目录

[第一章 2](#_Toc502666382)

[1.1、计算机的发展历史 2](#_Toc502666383)

[1.2计算机分类 3](#_Toc502666384)

[1.3计算机系统 3](#_Toc502666385)

[1.4计算机的主要部件---存储器 3](#_Toc502666386)

[1.5计算机的性能指标 4](#_Toc502666387)

[1.6冯诺依曼计算机---现代计算机之父 5](#_Toc502666388)

[1.7大数据的特点 5](#_Toc502666389)

[第二章 6](#_Toc502666390)

[2.1二进制的基本运算 6](#_Toc502666391)

[2.2二进制数据基本运算的硬件实现 6](#_Toc502666392)

[2.3二进制数据运算的应用-二四译码器 7](#_Toc502666393)

[2.4其他进制数据 7](#_Toc502666394)

[2.5机器数 8](#_Toc502666395)

[第三章 9](#_Toc502666396)

[3.1计算机编程的基本概念 9](#_Toc502666397)

[3.2语言的层次 10](#_Toc502666398)

[3.3初窥高级语言 10](#_Toc502666399)

[3.4Python 12](#_Toc502666400)

[3.5列举高级语言 16](#_Toc502666401)

[第四章 17](#_Toc502666402)

[4.1程序设计 17](#_Toc502666403)

[4.2基本的数据结构 17](#_Toc502666404)

[4.3程序设计方法 21](#_Toc502666405)

[4.4软件生命周期 23](#_Toc502666406)

[第五章 25](#_Toc502666407)

[5.1认识操作系统 25](#_Toc502666408)

[5.2操作系统分类 26](#_Toc502666409)

[5.3操作系统对硬件的管理—CPU 27](#_Toc502666410)

[5.4文件系统 30](#_Toc502666411)

[第六章 30](#_Toc502666412)

[6.1无远弗届的网络 30](#_Toc502666413)

[6.2Web 35](#_Toc502666414)

[6.3初窥物联网 36](#_Toc502666415)

|  |  |  |
| --- | --- | --- |
| 章节 | 章节内容 | 小节内容 |
| 第一章 | 计算机概述 | 1.1计算机的发展历史  1.2计算机分类  1.3\*计算机系统  1.4计算机的主要部件  1.5计算机的性能指标  1.6冯诺依曼计算机  1.7大数据 |
| 第二章 | 计算机中的数据 | 2.1二进制数据的基本运算  2.2二进制数据进本运算的硬件实现  2.3二进制数据运算的应用  2.4其他进制数据  2.5机器数及有符号数据在上计算机中的存储 |
| 第三章 | 计算机编程 | 3.1计算机编程的基本概念  3.2语言的层次  3.3初窥高级语言  3.4Python  3.5列举高级语言  3.6算法 |
| 第四章 | 程序设计导引 | 4.1程序设计的四个步骤  4.2基本的数据结构  4.3程序设计方法  4.4软件生命周期 |
| 第五章 | 操作系统 | 5.1认识操作系统  5.2操作系统的分类  5.3操作系统对硬件的管理-CPU  5.4文件系统 |
| 第六章 | 网络与物联网 | 6.1无远弗届的网络  6.2Web  6.3初窥物联网 |

# 第一章

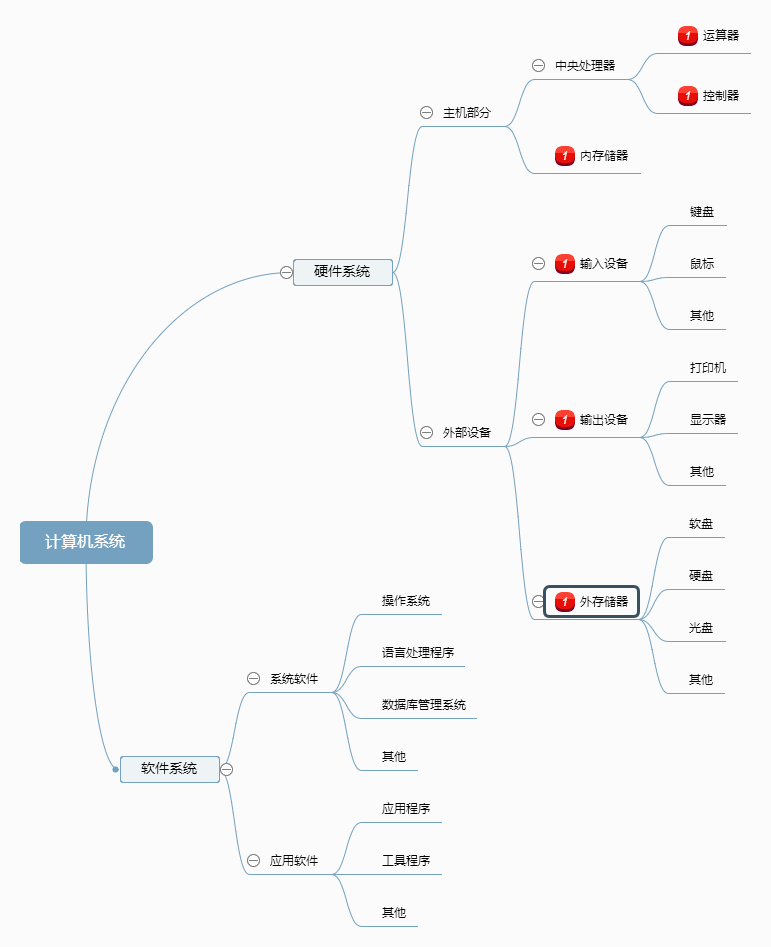
1.1、计算机的发展历史

1. 电子管计算机时代---世界上第一台电子管计算机产生于1946年宾夕法尼亚大学，名为ENIAC
2. 晶体管时代---世界上第一台晶体管计算机产生于1954年贝尔实验室，名为TRADIC
3. 小、中规模集成电路时代
4. 大、超大规模（甚大、极大规模）集成电路计算机时代

1.2计算机分类

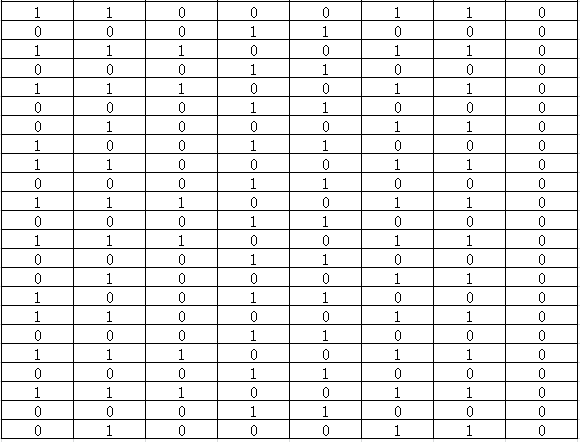
1. 按运算方式：电子数字计算机、电子模拟计算式电计算机
2. 按规模和功能：超级计算机、大型计算机、小型计算机、工作站、微型计算机、笔记本计算机

1.3计算机系统

 注：计算机的主要组成部分为：运算器、控制器、存储器、输入设备、输出设备。总线不属于五大部件。

1.4计算机的主要部件---存储器

1.存储器是存放程序与数据，是计算机实现“存储程序控制”的基础。

2.

一小格是一位（简称b），一行是八位，每八位是一个字节（简称B）。

3.内存以字节为基本单位，按字节编制；内存使用时常以若干字节为一个逻辑单位。

4.一个字节通常作为识别数据的基本单位，且每个字节均有一个地址。

5.一个字节包括八位，这八个二进制位有2^8个可能值，即2^8个不同的二进制串。

6.计算机内部采用二进制来表示数据和指令。

7.将编好的程序和原始数据事先存入存储器中，然后再启动计算机工作，程序就可自动执行。

1.5计算机的性能指标

1. 机器字长：机器字长是指参与运算的书的基本位数，它是由加法器、寄存器、数据总线的位数决定的。最少要八位字节。
2. 数据通路宽度：数据总线一次所能并行传送信息的位数，成为数据通路宽度。
3. 主存容量：一个主存储器所能存储的全部信息量 （衡量主存容量的单位：a.字节数；b.字数\*字长）
4. 运算速度：以MIPS和MFLOPS作为计量单位来衡量运算速度。（MIPS便是每秒钟百万条指令；）MIPS=指令条数/（执行时间\*10^6）=1/（CPI\*时钟周期\*10^6） MFLOPS=浮点操作次数/（执行时间\*10^6） 主频\*时钟周期=1 CPI指执行一条指令所需时钟周期数

1.6冯诺依曼计算机---现代计算机之父

1. 五大部件：运算器、控制器、存储器、输入设备、输出设备
2. 内部采用二进制表示指令和数据
3. 存储程序的思想（最核心）

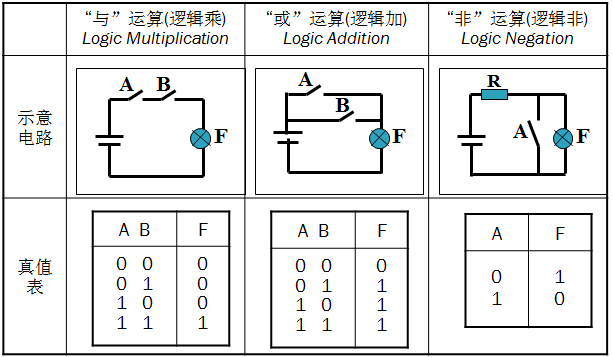
1.7大数据的特点

volume（大量）、velocity（高速）、variety（多样）、value（价值）

# 第二章

2.1二进制的基本运算

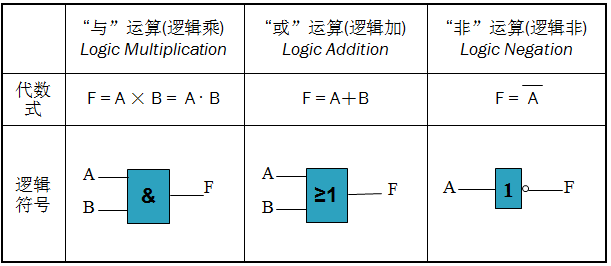
1. 二进制运算的示意电路和真值表



（“1”表示“真”，“0”表示“假”）

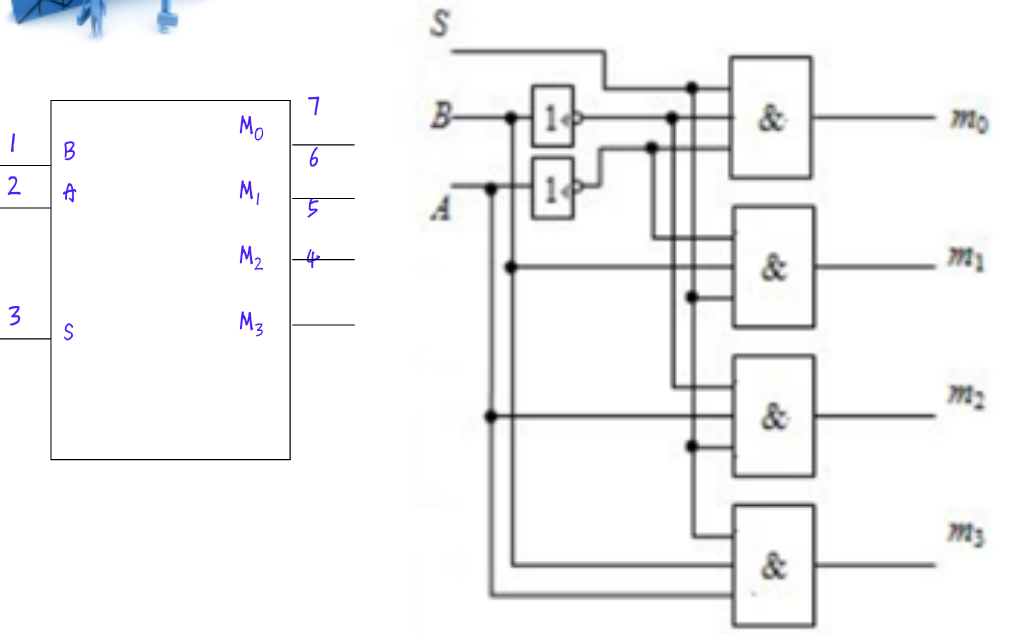
2.2二进制数据基本运算的硬件实现

1.代数式和逻辑符号



2.3二进制数据运算的应用-二四译码器

1. 一般地，n根地址线可产生2^n个输出，即可区分2^n个内存块。
2. 1K=210 , 1M=210 K, 1G=210M
3. 二四译码器的实现原理



2.4其他进制数据

1. 八进制计数制（0-7，逢8进1，后缀Q）

十六进制计数制（0-15，逢16进1，后缀H）

1. 十进制转换成R进制

整数部分除R取余，至商为零，倒取余数；小数部分乘R取整数部分，至小数部分为零，正取整数 部分。

1. 八进制转换成二进制

将二进制数据从小数点开始，分别向前向后三位分成一组，不足三位补零，然后写出对应的八进制即可。

1. 十六进制转换成二进制

将二进制数据从小数点开始，分别向前向后四位分成一组，不足四位补零，然后写出对应的十六进制即可。

2.5机器数

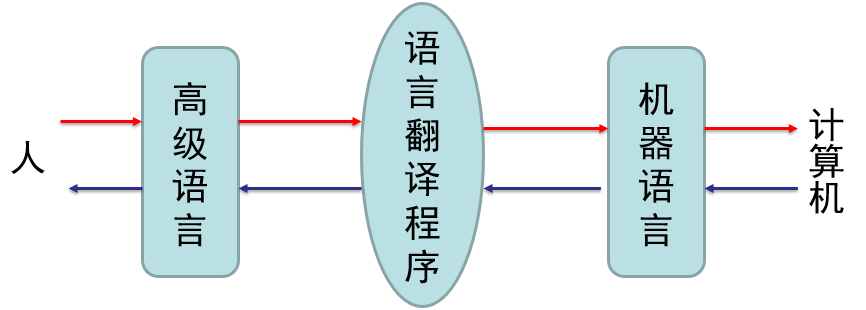
1. 机器数是有符号数据在计算机中的存储，0表示正号，1表示负号。
2. 规定：以后如果没有特别指明的话，规定一个字节，即八位来存储整数。
3. 原码：符号占一位，0表正，1表负，数值部分按二进制书写（占剩下部分）。
4. 补码：如果是正数，与原码相同；如果是负数，符号位不变，数据位取反。（补码中，一个字节能表示-128~+127这256个数，最大值为01111111，最小值为10000000）
5. 反码：如果真值为正数，与原码相同；如果真值是负数，符号位不变，数据位取反。（反码中，一个字节可以表示-127~+127这255个数）
6. N位存储空间表示补码时，可表示的数的范围是[-2^（n+1），+2^（n-1）-1]

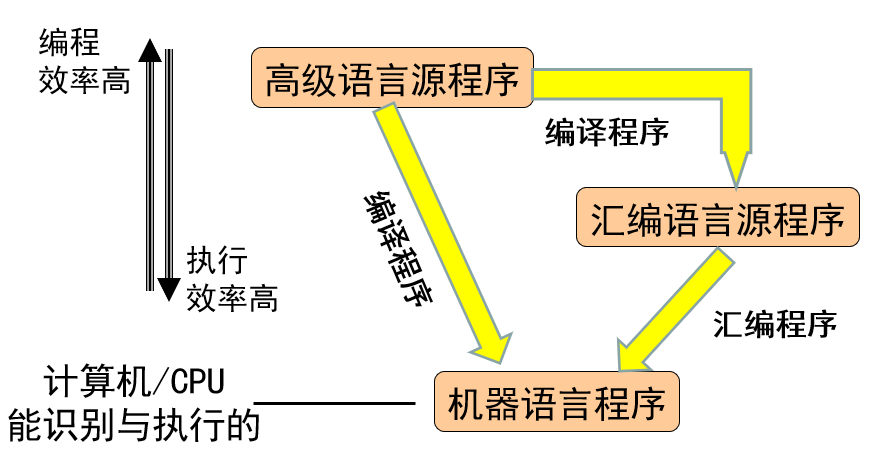
# 第三章

3.1计算机编程的基本概念

1. 程序：为解决某一问题而设计的一系列指令，能被计算机识别和执行。
2. 程序设计语言：用于书写计算机程序的语言，人与计算机打交道是交流信息的一类媒介和工具，又语句组成。
3. 计算机程序设计语言经过一个从低级语言到高级语言的发展过程。
4. 语言处理程序包括：机器语言、汇编语言等低级语言和高级语言等高级语言。
5. 机器语言：计算机直接使用的二进制形式的程序语言或机器代码。（执行效率最高，编程效率最低）
6. 汇编语言：一种面向机器的用符号表示的低级程序设计语言。相当于机器指令的助记符号，与机器语言很接近。
7. 高级语言：是易为人们所理解的完全符号化的程序设计语言。（执行效率最低，编程效率最低）

3.2语言的层次





1. 编译系统

由于计算机只能识别和执行机器语言，高级语言写的程序仍不能直接被计算机识别，必须经过“翻译”才能被执行。

1. 翻译方式：编译、解释
2. 负责这种翻译工作的程序称为语言处理程序：编译程序、解释程序，它们均是系统程序。

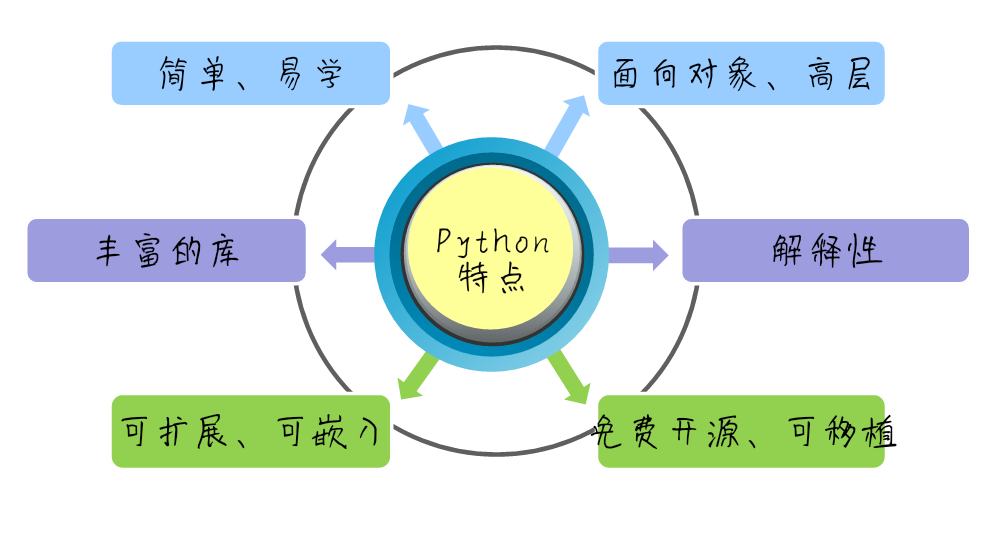
3.3初窥高级语言

1. 高级语言的特点
2. 高度封装，与低级语言相对
3. 使用一般人易于接受的文字来表示
4. 不是特指的某一种语言，而是包括很多种
5. 与计算机的硬件和指令系统无关
6. 执行速度慢
7. 高级语言的语法：①词法规则②语法规则
   1. 词法规则：字母表就是一个有穷字符集；词法规则试制单词符号的形成规则（字母、下划线打头的字母、数字和下划线构成的符号串。如a1、ave）
   2. 语法规则：语法规则规定了如何从单词符号形成更大的结构，换言之，语法规则是语法单位的形成规则。
8. 最常用的三种语句：
9. 表达式语句：由表达式组成，表达式是由数据、运算符、数字、分组符号（括号）、变量等组成的有意义的序列，且可求得数值。
10. 函数调用语句：由函数名和函数的实际参数组成。
11. 控制语句：有for、while、if语句（顺序结构、选择结构、循环结构）

|  |  |  |
| --- | --- | --- |
| 顺序结构 | 选择结构 | 循环结构 |
| b14 | b16  b15 |  |

3.4Python

1. Python特点



1. 在Python中要用半角英文，不可人为加入空格
2. Python的数据类型

\*数值型

\*布尔类型

\*列表

\*字符串类型

1. 通常Python整数类型的表示范围是-2147493648到+2147493647
2. 操作符：+ - \* / // \*\*（除法‘/’所得到的结果不是整数类型，而是浮点型，要想得到整数3，需使用‘//’）
3. 生成随机数：在Python在要产生随机数，首先要引入random模板



1. 布尔类型：

•产生的可能值：Ture、False

•布尔类型比较运算符：< > <= >= == !=

•布尔类型的逻辑运算符：not、and、or

1. 列表：序列的一种，数据元素类型可以不一样
2. 列表的声明形式L=[]（空列表）L=[1,3,5]
3. 列表中的元素以“，”相隔
4. 序列的通用操作
5. 索引：序列中所有元素都有索引号，索引号为正数时，从0开始递增，索引号为负数时，表示从序列的最后一个元素开始计数。
6. 分片：

•L[index1:ndex2]

•L[index1:]

•[:index2]

•[:]

•[index1:index2:stride]

1. 序列相加连接，序列乘数字表示序列重复多次。
2. 列表专有方法：S=[1,2]





1. 字符串类型

双引号表示的字符串中可以有单引号；

双引号表示的字符串中不能有双引号；

单引号表示的字符串中不能有单引号。

1. 字符串类型与数值型相互转化

---数值型转化成字符串类型：str()

---字符串类型转化成数值型：int() float()

1. Python语言的突出特点
2. 软件质量高：Python高度重视程序的可读性、一致性、代码的可重用性、可维护性高。
3. 提高开发效率：语法简单，使用方便。
4. 程序可移植性强
5. 标准库的支持：Python提供了强大的标准库支持，支持一系列复杂的编程任务。

3.5列举高级语言

Jave FORTRAN PPASCAL C语言 C++ C# Python Web

3.6算法

1.算法：算法是编程的第一步，在有限的步骤内求解某一个问题所使用的一组定义明确的规则。

2.算法的重要特征：

-有限性

-确切性

-具有输入项

-输出项

-可行性

3.算数的描述方法

-伪代码

-流程图

-自然语言

4.算法实例：解平方根算法

算法一：趋近法

算法二：二分法

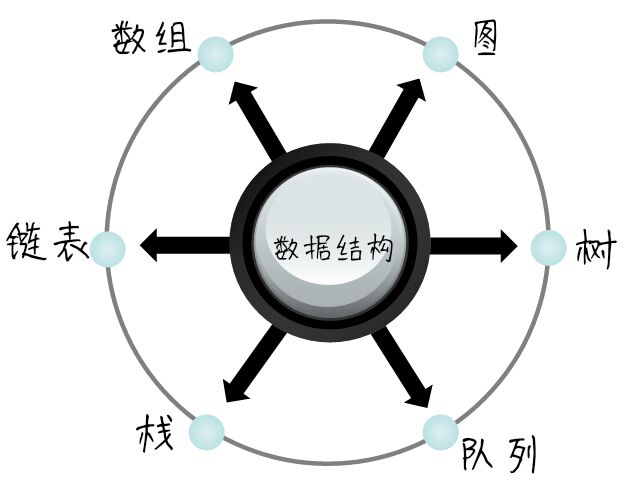
算法三：牛顿迭代法逼近近似解

# 第四章

4.1程序设计

1. 程序设计的四个步骤
2. 分析问题
3. 设计算法
4. 编码
5. 调试程序

4.2基本的数据结构



1. 线性表

•最常用且最贱的一种能够数据结构

•特点：（四个唯一）

①存在唯一的一个被称作“第一个”的数 据元素

②存在唯一的一个被称作“最后一个”的数 据元素

③除第一个之外的数据元素均只有一个前驱

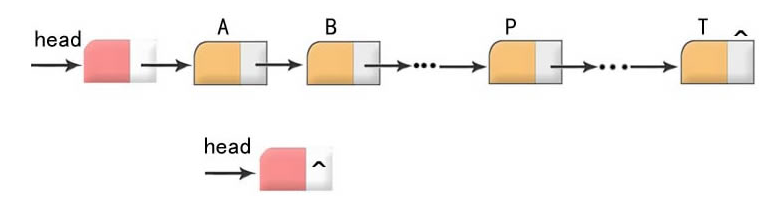
④除最后一个之外的数据元素均只有一个后继

•遍历

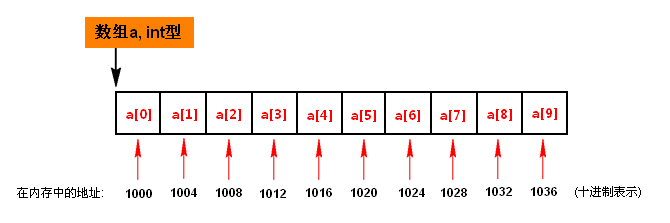
•插入

•删除

1. 链表（遍历，插入，删除）



1. 插入
   1. 判断插入位置是否合法
   2. 找到插入位置得前一个结点
   3. 在内存空间中分得一块空间，将待插入的值赋值给数据域，然后修改指针（先修改新结点的指针域，再修改原链表中对应节点的指针域）
2. 头结点：头结点不一定必须有，如果没有头结点，头指针在第一个元素所在结点的内存位置。
3. 链表优点：插入、删除很方便，无需挪动，仅需修改指针域。
4. 链表缺点：指针域会额外的占用内存空间，查找麻烦。
5. 数组



1. 遍历

每个元素的值均要被访问，并且仅被访问一次（不漏下也不重复）

1. 插入

①判断插入位置是否合法

②判断数组空间是否够用

③挪动位置（注意在最后一个合法位置上插入无需挪动；否则，从最后一个元素开始，依次向后挪动一个位置）

④对应位置元素赋值，长度加一

1. 删除

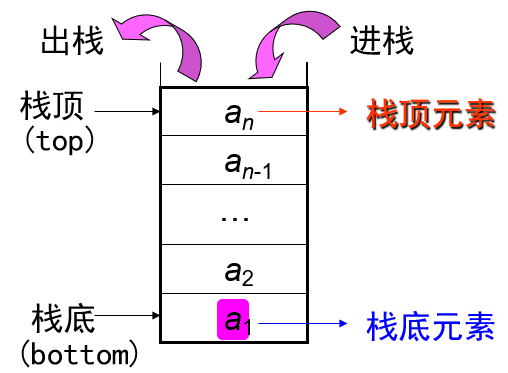
①判断删除位置是否合法

②挪动位置（注意，删除最后一个元素时无需挪动；否则，从对应位置开始依次向后挪动）

1. 长度减一
2. 栈(线性表)

•限定仅在表尾进行插入或删除操作

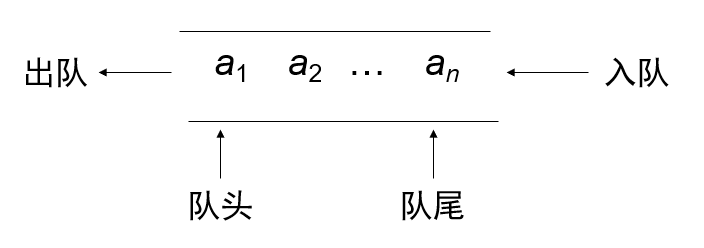
•后进先出（LIFO结构）

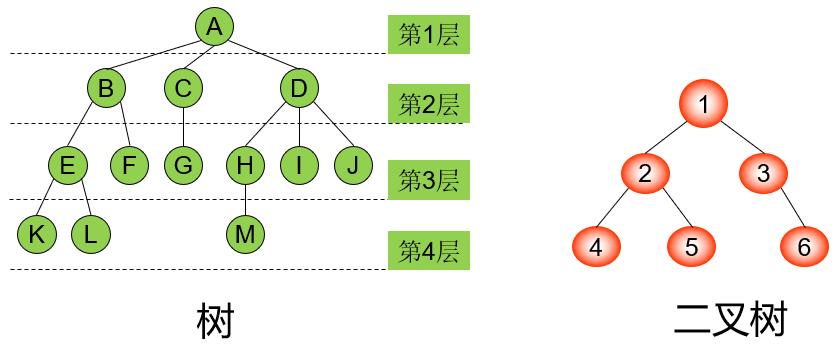


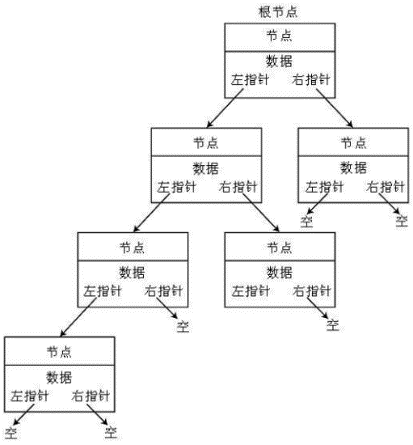
1. 队列（线性表）

•限定在表的一端插入，另一端删除

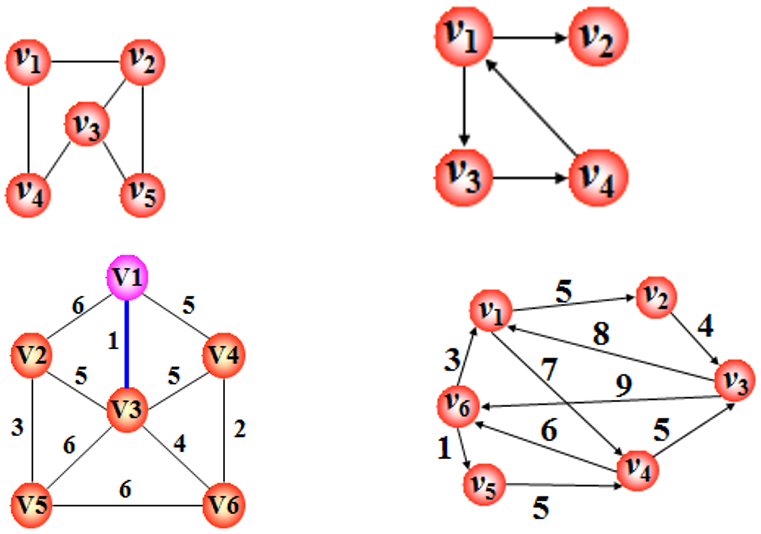
•后进先出（LIFO结构）



1. 树
   1. 
   2. 二叉树若为k层，则其结点数有2^k-1个，且为满二叉树。
   3. 树的存储



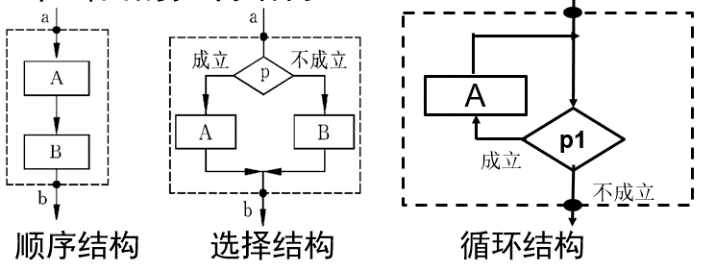
1. 图



4.3程序设计方法

1.面向过程（结构化）程序设计

•采用自上而下、逐步求精的设计方法和单入口、单出口的控制结构。



•模板易于识别，每个模板符合单入口单出口

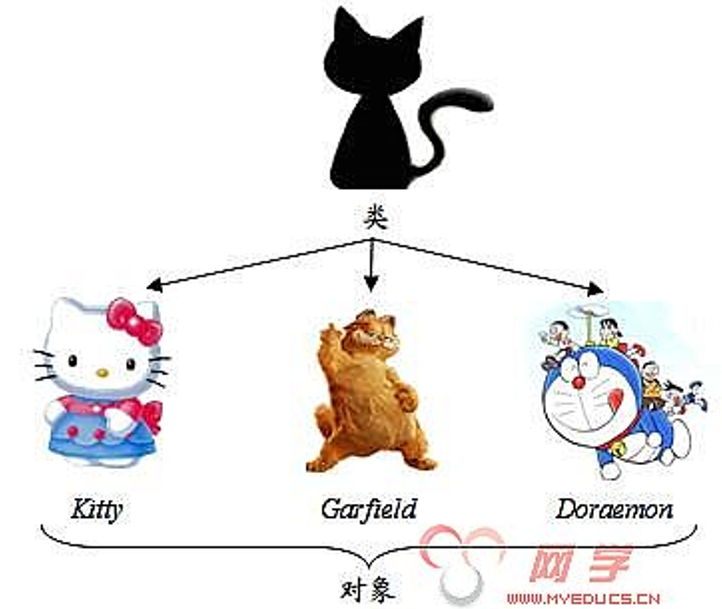
•应该控制goto语句使用

1. 面向对象程序设计

面向对象方法的出发点和基本原则，是尽可能地模拟现实世界中人类的思维方式，使开发软件的方法和过程尽可能地接近人类解决问题的方法和过程。它具备四个要点：

* 1. 对象是组成客观世界的基本元素
  2. 对象是属于某个类的
  3. 继承性
  4. 消息传递

1. 对象和类



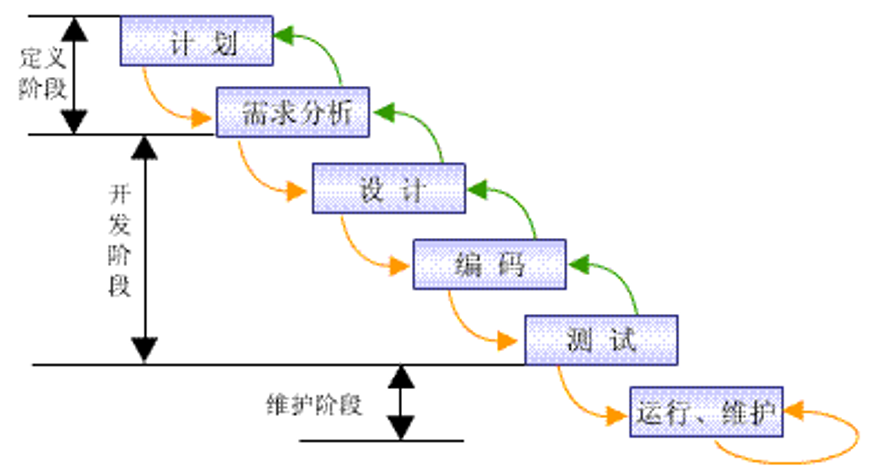
1. 基本手段—抽象

抽象是人类认识问题的基本手段之一，是选择性忽略。抽象的过程，就是对问题进行分析和认识的过程。

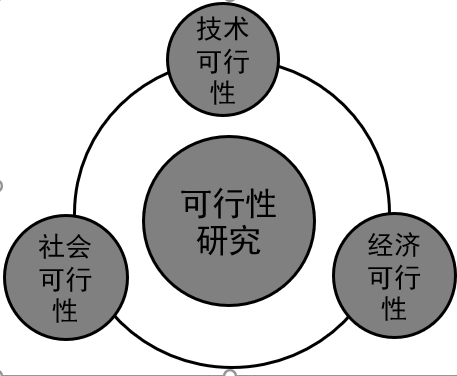
1. 面向对象的三大特征
   1. 封装
   2. 继承
   3. 多态

4.4软件生命周期

1.软件生命周期的层次—瀑布模型



1. 可行性研究



1. 需求分析
2. 软件需求分析就是把软件计划期间建立的软件可行性分析求精和细化，分析各种可能的解法。是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。
3. 软件需求包括三个不同的层次：业务需求、用户需求和功能需求（也包括非功能需求）
4. 软件测试

单元测试、白盒异常测试、接口测试、集成测试、稳定性测试、性能测试、仿真测试

1. 瀑布模型的优点
2. 为项目提供了按阶段划分的检查点
3. 当前一阶段完成后，您只需要去关注后续阶段
4. 瀑布模型的缺点
5. 在项目各个阶段之间很少有反馈
6. 只有在项目生命后期才能看到结果
7. 通过过多的强制完成日期和里程碑来跟踪各个项目阶段
8. 不适应用户需求的变化，对软件的开发影响较大
9. 软件过程
10. RUP模型：是一个面向对象且基于网络的程序开发方法论。
11. Scrum是一种迭代式增量软件开发过程，通常用于敏捷软件开发。Scrum包括了一系列实践和预定义角色的过程骨架。
12. ICONIX是尽早进入编码阶段，缩短分析设计 周期的软件开发方法

ICONIX过程是一种以最少步骤实现从用例到代码OOAD[object-oriented analysis and design]方法学，开覆盖了软件过程中所有关键的环节。

# 第五章

5.1认识操作系统

1.操作系统：是控制和管理计算机系统内各种硬件和软件资源，合理有效的组织计算机系统的工作，为用户提供一个使用方便可扩展的工作环境，从而起到连接计算机和用户的接口租用。

2.计算机的启动过程

-启动自检阶段

-初始化启动阶段

-启动加载阶段

-内核装载阶段

-登陆阶段

5.2操作系统分类

•批处理操作系统

•分时操作系统

•实时操作系统

•嵌入式操作系统

1.批处理操作系统

批处理是指计算机系统对一批作业自动进行处理的技术。 由于系统资源为多个作业所共享，其工作方式是作业之间自动调度执行。并在运行过程中用户不干预自己的作业，从而大大提高了系统资源的利用率和作业吞吐量 。

2.分时操作系统

将计算机系统的CPU时间划分成一些小的时间片，按时间片轮流把处理机分给各联机作业使用。

1. 特点：

-交互性

-多路性

-独立性

-及时性

3.实时操作系统

所谓"实时",即"及时",是指系统能及时(或即时)响应外部 事件的请求,在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。它必须保证实时性和高可靠性，对系统的效率则放在第二位。

1. 嵌入式操作系统

是指用于嵌入式系统的操作系统。嵌入式操作系统是一种用途广泛的系统软件，通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。嵌入式操作系统负责嵌入式系统的全部软、硬件资源的分配、任务调度，控制、协调并发活动。它必须体现其所在系统的特征，能够通过装卸某些模块来达到系统所要求的功能。目前在嵌入式领域广泛使用的操作系统有：嵌入式Linux、Windows Embedded、VxWorks等，以及应用在智能手机和平板电脑的Android、iOS等。

5.3操作系统对硬件的管理—CPU

1.操作系统硬件管理方式

•CPU

•内存和外存

•1/0设备

2.CPU管理

•为了满足系统的性能要求，提高任务处理的效率，现在主流的计算机通常都有一个后者多个CPU，每个CPU中又有多个核。

•然而核的数量是远远小于需要执行的程序的数量。

•多任务在同一个核上进行执行

-分时间片

-中断—进程

-调度—调度策略

3.进程

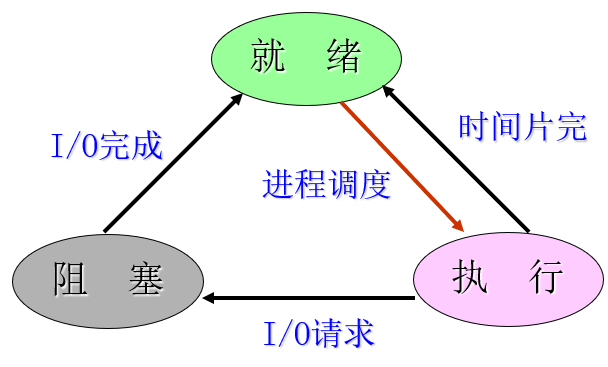
•在多道程序设计的环境下，为了描述程序在计算机系统内的执行情况，必须引入新的概念—进程。

•进程：是一个程序的一次执行，包含了其执行时所有的环境信息。

•进程是执行中的程序。

•进程的三种状态：

1. 就绪状态：进程已获得除CPU之外的所有必需的资源，一旦得到CPU控制权，立即可以运行。
2. 运行状态：进程已获得运行所必需的资源，它的程序正在处理机上执行。
3. 阻塞状态：正在执行的进程由于发生某事件而暂时无法执行，便放弃处理机而处于暂停状态，称该进程处于阻塞状态。



1. 进程调度

•调度程序，按照调度策略，动态地把CPU分配 给处于就绪队列中的进程，并将该进程从就绪态转换到运行态。

•对于不同的系统和系统目标，通常采用不同的调度算法。衡量调度策略的好坏，一个重要的指标是：周转时间（平均周转时间）

•进程调度策略

-先来先服务调度算法（FCFS）

-短任务优先调度算法（SJF）

5.4文件系统

1. 目录树结构

•文件目录用于标识系统中的文件及其物理地址，供检索时使用。

•对目录管理的要求如下：

-实现按名存取

-提高对目录的检索速度

-文件共享

-允许文件重名

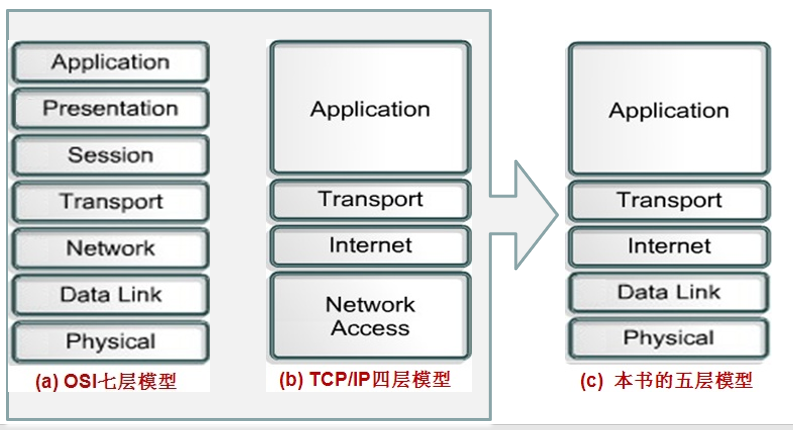
1. 目录查询技术

•当用户要访问一个已存文件时，系统首先会利用用户提供的文件名对目录进行查询，找出该文件控制块或对应索引结点，然后根据FCB或索引结点中所记录的文件物理地址，换算出文件在磁盘上的物理位置，最后通过磁盘驱动程序，将所需文件读入内存。

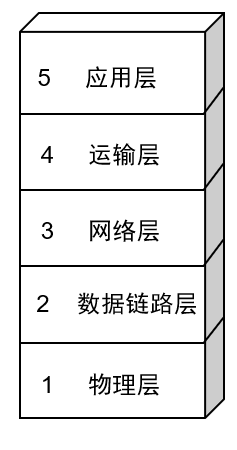
# 第六章

6.1无远弗届的网络

1. 计算机网络的重要功能
2. 连通性：彼此连通，交换信息
3. 共享：信息共享，软件共享，硬件共享
4. 物理的网络---五层模型
5. 逻辑的网络---万维网
6. 新型网络---物联网
7. 计算机网络的体系结构---物理网络的建立
8. 计算机网络体系结构：计算机网络各层及其协议的集合就是计算机网络及其部件所完成的功能的精码定义。
9. 协议：数据交换遵守的规则标准或约定。



1. 五层模型



1. 物理层：协调在物理媒体中传送比特流所需要的各种结构
2. 数据链路层：在两个临界点之间传送数据。
3. 网络层：在不同逐级之间的通信提供服务。
4. 运输层：负责向两个主机中进程之间的通信提供服务。
5. 分层的必要性

-提高工作效率

-提高容错性

-增强可扩展性

1. 细数五层模型---物理层

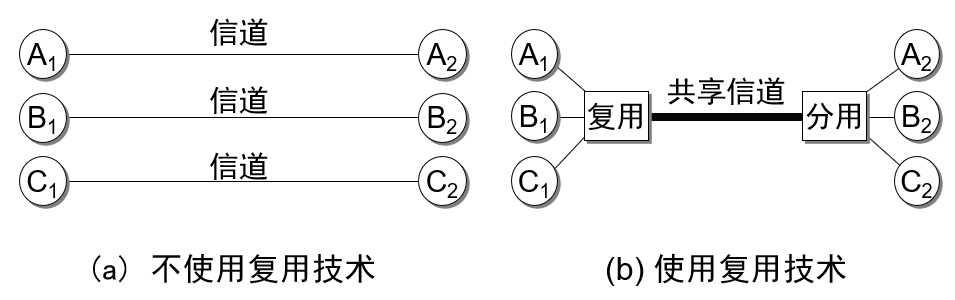
•物理层解决如何在连接各种计算机的传输媒体上传输数据比特流。

-网线

-无线

-数据线

•信道复用：传输线路同时传输多路信号。



•频分复用：用户在分配到一定的频带后，在通信过程中自始至终都占用这个频带。

•频分复用的所有用户在同样的时间占用不同的宽带资源。

1. 细数五层模型---数据链路层

•封装成帧

•透明传输

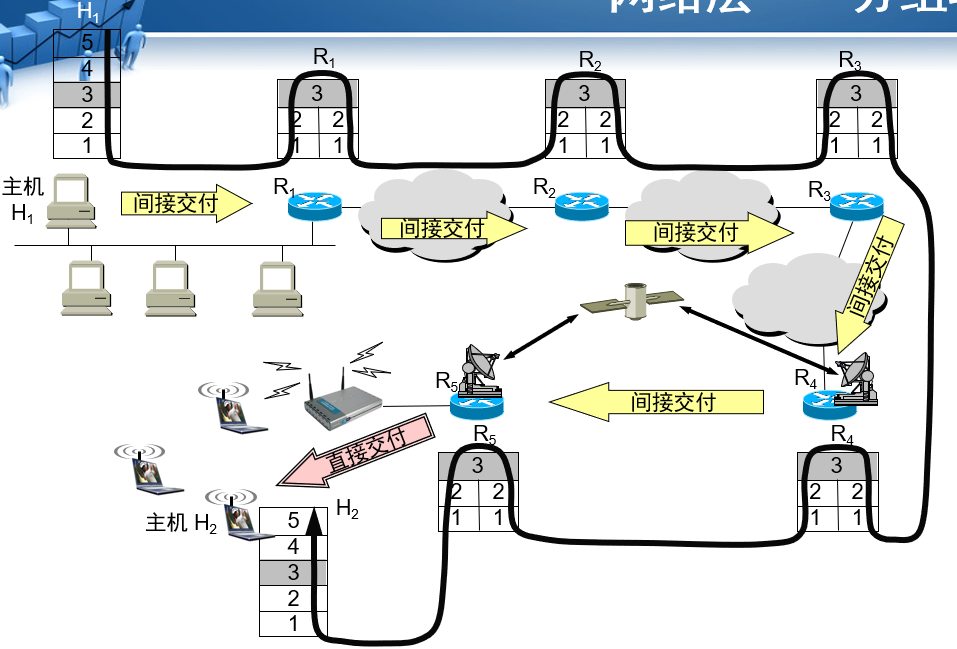
•差错检测（比特差错）

1. 细数五层模型---网络层

•网络层关注的是如何将分组从源端沿着网络路径送达目的端。

•网络层向上只提供简单灵活的、无连接的、尽最大努力交付的数据报服务。

•网络层-分组转发



•网络层-IP地址

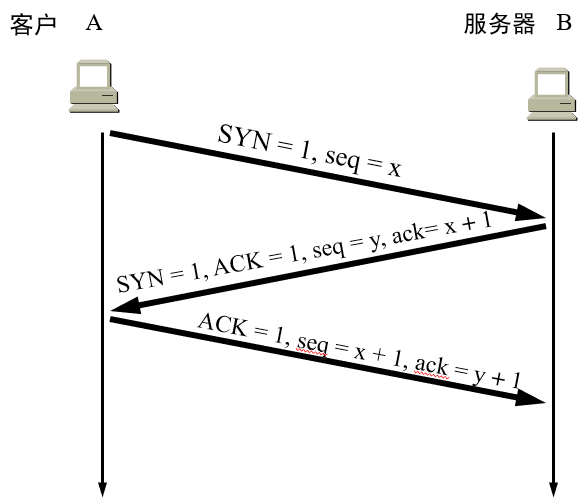
•网络层-路由表

1. 细数五层模型---传输层

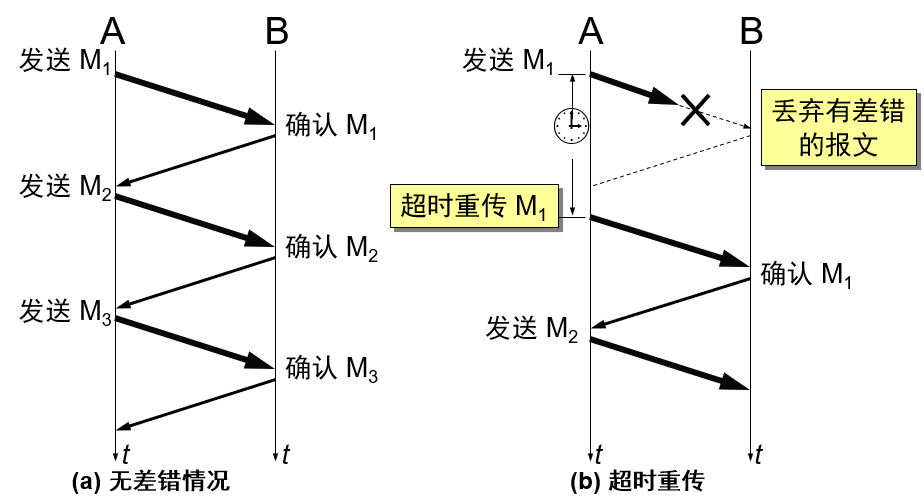
•传输层为相互通信的应用进程提供了逻辑通信。

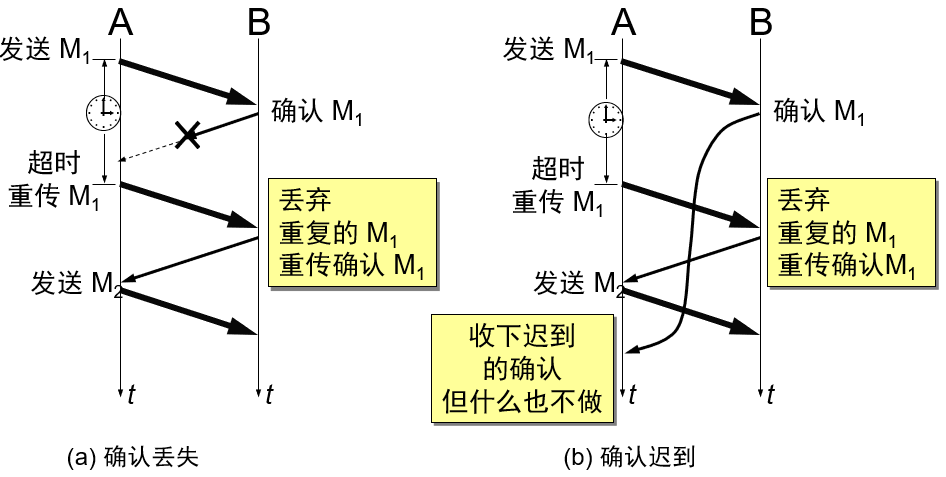
•传输层提供面向连接和无连接的服务。

•TCP三次握手



•可靠传输-停止等候协议



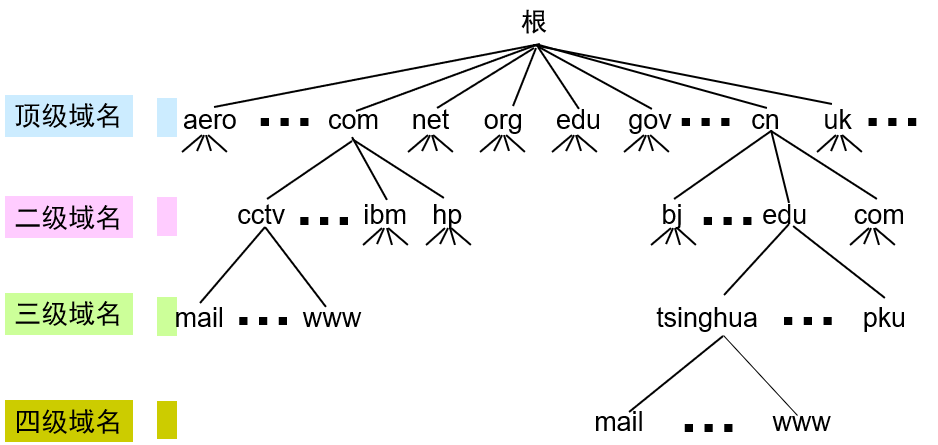


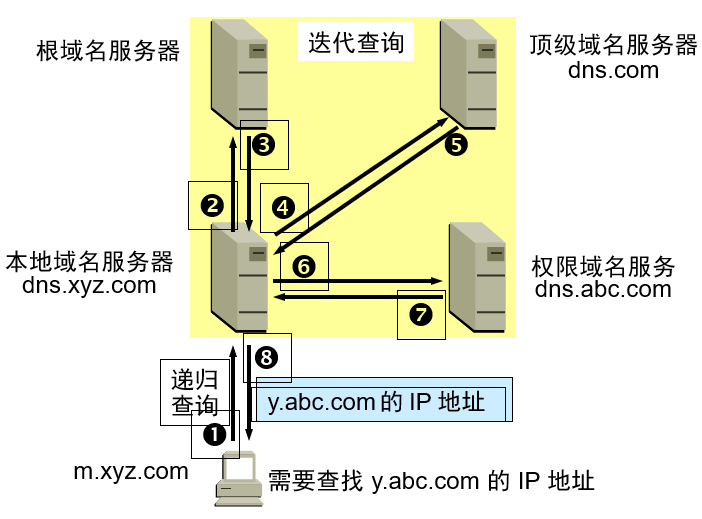
1. 细数五层模型---应用层

•应用层协议都是为了解决某一类实际应用题。

•网络当中解决某一类实际问题就是通过位于不同主机中应用层的多个进程之间的通信来完成。

1. 域名分析





6.2Web

1. 什么是URL？

URL就是统一资源定位符，是互联网上标准资源的地址，可以从互联网上得到的资源的位置和访问方法。

6.3初窥物联网

1. 物联网：物物相连的互联网。物联网的核心和基础仍然是互联网，是在互联网基础上的延伸和扩展的网络。
2. 寻址：每个物体都能寻址，IP地址数量支持
3. 通信：每个物体都能控制