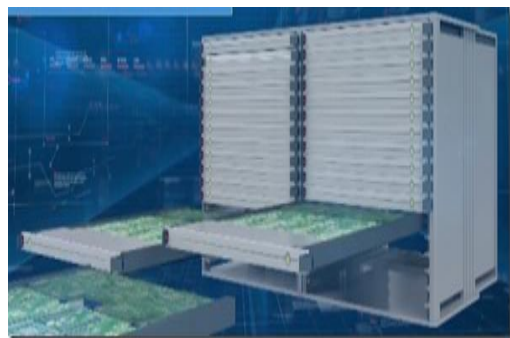
单节点260核



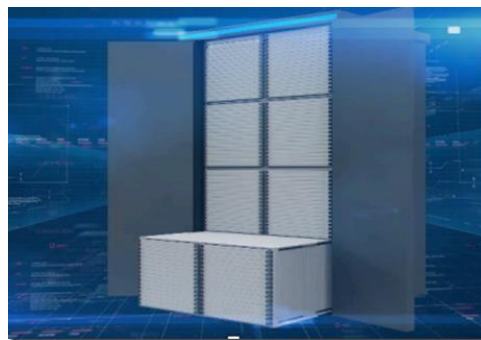
1个卡上2个节点



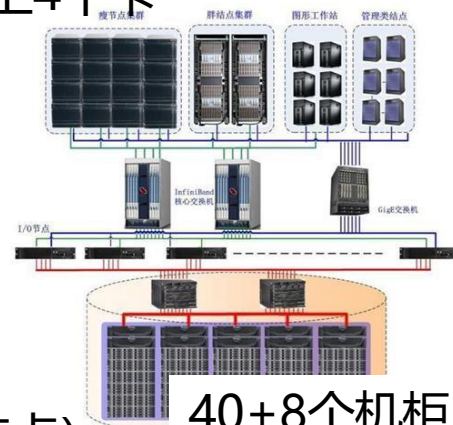
一个板上4个卡



一个超级节点32个板（256节点）



一个机柜4个超级节点（1024节点）

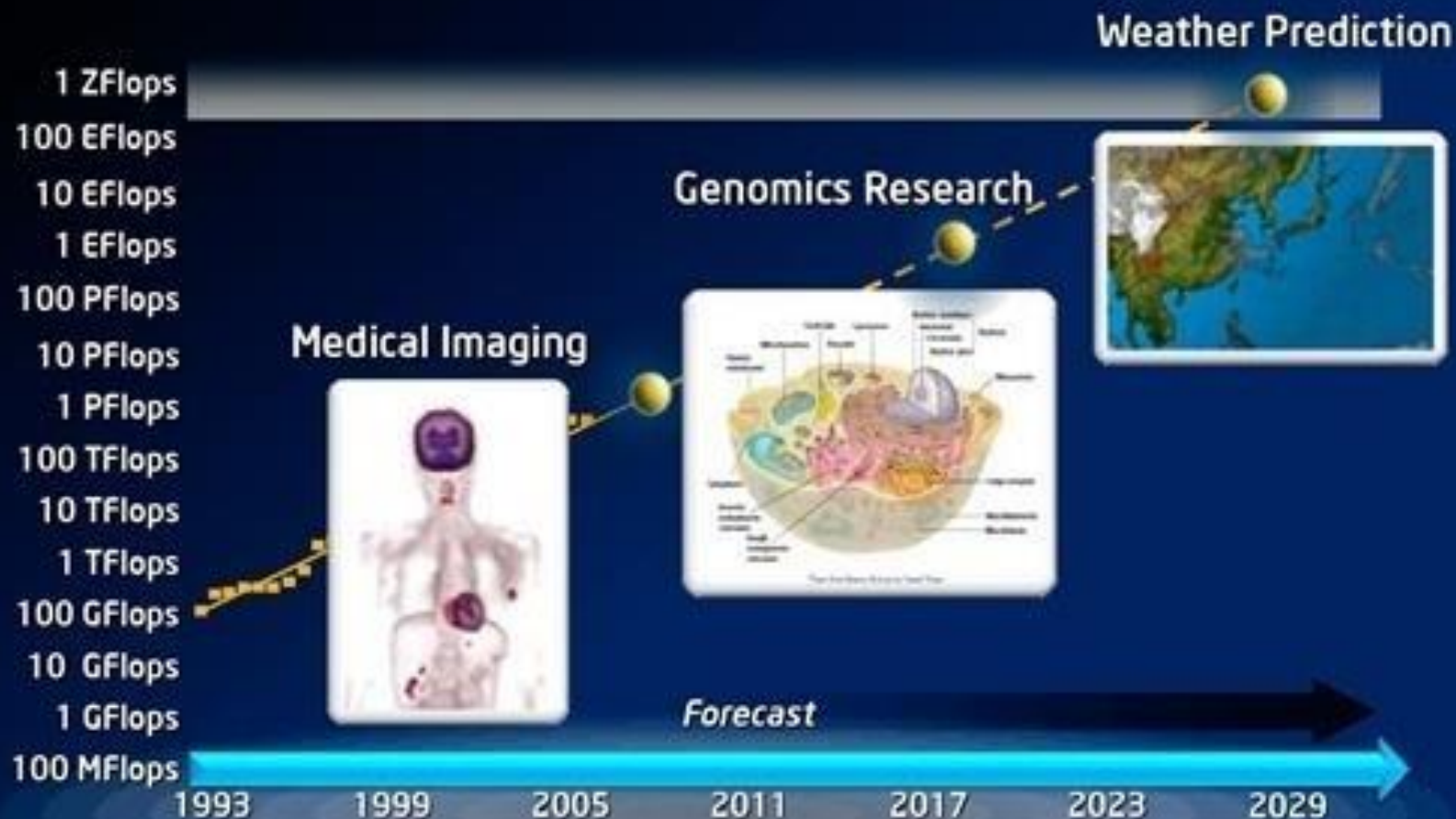


40+8个机柜



An Insatiable Need For Computing

Intel: 2020年前实现百亿亿次超算



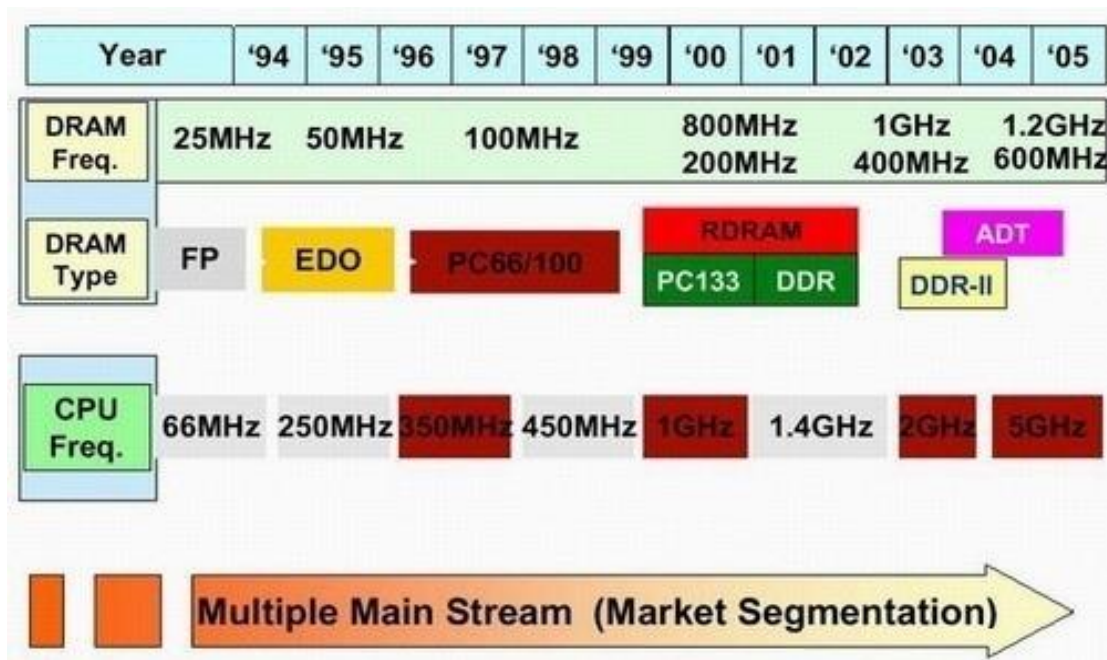
*Exascale Problems Cannot Be Solved Using the
Computing Power Available Today*

计算机科班出身应该关注什么

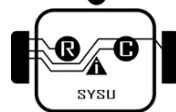
□ 成功的程序员关心其所写的程序性能如何

■ 上世纪60~70年代，计算机性能受制于内存容量。

程序员尽可能少地用内存提高程序执行速度



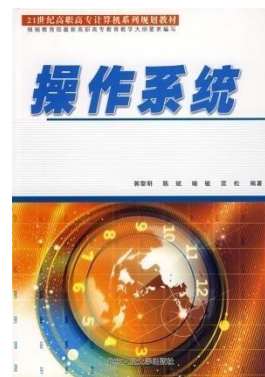
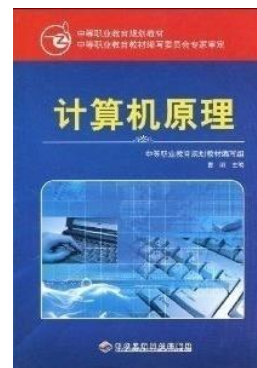
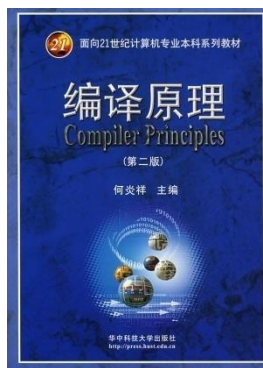
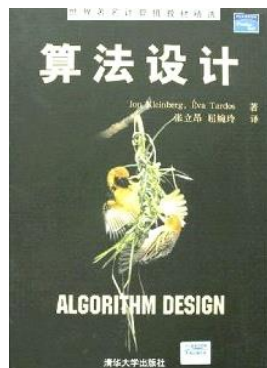
■ 现代，程序员需要理解的是内存的层次化特性和处理器的并行化特点



计算机科班出身应该关注什么

□ 成功的程序员关心其所编写的程序性能如何

- 决定程序性能的主要因素
 - 程序中使用的算法——涉及数据结构、算法设计
 - 创建程序并翻译成机器指令的软件——涉及编译原理
 - 计算机各部件执行效率——涉及计算机组成原理、操作系统



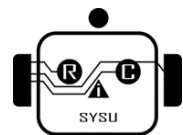
计算机科班出身应该关注什么

□ 如何理解程序性能

软硬件部件	如何影响性能	涉及的章节
算法(Algorithm)	决定源程序语句数量及执行I/O操作数量	参考相关资料
程序设计语言、编译器(Compiler)和体系结构(Architecture)	决定每一条源程序语句所对应的机器指令	第2章
处理器(Processor)、存储器系统(Memory System)	决定指令的执行速度	第3、4、5章
I/O系统(硬件和操作系统)	决定I/O操作的执行速度	第6章

本讲内容

- 计算机系统概述
 - 技术发展历程
 - 国内计算机系统
- 计算机的基本组成
- 计算机拆解



计算机软硬件

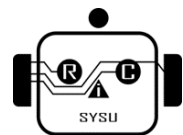
□ 计算机系统

- ✓ 一个由“硬件 + 软件”组成的复杂的自动化电子设备
- ✓ 包含硬件系统和软件系统两大部分
 - 硬件：计算机的实体部分，由看得见摸得着的各种电子元器件，各类光、电、机设备的实物组成
 - 软件：由人们事先编制的具有各类特殊功能的程序组成，看不见摸不着，一般通过各类媒介（如光盘、磁盘等）存放

□ 硬件与软件的关系

- ✓ 硬件是躯体，是物质基础
- ✓ 软件是灵魂，是硬件功能的完善和补充

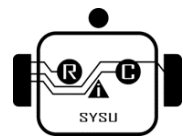
计算机性能的好坏取决于“软”、“硬”件功能的总和



硬件与软件的逻辑等价性

□ 计算机硬件和软件的逻辑等价

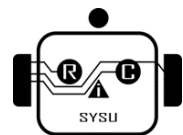
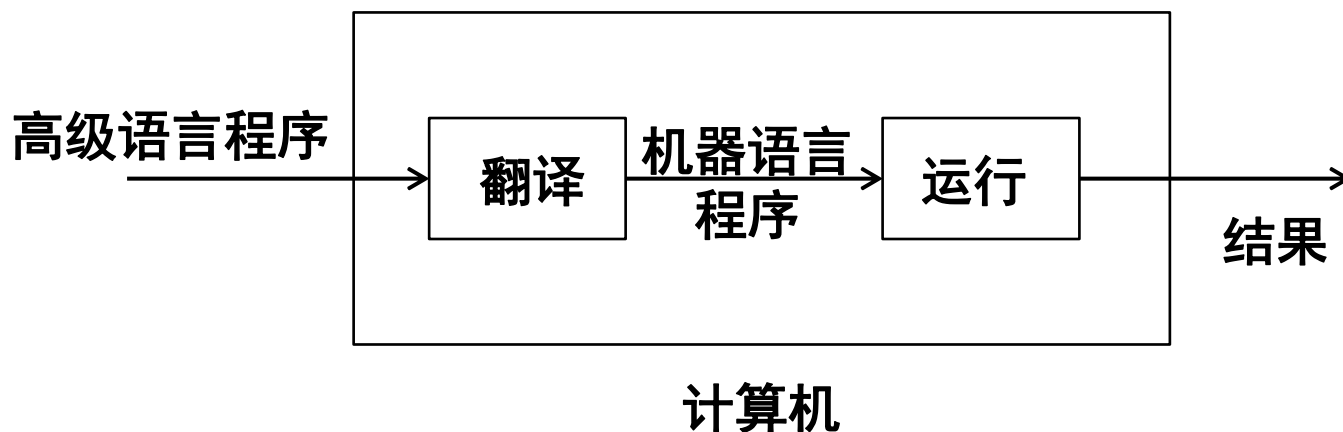
- ✓ 计算机中，许多功能既可以直接由硬件实现，也可以在基本硬件的支持下利用软件来实现
 - 如乘法运算，可以使用硬件乘法电路实现，也可以用乘法子程序实现
- ✓ 硬件和软件间的功能分配，随技术发展而变化，没有固定的模式
 - 取决于设计目标、系统性价比以及当时的技术水平
 - 早期，**“硬件软化”**，降低计算机造价
 - 集成电路技术的发展，**“软件硬化”**
- ✓ 软件固化：将软件的一些功能固化地保存在只读存储器中，称为**固件**
 - 功能上是软件，形态上是硬件
 - 一些系统软件的核心部分（如操作系统内核、常用软件的固定部分）常被固化在存储芯片中



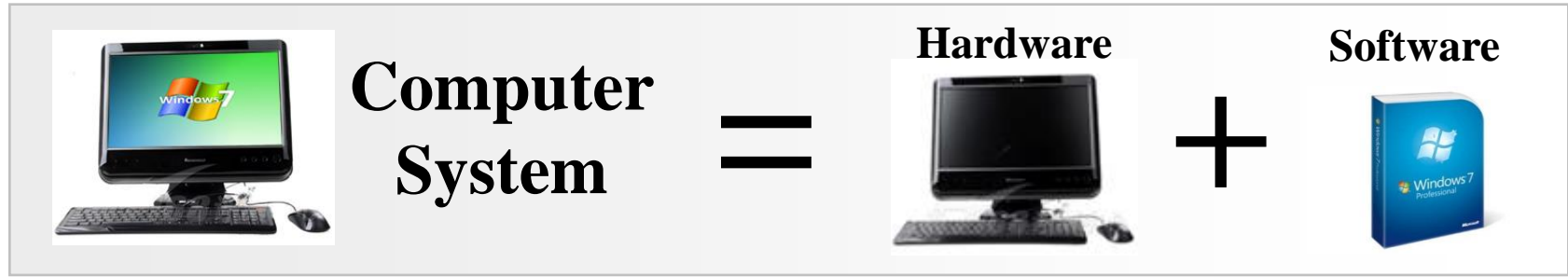
运行过程

□ 现代计算机的解题处理过程

- ✓ 首先，用户用高级语言编写程序，描述问题的求解过程
- ✓ 然后，将程序与数据一起送入计算机中，由计算机将其翻译成机器语言程序
- ✓ 最后，计算机自动运行机器语言程序，输出计算结果



计算机软硬件



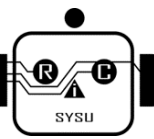
硬件
Hardware

tangible objects , 具体的物理器件
e.g integrated circuit, printed circuit board,
cables, power supplies, memories, printer

软件
Software

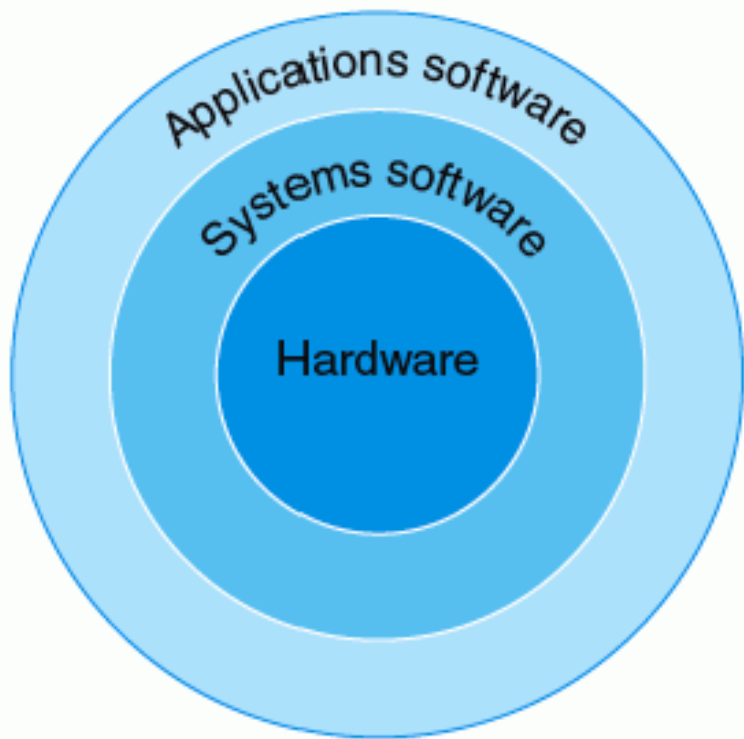
encoded symbol , 对抽象概念进行符号编码
Software = program + document
program = algorithms + data
program 是由若干条指令或语句组成的

我们对计算机讲什么语言，可以让计算机理解呢？



程序的表象之下

- 简单的软硬件层次化结构



- 隐藏低层层次的实现细节，
简化各层次上用户的使用

- 每一层次都为上一层隐藏了自己的技术细节——“抽象”

一个典型的应用程序，需要经过几个软件层次，才能将我们人类复杂的操作逐步翻译成简单的不同计算机指令的组合

计算机软件

- 系统软件：提供公共服务的程序

操作系统

管理和控制计算机软、硬件资源的程序。是用户程序和计算机硬件的接口

功能：处理基本的输入输出操作、分配存储空间、为多个程序同时使用计算机提供支持



编译器

将高级语言语句翻译成与硬件特征有关的汇编语言语句的程序



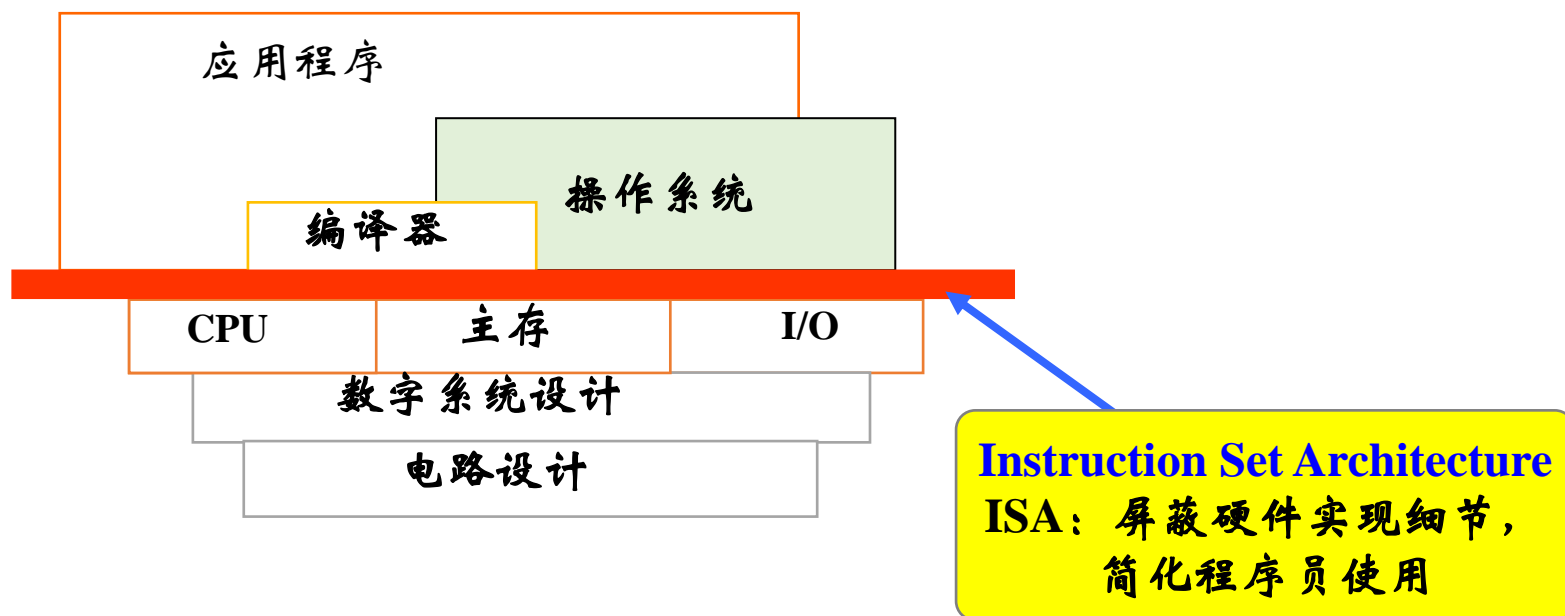
汇编器

将汇编语言中的符号指令翻译成计算机能够识别的二进制指令的程序

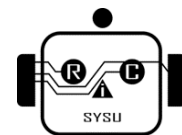


计算机的层次结构

Computer Hierarchy (计算机的层次结构)

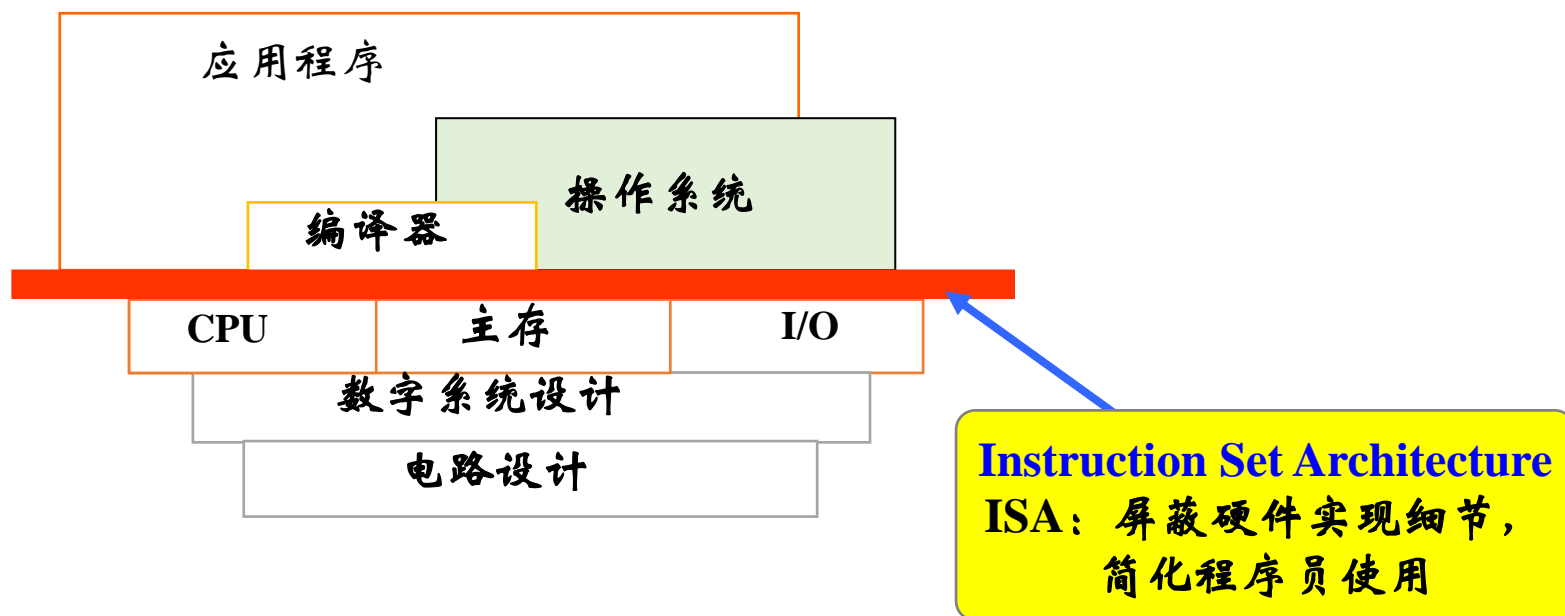


■ 计算机系统层次结构中, 指令系统是软/硬件的交界面

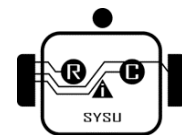


计算机的层次结构

Computer Hierarchy (计算机的层次结构)

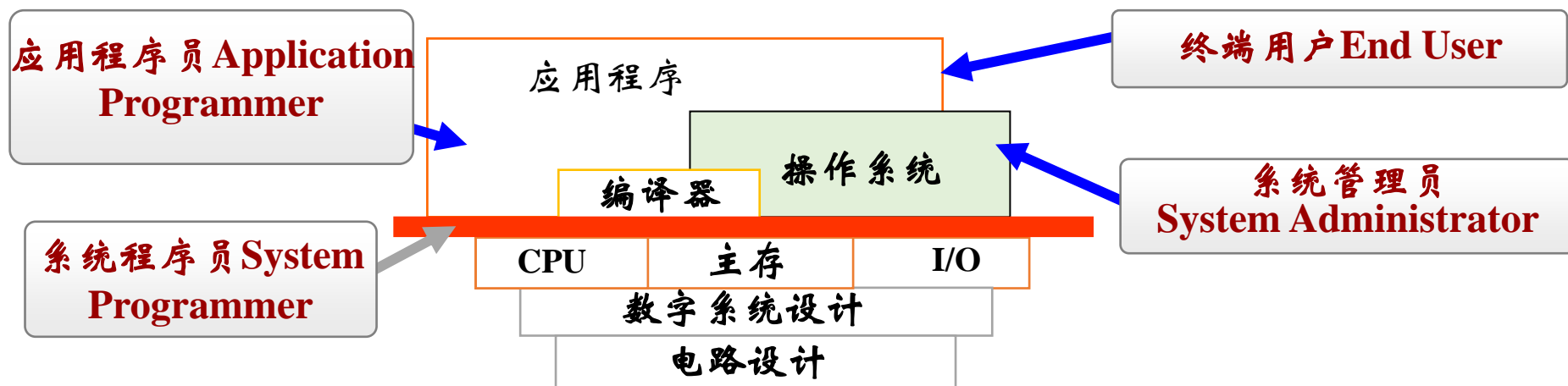


- 计算机系统层次结构中, 指令系统是软/硬件的交界面
- 不同用户工作在不同层次, 看到的计算机是不一样的

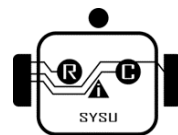


计算机的层次结构

Computer Hierarchy (计算机的层次结构)



- 计算机系统层次结构中，**指令系统是软/硬件的交界面**
- 不同用户工作在不同层次，看到的计算机是不一样的
- 计算机的功能和性能是由机器的指令系统集中体现出来



从程序到电子信号

□分析：如何使用计算机求解一个问题？

□例：计算整数18和40的和

- 需要将我们对该问题的求解要求通过**输入设备**告诉计算机；
- 计算机调用相应的**运算部件**对问题进行求解；
- 并将最后的计算机结果从**输出设备**输出



输入设备



中央处理器



输出设备

从程序到电子信号

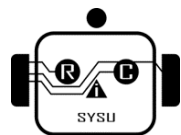
□ 例1.1：计算整数18和40的和

□ 问题的高级语言描述

- 高级语言包括C语言、Pascal、Java、Basic等
- 例1.1的C语言描述 (*sum.c*)

```
#include <stdio.h>
int main()
{
    int    c;
    c = 18 + 40;
    printf("result is : %d\n", c);
}
```

当执行这个程序时，
计算机内部发生了什么
&为什么？



从程序到电子信号

□ 例1.1：计算整数18和40的和

□ 问题的高级语言描述

□ 高级语言包括C语言、Pascal、Java、Basic等

□ 例1.1的C语言描述 (*sum.c*)

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int    c;
```

```
    c = 18 + 40;
```

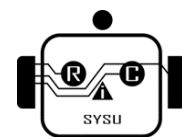
```
    printf("result is : %d\n", c);
```

```
}
```

预处理阶段



编译阶段



从程序到电子信号

□ (1) 高级语言转换成汇编语言

□ gcc -S sum sum.c

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int    c;
```

```
    c = 18 + 40;
```

```
    printf("result is : %d\n", c);
```

```
}
```

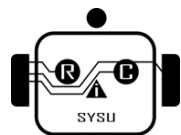
```
...  
DW    C    EQU    ?
```

```
MOV    A,    12H
```

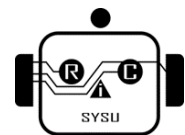
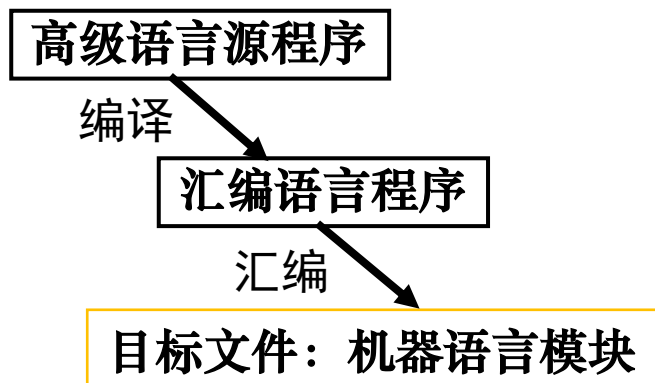
```
ADD    A,    28H
```

```
MOV    C,    A
```

```
...
```



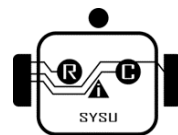
从程序到电子信号



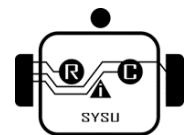
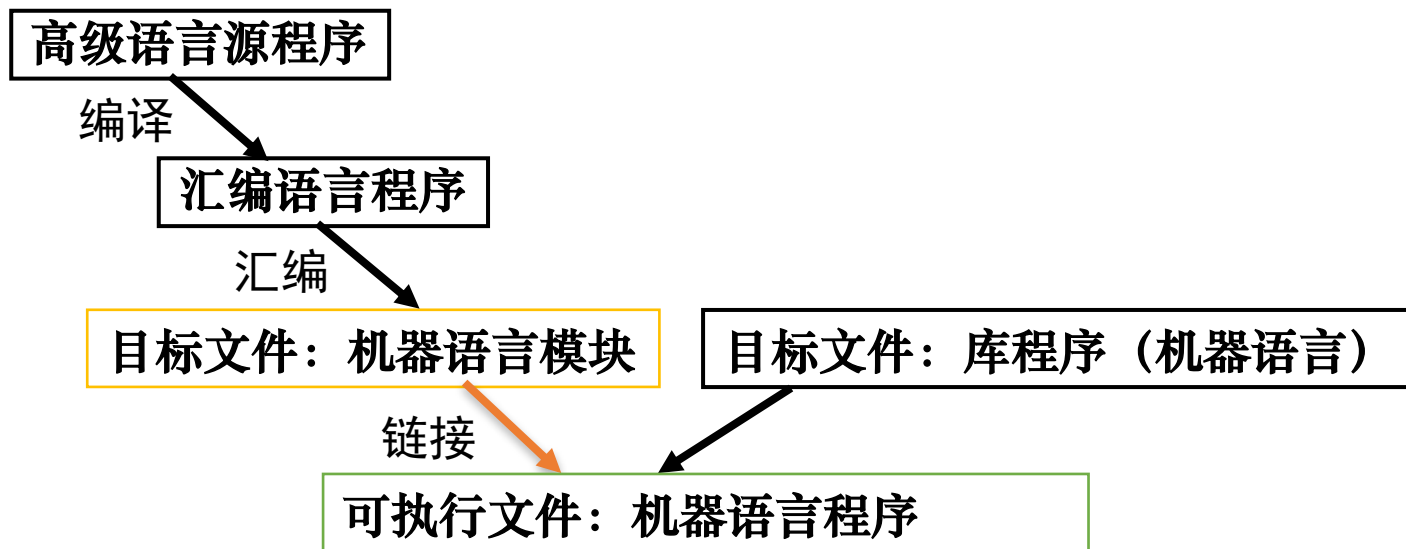
从程序到电子信号

■ (2) 汇编语言转换为机器语言

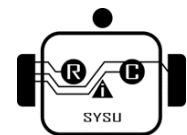
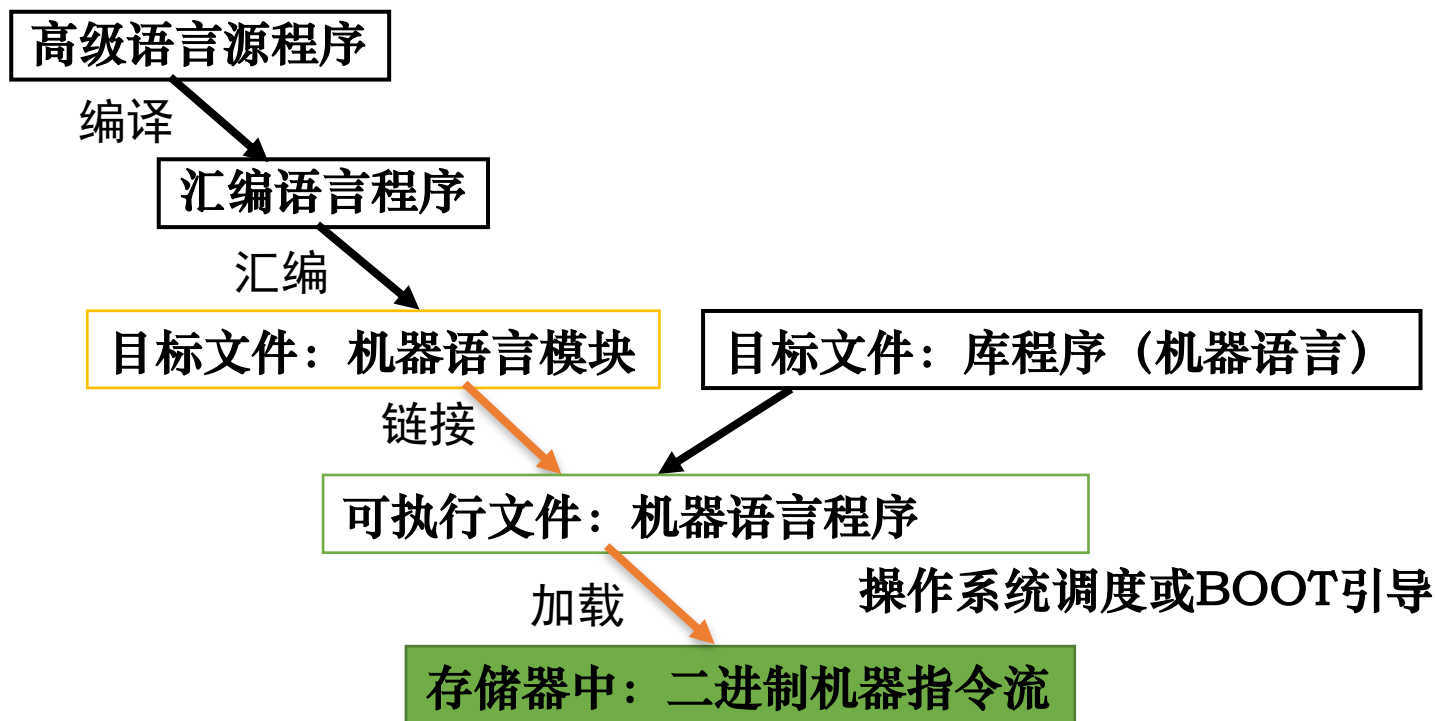
主存地址	指令或数据		说明
00000	B8	12H	取数: $A \leftarrow 12H$
00002	05	28H	加法: $A \leftarrow A + 28H$
00004	A3	30H	存数: $(30H) \leftarrow A$
00006	F4		停机
.....		
00000	初始数据区		...
00B02			DW C EQU ?
00030			MOV A, 12H ADD A, 28H MOV C, A
	3AH	结果数据	...



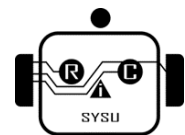
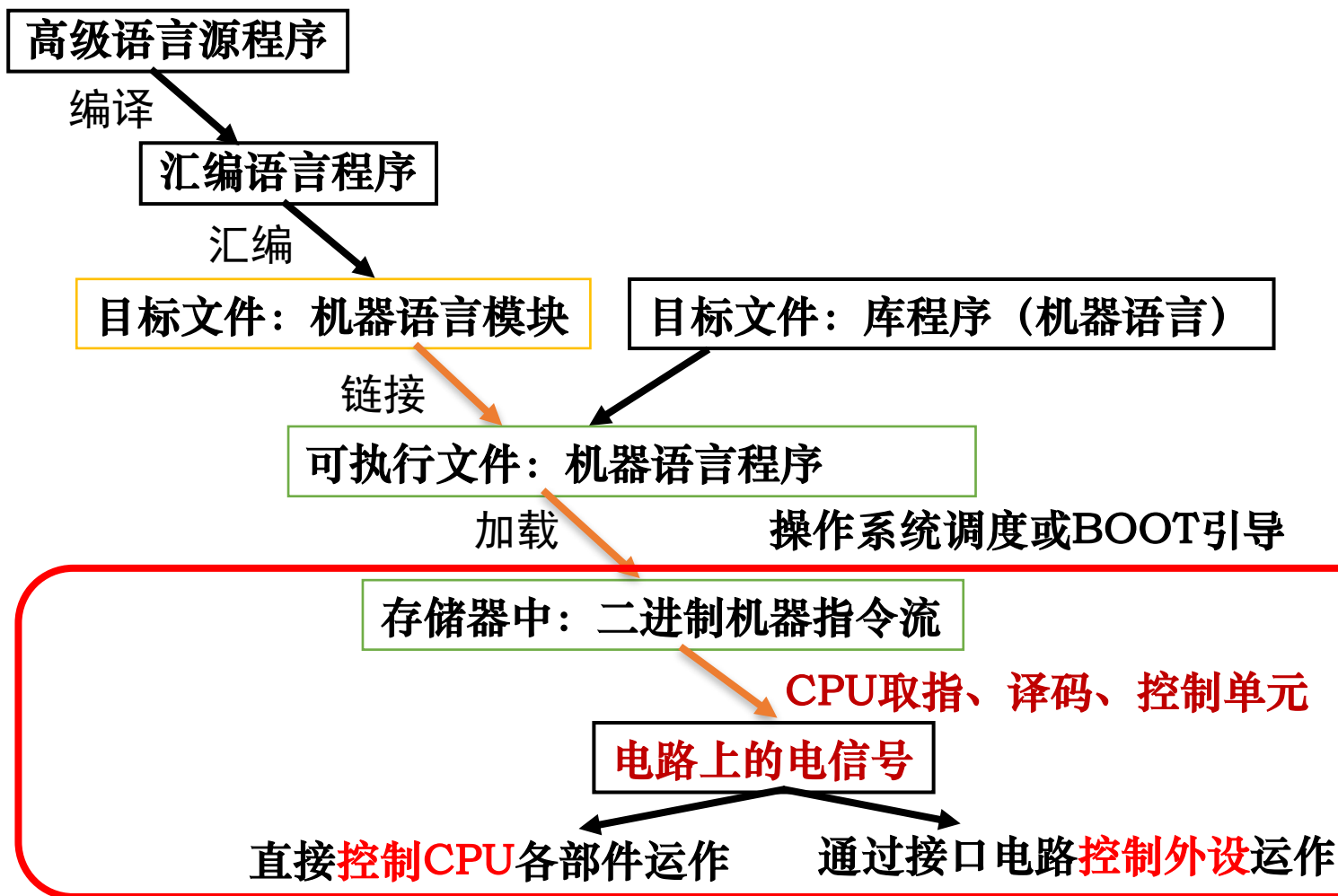
从程序到电子信号



从程序到电子信号



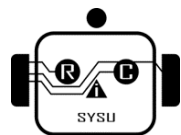
从程序到电子信号



从程序到电子信号

□ 解题过程的解释

- ✓ 把原始数据及解题步骤记录在纸上，即在纸上“存储”了解题的原始信息，纸作为具有记忆功能的部件——**存储器**
- ✓ 对数据进行了加、减、乘等算术运算，人脑作为具有计算功能的部件——**运算器**
- ✓ 用笔将解题信息写在纸上或将运算结果写出来——**输入/输出设备**
- ✓ 整个过程在人脑的控制下有序进行——**控制器**



从程序到电子信号

□ 计算机硬件的五大组成部件

✓ 运算器

- 完成算术运算和逻辑运算，暂存中间结果

✓ 存储器

- 存放数据和程序

✓ 控制器

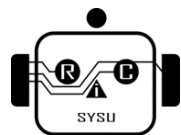
- 控制、指挥程序和数据的输入、运行以及处理运算结果

✓ 输入设备

- 将人们熟悉的信息形式转换成机器能识别的形式

✓ 输出设备

- 将机器运算结果转换为人们熟悉的信息形式展示出来



冯·诺依曼计算机组成结构

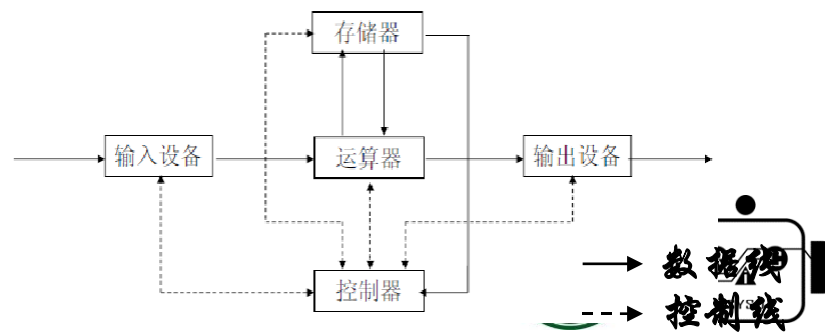
□ 冯·诺依曼机

✓ 1945年，冯·诺依曼在研究EDVAC时提出“存储程序”概念，以此概念为基础研制的计算机统称为冯·诺依曼机



✓ 特点：

- 五大组成部件，以运算器为中心
- 数据和指令用二进制数表示，以同等地位存放于存储器中，按地址访问
- 指令由操作码和地址码组成，在存储器中按顺序存放



冯·诺依曼结构 VS. 哈佛结构

□ 冯·诺依曼结构也叫做普林斯顿结构

□ 哈佛结构

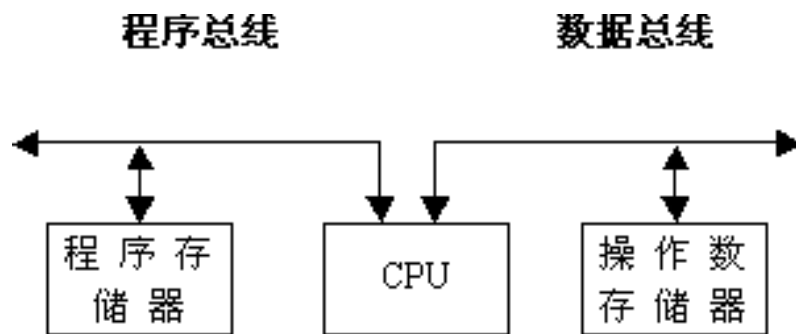
✓ 将指令和数据分开存储的结构

□ 两者的主要区别

✓ 指令和数据是否分别存储

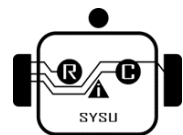
✓ 是否使用两条独立的总线，分别作为CPU与每个存储器间的专用通信通道，两条总线间毫无关联

✓ 哈佛结构目前较多出现在嵌入式应用中

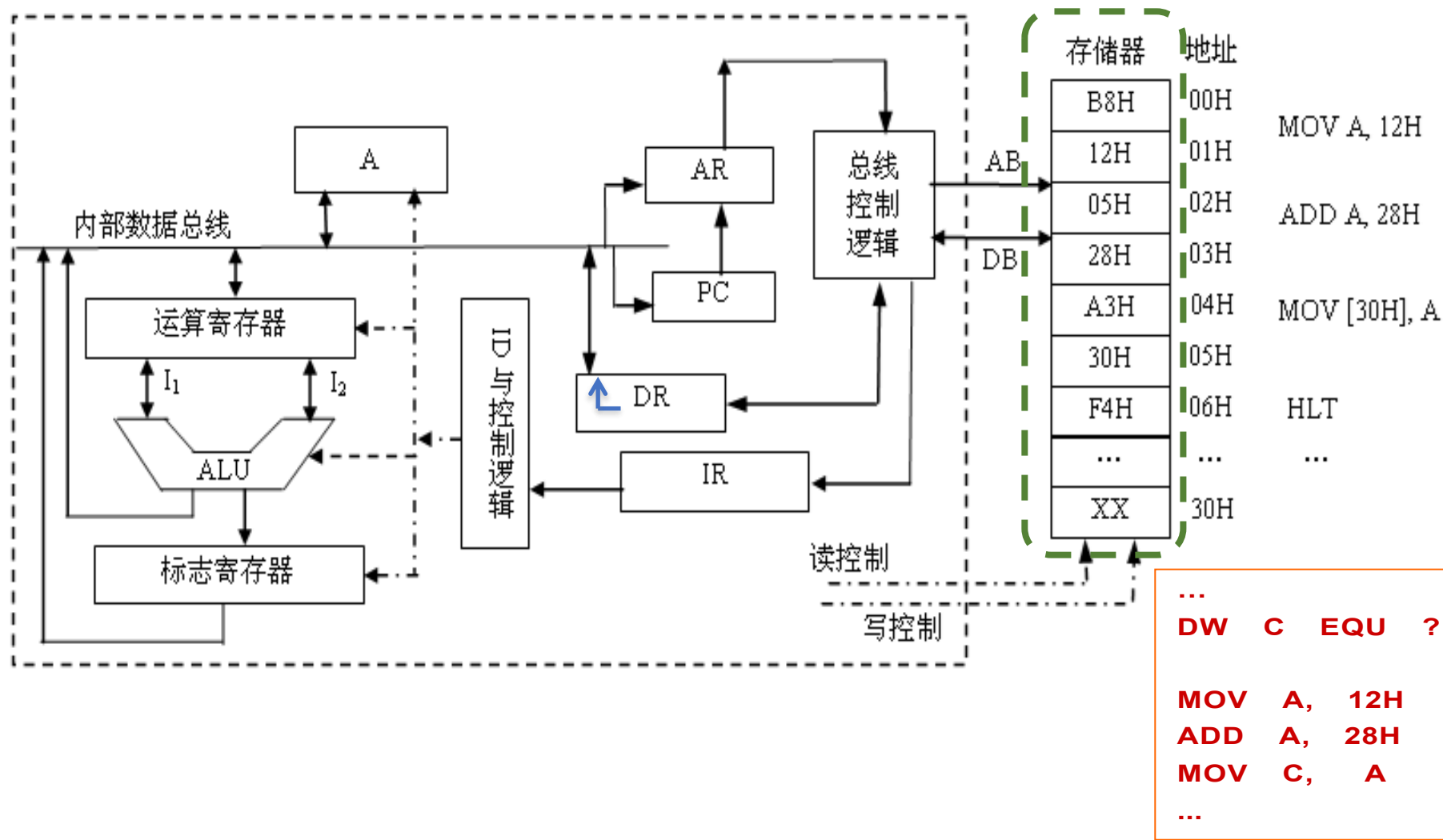


从程序到电子信号

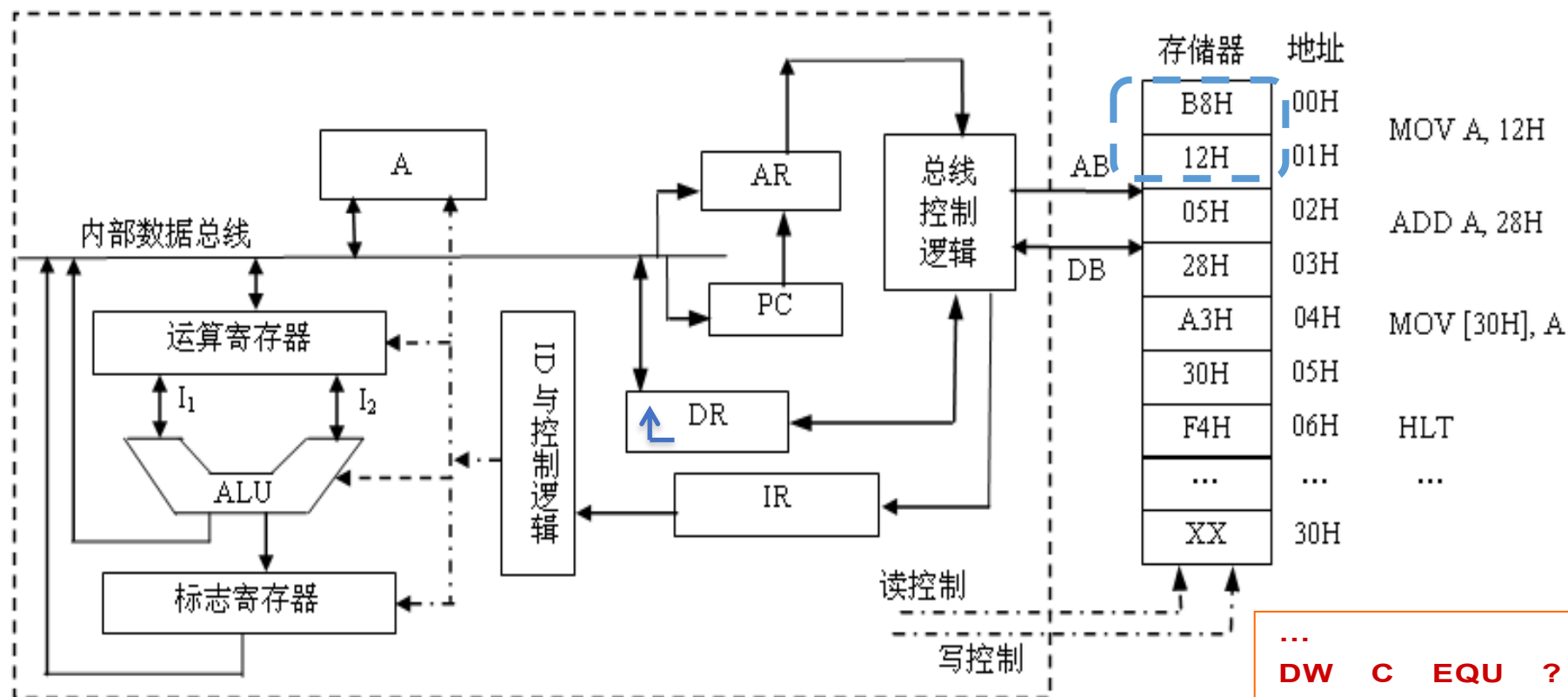
- 计算机内部工作过程：**逐条执行**加载到**内存中的二进制机器指令流**的过程。
- 一条指令的执行过程可简单地分为两个操作阶段：
 - **取指阶段**，CPU从内存中读取指令，程序计数器**保存**要被取出的**下一条指令的地址**，除非遇到跳转指令等情况，否则，PC一般都是在每次取指后加上一个增量（当前指令的字节数）；
 - **执行阶段**，对取出的指令先**译码**，解释指令的功能，然后执行译码好的指令，这期间可能会读写存储器或端口来获取操作数或者存放结果。
- 程序的执行过程就是周期性和重复性地进行取指令和执行指令两个操作。



计算机系统基本执行结构



计算机系统基本执行结构

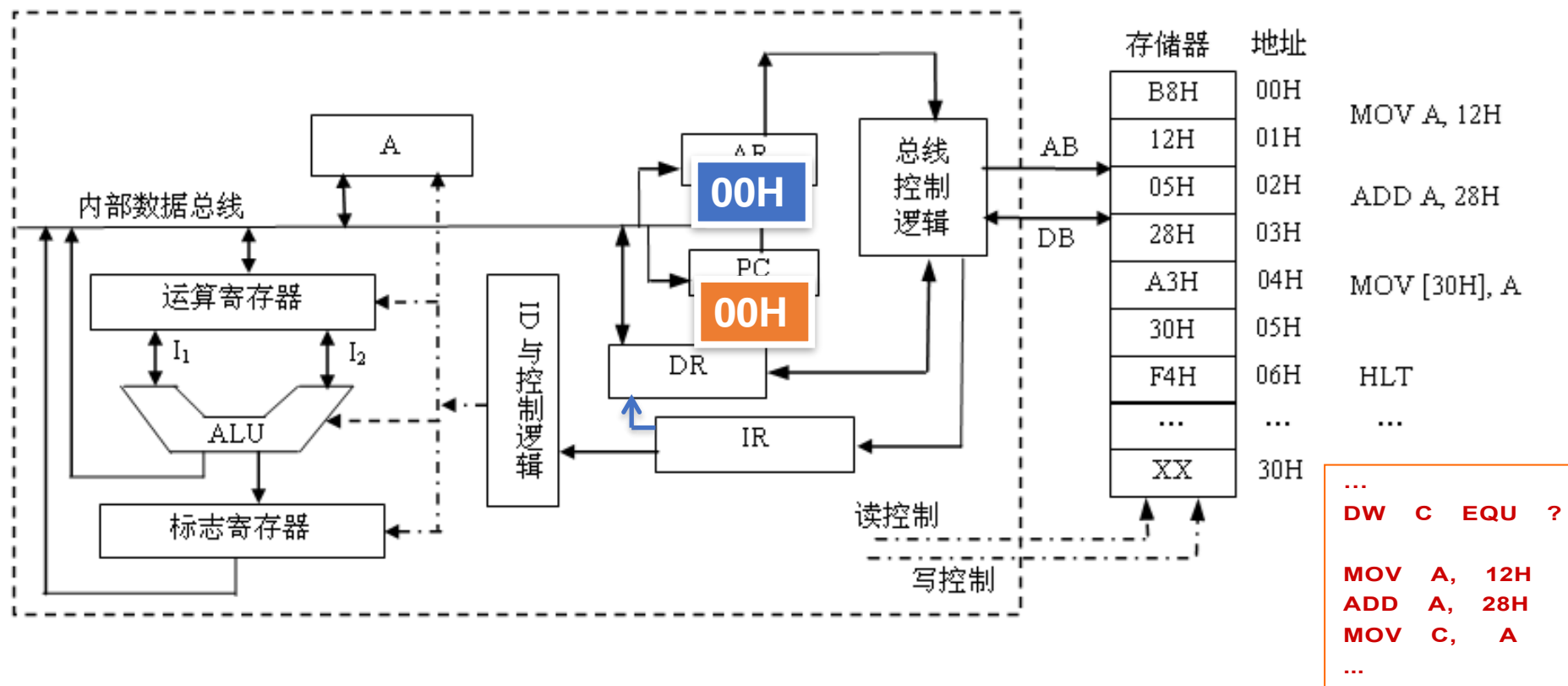


例：16位指令，存放在两个字节单元

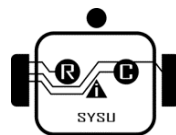
...
DW C EQU ?

MOV A, 12H
ADD A, 28H
MOV C, A
...

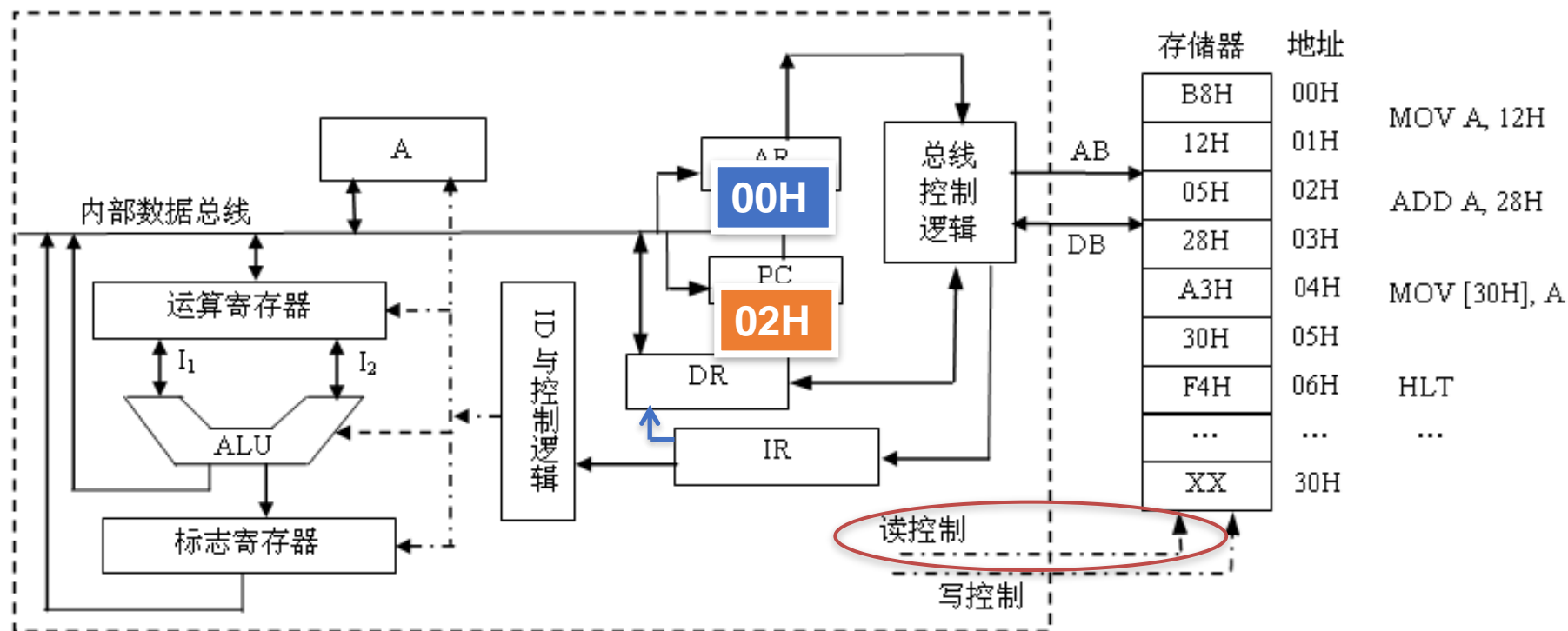
计算机系统基本执行结构



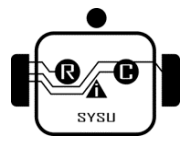
例：16位指令，存放在两个字节单元



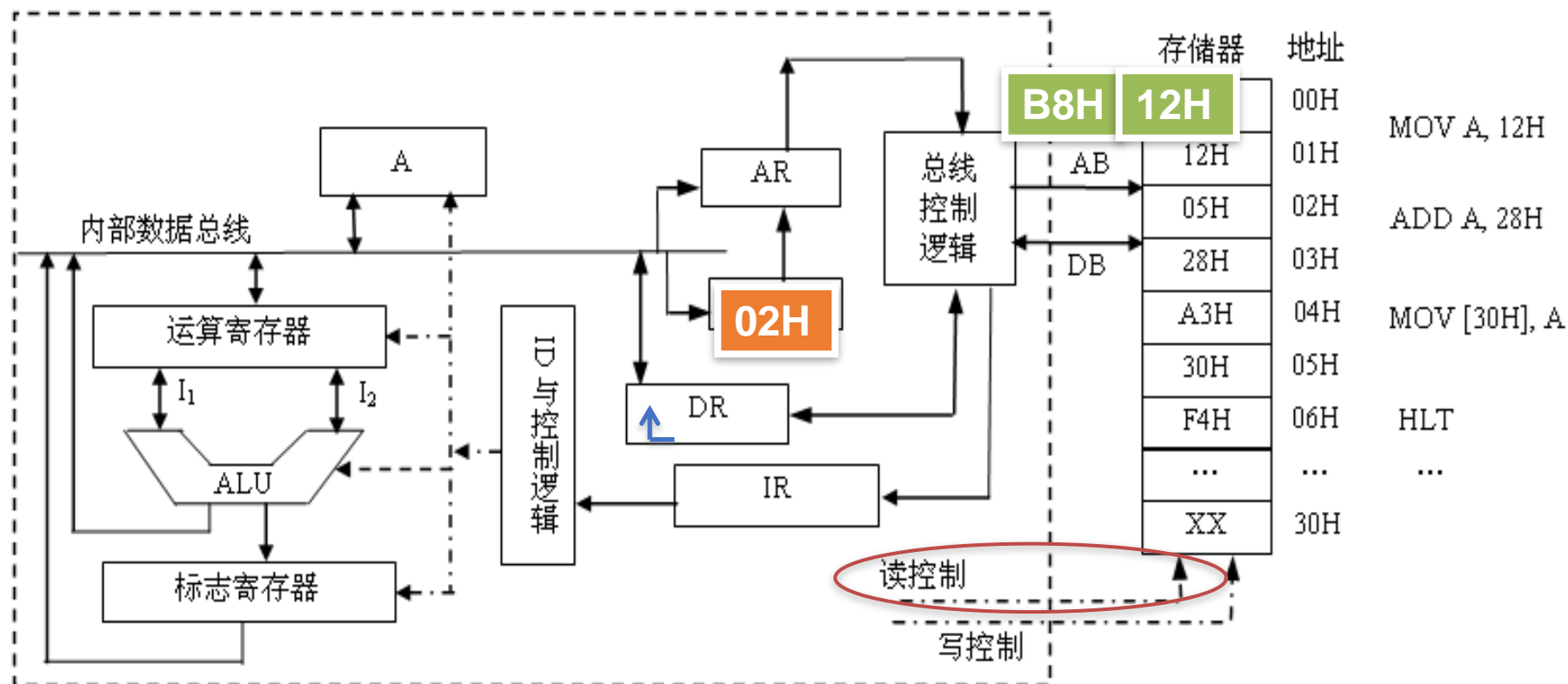
计算机系统基本执行结构——取指过程



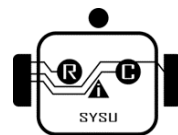
例：16位指令，存放在两个字节单元



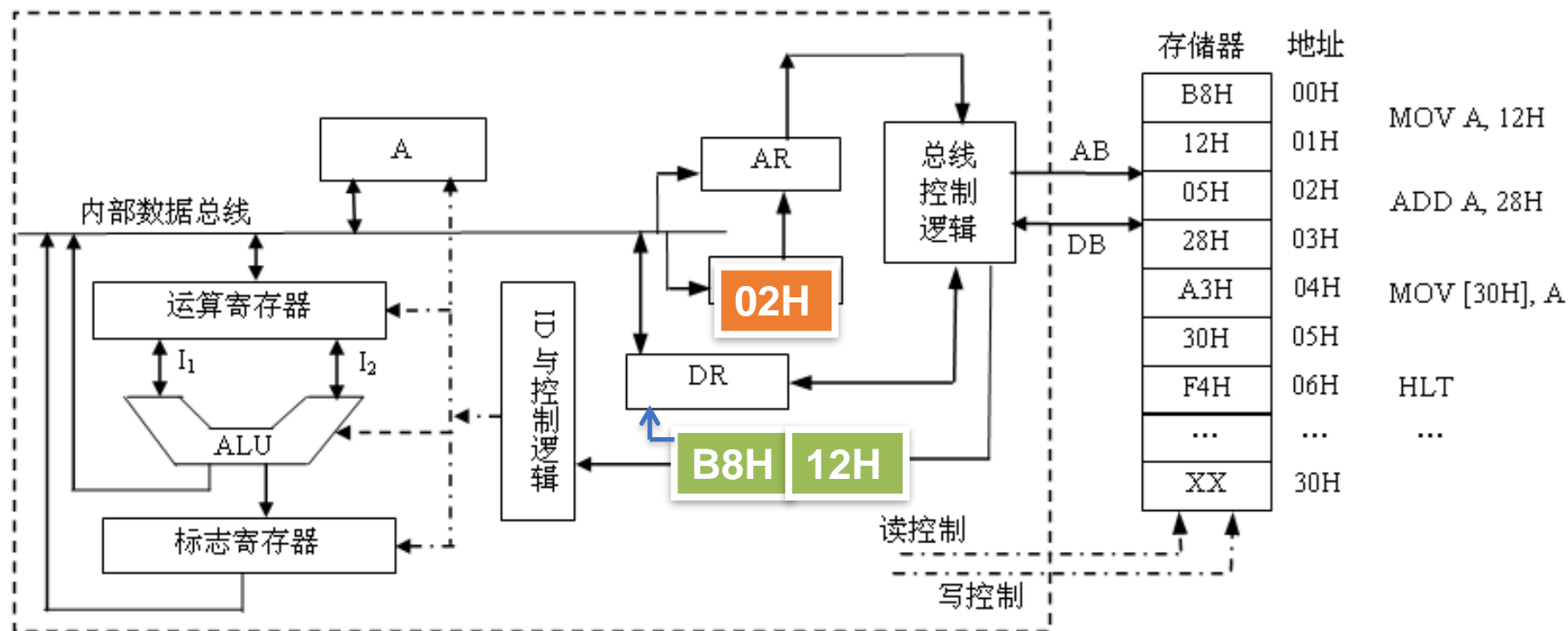
计算机系统基本执行结构——取指过程



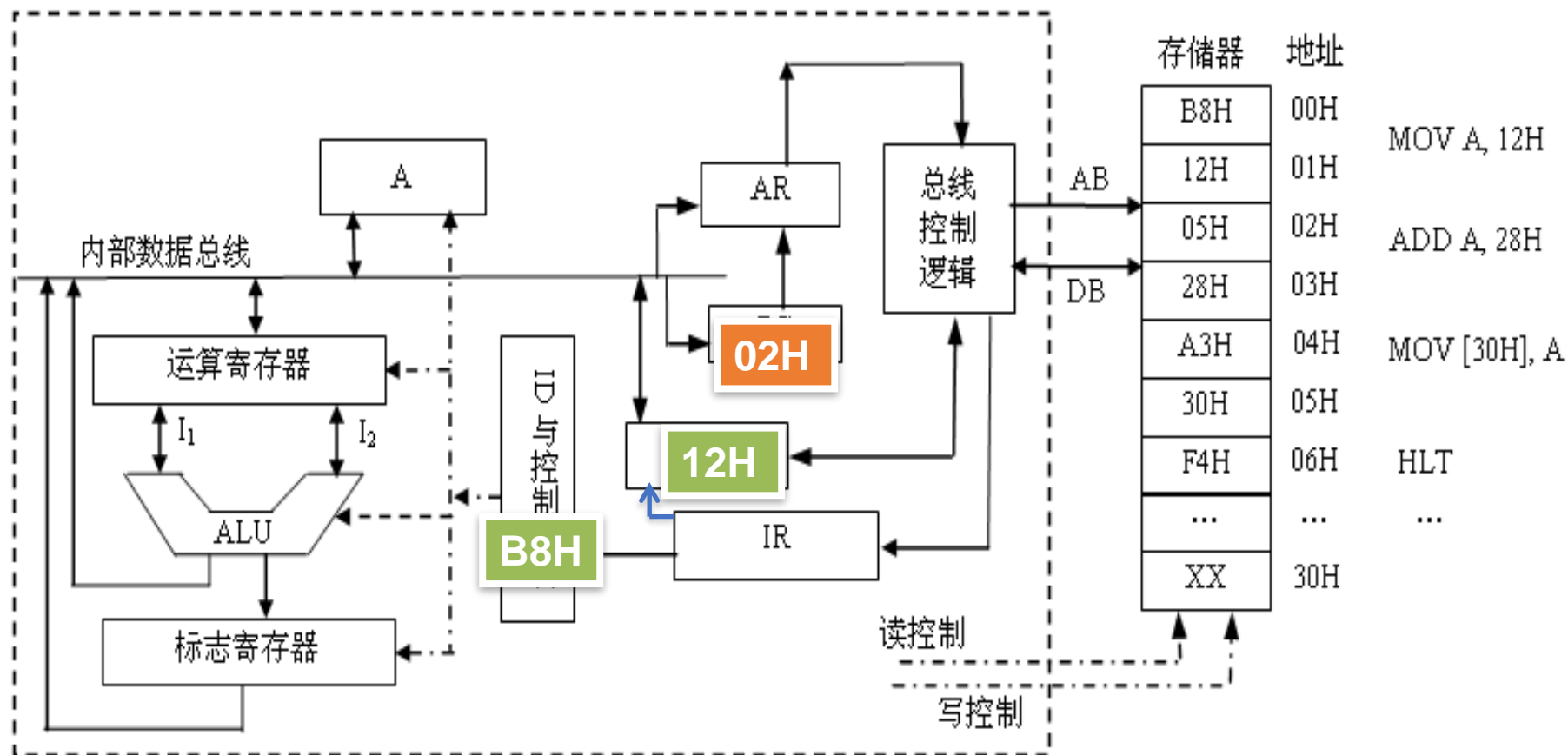
例：16位指令，存放在两个字节



计算机系统基本执行结构——执行过程



计算机系统基本执行结构——执行过程



程序的表象之下

□ 一个典型程序的转换处理过程

经典的 “hello.c” C-源程序

hello.c的ASCII文本表示

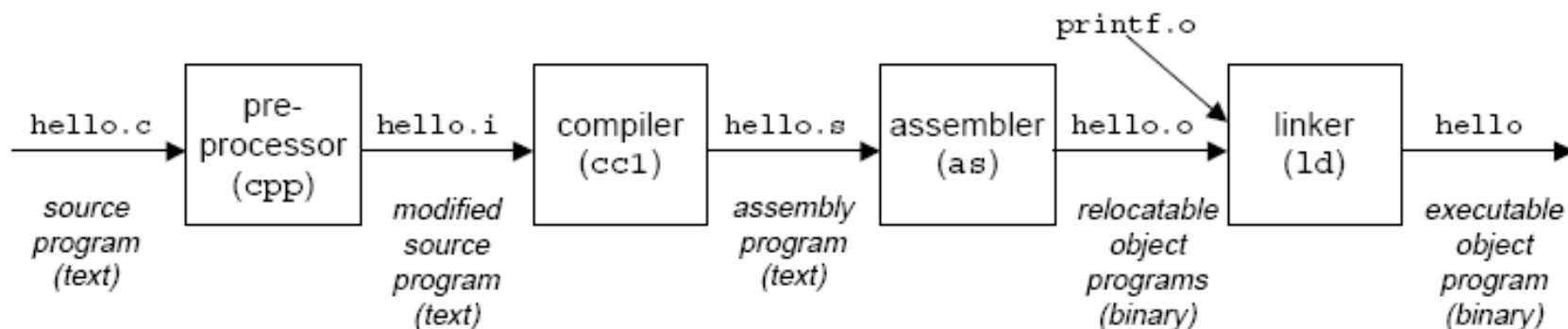
```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }
```

```
# 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
```

程序的功能是：
输出“hello,world”

计算机能够直接识别hello.c源程序吗？

不能，需要转换为机器语言代码！即：编译



程序的执行过程

● 从hello程序说起

经典的hello.c 源程序:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }
```

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
```

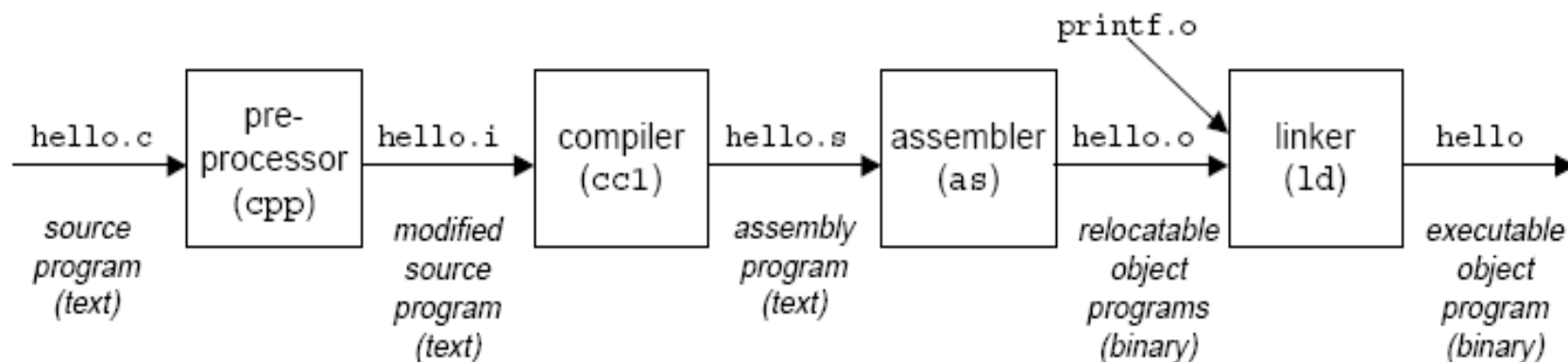
- 由ASCII字符构成的文件，被称为**文本文件**
- 所有其他文件都被称为**二进制文件**
- 系统中所有信息都是由一串**位(bit)**表示的，8个位被组织成一组，称为**字节(Byte)**
- 区分不同数据对象的唯一方法是读这些数据对象的**上下文**

问题：计算机能够直接识别hello.c源程序吗？

程序被翻译成不同的格式

● 看hello程序的生命周期

C语言程序需要由编译系统转化为机器语言代码!



■ 从hello.c到目标文件的转化是由编译系统完成

`unix>gcc -o hello hello.c`

■ 编译系统是由：预处理器、编译器、汇编器和链接器构成

■ 翻译过程是由：预处理、编译、汇编和链接这四个阶段完成

hello程序的执行过程

■ Unix 操作系统启动hello的shell命令行:

```
unix> ./hello [Enter]
```

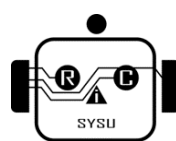
```
hello, world
```

```
unix>
```

shell是一个命令行解释器。输出一个提示符，等待输入一个命令行，然后执行这个命令

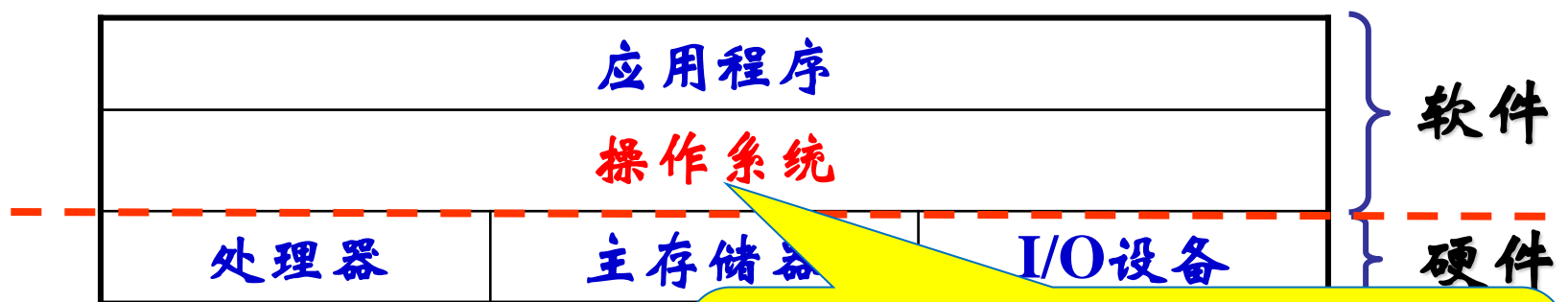
hello程序被启动后，计算机的动作过程如下：

- ① shell程序读取字符串“./hello”中字符到寄存器，然后存放到主存
- ② “Enter”键输入后，shell根据主存中的字符串“hello”到磁盘上找到特定的hello目标文件，将其包含的指令代码和数据(“hello, world\n”)从磁盘读到主存
- ③ 处理器从hello主程序的指令代码开始执行
- ④ hello程序将“hello, world\n”串中的字节从主存读到寄存器，再从寄存器输出到显示器上

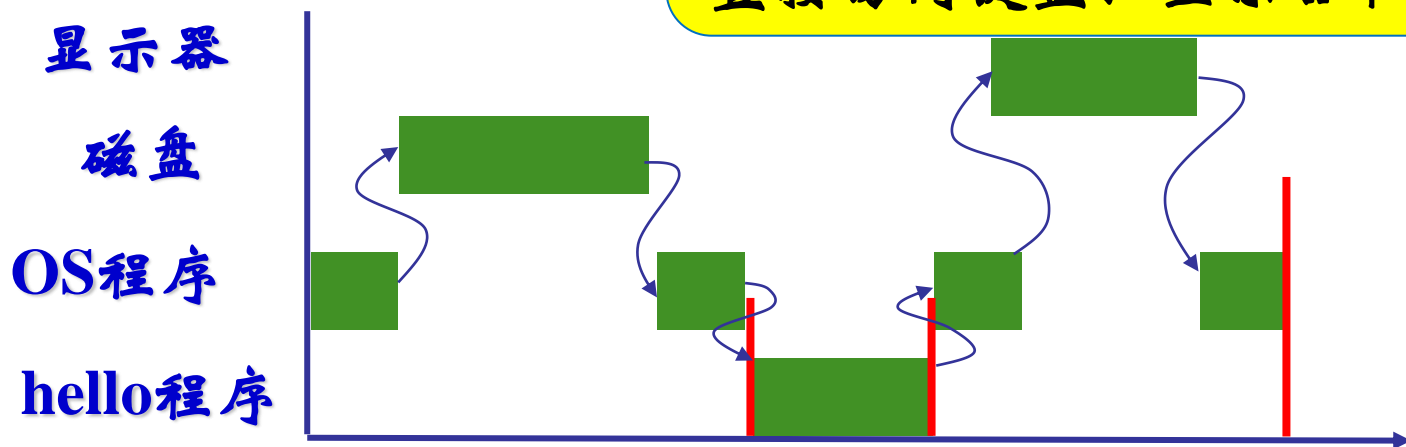


操作系统管理硬件

□ 计算机系统的分层视图



□ 理解操作系统的基本原理



操作系统提供的抽象表示

□ 操作系统的基本功能

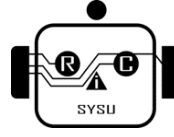
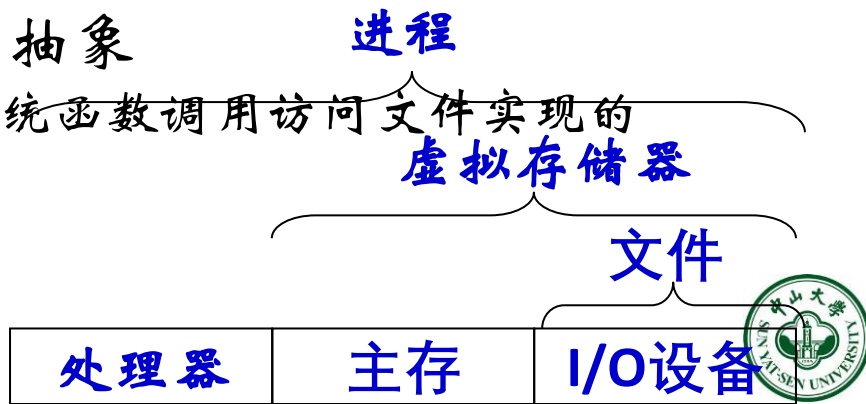
- 防止硬件被失控的应用程序滥用
- 向应用程序提供简单一致的机制来控制复杂的低层硬件设备

□ 操作系统提供的基本抽象概念

- 进程：是对一个正在运行程序的抽象
 - ◆ 操作系统实现交错执行的机制，称之为上下文切换

- 虚拟存储器：是对程序存储器的抽象
 - ◆ 每个进程看到的是一致的存储器——虚拟地址空间

- 文件：是对I/O设备的抽象
 - ◆ I/O是通过OS提供的系统函数调用访问文件实现的



进程 (processes)

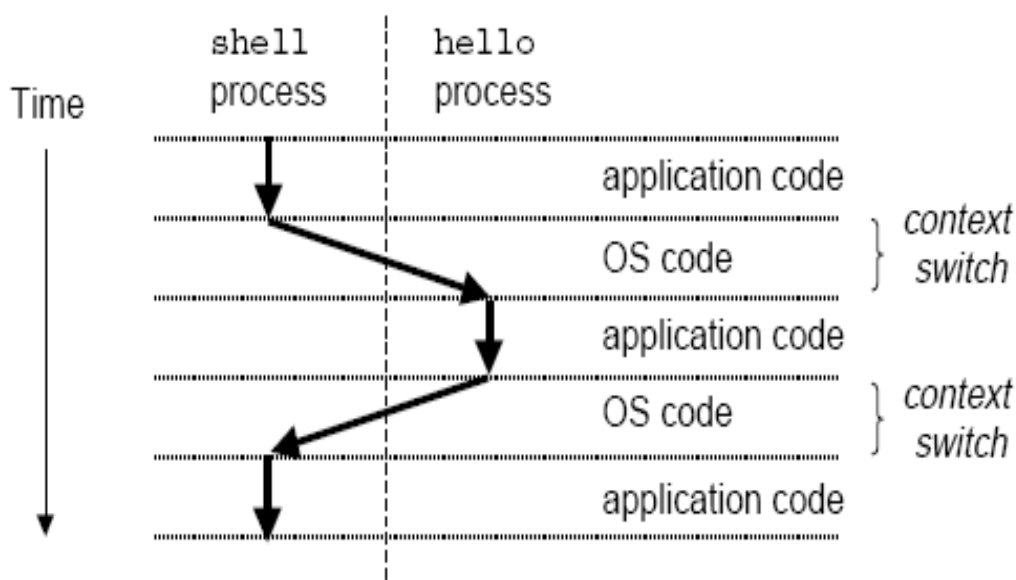
- hello程序运行时，该程序会以为(错觉)

```
unix> ./hello [Enter]  
hello, world  
unix>
```

 - 所有系统资源都被自己独占使用
 - 处理器始终在执行本程序的一条条指令
- **进程**：是一个“**执行中的程序**”，是OS对运行程序的一种抽象(虚拟)
 - 一个系统上可以同时运行很多进程，但每个进程都好像自己是独占使用系统
 - 实际上，OS让处理器交替执行多个进程中的指令
 - OS实现交替指令执行的机制，称之为“**上下文切换**”
- 进程的上下文
 - 指进程运行所需的所有状态信息。例如：PC、寄存器堆的当前值、主存的内容、段/页表
 - 系统中有一套状态单元存放当前运行进程的上下文
- **线程**：轻量级进程(共享数据存储)，是进程的一个实体。一个进程可以包含若干个线程

Hello程序执行过程中进程间的切换

- 上下文切换过程(任何**时刻**，单CPU系统中只有一个进程**正在运行**)
 - 上下文切换指把正在运行的进程换下，换上一个新进程到处理器执行。上下文切换时，必须保存换下进程上下文，同时恢复换上进程上下文



开始shell进程等待命令行输入；输入“hello”后shell进行系统调用；OS保存shell上下文，创建并换入hello进程；hello进程中止时，进行系统调用；OS恢复shell进程上下文，并换入shell进程

在一个进程的整个生命周期中，有不同进程在处理器交替运行，故运行时间很难准确测量

联系方式

□Acknowledgements:

□This slides contains materials from following lectures:

- Computer Architecture (NUDT, USTC)

□Research Area:

- 计算机视觉与机器人应用计算加速,
- 人工智能和深度学习芯片及智能计算机

□Contact:

- 中山大学计算机学院
- 管理学院D101 (图书馆右侧)
- 机器人与智能计算实验室
- cheng83@mail.sysu.edu.cn

2022/08/31

