

# 1.1 存储程序原理

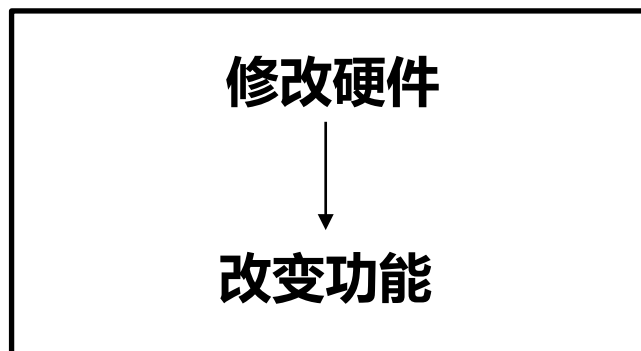
# 1.1 存储程序原理

## 存储程序原理：

- 将描述计算机的解题过程的程序事先设计好
- 将程序对应的指令序列以二进制形式存储在机器中
- 计算机工作时自动有序地从机器中逐条取出指令加以执行

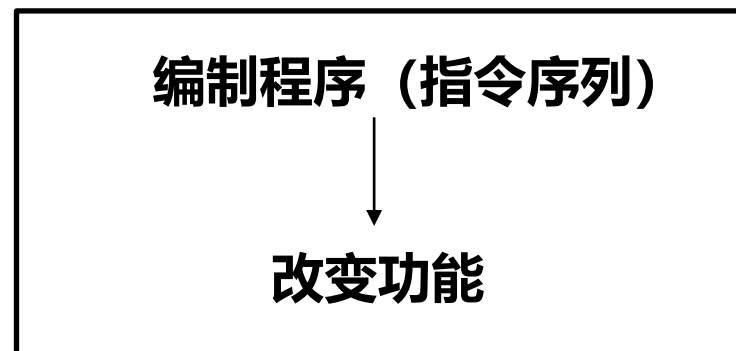


### ENIAC



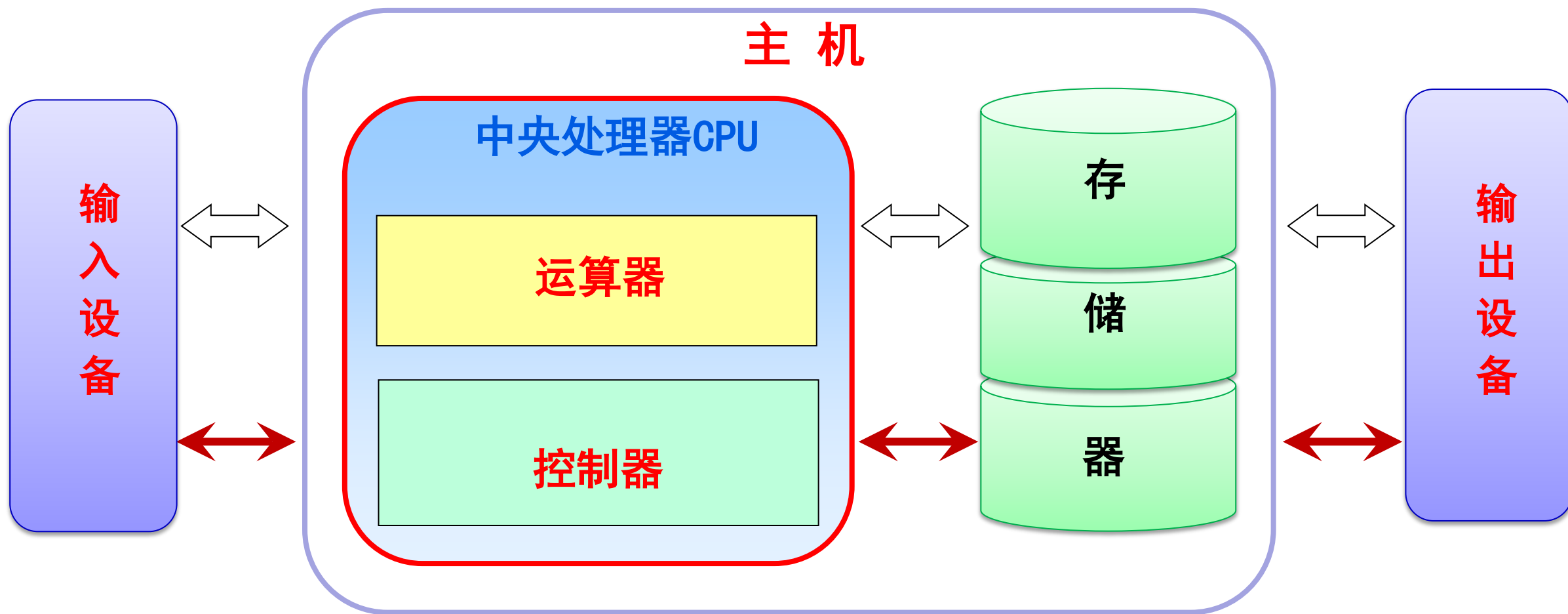
VS

### 存储程序原理计算机



# 1.1 存储程序原理

冯·诺伊曼体系结构 → 现代计算机的五大组件



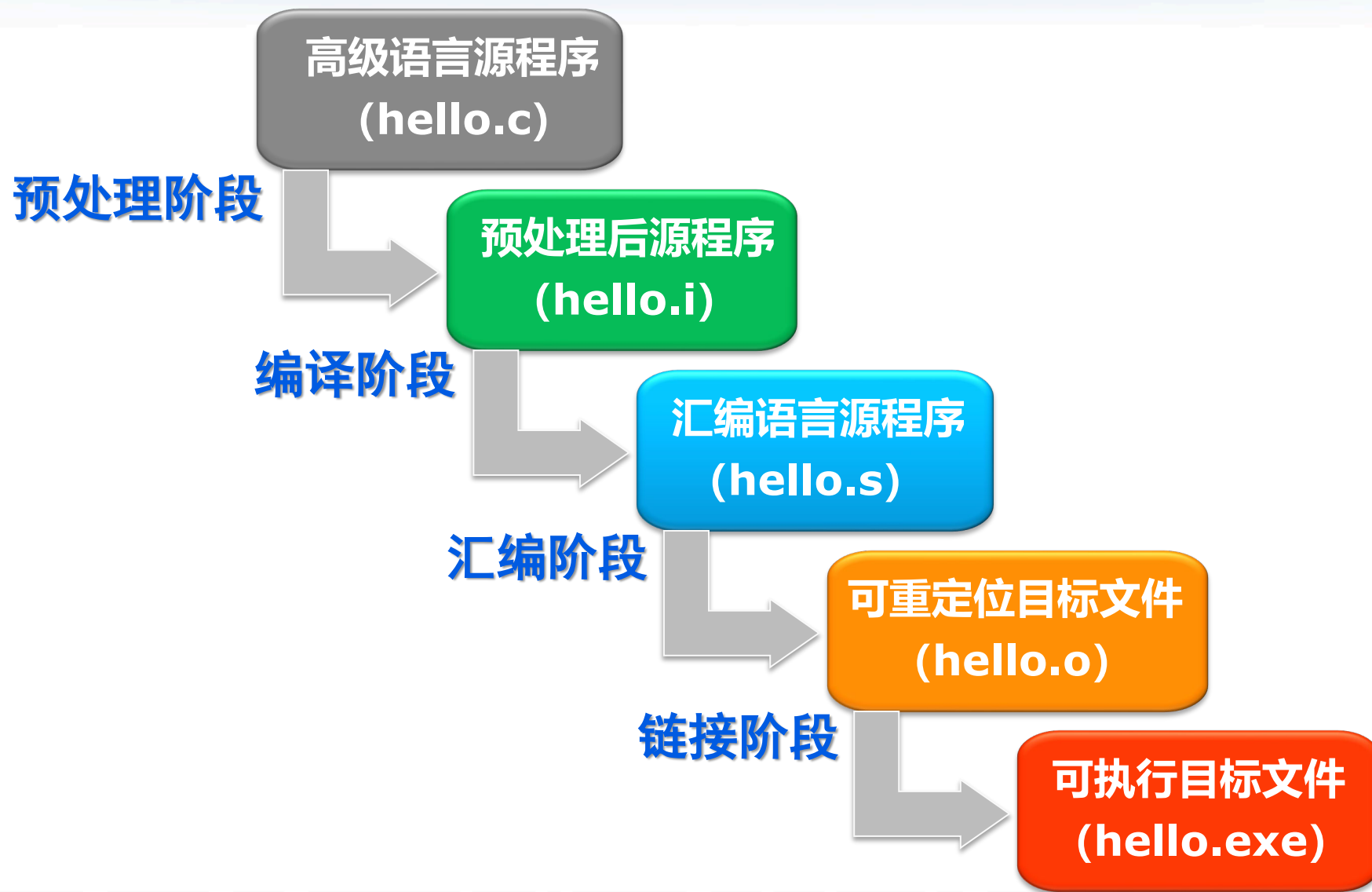
# 1.1 存储程序原理

典型台式机



# 程序从高级语言到可执行代码的转换

# 程序从高级语言到可执行代码的转换



# 补充：Hello World之从高级语言到机器代码

## ● GCC常用命令

`gcc -E hello.c -o hello.i`

`gcc -S hello.i -o hello.s`

`gcc -c hello.s -o hello.o`

`gcc hello.o -o hello.exe`

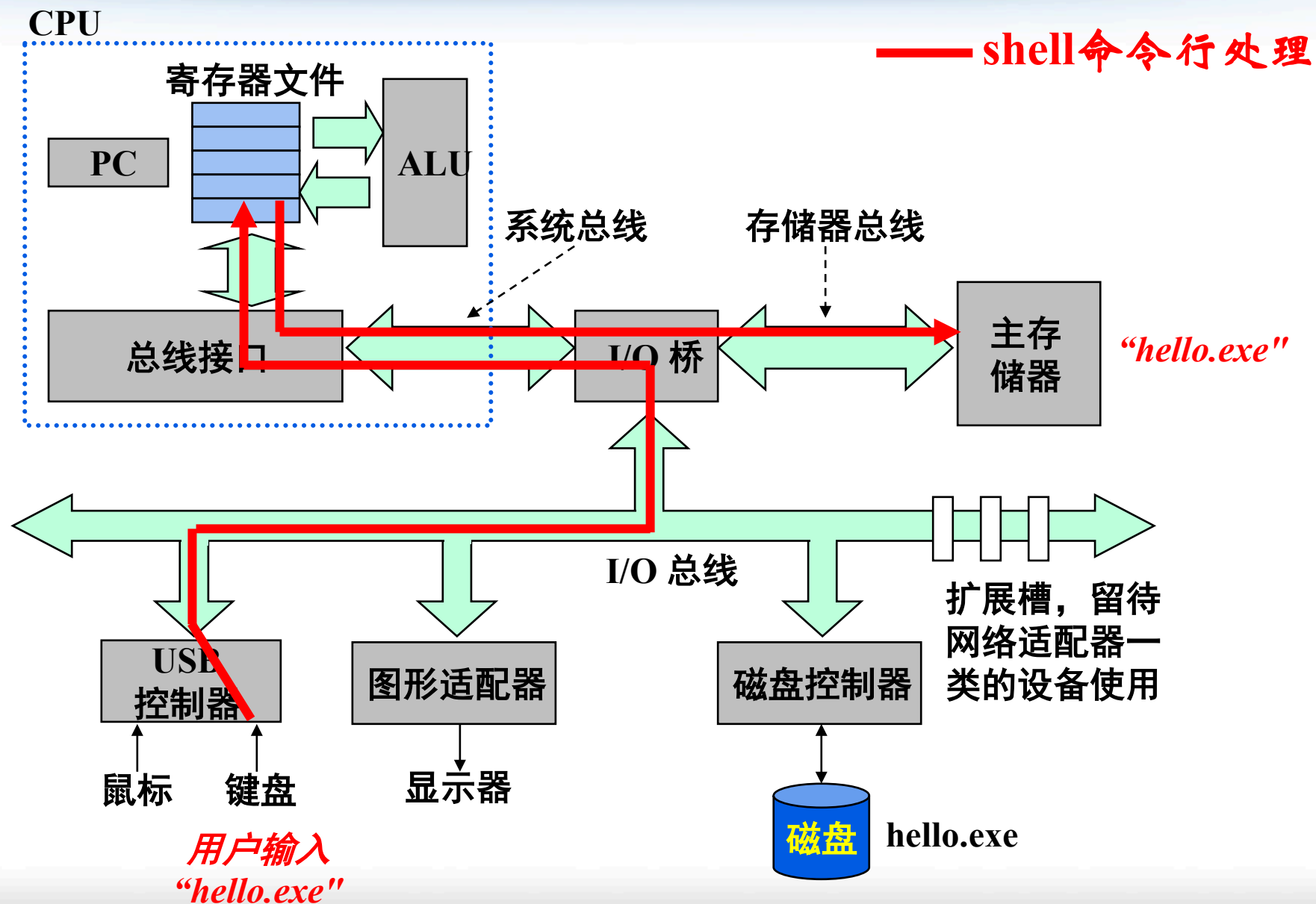
-E：对文件作预处理  
-o：指定输出文件名  
-S：对文件进行编译  
-c：对文件进行汇编



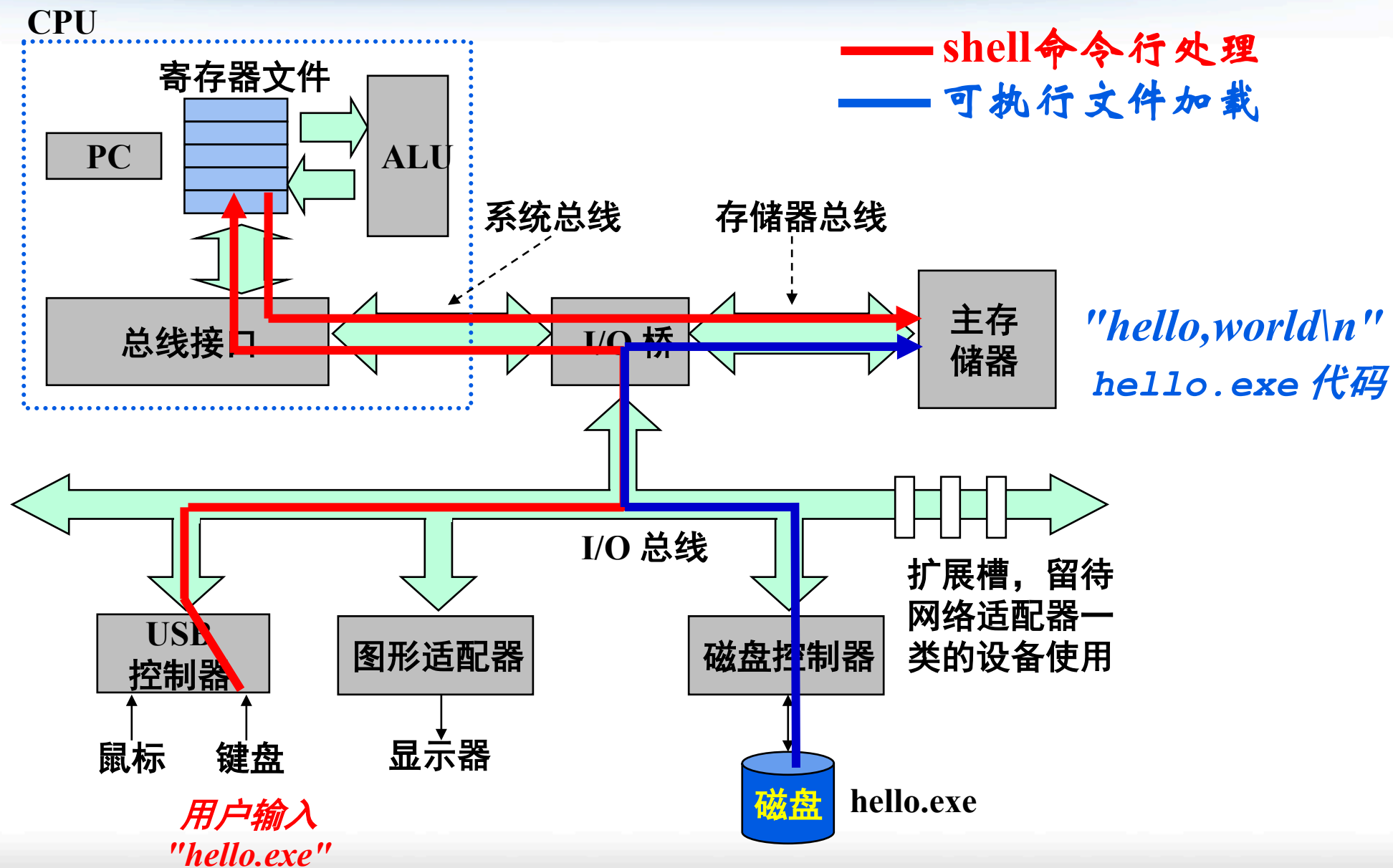
## 1.2 计算机的基本硬件组成



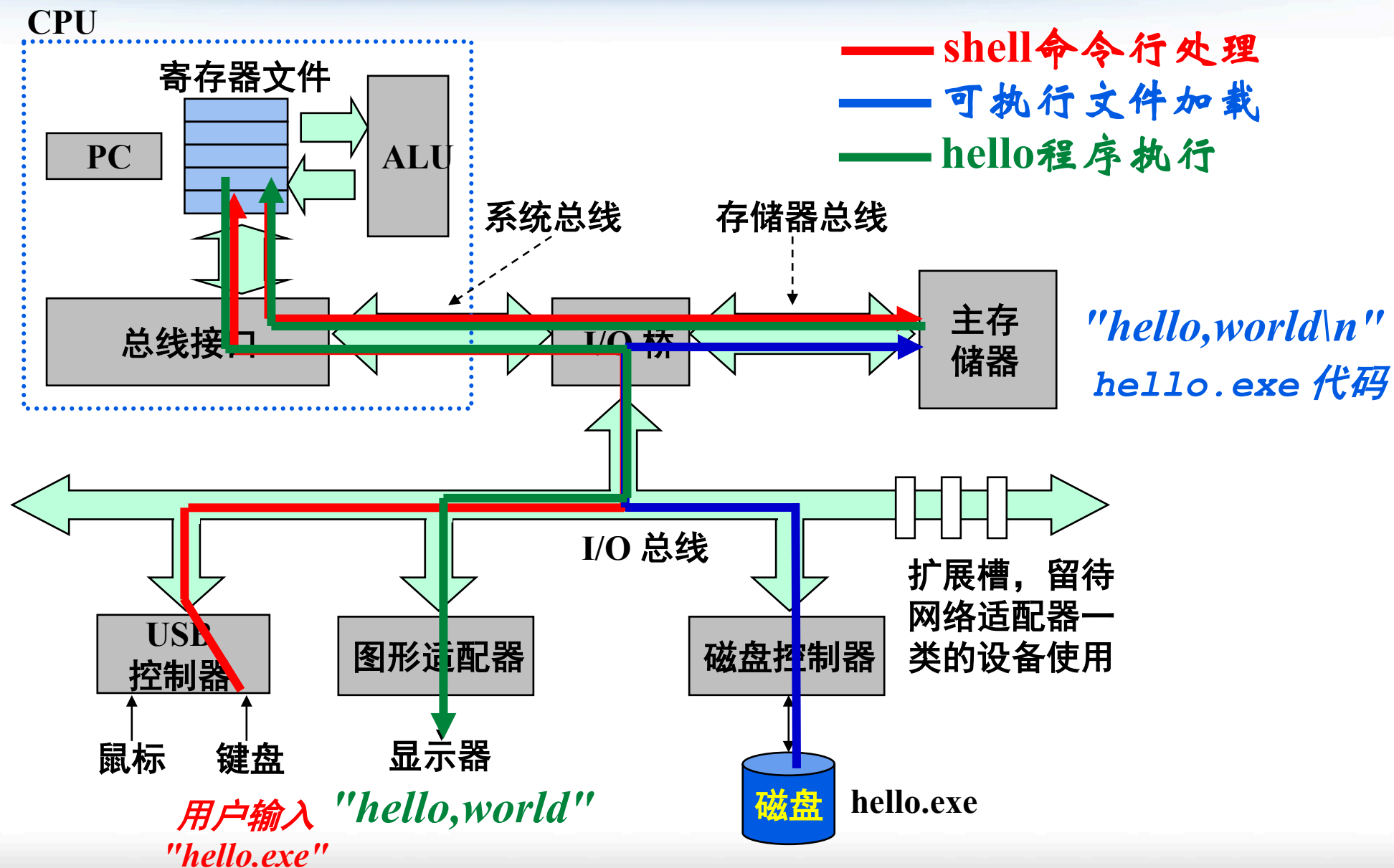
## 1.2 计算机的基本硬件组成



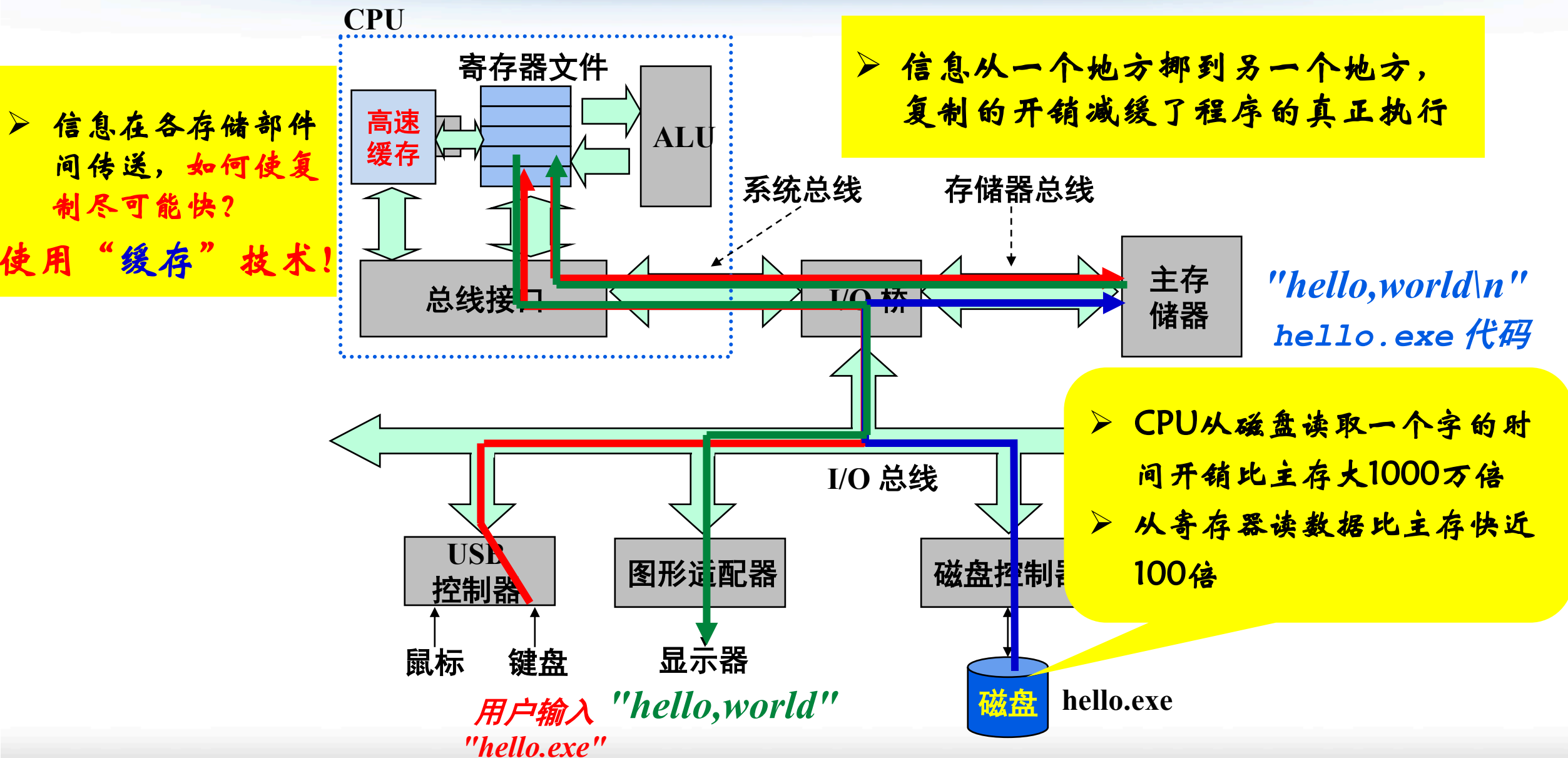
## 1.2 计算机的基本硬件组成



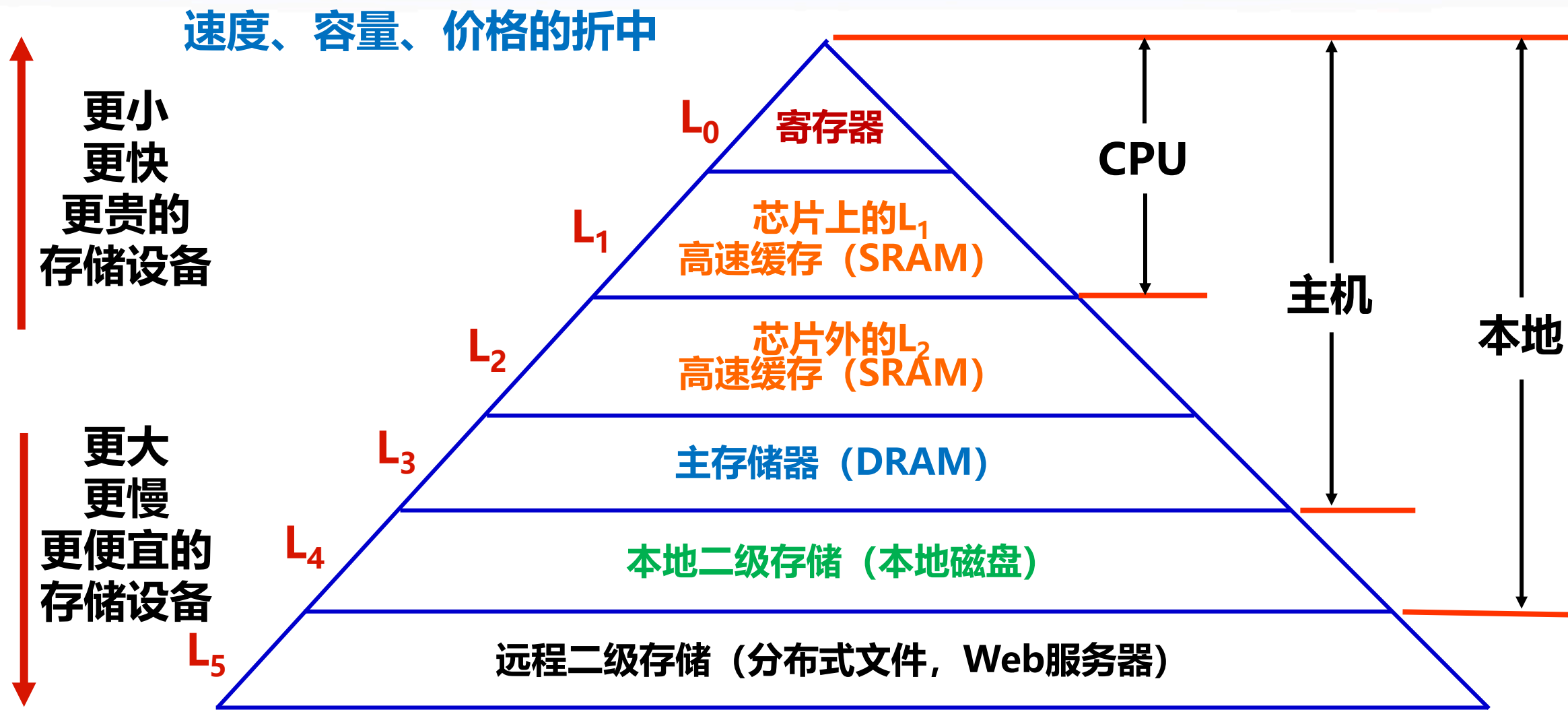
## 1.2 计算机的基本硬件组成



## 1.2 计算机的基本硬件组成



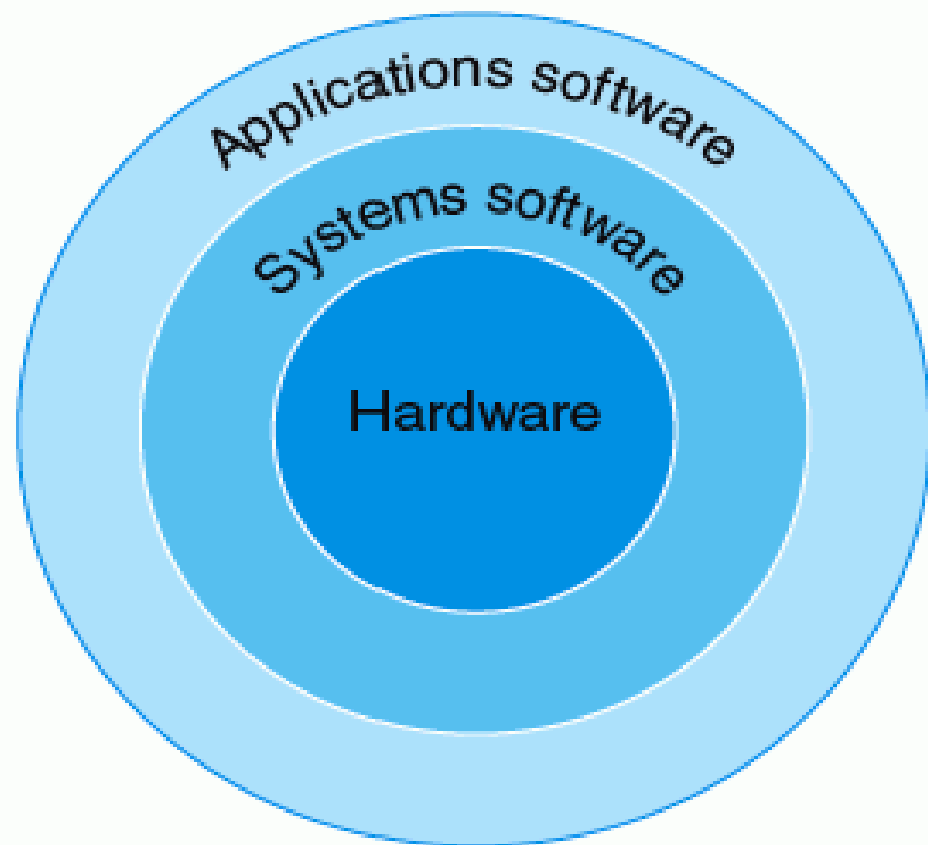
## 1.2 计算机的基本硬件组成



## 1.3 计算机的层次结构

## 1.3 计算机的层次结构

- 简单的软硬件层次化结构



- 隐藏低层次的实现细节，  
简化上层用户的使用
- 每一层都为上一层隐藏了  
自己的技术细节——“抽象”



# 1.3 计算机的层次结构

- 系统软件：提供公共服务程序

## 操作系统

用户程序和硬件的接口，  
提供计算机资源管理

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

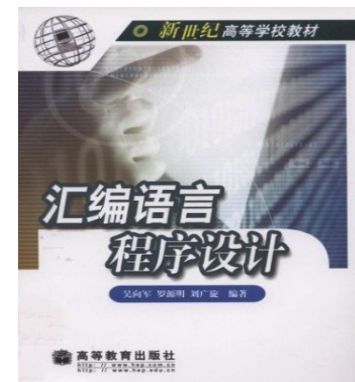
将高级语言语句翻译成汇编语言语句的程序



## 汇编器

汇编语言是一台计算机指令系统的符号化表示

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



## 1.3 计算机的层次结构

### 编译器

将高级语言（如C、C++、Java等）  
编写的程序翻译成硬件能执行的指令  
（助记符表示）

### 汇编器

将指令由助记符形式翻译成二进制形式

通过编写一个程序来将更强大的  
高级语言翻译成计算机指令是计  
算机发展早期的一个重大突破

高级语言程序  
(C语言)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

编译器

汇编语言程序

```
swap:
  LSL  X10, X1, 3
  ADD  X10, X0, X10
  LDUR X9, [X10, 0]
  LDUR X11, [X10, 8]
  STUR X11, [X10, 0]
  STUR X9, [X10, 8]
  BR   X10
```

汇编器

```
000000001010001000000000100011000
二进制机器语言程序 000000100000100001000000100001
( ARMv8指令集) 10001101111000100000000000000000
1000111000010010000000000000000100
1010111000010010000000000000000000
1010110111100010000000000000000100
000000111110000000000000000001000
```

## 1.4 计算机的应用发展

## 1.4.1 现代计算机的分类和特点

### 个人计算机

- 用于个人使用的计算机，通常包含图形显示器、键盘和鼠标等
- 侧重于为单个用户提供良好的性能，保证低廉的价格，并通常运行第三方软件



联想扬天A4600K台式机

### 服务器

- 能同时为多个用户使用、运行大型程序的计算机，一般通过网络访问
- 超级计算机可视为高端服务器



IBM X3400塔式服务器

“天河二号”超级计算机

### 嵌入式计算机

- 嵌入到其他设备中的计算机，运行预先定义的一个应用或者一组软件
- 在保证达到功能最低要求的前提下严格控制成本或功耗



## 1.4.2 后PC时代

### 何为“后PC时代”？

- ① 后PC时代是指将计算机、通信和消费产品的技术结合起来，以3C产品的形式通过Internet进入家庭。
- ② 以公元2000年作为一个分水岭，之前称为“PC”（Personal Computer）时代；之后称为“后PC”（Post-Personal Computer）时代。

## 1.4.2 后PC时代

### 后PC时代的标志

#### 个人移动设备 (PMD)

- PMD(Personal Mobile Device, PMD): 连接到英特网上的小型无线设备, 由电池供电, 通过下载App安装软件。典型例子包括智能手机和平板电脑。

#### 云计算(Cloud Computing)

- 通过英特网提供服务的大规模服务器集群, 一些运营商可以动态地出租不同数量的服务器供租借者使用

## 1.4.2 后PC时代

### 云计算

- 依赖于被称为仓储规模计算机（Warehouse Scale Computer, WSC）的巨型数据中心
- 通过云计算实现的软件即服务（Software as a Service, SaaS）革命性地影响了软件工业的发展



# 计算机系统设计中的8个伟大思想

# 1. 面向摩尔定律的设计

## 摩尔定律

集成电路上可容纳的  
晶体管数每18~24个  
月翻一番

## 面向摩尔定律的设计

计算机系统结构设计  
师应当预测其**设计完  
成时**的工艺和技术水  
平，而不是**设计开始  
时**的工艺



## 2. 使用抽象简化设计

### 使用抽象简化设计

- 使用抽象 (abstraction) 来表征不同级别的设计
- 低层将细节隐藏起来，呈现给高层的只是一个简化的模型
- 提高硬件和软件开发效率



### 3. 加速大概率事件

#### 加速大概率事件

- 加速大概率事件（common case fast）远比优化小概率事件更能提高性能
- 大概率事件通常比小概率事件简单，因而更易于对其进行优化
- 通过仔细的实验与测量找出大概率事件



## 4. 通过并行提高性能

### 通过并行提高性能

- 从计算诞生开始，计算机架构师就给出了通过并行执行操作来提高性能的设计方案



## 5. 通过流水线提高性能

### 通过流水线提高性能

- **流水线(pipelining)**是计算机中一种重要的并行技术



## 6. 通过预测提高性能

### 通过预测提高性能

- “求人准许不如求人原谅”
- 假设预测错误后恢复的代价不大，并且预测的准确率相对较高，那么通过猜测的方式提前开始工作，要比等到确定知道能执行时才启动要效率高一些
- 也体现了加速大概率事件的思想





# 7. 存储层次

## 存储器层次

- 存储器速度影响程序的性能，容量限制了解题的规模
- 通过存储器层次（hierarchy of memory）可以来缓解对存储器需求的矛盾
- 存储器层次中，位于顶层的存储器速度最快、容量最小，但每位价格最昂贵。反之，处于最底层的存储器，速度最慢、容量最大，但每位价格最便宜。



## 8. 通过冗余提高可靠性

### 通过冗余提高可靠性

- 通过增加冗余器件可以提高系统的可靠性
- 当发生失效时，冗余器件可以替代失效器件并可以帮助检测错误



DEPENDABILITY

谢 谢！