1、码制的表示

码制	定点整数	定点小数	数码个数
原码	-（2^{n-1}-1）～+（2^{n-1}-1）	-（1-$2^{-(n-1)}$）～+（1-$2^{-(n-1)}$）	2^n-1
反码	-（2^{n-1}-1）～+（2^{n-1}-1）	-（1-$2^{-(n-1)}$）～+（1-$2^{-(n-1)}$）	2^n-1
补码	-2^{n-1}～+（2^{n-1}-1）	-1～+（1-$2^{-(n-1)}$）	2^n
移码	-2^{n-1}～+（2^{n-1}-1）	-1～+（1-$2^{-(n-1)}$）	2^n

2、浮点数的表示

（1）浮点数格式

阶码决定范围，阶码越长，范围越大；

尾数决定精度，尾数越长，精度越高。

（2）浮点数运算过程

对阶→尾数计算→格式化；

对阶：小数像大数看齐，尾数右移。

3、校验码

检验方式	校验码位数	校验码位置	检错	纠错	校验方式
奇偶校验	1	一般拼接在头部	可检奇数位错	不可纠错	奇校验：最终 1 的个数是奇数个； 偶校验：最终 1 的个数是偶数个；
CRC 循环冗余校验	生成多项式最高次幂决定	拼接在信息位尾部	可检错	不可纠错	模二除法求余数，拼接作为校验位
海明校验	$2^r \geqslant m+r+1$	插入在信息位中间	可检错	可纠错	分组奇偶校验

4、CPU 组成

CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成。

（1）运算器

- 算术逻辑单元 ALU：执行算术运算和逻辑运算。
- 累加寄存器 AC：暂存数据，为 ALU 提供工作区。
- 数据缓冲寄存器 DR
- 状态条件寄存器 PSW 归属有争议

（2）控制器

- 程序计数器 PC：存储下一条要执行指令的地址。
- 指令寄存器 IR：存储即将执行的指令。
- 指令译码器 ID。
- 时序部件。

5、CISC 与 RISC

CISC（复杂指令集）的特点：指令数量多，指令频率差别大，可变长格式，多种寻址方式，使用微码（微程序）实现，研制周期长。

RISC（精简指令集）的特点：指令数量少，频率接近，定长格式，单周期，多寄存器寻址，多通用寄存器，硬布线逻辑控制，适用于流水线。有效支持高级程序语言，优化编译。

6、流水线技术

流水线建立时间：第 1 条指令执行时间。

流水线周期：指令分段后，最长段时间。

流水线执行时间（默认使用理论公式，无答案时考虑实践公式）。

理论公式：流水线建立时间+（指令条数-1）*流水线周期。

实践公式：指令段数*流水线周期+（指令条数-1）*流水线周期。

吞吐率=指令条数/流水线执行时间。

最大吞吐率=流水线周期的倒数。

7、局部性原理

时间局部性：指程序中的某条指令一旦执行，不久以后该指令可能再次执行，典型原因是由于程

序中存在着大量的循环操作。

空间局部性：指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内，其典型情况是程序顺序执行。

8、常见存储器

（1）按内容存取

- 相联存储器（如 Cache）

（2）按地址存取

- 随机存取存储器（如内存）
- 顺序存取存储器（如磁带）
- 直接存取存储器（如磁盘）

（3）工作方式

- 随机存取存储器 RAM（如内存 DRAM）
- 只读存储器 ROM（如 BIOS）

9、Cache

在计算机的存储系统体系中，Cache 是（除寄存器以外）访问速度最快的层次。解决 CPU 与主存之间速度容量不匹配问题。

Cache 与主存映射三种方式：

	冲突率 （高、折中、低）	电路复杂度 （复杂、简单、折中）	其他
直接相联映象	高	简单	对应位置有数据即冲突
全相联映象	低	复杂	所有位置有数据即冲突
组相联映象	折中	折中	

10、主存编址计算

内存单元数个数=最大地址+1-最小地址。

内存编址内容：按字编址（每个存储单元存放内容为机器字长—题干定义）、按字节编址（每个存储单元内容为 1 字节即 8bit）。

内存总容量=存储单元个数*编址内容。

内存总容量=单位芯片容量*芯片片数。

芯片片数=内存总容量/单位芯片容量。

单位芯片容量=内容总容量/芯片片数。

11、输入输出技术

程序控制（查询）方式：分为无条件传送和程序查询方式。方法简单，硬件开销小，但 I/O 能力不高，严重影响 CPU 的利用率。

程序中断方式：与程序控制方式相比，中断方式因为 CPU 无需等待而提高了传输请求的响应速度。

DMA 方式：DMA 方式是为了在主存与外设之间实现高速、批量数据交换而设置的。DMA 方式比程序控制方式与中断方式都高效。

12、中断

中断处理（CPU 无需等待也不必查询 I/O 状态）：

（1）当 I/O 系统准备好以后，发出中断请求信号通知 CPU；

（2）CPU 接到中断请求后，保存正在执行程序的现场（保存现场），打断的程序当前位置即为断点；

（通过中断向量表-保存中断服务程序的入口地址）

（3）转入 I/O 中的服务程序的执行，完成 I/O 系统的数据交换；

（4）返回被打断的程序继续执行（恢复现场）。

13、可靠性

串联系统计算：R 总=R1*R2*...*Rn。

并联系统计算：R 总=1-(1-R1)(1-R2)...(1-Rn)。

N 模混联系统：先将整个系统划分为多个部分串联 R1、R2...等，再计算 R1、R2 内部的并联可靠性，带入原公式。

可靠性表示：MTTF/(1+MTTF)。

相关参数计算

（1）失效率计算

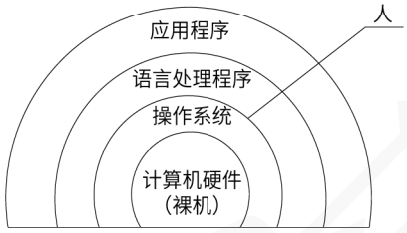
比如：假设统一型号的 1000 台计算机，在规定的条件下工作 1000 小时，其中 10 台故障。

其失效率λ=10/(1000*1000)=1*10-5

（2）千小时可靠度计算

千小时可靠性 R(t)=1-t*λ=1-1000*(1-10-5)=1-0.01=0.99

14、操作系统位置和功能



管理系统的硬件、软件、数据资源，控制程序运行，人机之间的接口，应用软件与硬件之间的接口。

15、嵌入式操作系统

特点：微型化、可定制（针对硬件变化配置）、实时性、可靠性、易移植性(硬件抽象层 HAL 和板级支撑包 BSP 支持)。

初始化过程：片级初始化→板级初始化→系统初始化。

16、线程

同一个进程当中的各个线程，可以共享该进程的各种资源，如内存地址空间、代码、数据、文件等，线程之间的通信与交流非常方便。

对于同一个进程当中的各个线程来说， 他们可以共享该进程的大部分资源。每个线程都有自己独立的 CPU 运行上下文和栈，这是不能共享的（程序计数器、寄存器和栈不能共享）。

17、PV 操作

P 操作：S=S-1(申请并锁定资源)；S<0(检查资源是否足够)。

V 操作：S=S+1(释放资源)；S<=0(检查是否有进程排队并通知排队进程)。

S 信号量：表示资源数，初值即为初始状态无操作时，资源的数量；信号量小于 0 的时候，还可以表示排队的进程数量。

18、前趋图与 PV 操作分析题技巧

针对箭线标注信号量，箭线的起点位置是 V 操作（即前趋活动完成后以 V 操作通知后继活动）；箭线的终点位置是 P 操作（即后继活动开始前以 P 操作检查前趋活动是否完成）。

19、死锁

死锁四大条件：互斥、保持和等待、不剥夺、环路等待。

假设 m 个进程各自需要 w 个 R 资源，系统中共有 n 个 R 资源，此时不可能形成死锁的条件是：m*(w-1)+1<=n。

20、页式存储的淘汰原则

页面淘汰时，主要依据原则（考试中默认按照此原则进行淘汰）：先淘汰最近未被访问的（访问位为 0），其次多个页面访问位为 0 时，则淘汰未被修改的（即修改位为 0，因为修改后的页面淘汰时代价更大）。

21、树形目录结构(多级目录结构)

绝对路径从根目录开始写起，并且该文件的全名即为绝对路径+文件名。

相对路径从当前位置下一级目录开始写起。

22、I/O 管理软件

硬件：完成具体的 I/O 操作。

中断处理程序：I/O 完成后唤醒设备驱动程序。

设备驱动程序：设置寄存器，检查设备状态。

设备无关 I/O 层：设备名解析、阻塞进程、分配缓冲区。

用户级 I/O 层：发出 I/O 调用。

23、分布式透明性

分片透明：用户不必关心数据是如何分片的即如何分片对用户是透明的。

复制透明：用户不用关心数据库在网络中各个结点的复制情况，被复制的数据的更新由系统自动完成。

位置透明：用户不必知道所操作的数据放在何处，即数据分配到哪个或哪些站点存储对用户是透明的。

局部映像透明性（逻辑透明）：用户不必关心局部 DBMS 支持哪种数据模型、使用哪种数据操纵语言，数据模型和操纵语言的转换是由系统完成的。

24、数据库三级模式两级映像

外模式-视图；模式-基本表；内模式-文件。

外模式-模式映射，保证数据逻辑独立性，即数据的逻辑结构发生变化后，用户程序也可以不修改。只需要修改外模式和概念模式之间的映像。

模式-内模式映射，保证数据物理独立性，即当数据的物理结构发生改变时，应用程序不用改变。只需要修改概念模式和内模式之间的映像。

25、数据库设计过程

需求分析阶段产物：数据流图、数据字典、需求说明书。

概念设计阶段产物：E-R 模型。

逻辑设计阶段产物：关系模式。设计依据：需求分析、E-R 模型、转换原则、规范化理论。（关系规范化是逻辑设计阶段的任务）

26、关系模式基本概念

简单属性：是原子的，不可再分的。

复合属性：可以细分为更小的部分（即划分为别的属性）。

单值属性：定义的属性对于一个特定的实体都只有单独的一个值。

多值属性：在某些特定情况下，一个属性可能对应一组值。

NULL 属性：表示无意义或不知道。

派生属性：可以从其他属性得来。

目或度：关系模式中属性的个数。

候选码（候选键）：唯一标示元组的属性集合，可以有多个。

主码（主键）：从候选键选择一个。

主属性与非主属性：组成候选码的属性就是主属性，其它的就是非主属性。

外码（外键）：其他关系模式的主键。

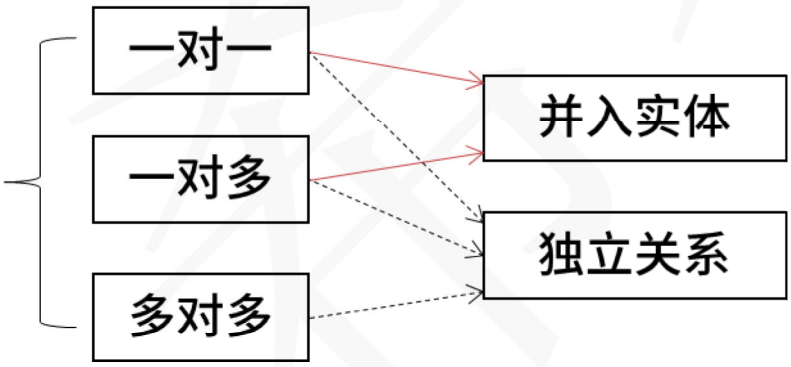
全码（ALL-Key）：关系模式的所有属性组是这个关系的候选码。

27、候选键

选择入度为 0（无函数依赖可推导得出的属性入度为 0）的属性集合，从该集合尝试推导出全部属性（可通过传递函数依赖等进行传递推导），如果可以，该集合为候选键，否则，该集合依次添加既有入度也有出度（既可被推导得出也可推导出其他属性）的中间结点，直到推导出所有属性为止，最终集合即为候选键。

28、E-R 图转关系模式转换原则

实体必须单独转换为 1 个关系模式。



联系根据类型不同：

（1）一对一联系的转换有 2 种方式。

独立的关系模式：并入两端主键及联系自身属性。（主键：任一端主键）

归并（任意一端）：并入另一端主键及联系自身属性。（主键：保持不变）

（2）一对多联系的转换有 2 种方式。

独立的关系模式：并入两端主键及联系自身属性。（主键：多端主键）

归并（多端）：并入另一端主键及联系自身属性。（主键：保持不变）

（3）多对多联系的转换只有 1 种方式

独立的关系模式：并入两端主键及联系自身属性。（主键：两端主键的组合键）

29、关系代数

笛卡尔积×：结果的属性列数是二者之和，结果的元组行数是二者乘积。

投影π：对垂直方向的属性列进行筛选。

选择σ：对水平方向的元组行进行筛选。

自然连接⋈：结果的属性列数是二者之和减去重复列数，结果元组是同名属性列取值相等的元组。

30、Amstrong 公理体系

A1.自反律（Reflexivity）：若 Y⊆X⊆U，则 X→Y 成立。

A2.增广律（Augmentation）：若 Z⊆U 且 X→Y，则 XZ→YZ 成立。

A3.传递律（Transitivity）：若 X→Y 且 Y→Z，则 X→Z 成立。

合并规则：由 X→Y，X→Z，有 X→YZ。（A2， A3）。

伪传递规则：由 X→Y，WY→Z，有 XW→Z。（A2， A3）。

分解规则：由 X→Y 及 Z⊆Y，有 X→Z。（A1， A3）。

31、规范化程度判断即范式判定依据

1NF：属性值都是不可分的原子值。（基本二维表）

2NF：在 1NF 基础上，消除了非主属性对候选键的部分函数依赖。（候选键是单属性至少满足 2NF）

3NF：在 2NF 基础上，消除了非主属性对候选键的传递函数依赖。（没有非主属性至少满足 3NF）

BCNF：在 3NF 基础上，消除了主属性对候选键的部分函数依赖和传递函数依赖。

32、查询

SELECT [ALL|DISTINCT] <目标表达式> [， <目标表达式>]…

FROM <表名> [，<表名>]…

[WHERE <条件表达式>]

[GROUP BY <列名 1> [HAVING <条件表达式>]]

[ORDER BY <列名 2> [ASC|DESC] …];

处理类型	处理子类	示例/语法

结果排序	升序或降序	ORDER BY　字段名　DESC|ASC
集函数	统计	COUNT([DISTINCT|ALL] <列名>)
	计算一列中值的总和	SUM([DISTINCT|ALL] <列名>)
	计算一列值的平均值	AVG([DISTINCT|ALL] <列名>)
	求一列值中的最大值	MAX([DISTINCT|ALL] <列名>)
	求一列值中的最小值	MIN([DISTINCT|ALL] <列名>)
对结果分组	将查询结果按列值分组	GROUP BY <列名>
对分组结果筛选	对分组结果筛选	HAVING <条件列达式>

33、事务特性（ACID）

原子性 A：事务是原子的，要么都做，要么都不做。

一致性 C：事务执行的结果必须保证数据库从一个一致性状态变到另一个一致性的状态。

隔离性 I：事务相互隔离，当多个事务并发执行时，任一事务的更新操作直到其成功提交的整个过程，对其他事务都是不可见的。

持续性 D：一旦事务成功提交，即使数据库崩溃，其对数据库的更新操作也将永久有效。

34、封锁协议

共享锁（S 锁、读锁）：若事务 T 对数据对象 A 添加了 S 锁，则只允许 T 读取 A，但不能修改 A。并且其他事务只能对 A 加 S 锁，不能加 X 锁。

排他锁（X 锁、写锁、独占锁）：若事务 T 对数据对象 A 添加了 X 锁，则只允许 T 读取和修改 A，其他事务不能再对 A 加任何锁。

35、OSI 七层模型

层次	名称	主要功能	主要设备及协议
7	应用层	实现具体的应用功能	POP3、FTP、HTTP、Telnet、SMTP DHCP、TFTP、SNMP、DNS
6	表示层	数据的格式与表达、加密、压缩	

5	会话层	建立、管理和终止会话	
4	传输层	端到端的连接	TCP、UDP
3	网络层	分组传输和路由选择	三层交换机、路由器 ARP、RARP、IP、ICMP、IGMP
2	数据链路层	传送以帧为单位的信息	网桥、交换机（多端口网桥）、网卡 PPTP、L2TP、SLIP、PPP
1	物理层	二进制传输	中继器、集线器（多端口中继器）

36、TCP/IP 协议簇四层模型

<table>
<tr><th>OSI 七层模型</th><th>TCP/IP 四层模型</th><th colspan="2">协议分层</th></tr>
<tr><td>应用层</td><td rowspan="3">应用层</td><td rowspan="3">POP3
110
FTP HTTP
20/21 80
Telnet SMTP
23 25</td><td rowspan="3">DHCP TFTP
67 69
SNMP DNS
161 53</td></tr>
<tr><td>表示层</td></tr>
<tr><td>会话层</td></tr>
<tr><td>传输层</td><td>传输层</td><td>TCP</td><td>UDP</td></tr>
<tr><td>网络层</td><td>网际层</td><td colspan="2">IP ICMP IGMP ARP RARP</td></tr>
<tr><td>数据链路层</td><td rowspan="2">网络接口层</td><td colspan="2" rowspan="2">CSMA/CD TokingRing</td></tr>
<tr><td>物理层</td></tr>
</table>

37、常见协议功能

协议名	默认端口	功能	特殊说明
HTTP	80	超文本传输协议，网页传输	不安全，结合 SSL 的 HTTPS 协议是安全的超文本传输协议，默认端口 443

Telnet	23	远程协议	不安全，SSH 是安全的远程协议
FTP	20 数据 21 控制	文件传输协议	不安全，结合 SSL 的 SFTP 是安全的文件传输协议。
POP3	110	邮件收取	附加多媒体数据时需采用 MIME（MIME 不安全，结合 SSL 的 MIME/S 是安全的多媒体邮件协议）。使用 WEB 方式收发电子邮件时必须设置账号密码登录。
SMTP	25	邮件发送	
DNS	53	域名解析协议，记录域名与 IP 的映射关系	本地客户端主机首查本机 hosts 文件 域名服务器首查本地缓存
DHCP	67	IP 地址自动分配	169.254.X.X 和 0.0.0.0 是无效地址
SNMP	161	简单网络管理协议	服务器仅发送消息给当前团体

协议名	**功能**	**特殊说明**
ARP	地址解析协议， IP 地址转换为 MAC 地址	ARP Request 请求采用广播进行传送 ARP Response 响应采用单播进行传送
RARP	反向地址解析协议， MAC 地址转 IP 地址	无
ICMP	因特网控制协议	PING 命令来自该协议
IGMP	组播协议	无

38、常见网络诊断命令

（1）ping：用于检查网络是否连通。

检查错误时，使用由近及远的原则，首先用 ping127.0.0.1 来检查本机 TCP/IP 协议栈，能 PING 通，说明本机协议栈无问题。

（2）tracert（linux: traceroute）：用于确定 IP 数据包访问目标所采取的路径，若网络不通，能定位到具体哪个结点不通；

（3）nslookup（查询 DNS 记录）；

（4）netstat：用于显示网络连接、路由表和网络接口信息。

（5）ipconfig（ linux: ifconfig）：显示 TCP/IP 网络配置值，如：IP 地址，MAC 地址，网关地址等。

① ipconfig 显示简要信息，不能查看 DHCP 服务开启情况。

② ipconfig /all 显示本机 TCP/IP 配置的详细信息，能为 DNS 和 WINS 服务器显示它已配置且所要使用的附加信息（如 IP 地址等），并且显示内置于本地网卡中的物理地址。

③ ipconfig 在 DHCP 服务中的应用：

ipconfig /all 显示详细信息 ，可查看 DHCP 服务是否已启用。

ipconfig /renew 更新所有适配器。DHCP 客户端手工向服务器刷新请求。

ipconfig /release 释放 IP 地址租约，只能在向 DHCP 服务器租用其 IP 地址的计算机上起作用。

④ ipconfig 在 DNS 服务中的应用：

ipconfig/ flushdns：清除本地 DNS 缓存。

ipconfig/ displaydns：显示本地 DNS 内容。

ipconfig/ registerdns：DNS 客户端手工向服务器进行注册。

39、特殊的 IP 地址

IP	说明
127 网段	回播地址，本地环回地址
主机号非全 0 和非全 1	可作为子网中的主机号使用
主机号全 0 地址	代表这个网络本身，可作为子网地址使用
主机号全 1 地址	特定子网的广播地址
169.254.0.0	保留地址，用于 DHCP 失效(Win)

0.0.0.0	保留地址，用于 DHCP 失效(Linux)

40、层次化网络

核心层：主要是高速数据交换，实现高速数据传输、出口路由，常用冗余机制。

接入层：主要是针对用户端，实现用户接入、计费管理、MAC 地址认证、MAC 地址过滤、收集用户信息，可以使用集线器代替交换机。

汇聚层 ：网络访问策略控制、数据包处理和过滤、策略路由、广播域定义寻址。

41、URL

URL：协议名://主机名.组名.最高层域名

组织模式	含义	地理模式	含义
com	商业组织	cn	中国
edu	教育机构	hk	中国香港
gov	政府机构	mo	中国澳门
mil	军事部门	tw	中国台湾
net	主要网络支持中心	us	美国
org	上述以外组织	uk	英国
int	国际组织	jp	日本

42、加密算法

常见对称密钥加密算法（共享密钥加密技术）：DES、 3DES(三重 DES)、 RC-5、IDEA、AES 算法。

常见非对称密钥加密算法（公开密钥加密技术）： RSA、ECC。

常见的摘要算法：MD5(128 位)，SHA(160 位)。

43、加密技术应用

数字信封：用接收方公钥加密使用的对称密钥。

数字签名：用发送方私钥签名，保证发送方身份真实性，发送者不可抵赖。与信息摘要结合，可防篡改。

信息摘要：单向散列值函数，防篡改，保证消息完整性。

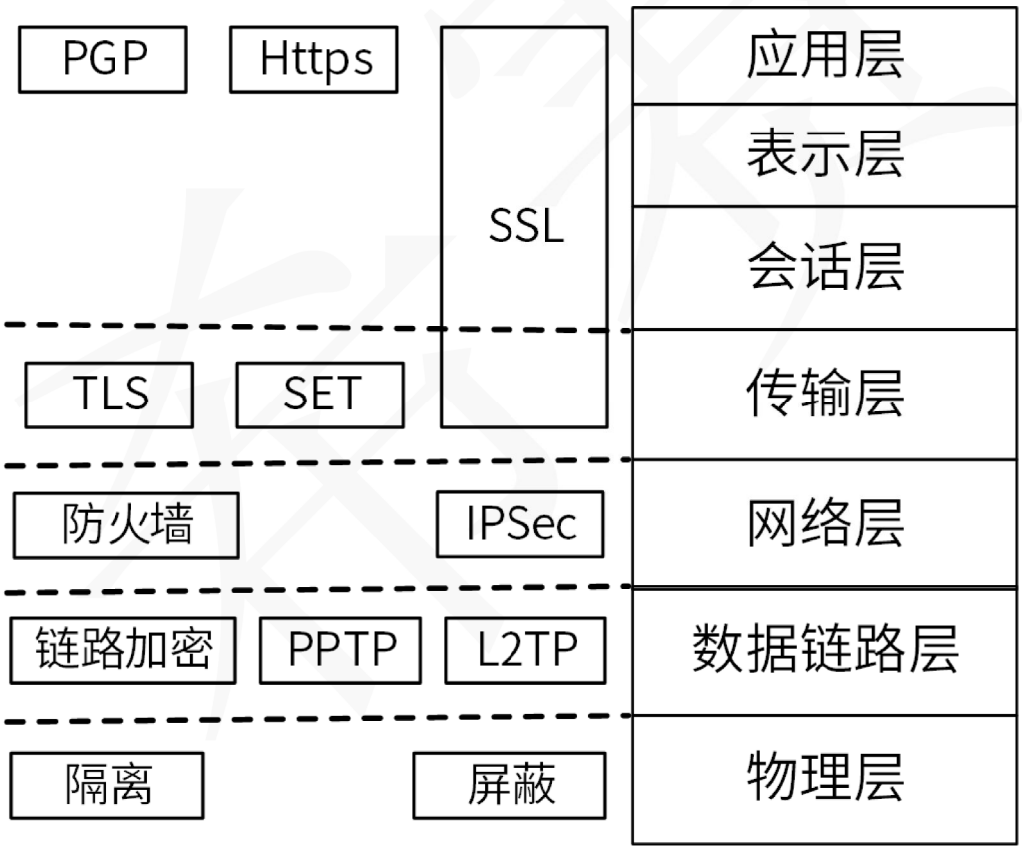
数字证书

数字证书的内容包括：CA 签名、用户信息（用户名称）、用户公钥等。

证书中的 CA 签名验证数字证书的可靠性、验证网站真伪。

用户公钥：客户端利用证书中的公钥加密，服务器利用自己的私钥解密。

44、网络安全协议分层



HTTPS：是 HTTP 协议与 SSL 协议的结合，默认端口号 443。

PGP：是邮件安全协议。

SET：是电子商务安全协议，涉及电子交易安全。

SSH：为建立在应用层基础上的安全协议。SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议。

45、网络攻击

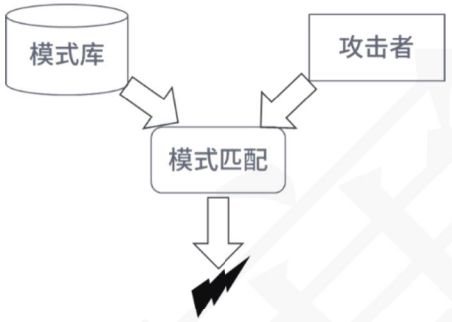
攻击类型	攻击名称	描述
被动攻击	窃听（网络监听）	用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。
	业务流分析	通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从而发现有价值的信息和规律。
	非法登录	有些资料将这种方式归为被动攻击方式。
主动攻击	假冒身份	通过欺骗通信系统（或用户）达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。黑客大多是采用假冒身份进行攻击。
	抵赖	这是一种来自用户的攻击，比如：否认自己曾经发布过的某条消息、伪造一份对方来信等。
	旁路控制	攻击者利用系统的安全缺陷或安全性上的脆弱之处获得非授权的权利或特权。
	重放攻击	所截获的某次合法的通信数据拷贝，出于非法的目的而被重新发送。【时间戳可防御】
WEB 服务器常见威胁	拒绝服务（DOS）	对信息或其它资源的合法访问被无条件地阻止。
	SQL 注入	SQL 注入攻击，就是通过把 SQL 命令插入到 Web 表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。其首要目的是获取数据库访问权限。 对用户输入做关键字过滤、Web 应用防火墙、定期扫描系统漏洞并及时修复都可以有效防御 SQL 注入攻击。

46、网络防御

防火墙技术：防外不防内，对于 DMZ 非军事区主要放置应用服务器（如邮件服务器，WEB 服务器）。

漏洞扫描：入侵者可以利用系统漏洞侵入系统，系统管理员可以通过漏洞扫描技术，及时了解系统存在的安全问题，并采取相应措施来提高系统的安全性。

入侵检测 IDS：根据攻击者行为和模式库记录的行为进行模式匹配，对攻击行为报警。



47、常见的软件开发模型

（1）瀑布模型

容易理解，管理成本低，每个阶段都有对应的成果产物，各个阶段有明显的界限划分和顺序要求，一旦发生错误，整个项目推倒重新开始。

适用于需求明确的项目，一般表述为需求明确、或二次开发，或者对于数据处理类型的项目。

（2）V 模型

强调测试贯穿项目始终，而不是集中在测试阶段。是一种测试的开发模型。

（3）喷泉模型

以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。

特点是迭代、无间隙。

（4）原型模型

典型的原型开发方法模型。适用于需求不明确的场景，可以帮助用户明确需求。

（5）增量模型

可以有多个可用版本的发布，核心功能往往最先完成，在此基础上，每轮迭代会有新的增量发布，核心功能可以得到充分测试。强调每一个增量均发布一个可操作的产品。

（6）螺旋模型

典型特点是引入风险分析。结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

48、敏捷方法

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法，适用于小团队和小项目，具有小步快跑的思想。

敏捷方法	特点
极限编程 XP	4 大价值观、5 个原则、12 个最佳实践
水晶法(Crystal)	认为每一个不同的项目都需要一套不同的策略、约定和方法论，认为人对软件质量有重要的影响（以人为本），因此随着项目质量和开发人员素质的提高，项目和过程的质量也随之提高。通过更好地交流和经常性交付，软件生产力得到提高。
开放式源码	程序开发人员在地域上分布很广
并列争球法（SCRUM）	把每 30 天一次的迭代称为一个“冲刺”，并按需求的优先级来实现产品。多个自组织和自治的小组并行地递增实现产品。协调是通过简短的日常情况会议来进行，就像橄榄球中的“并列争球”。
功用驱动开发方法 FDD	首席程序员和“类”程序员
自适应软件开发 ASD	核心是三个非线性的、重叠的开发阶段：猜测、合作与学习。ASD 有 6 个基本的原则：有一个使命作为指导；特征被视为客户价值的关键点；过程中等待是很重要的，因此“重做”与“做”同样关键；变化不被视为改正，而是被视为对软件开发实际情况的调整；确定的交付时间迫使开发人员认真考虑每一个生产的版本的关键需求；风险也包含其中。
敏捷统一过程 AUP	采用“大型连续”以及在“小型迭代”原理来构建软件系统，采用经典的 UP 阶段性（初始化、精化、构建、转换）

49、极限编程

极限编程是一种轻量级的开发方法。

它提出了四大价值观：沟通、简单、反馈、勇气。

五大原则：快速反馈、简单性假设、逐步修改、提倡更改、优质工作。

十二个最佳实践：计划游戏、隐喻、小型发布、简单设计、测试先行、重构、结对编程、集体代

码所有制、持续集成、每周工作 40 小时、现场客户和编码标准。

计划游戏：快速制定计划、随着细节的不断变化而完善。

小型发布：系统的设计要能够尽可能早地交付。

隐喻：找到合适的比喻传达信息。

简单设计：只处理当前的需求，使设计保持简单。

测试先行：先写测试代码，然后再编写程序。

重构：重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求。

结对编程。

集体代码所有制。

持续集成：可以按日甚至按小时为客户提供可运行的版本。

每周工作 40 小时。

现场顾客：系统最终用户代表应该全程配合 XP 团队。

编码标准。

50、开发方法

结构化开发方法：用户至上，严格区分工作阶段，每阶段有任务和结果，强调系统开发过程的整体性和全局性，系统开发过程工程化，文档资料标准化，自顶向下，逐步求精。

原型开发方法：适用于需求不明确的情况。

面向对象开发方法：更好的复用性，关键在于建立一个全面、合理、统一的模型，分析、设计、实现三个阶段界限不明确。

51、结构化设计任务

体系结构设计：定义软件系统各主要部件之间的关系。

数据设计：基于 E-R 图确定软件涉及的文件系统的结构及数据库的表结构。

接口设计（人机界面设计）：软件内部，软件和操作系统间以及软件和人之间如何通信。

过程设计：系统结构部件转换成软件的过程描述。确定软件各个组成部分内的算法及内部数据结构，并选定某种过程的表达形式来描述各种算法。

52、模块设计原则

保持模块的大小适中。

尽可能减少调用的深度。

多扇入，少扇出。

单入口，单出口。

模块的作用域应该在模块之内。

功能应该是可预测的。

53、内聚性

内聚类型	描　　述
功能内聚	完成一个单一功能，各个部分协同工作，缺一不可
顺序内聚	处理元素相关，而且必须顺序执行
通信内聚	所有处理元素集中在一个数据结构的区域上
过程内聚	处理元素相关，而且必须按特定的次序执行
瞬时内聚（时间内聚）	所包含的任务必须在同一时间间隔内执行
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚（巧合内聚）	完成一组没有关系或松散关系的任务

54、耦合性

耦合类型	描　　述
非直接耦合	两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的
数据耦合	一组模块借助参数表传递简单数据
标记耦合	一组模块通过参数表传递记录信息（数据结构）
控制耦合	模块之间传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	一组模块都访问同一全局简单变量，而且不是通过参数表传递该全局变量的信息
公共耦合	多个模块都访问同一个公共数据环境

内容耦合	一个模块直接访问另一个模块的内部数据；一个模块不通过正常入口转到另一个模块的内部；两个模块有一部分程序代码重叠；一个模块有多个入口

55、测试分类

（1）静态测试

桌前检查、代码走查、代码审查。

（2）动态测试

- 黑盒测试

等价类划分(确定无效与有效等价类，设计用例尽可能多的覆盖有效类，设计用例只覆盖一个无效类)、边界值分析(处理边界情况时最容易出错，选取的测试数据应该恰好等于、稍小于或稍大于边界值)、错误推测、因果图。

- 白盒测试：语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖、路径覆盖。

56、白盒测试

	定义	特点
语句覆盖	被测试程序中的每条语句至少执行一次。	对执行逻辑覆盖很低，一般认为是很弱的逻辑覆盖。
判定覆盖（分支覆盖）	被测程序每个判定表达式至少获得一次“真”值和“假”值（或者程序中每一个判定取“真”分支和取“假”分支至少通过一次。）	判定覆盖比语句覆盖更强一些。 判定可以是 1 个条件，也可以是多个条件的组合。
条件覆盖	每一个判定语句中每个逻辑条件的各种可能的值至少满足一次。	条件覆盖和判断覆盖没有包含关系。
判断/条件覆盖	判定中每个条件的所有可能取值（真/假）至少出现一次，并使每个判定本身的判定结果（真/假）也至少出现一次。	同时满足判定覆盖和条件覆盖

条件组合覆盖	每个判定中的各种可能值的组合都至少出现一次。	同时满足判定覆盖、条件覆盖、判定/条件覆盖。
路径覆盖	覆盖被测试程序中所有可能的路。	
基本路径测试	每一条独立路径都执行过（即程序中可执行语句至少执行一次）。	测试用例个数与环路复杂度一致。判定为关键控制结点，必须出现在基本路径中。
循环覆盖	循环中每个条件都得到验证。	注意数组参数可循环验证

57、特殊的测试阶段及任务

验收测试：有效性测试、软件配置审查、验收测试。

系统测试：恢复测试、安全性测试、强度测试、性能测试、可靠性测试和安装测试。

集成测试：模块间的接口和通信。

单元测试：模块接口、局部数据结构、边界条件、独立的路径、错误处理。

回归测试：修改软件后进行的测试，防止引入新的错误。

负载测试：对软件负载能力的测试。

压力测试：对软件超负荷条件下运行情况的测试。

58、McCabe 复杂度计算

McCabe 复杂度计算公式：V(G)=m-n+2，其中 m 是有向弧的条数，n 是结点数。

对于伪代码可以先转换为程序流程图，对程序流程图可以最终转换为结点图处理，转换时注意将交点的地方标注为新的结点，以最终的结点图带入公式结算其 McCabe 复杂度。

59、维护

更正性维护：针对真实存在并已经发生的错误进行的维护行为。

预防性维护：针对真实存在但还未发生的错误进行的维护。

适应性维护：指使应用软件适应信息技术变化和管理需求变化而进行的修改。企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。

完善性维护：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计

阶段中没有规定的功能与性能特征。

60、质量属性与其依从属性

功能性：适合性、准确性、互操作性、安全保密性。

可靠性：成熟性、容错性、易恢复性。

易用性：易理解性、易学性、易操作性。

效率：时间特性、资源利用性。

维护性：易分析性、稳定性、易测试性、易改变性。

可移植性：适应性、易安装性、一致性、易替换性。

61、CMMI（能力成熟度模型集成）阶段式

初始的：过程不可预测且缺乏控制。

已管理的：过程为项目服务。

已定义的：过程为组织服务。

定量管理的：过程已度量和控制。

优化的：集中于过程改进。

62、CMMI（能力成熟度模型集成）连续式

CL0（未完成的）：过程域未执行或未得到 CL1 中定义的所有目标。

CL1（已执行的）：其共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

CL2（已管理的）：其共性目标是集中于已管理的过程的制度化。

CL3（已定义级的）：其共性目标集中于已定义的过程的制度化。

CL4（定量管理的）：其共性目标集中于可定量管理的过程的制度化。

CL5（优化的）：使用量化（统计学）手段改变和优化过程域，以满足客户的改变和持续改进计划中的过程域的功效。

63、PERT 图

关键路径是图中源点至汇点的最长路径，关键路径的时间称之为项目工期，也表述为项目完成所需的最少时间。

总时差：在不延误总工期的前提下，该活动的机动时间。一般在图中，以最晚结束时间减去最早结束时间求取，或以最晚开始时间减去最早开始时间求取。

64、风险管理

（1）风险的特性：具有不确定性，可能会造成损失。

（2）风险的类别

项目风险涉及到各种形式的预算、进度、人员、资源以及客户相关的问题，并且可能导致项目损失。

技术风险涉及到技术相关的可能会导致项目损失的问题。

商业风险与市场因素相关。

社会风险涉及到政策、法规等因素。

（3）风险曝光度(RiskExposure)=错误出现率(风险出现率) X 错误造成损失(风险损失)。

65、沟通路径

有主程序员：n 个成员小组，1 个主程序员，普通程序员只需要与主程序员沟通。沟通路径：n-1。

无主程序员：n 个成员的项目小组，相互之间都可以沟通。沟通路径：n(n-1)/2。

66、COCOMO II 模型

67、数据流图

（1）数据流常见的 3 种错误

黑洞：加工只有输入没有输出；

奇迹：加工只有输出没有输入；

灰洞：加工中输入不足以产生输出。

（2）子图与父图保持平衡

父图与子图之间平衡是指任何一张 DFD 子图边界上的输入/输出数据流必须与其父图对应加工的输入/输出数据了保持一致。如果父图中某个加工的一条数据流对应于子图中的几条数据流，而子图中组成这些数据流的数据项全体正好等于父图中的这条数据流，那么它们仍然是平衡的。

68、面向对象基本概念

面向对象：对象+分类+继承+通过消息的通信。

对象：属性（数据）+方法（操作）+对象 ID。

封装：隐藏对象的属性和实现细节，仅对外公开接口（信息隐藏技术）。

类（实体类/控制类/边界类）：对对象的抽象。

接口：一种特殊的类，他只有方法定义没有实现。

继承与泛化：复用机制。

消息和消息通信：对象之间进行通信的一种构造叫做消息。消息是异步通信的。

重置/覆盖：在子类中重新定义父类中已经定义的方法。

重载：一个类可以有多个同名而参数类型不同的方法。

动态绑定：根据接收对象的具体情况将请求的操作与实现的方法进行连接（运行时绑定）。

多态：不同对象收到同样的消息产生不同的结果（软设一般只涉及过载多态-同一个名字在不同的上下文中所代表的含义不同）。

69、面向对象设计原则

单一职责原则：设计目的单一的类。

开放-封闭原则：对扩展开放，对修改封闭。

李氏（Liskov）替换原则：子类可以替换父类。

依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程。

接口隔离原则：使用多个专门的接口比使用单一的总接口要好。

组合重用原则：要尽量使用组合，而不是继承关系达到重用目的。

迪米特（Demeter）原则（最少知识法则）：一个对象应当对其他对象有尽可能少的了解。

重用发布等价原则：重用的粒度就是发布的粒度。

共同封闭原则：包中的所有类对于同一性质的变化应该是共同封闭的。一个变化若对一个包产生影响，则将对该包里的所有类产生影响，而对于其他的包不造成任何影响。

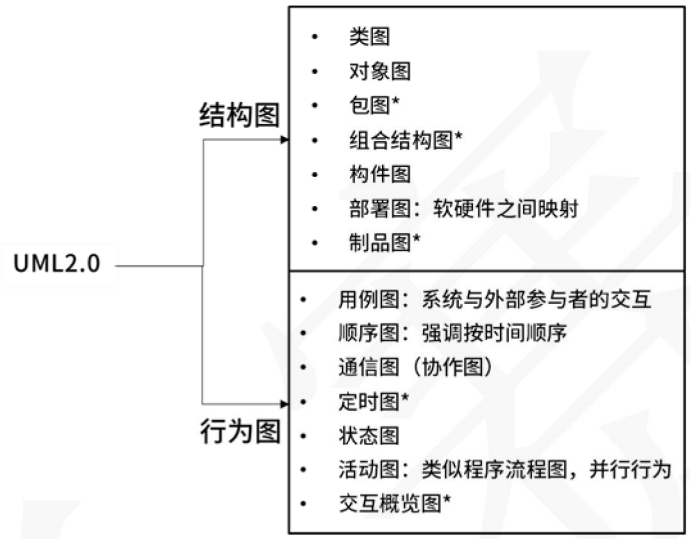
共同重用原则：一个包里的所有类应该是共同重用的。如果重用了包里的一个类，那么就要重用包中的所有类。

无环依赖原则：在包的依赖关系图中不允许存在环，即包之间的结构必须是一个直接的无环图形。

稳定依赖原则：朝着稳定的方向进行依赖。

稳定抽象原则：包的抽象程度应该和其稳定程度一致。

70、UML 图分类



71、类图关系

依赖关系：一个事物发生变化影响另一个事物。

泛化关系：特殊/一般关系。

关联关系：描述了一组链，链是对象之间的连接。

聚合关系：整体与部分生命周期不同。

组合关系：整体与部分生命周期相同。

实现关系：接口与类之间的关系。

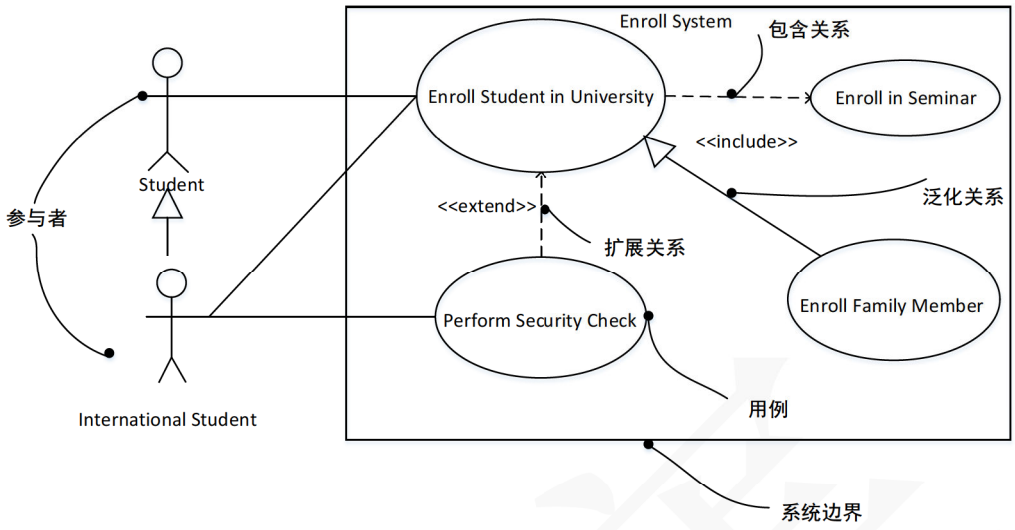
72、用例关系

包含关系：其中这个提取出来的公共用例称为抽象用例，而把原始用例称为基本用例或基础用例系，当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。

扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。

泛化关系：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，

子用例继承了父用例所有的结构、行为和关系。



73、设计模式分类

	创建型	结构型	行为型	
类	factory method 工厂方法模式	adapter 适配器模式（类和对象）	template method 模板方法模式	interpreter 解释器模式
对象	abstract factory 抽象工厂模式 prototype 原型模式 singleton 单例模式 builder 构建器模式	bridge 桥接模式 composite 组合模式 decorator 装饰模式 facade 外观模式 flyweight	chain of responsibility 职责链模式 command 命令模式 iterator 迭代器模式 mediator 中介者模式	memento 备忘录模式 observer 观察者模式 state 状态模式 strategy 策略模式 visitor

		享元模式 proxy 代理模式		访问者模式

74、创建型设计模式应用场景

设计模式名称	简要说明	速记关键字
Factory Method 工厂方法模式	定义一个创建对象的接口，但由子类决定需要实例化哪一个类。工厂方法使得子类实例化的过程推迟	动态生产对象
Abstract Factory 抽象工厂模式	提供一个接口，可以创建一系列相关或相互依赖的对象，而无需指定它们具体的类	生产成系列对象
Builder 构建器模式	将一个复杂类的表示与其构造相分离，使得相同的构建过程能够得出不同的表示	复杂对象构造
Prototype 原型模式	用原型实例指定创建对象的类型，并且通过拷贝这个原型来创建新的对象	克隆对象
Singleton 单例模式	保证一个类只有一个实例，并提供一个访问它的全局访问点	单实例

75、结构型设计模式应用场景

设计模式名称	简要说明	速记关键字
Adapter 适配器模式	将一个类的接口转换成用户希望得到的另一种接口。它使原本不相容的接口得以协同工作	转换接口

Bridge 桥接模式	将类的抽象部分和它的实现部分分离开来，使它们可以独立地变化	继承树拆分
Composite 组合模式	将对象组合成树型结构以表示“整体-部分”的层次结构，使得用户对单个对象和组合对象的使用具有一致性	树形目录结构
Decorator 装饰模式	动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代，比派生一个子类更加灵活	动态附加职责
Facade 外观模式	定义一个高层接口，为子系统中的一组接口提供一个一致的外观，从而简化了该子系统的使用	对外统一接口
Flyweight 享元模式	提供支持大量细粒度对象共享的有效方法	汉字编码
Proxy 代理模式	为其他对象提供一种代理以控制这个对象的访问	快捷方式

76、行为型设计模式应用场景

设计模式名称	简要说明	速记关键字
Chain of Responsibility 职责链模式	通过给多个对象处理请求的机会，减少请求的发送者与接收者之间的耦合。将接收对象链接起来，在链中传递请求，直到有一个对象处理这个请求	传递职责
Command 命令模式	将一个请求封装为一个对象，从而可用不同的请求对客户进行参数化，将请求排队或记录请求日志，支持可撤销的操作	日志记录，可撤销
Interpreter 解释器模式	给定一种语言，定义它的文法表示，并定义一个解释器，该解释器用来根据文法表示来解释语言中的句子	虚拟机的机制
Iterator 迭代器模式	提供一种方法来顺序访问一个聚合对象中的各个元素，而不需要暴露该对象的内部表示	数据集
Mediator 中介者模式	用一个中介对象来封装一系列的对象交互。它使各对象不需要显式地相互调用，从而达到低耦合，还可以	不直接引用

	独立地改变对象间的交互	
Memento 备忘录模式	在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，从而可以在以后将该对象恢复到原先保存的状态	游戏存档
Observer 观察者模式	定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动更新	联动
State 状态模式	允许一个对象在其内部状态改变时改变它的行为	状态变成类
Strategy 策略模式	定义一系列算法，把它们一个个封装起来，并且使它们之间可互相替换，从而让算法可以独立于使用它的用户而变化	多方案切换
Template Method 模板方法模式	定义一个操作中的算法骨架，而将一些步骤延迟到子类中，使得子类可以不改变一个算法的结构即可重新定义算法的某些特定步骤	框架
Visitor 访问者模式	表示一个作用于某对象结构中的各元素的操作，使得在不改变各元素的类的前提下定义作用于这些元素的新操作	数据与操作分离

77、顺序表和链表对比

性能类别	具体项目	顺序存储	链式存储
空间性能	存储密度	=1，更优	<1
	容量分配	事先确定	动态改变，更优
时间性能	查找运算	O（n）	O（n）
	读运算	O（1），更优	O（n），最好情况为1，最坏情况为 n

	插入运算	O（n），最好情况为 0，最坏情况为 n	O（1），更优
	删除运算	O（n）	O（1），更优

78、树的基本概念

双亲、孩子和兄弟：结点的子树的根称为该结点的孩子；相应地，该结点称为其子结点的双亲。具有相同双亲的结点互为兄弟。

结点的度：一个结点的子树的个数记为该结点的度。

叶子结点：也称为终端结点，指度为 0 的结点。

内部结点：度不为 0 的结点，也称为分支结点或非终端结点。除根结点之外，分支结点也称为内部结点。

结点的层次：根为第一层，根的孩子为第二层，依次类推，若某结点在第 i 层，则其孩子结点在第 i+1 层。

树的高度：一棵树的最大层次数记为树的高度（深度）。

79、二叉树的特性

在二叉树的第 i 层上最多有2^{i-1}个结点（i≥1）。

深度为 k 的二叉树最多有2^k-1 个结点（k≥1）。

对任何一棵二叉树，如果其叶子结点数为 n0，度为 2 的结点数为 n2，则 n0=n2+1。

对一棵有 n 个结点的完全二叉树的结点按层序编号，即从第 1 层到$[log_2 n]+1$层，每层从左到右依次编号。

80、特殊的二叉树

满二叉树：任何结点，或者是树叶，或者恰有两棵非空子树。

完全二叉树：最多只有最小面的两层结点的度可以小于 2，并且最下面一层的结点全都集中在该层左侧的若干位置。

平衡二叉树：树中任一结点的左右子树高度之差不超过 1。

查找二叉树：又称之为排序二叉树。任一结点的权值，大于其左孩子结点，小于其右孩子结点。中序遍历结果有序。

线索二叉树：在每个结点中增加两个指针域来存放遍历时得到的前驱和后继信息。

81、最优二叉树的概念

最优二叉树：又称为哈弗曼树，它是一类带权路径长度最短的树。

路径是从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

树的路径长度是从树根到每一个叶子之间的路径长度之和。结点的带权路径长度为从该结点到树根之间的路径长度与该结点权值的乘积。

树的带权路径长度为树中所有叶子结点的带权路径长度之和。

82、二叉树的遍历操作

前序遍历：又称为先序遍历，按根→左→右的顺序进行遍历。

后序遍历：按左→右→根的顺序进行遍历。

中序遍历：按左→根→右的顺序进行遍历。

层次遍历：按层次顺序进行遍历。

83、图的概念

（1）完全图

在无向图中，若每对顶点之间都有一条边相连，则称该图为完全图（complete graph）。

在有向图中，若每对顶点之间都有二条有向边相互连接，则称该图为完全图。

（2）强连通图：在有向图中，对于每一对顶点，从顶点 vi 到顶点 vj 和从顶点 vj 到顶点 vi 都存在路径，则称为强连通图。

84、图的遍历特点

深度优先遍历：

当以邻接矩阵作为存储结构时，深度优先搜索遍历图的时间复杂度为 $O(n^2)$

当以邻接表作为存储结构时，深度优先搜索遍历图的时间复杂度为 O(n+e)

广度优先遍历和深度优先搜索遍历图的运算时间复杂度相同，其不同之处仅仅在于对顶点的访问次序不同。

85、最小生成树与最短路径

（1）最小生成树解决方案

普里姆算法：找最短的边，直到把所有点连起来。注意：不能形成闭环。【贪心策略】

迪杰斯特拉算法：每一次只考虑从上一层节点到当前结点的最短路径。【贪心策略】

（2）最短路径问题解决方案

克鲁斯卡尔算法：从某个点开始，找现有点集合中最短的边。注意：不能形成闭环。【贪心策略】

86、常见算法策略

算法名称	关键点	特征	典型问题
分治法	递归技术	把一个问题拆分成多个小规模的相同子问题，一般可用递归解决。	归并排序、快速排序、二分搜索
动态规划法	最优子结构和递归式	划分子问题（最优子结构），并把子问题结果使用数组存储，利用查询子问题结果构造最终问题结果。	矩阵乘法、背包问题、LCS 最长公共子序列
贪心法	一般用于求满意解，特殊情况可求最优解（部分背包）	局部最优，但整体不见得最优。每步有明确的，既定的策略。	背包问题（如装箱）、多机调度、找零钱问题
回溯法	探索和回退	系统的搜索一个问题的所有解或任一解。有试探和回退的过程。	N 皇后问题、迷宫、背包问题

87、常见的算法执行所需时间的度量

$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<O(2^n)$

88、常见排序算法对比

类别	排序方法	时间复杂度		空间复杂度	稳 定 性
		平均情况	特殊情况	辅助存储	
插入排序	直接插入	O（n^2）	基本有序最优 O（n）	O（1）	稳定
	Shell 排序	O（$n^{1.3}$）	-	O（1）	不稳定
选择排序	直接选择	O（n^2）	-	O（1）	不稳定
	堆排序	O（$nlog_2n$）	-	O（1）	不稳定
交换排序	冒泡排序	O（n^2）	基本有序最优 O（n）	O（1）	稳定
	快速排序	O（$nlog_2n$）	基本有序最差（n^2）	O（1）	不稳定
归并排序		O（$nlog_2n$）	--	O（n）	稳定
基数排序		O（d（n+rd））	--	O（rd）	稳定

89、常见排序算法适用常见对比

若待排序列的记录数目 n 较小，可采用直接插入排序和简单选择排序。由于直接插入排序所需的记录移动操作较简单选择排序多，因而当记录本身信息量大时，用简单选择排序方法较好。

若待排记录按关键字基本有序，宜采用直接插入排序或冒泡排序。

当 n 很大且关键字位数较少时，采用基数排序较好。

若 n 很大，则应采用时间复杂度为 $O(nlog_2n)$的排序方法，例如快速排序、堆排序或归并排序。

快速排序目前被认为是内部排序中最好的方法，当待排序的关键字为随机分布时，快速排序的平均运行时间最短。

堆排序只需要一个辅助空间，并且不会出现在快速排序中可能出现的最快情况。

快速排序和堆排序都是不稳定的排序方法，若要求排序稳定，可选择归并排序。

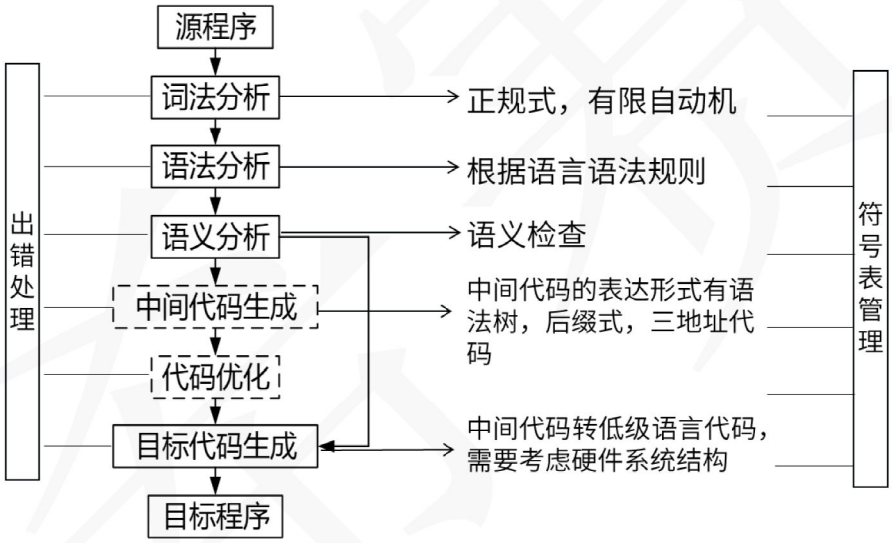
90、编译与解释的区别

编译方式下机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程，因此执行时效率较高。

解释方式下解释程序和源程序（或某种等价表示）要参与到程序的运行过程中，边解释边执行，执行效率较低。

即：解释方式，翻译程序不生成独立的目标程序，而编译方式则生成独立保持的目标程序。

91、编译过程



（1）符号表

符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的存在可以贯穿编译所有阶段。

（2）错误管理

静态错误：编译时所发现的程序错误，分为语法错误和静态语义错误。

语法错误包含：单词拼写错误、标点符号错误、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。

静态语义分析：运算符与运算对象类型不合法等错误。

动态错误：发生程序运行时，也叫动态语义错误。包括死循环、变量取零时做除数、引用数组元

素下标越界等错误。

92、文法和正规式

一般的程序设计语言属于上下文无关文法。

正规文法，表示的语言集合是正规集，正规集的规律可以用正规式表示。

正规式	正规集	举例
ab	字符串 ab 构成的集合	{ab}
a|b	字符串 a、b 构成的集合	{a , b }
a*	由 0 或多个 a 构成的字符串集合	{空 , a , aa , aaa , a…a(n 个 a)}
(a|b)*	所有字符 a 和 b 构成的串的集合	{空 , a , b , ab , aab , abb , baa, aba , …}
a(a|b)*	以 a 为首字符的 a、b 字符串的集合	{a , aa , ab , aab, aba , aaab , aaba , …}
(a|b)*abb	以 abb 结尾的 a、b 字符串的集合	{abb, aabb, babb, abaabb, abaabb, …}

93、传值调用和引用调用

传递方式	主要特点
传值调用	形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变
传址调用（引用调用）	形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值

94、常见的程序设计语言

Fortran 语言(第一个高级程序设计语言，科学计算，执行效率高)

Pascal 语言（结构化程序设计语言，表达能力强，Delphi）

C 语言（通用、结构化程序设计语言，指针操作能力强，高效）

Lisp 语言（函数式程序语言，符号处理，人工智能）

C++语言（C 语言基础上增加了类机制，面向对象，高效，与 C 兼容）

Java 语言（面向对象，中间代码，跨平台，通用的程序设计语言）

Python（面向对象，解释型程序设计语言，胶水语言，通用的脚本语言）

PHP（服务器端脚本语言，制作动态网页）

Ruby（简单快捷、面向对象、脚本语言）

Delphi（快速应用程序开发工具，可视化编程环境）

COBOL（数据处理领域最为广泛的程序设计语言，高级编程语言）

XML（可扩展标记语言，标准通用标记语言的子集 ）

PROLOG（逻辑式语言，间接性，表达能力强，建造专家系统、数据库、自然语言理解、智能知识库等）

注：C/C++常被用于操作系统开发；脚本语言是解释性语言。

95、保护范围和保护对象

法律法规名称	保护对象及范围	注意事项
著作权法	著作权 文学、绘画、摄影等作品	1、不需要申请，作品完成即开始保护 2、绘画或摄影作品原件出售（赠予）著作权还归原作者，原件拥有者有：所有权、展览权。
软件著作权法 计算机软件保护条例	软件著作权 软件作品	1、不需要申请，作品完成即开始保护 2、登记制度便于举证
专利法	专利权	需要申请，专利权有效期是从申请日开始计算
商标法	商标权	需要申请，核准之日起商标受保护
反不正当竞争法	商业秘密权	1、商业秘密包括技术与经营两个方面 2、必须有保密措施才能认定商业秘密

96、保护期限

客体类型	权力类型	保护期限
公民作品	署名权、修改权、保护作品完整权	没有限制
	发表权、使用权和获得报酬权	作者终生及其死亡后的 50 年（第 50 年的 12 月 31 日）
单位作品	发表权、使用权和获得报酬权	50 年（首次发表后的第 50 年的 12 月 31 日）
公民软件产品	署名权、修改权	没有限制
	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	作者终生及死后 50 年（第 50 年 12 月 31 日）。合作开发，以最后死亡作者为准
单位软件产品	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	50 年（首次发表后的第 50 年的 12 月 31 日）
注册商标		有效期 10 年（若注册人死亡或倒闭 1 年后，未转移则可注销，期满后 6 个月内必须续注）
发明专利权		保护期为 20 年（从申请日开始）
实用新型和外观设计专利权		保护期为 10 年（从申请日开始）
商业秘密		不确定，公开后公众可用

97、知识产权人确定-职务作品判定

情况说明		判断说明	归属
作品	职务	利用单位的物质技术条件进行创作，并由	除署名权外其他著作权归单位

	作品	单位承担责任的	
		有合同约定，其著作权属于单位	除署名权外其他著作权归单位
		其他	作者拥有著作权，单位有权在业务范围内优先使用
软件	职务作品	属于本职工作中明确规定的开发目标	单位享有著作权
		属于从事本职工作活动的结果	单位享有著作权
		使用了单位资金、专用设备、未公开的信息等物质、技术条件，并由单位或组织承担责任的软件	单位享有著作权
专利权	职务作品	本职工作中作出的发明创造	单位享有专利
		履行本单位交付的本职工作之外的任务所作出的发明创造	单位享有专利
		离职、退休或调动工作后 1 年内，与原单位工作相关	单位享有专利

98、知识产权人确定-其他

情况说明		判断说明	归属
作品软件	委托创作	有合同约定，著作权归委托方	委托方
		合同中未约定著作权归属	创作方
	合作开发	只进行组织、提供咨询意见、物质条件或者进行其他辅助工作	不享有著作权
		共同创作的	共同享有，按人头比例 成果可分割的，可分开申请
商标		谁先申请谁拥有（除知名商标的非法抢注） 同时申请，则根据谁先使用（需提供证据）	

	无法提供证据，协商归属，无效时使用抽签（但不可不确定）
专利	谁先申请谁拥有 同时申请则协商归属，协商不成则同时驳回双方的专利申请，最好的方式就是双方作为一个团体一起申请、享有专利

99、侵权判断的特殊要求

中国公民、法人或者其他组织的作品，不论是否发表，都享有著作权。

开发软件所用的思想、处理过程、操作方法或者数学概念不受保护。

著作权法不适用于下列情形：

- 法律、法规，国家机关的决议、决定、命令和其他具有立法、行政、司法性质的文件，及其官方正式译文；
- 时事新闻；
- 历法、通用数表、通用表格和公式。

100、典型的合理引用和侵权行为

不侵权	侵权
个人学习、研究或者欣赏； 适当引用； 公开演讲内容 用于教学或科学研究 复制馆藏作品； 免费表演他人作品； 室外公共场所艺术品临摹、绘画、摄影、录像； 将汉语作品译成少数民族语言作品或盲文出版。	未经许可，发表他人作品； 未经合作作者许可，将与他人合作创作的作品当作自己单独创作的作品发表的； 未参加创作，在他人作品署名； 歪曲、篡改他人作品的； 剽窃他人作品的； 使用他人作品，未付报酬； 未经出版者许可，使用其出版的图书、期刊的版式设计的。

注：合理使用不侵权，不需要通知作者、不需要付费，但仍然受著作权法保护。

更多备考资料和学习福利，可扫码添加希赛萧萧老师，申请入群