

计算机科学导论

林川





目录

1. 绪论
2. 数字系统
3. 数据存储
4. 数学运算
5. 计算机组成
6. 计算机网络
7. 操作系统



绪论

1. 图灵机
2. 冯·诺依曼模型
3. 计算机发展史

图灵机

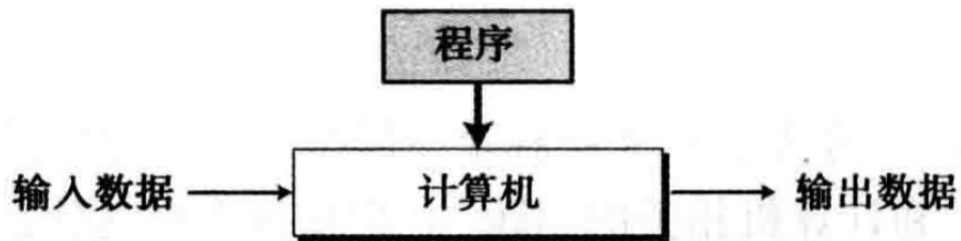
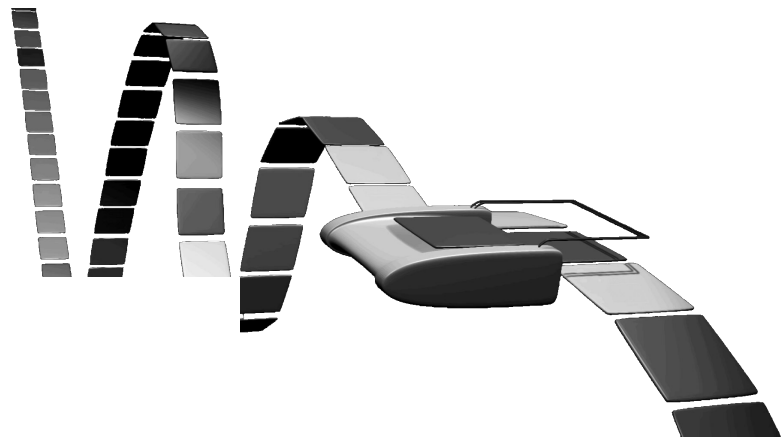


图 1-2 可编程数据处理器模型

同一程序，相同输入相同输出

<https://zh.wikipedia.org/wiki/%E5%9B%BE%E7%81%B5%E6%9C%BA%E5%9B%BE%E7%81%B5%E6%9C%BA%E7%9A%84%E6%AD%A3%E5%BC%8F%E5%AE%9A%E4%B9%89>

冯·诺依曼模型

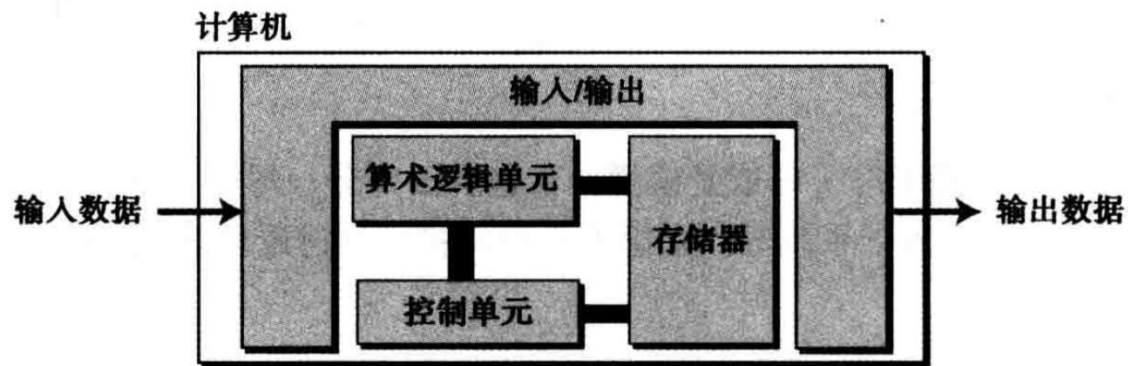


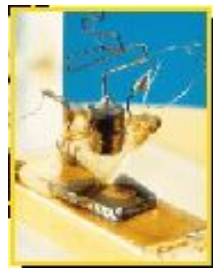
图 1-5 冯·诺依曼模型

计算机发展史

第一代
(1946~1957)
电子管
5千~4万(次/秒)
1964年911机



第二代
(1958~1964)
晶体管
几十万~百万(次/秒)



第三代
(1965~1971)
集成电路
百万~几百万(次/秒)



第四代
(1972~今)
大规模集成电路
几百万~几亿(次/秒)



数字系统-数制

◆ 数制

按进位的原则进行计数称为进位计数制，简称数制。

- 十进制：逢十进一（常用的）
- 八进制：逢八进一
- 二进制：逢二进一
- 十六进制：逢十六进一

特点：数字的总个数等于基数；最大的数字比基数小一；每个数字都要乘以基数的次幂。

进制转换一位权展开式

数	(<u>3</u>	<u>6</u>	<u>5</u>)	₁₀
位权		10^2		10^1		10^0
位权展开式		$3 \times 10^2 +$		$6 \times 10^1 +$		5×10^0

任何一个数都可以用位权展开式表示，位权展开式又称为乘权求和。

$$(365)_8$$

$$(365)_{16}$$

$$(1011)_2$$



进制转换-R进制转换为10进制

A: 10 1010
C: 12 1100

转换方法: 按位权展开求和

例:

$$\begin{aligned} 1) (1001.01)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 0 + 0 + 1 + 0 + 0.25 \\ &= 9.25 \end{aligned}$$

$$2) (25)_8 = 2 \times 8^1 + 5 \times 8^0 = 16 + 5 = 21$$

$$3) (2AB)_{16} = 2 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 = 683$$



练习-R进制转10进制

1. 二进制整数111110转换成十进制数是_____
2. 5位二进制整数最大能表示的十进制整数是_____

A、62

B、60

C、58

D、56

A、64

B、63

C、32

D、31



练习-R进制转10进制-续

数111110

111110

$2+4+8+16+32$

111111

1000000

$64-2 = 62$

11111

$1+2+4+8+16 = 32-1 = 31$



练习-R进制与10进制互相转换

1. 十进制数 215 等于二进制数_____。

A、11101011 B、11101010 C、11010111 D、11010110

2. 下列各进制的整数中, _____表示的值最大。

A、十进制数11 B、八进制数11
C、十六进制数11 D、二进制数11

3. 已知三个用不同数制表示的整数 $A=00111101$, $B=3C$, $C=64$, 则能成立的比较关系是_____。

A、 $A < B < C$ B、 $B < C < A$ C、 $B < A < C$ D、 $C < B < A$



0011 1101 3C $64 = 16 * 4 = 40$

3D

C > A > B



练习-R进制与10进制互相转换-续

- 215
- 255
- 65535



35

$32 + 3$

100011

$126 = 127 - 1$

1111110



$$0.634 = 0.5 + 0.134 = 0.125 + 0.009$$

$$0.5 \quad 0.25 \quad 0.125 \quad 0.0625 \quad 0.03125 \quad 0.015625 \quad 0.0078125$$

$$0.1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1$$

$$100 \ 1110 \ 0010$$

$$4 \ E \ 2$$

2进制16进制互相转换

0	1	2	3
0000	0001	0010	0011
4	5	6	7
0100	0101	0110	0111
8	9	A	B
1000	1001	1010	1011
C	D	E	F
1100	1101	1110	1111

$(1101\ 0100\ \blacksquare\ 0011\ 1010)_2$

$= (D\ 4.3\ A)_{16}$



作业

教材第二章 (<https://wiki.banyuan.club/pages/viewpage.action?pageId=13698223>) 练习:

1, 2, 3, 4, 6, 10

以上各题目只需完成a和c

14, 15, 19, 21



预习和笔记

- 预习重点反馈预习过程中的难点，不理解的地方，以便在课上重点讲解
- 笔记是对已学知识的提炼和理解，将书本的知识转化为自己的知识；好记性不如烂笔头
- 重内容，轻形式



数据存储

1. 计算机的存储
2. 如何存储文本
3. 如何存储数字



计算机存储

- 计算机内部只有bit
- 每个bit只能是0或者1
- 8个bit组成1个Byte
- b & B
- 1K = 1024B
- 1M = 1024K
- 1G = 1024M
- 1T = 1024G
- 1P = 1024T

2个16进制的数码表示范围：

1个数码是4bit 2个是8bit

0-255

```
Linc@pop-os:~$ echo "1234" > txt
Linc@pop-os:~$ od -x txt
00000000 3231 3433 000a
00000005
```

大小端



存储文本

- ASCII码(1111111-127, 7bit)
 - man ascii
 - 大小写字母值连续且分离
 - 如何进行大小写转换？

china 小写大写都是连续的，每个字符都是一个数字 $a \ A \Rightarrow a + (A - a) = A$

CHINA $x + (A - a) = X$

- Unicode
 - 16bit 65536 32bit ??
 - UTF-8

存储数字-整数

■ 码制

➤ 原码

用符号位和数值表示带符号数，正数的符号位用“0”表示，负数的符号位用“1”表示。

有+0(00000000)和-0(10000000)

➤ 反码

正数的反码与原码相同，负数的反码为对该数的原码除符号位外各位求反。

➤ 补码

正数的补码与原码相同，负数的补码为对该数的原码除符号位外各位求反，然后再在最后一位加1。

无符号数-没有负数

- 无符号整数 (定点数) 的表示
- 整数就是整数，没有小数。由于小数点始终隐含在个位数的右面，所以整数也叫做“定点数”
- 计算机中是如何表示（正）整数的？我们以 8 位的计算机为例说明：

1 1 1 1 1 1 1 1

最大的数

0 0 0 0 0 0 0 0

最小的数

- 最小的值： $(00000000)_2 = (0)_{10}$
- 最大的值： $(11111111)_2 = (255)_{10}$

0xFF

0 - 0xFFFF(65535)



有符号数-反码

- 带符号整数 (定点数) 的表示
- 最高位 (最左面的一位) 作为符号位, “0” 表示 “+” (正数), “1” 表示 “-” (负数), 其余各位表示数的绝对值

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

最大的数

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

最小的数

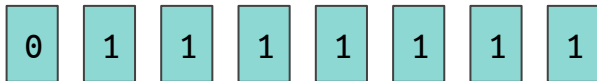
- 8 位计算机取值范围: $-127 \sim +127$
- $-2^7 + 1 \sim +2^7 - 1$

- 8个bit一共能存多少个数字? ---- 256
- 使用反码能存多少个数字? ---- $127 \times 2 + 1 = 255$



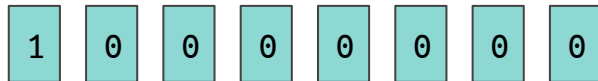
有符号数-补码

- 最高位(最左面的一位)作为符号位, “0”表示“+”(正数), “1”表示“-”(负数)
- 正数正常表示
- 负数是正数取反再+1
- 为什么使用补码???



最大的数
127

11111111 (-1)



最小的数
 $111111 + 1 =$
 $1000000 = 128$
-128



为什么要用补码

$$4 - 3 = 4 + (-3)$$

0000 0011 +3

1111 1100 +3去反

1111 1101 (-3)补码

0000 0100 4

0000 0001 1

$$4 - 3$$

0000 0100

- 0000 0011

0000 0001



补码示例

求-179的16bit补码的16进制表示 8bit表示补码范围:-128 +127

$$179 = 11 \times 16 + 3$$

$$= 0x00B3 = 0000\ 0000\ 1011\ 0011$$

取反为：

$$1111\ 1111\ 0100\ 1100$$

补码为：

$$1111\ 1111\ 0100\ 1101 = 0xFF4D$$



课堂练习

1. -1的16bit补码表示(2进制)

0000 0000 0000 0001

1111 1111 1111 1111

2. -128的8bit补码表示(16进制)

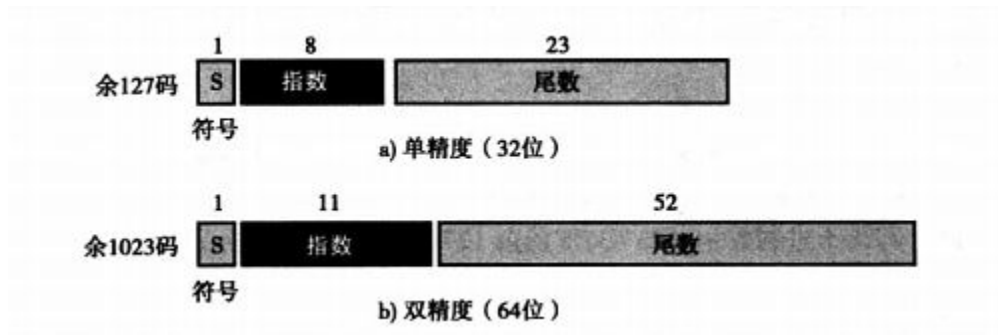
1000 0000

0111 1111

1000 0000

0-00

实数



1. 符号为正, 所以 $S=0$ 。
2. 十进制转换为二进制 $5.75=(101.11)_2$ 。
3. 规范化 $(101.11)_2=(1.0111)_2 \times 2^2$
4. $E=2+127=129=(10000001)_2$, $M=(0111)_2$ 。我们需要在M的右边增加19个0使之成为23位。
5. 该表示法如下所示:

计算机如何知道一个数是整数还是实数？

S E M

0 10000001 011100000000000000000000



作业

教材“数据存储”练习 (<https://wiki.banyuan.club/pages/viewpage.action?pageId=13698224>)

1, 2, 3, 6, 12, 15, 16, 24, 26



数学运算-逻辑运算

- 与 两者为真才为真
- 或 有一为真则为真
- 非 取反操作
- 异或 两着不同才为真

整数的bit运算

- 使用“与”操作去掉不想要的位数

求一8bit无符号整数 (x)对8取余??

$$60 / 8 = 7 \dots 4$$

0000 0111 & x -> 取x的低3bit
0-7

$$32 + 28 = 16 + 8 + 4$$

0011 1100

& 0000 0111

$$0000 0100 = 4$$



数学运算-左移右移

- 无符号
 - 逻辑移位
- 有符号
 - 算术移位
- 乘除操作



数学运算-算术

- 加
- 减
- 乘
- 除
- 余
- 溢出



作业

<https://wiki.banyuan.club/pages/viewpage.action?pageId=13698225>

1,2,3,4 只做a

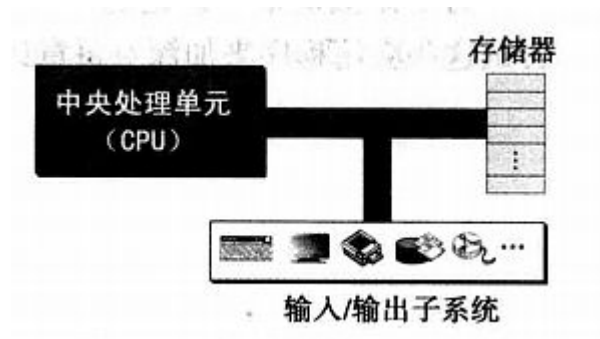
5, 6, 9, 10

13, 15 只做a

20, 21

计算机组成

- 中央处理单元(CPU)
- 主存储器(内存)
- 输入/输出子系统



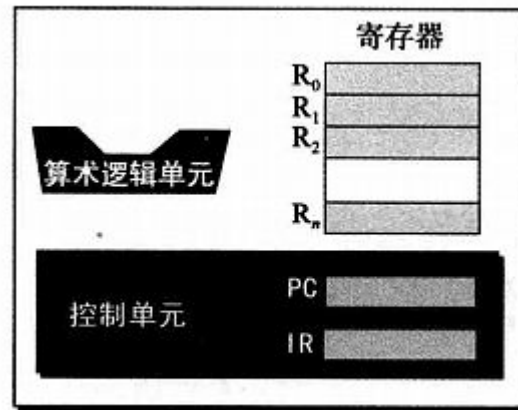


CPU

$10 + 200$

内存100地址的值 + 内存200地址的值

- ALU
 - 运算
- 寄存器
 - 数据寄存器
 - 指令寄存器
 - 程序计数器
- 控制单元
 - 连接存储和输入输出



中央处理单元 (CPU)



地址空间

16bit 65535 你只能访问到65535的内存空间

32bit 4 294 967 296 空间 你只能访问4G 字节的内存空间, win32早期, 一个文件大小不能超过4G

64bit 太大了,

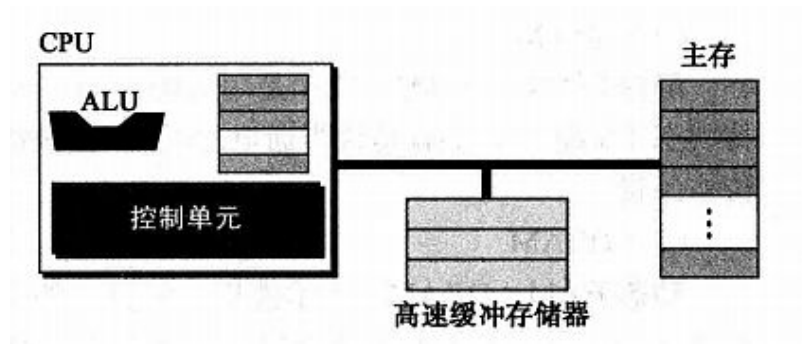
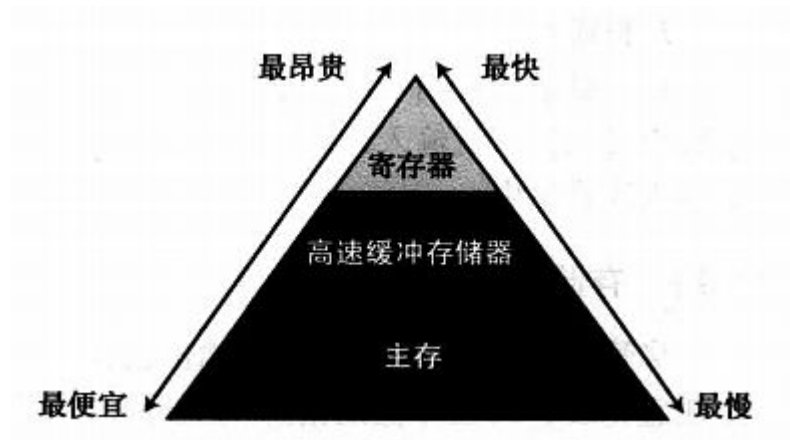
1023 0000001111111111

$2^{32} - 1$ 1111 1111 1111 1111 1111 1111 1111 1111 (32bit)

2^{32}

0(字节) 1 2 $2^{32} - 1$

存储器



CPU访问存储器

- 数据总线

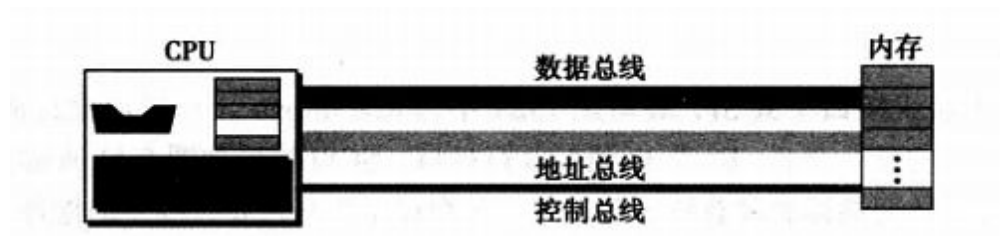
最大一次传输多少数据

- 地址总线

能访问的空间大小

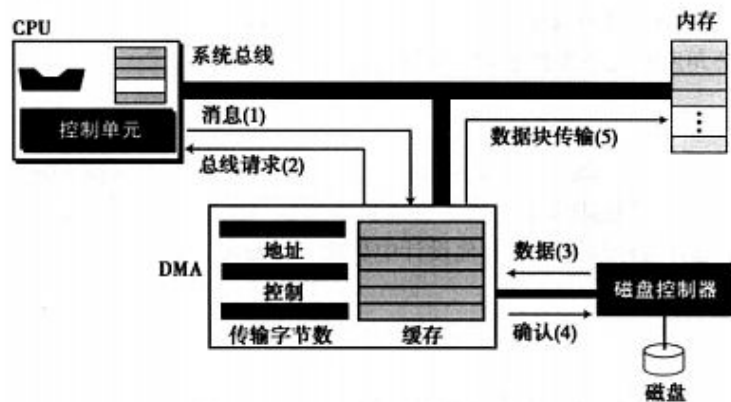
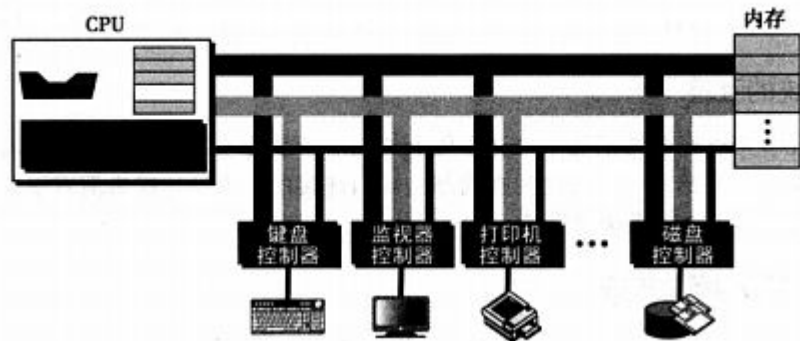
- 控制总线

定义CPU指令

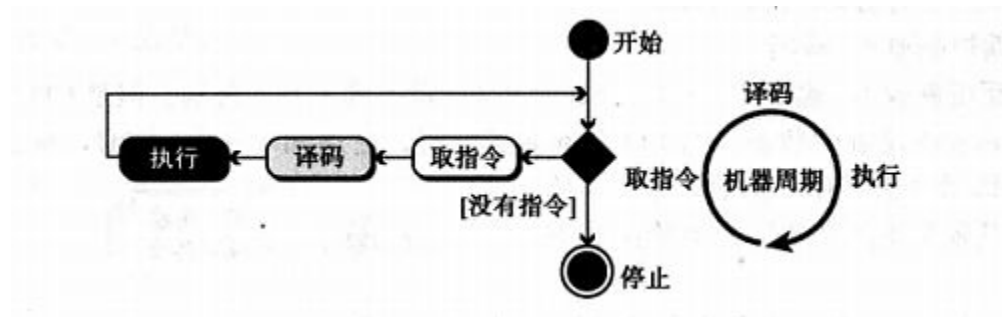


输入输出

- 非存储类
 - 显示器
 - 键盘
 - 打印机
- 存储类
 - 硬盘
 - 光盘
 - USB



CPU在做什么





作业

<https://wiki.banyuan.club/pages/viewpage.action?pageId=13698226#>

习题1, 2, 3



计算机网络

- 网络
 - 局域网
 - 广域网
 - Internet
- TCP/IP
 - 为什么要分层
 - 应用层
 - 传输层
 - 网络层
 - 链路层
 - 物理层

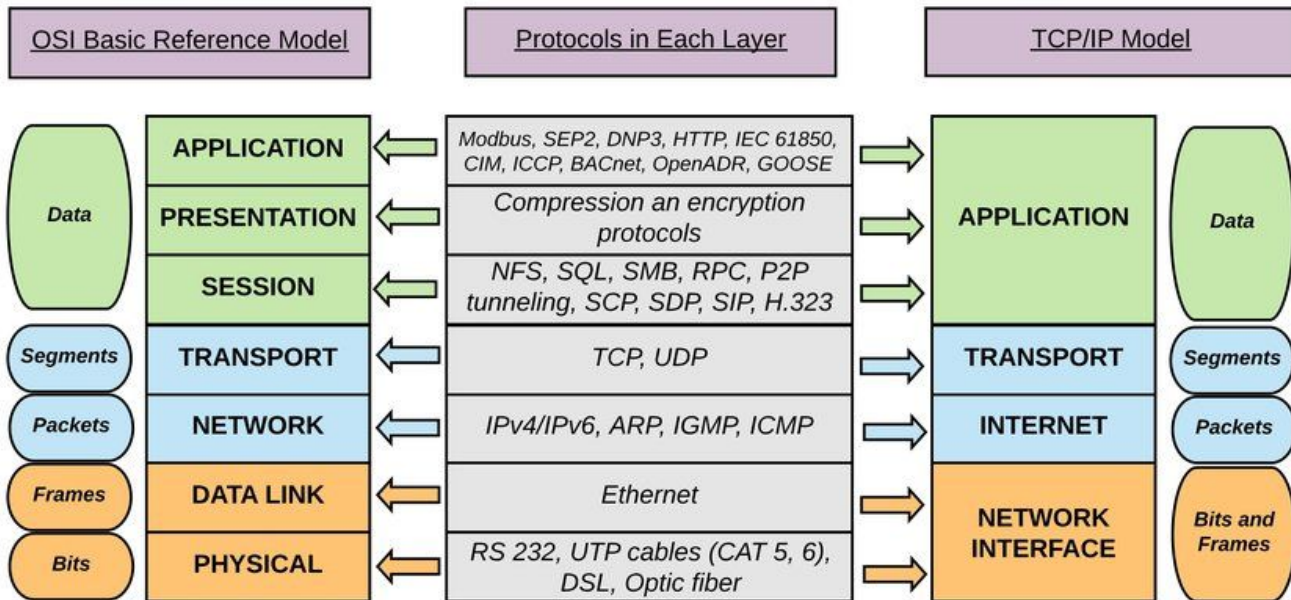


网络

- 局域网
 - MAC地址
 - 交换机
- 广域网
 - IP地址
 - 公有地址/私有地址
 - IPv4/IPv6
 - 路由器
- Internet
 - 巨大的广域网？



TCP/IP vs OSI





物理层

数字->信号

介质中传递的都是信号

介质的分类:

- 光纤
- 网线(双绞线, 超五类双绞线)
- 同轴电缆(E1)
- WIFI
- 4G/5G ??



链路层

- 标识LAN的地址
 - Mac地址
 - 4G/5G地址??
- 交换机
 - 转发Ethernet Frame



网络层

标识网络上的地址

- IP地址(IPv4)
 - 单播
 - 私有
 - 192.168.0.0/16
 - 172.16.0.0/12
 - 10.0.0.0/8
 - 共有
 - 共有地址不够怎么办？
 - 组播
 - 224.0.0.0/24
 - 广播
- 路由器
 - 转发IP报文
- 网关是什么？



传输层

- TCP
 - 面向连接
 - 3次握手4次挥手
 - Http, SSH, FTP
- UDP
 - 无连接
 - BT, magnet, DNS
- socket
 - IP地址+端口号
 - 什么是端口号？



应用层

- 用户可见的业务
- 你们写的代码都是在应用层



HTTP

- 客户端
- 服务器
- URL
- 域名
- hostname(主机名)



作业

<https://wiki.banyuan.club/pages/viewpage.action?pageId=13698227#>

复习题:8, 9

练习题:3, 9

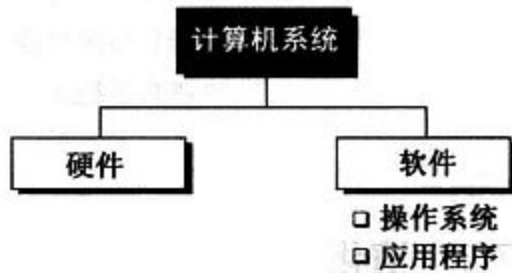
解释浏览器输入 `www.baidu.com` 并回车到底发生了些什么??



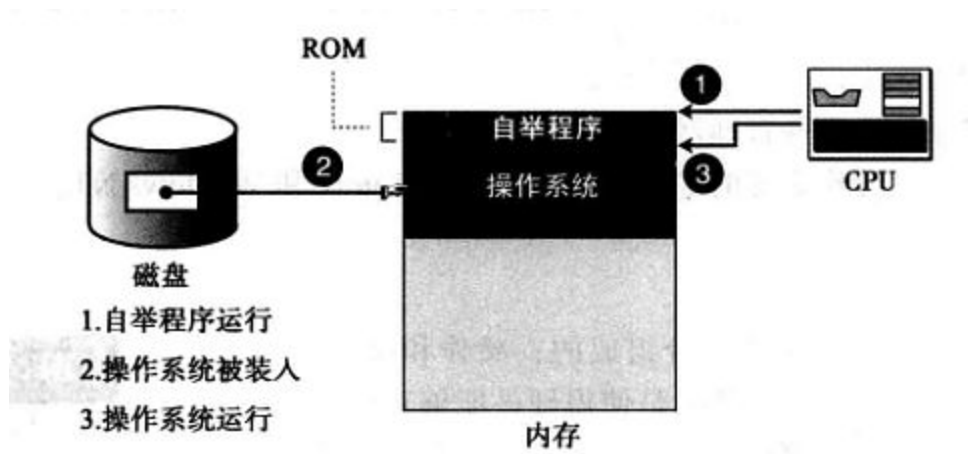
操作系统

- 自举过程
- 演化
- 组成部分
- 主流操作系统

计算机系统



自举过程



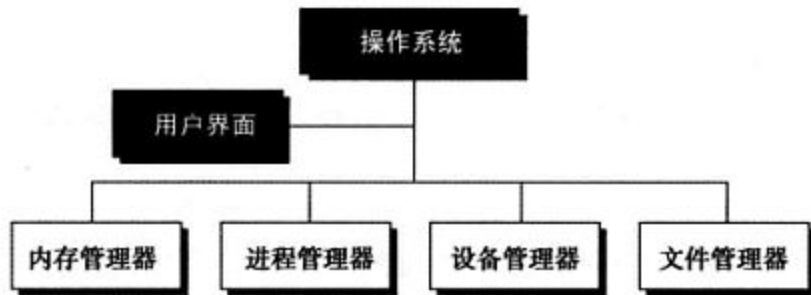


演化

- 批处理 DOS 每次做一个任务
- 分时 一个CPU的一个核, 提高效率, 在O等待的时候执行其他任务
- 个人 pad, 手机, 良好的UI
- 并行 多核多线程, 不同的核或线程做不同的任务
- 分布式 多点共同计算完成一个任务
- 实时 对响应速度要求高, 并不是CPU越厉害就能响应越快

组成部分

- 内存/虚拟内存
- 进程
 - 进程调度
 - 队列
 - 资源分配
 - 死锁
 - 饿死
- 文件系统





主流操作系统

Linux - 一切皆文件

- 内核
 - 内存管理
 - 进程管理
 - 设备管理
 - 文件管理
- 网络
- 安全



作业

复习题:9, 10

练习题:9, 10

确认Xcode正常工作