

网络工程师

冲刺提分秘籍



—— 2020年新版 ——

希赛内部资料

希赛网

希赛 | 专注在线教育19年

业务涵盖：

软考

PMP®

通信

建筑工程

金融财会

医药卫生

教资

学历

服务内容：

直播课堂

视频教程

智能题库

在线辅导

助教督学

社群交流

希赛
专注 **19** 年
软考在线培训

腾讯 新浪 网易
教育年度盛典获奖品牌

已服务企业**6万**家
培训学员**46万**人次

品质教学

蓄积深厚

服务无忧

值得信赖

更多软考学习干货、热点资讯、免费福利
尽在 **软考之家**



第一章 计算机硬件基础

1.1 本章重点

知识模块	知识点分布	重要程度
计算机体系结构	• 数据的表示（进制转换、原码、反码和补码）	• ★
	• 处理器组成（包括 ALU、AC、PC 等）	• ★★
	• 指令系统（寻址方式、CISC、RISC）	• ★★
	• 流水线	• ★★★
存储系统	• 高速缓存（实现、性能分析）	• ★
	• 主存储器（主存类型、主存容量计算）	• ★★★
	• 磁盘（接口类型）	• ★
可靠性计算	• 串联系统	• ★
	• 并联系统	• ★

1.2 考点精讲

1.2.1 进制的转换

数据的表示方法有二进制、八进制、十进制和十六进制等。网络工程师考试要求重点掌握这四种进制之间的数据转换方法。

R 进制，通常说法就是逢 R 进 1。可以用的数为 R 个，分别是 0, 1, 2, ..., R-1。例如八进制数的基数为 8，即可以用到的数码个数为 8，一位可以表示的数是 0, 1, 2, 3, 4, 5, 6, 7。二进制数的基数为 2，一位可以表示的数是 0 和 1。对于十六进制，一位可以表示的数为 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。

为了把不同的进制数分开表示，避免造成混淆，通常采用下标的方式来表示一个数的进制，如十进制数 18 表示为：(18)₁₀，八进制数 18 表示为：(18)₈。而如果是十六进制数，通常在数字的后面加大写“H”表示十六进制，例如，ABH 就表示十六进制数 AB。

进制间互转一般是无符号的数，三位二进制的数可以转换成一个八进制的数，四位二进制的数可以转换成一个十六进制数。

1.2.2 原码、反码、补码和移码

一个数在计算机中的二进制表示形式，叫作这个数的机器数。机器数是带符号的，在计

算机中用一个数的最高位作为符号位，正数为 0，负数为 1。

因为第一位是符号位，所以机器数的形式值就不等于真正的数值。

例如有机器数 10000011，其最高位 1 代表负，其真值是-3，而不是 131（10000011 转换成十进制等于 131）。

1. 原码

原码就是符号位加上真值的绝对值，即用第一位表示符号，其余位表示值。例如：假设用 8 位表示一个数值，则+11 的原码是 00001011，-11 的原码是 10001011。

2. 反码

反码通常是用来由原码求补码或者由补码求原码的过渡码。反码表示法和原码表示法一样是在数值前面增加了一位符号位（即最高位为符号位），正数的反码与原码相同，负数的反码符号位为 1，其余各位为该数绝对值的原码按位取反。

例如：+11 的反码是 00001011，-11 的反码为 11110100。

同理反码也存在“0”这个特殊的数值上，会有[0000 0000]反和[1111 1111]反两个编码表示 0。

3. 补码

在计算机系统中，**数值一律用补码来表示和存储**。原因在于，使用补码，可以将符号位和数值域统一处理；同时，加法和减法也可以统一处理。补码能唯一的表示一个数值。修复了 0 的存在两个编码的问题。

补码表示法和原码表示法一样是在数值前面增加了一位符号位（即最高位为符号位），正数的补码与原码相同，负数的补码是该数的反码加 1，这个加 1 就是“补”。

例如：+11 的补码是 00001011，-11 的补码为 11110101。

注意：其中负数补码转原码，符号位保留，其余各位取反+1。

4. 移码

移码（又叫增码）是符号位取反的补码，一般用指数的移码减去 1 来做浮点数的阶码，引入的目的是为了保证浮点数的机器零为全 0。

对于原码、反码和补码，假设用 n 位表示数值（二进制），则各种表示方法的表示范围如表 1-1 所示。

表 1-1 各种码制所表示数的范围

	定点整数	定点小数
原码	$-(2^{n-1}-1) \sim 2^{n-1}-1$	$-(1-2^{-(n-1)}) < X < 1-2^{-(n-1)}$
反码	$-(2^{n-1}-1) \sim 2^{n-1}-1$	$-(1-2^{-(n-1)}) < X < 1-2^{-(n-1)}$
补码	$-2^{n-1} \sim 2^{n-1}-1$	$-1 \leq X < 1-2^{-(n-1)}$

1.2.3 CPU 结构

1. 运算器

运算器通常是由 ALU（算术/逻辑单元，包括累加器、加法器等）、通用寄存器（不含地址寄存器）等组成。

ALU：进行算数运算和逻辑运算。

累加器 AC：暂时存放 ALU 运算的结果信息。

数据缓冲寄存器：用来暂时存放由内存储器读出的一条指令或一个数据字。反之，当向内储存存入一条指令或一个数据字时，也暂时将它们存放在数据缓冲寄存器中。

状态条件寄存器（PSW）保存由算术指令和逻辑指令运算的状态和程序的工作方式。

2. 控制器

控制器的组成包含程序计数器（PC）、指令寄存器（IR）、指令译码器、时序部件等。

程序计数器（PC）：存放的是下一条指令的地址。

指令寄存器（IR）：用来保存当前正在执行的一条指令。

指令译码器：指令中的操作码经过指令译码器译码后，即可向操作控制器发出具体操作的特定信号。

时序部件：为指令的执行产生时序信号。

3. 总线

微型计算机通过系统总线将各部件连接到一起，实现了微型计算机内部各部件间的信息交换。

数据总线 DB 用于传送数据信息。

地址总线 AB 是专门用来传送地址的，地址总线的位数决定了 CPU 可直接寻址的内存空间大小。一般来说，若地址总线为 n 位，则可寻址空间为 2^n 字节。

控制总线 CB 用来传送控制信号和时序信号。

1.2.4 指令系统

指令系统是中央处理器所有指令的集合，通常一条指令可分解为操作码和地址码两部分，操作码确定指令的操作类型，地址码确定指令所要处理操作数的位置。

1. 寻址方式

指令系统中采用不同寻址方式的目的是扩大寻址空间并提高编程灵活性。常见的寻址方式如图 1-1 所示：

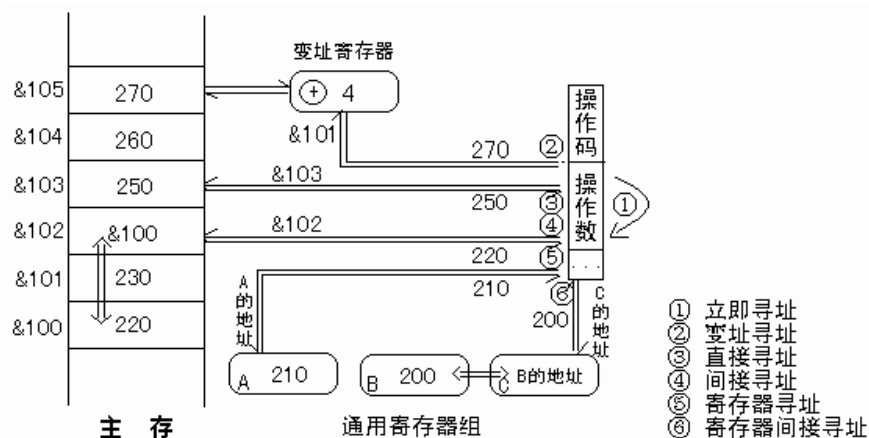


图 1-1 几种寻址方式比较

(1) 立即寻址方式：通常直接在指令的地址码部分给出操作数。

(2) 内存寻址

直接寻址方式：在指令中直接给出参加运算的操作数或运算结果所存放的主存地址。

间接寻址方式：在指令中给出操作数地址的地址。

变址寻址方式：变址寻址就是变址寄存器中的内容加地址码中的内容即可完成寻址。

(3) 寄存器寻址

寄存器直接寻址：指令在执行过程中所需要的操作数来源于寄存器。

寄存器间接寻址：寄存器存放的是操作数在主存的地址。

2. RISC 和 CISC

RISC 和 CISC 是目前设计制造 CPU 的两种典型技术。

表 1-2 RISC 与 CISC 比较

指令系统	指 令	寻址方式	实现方式
CISC	指令种类多，使用频率相差大，采用变长编码	支持多种，访（主）存指令多	微程序控制
RISC	指令种类少，定长，使用频率相互接近	只支持少量寻址方式，并且控制了访问内存的指令数量	硬布线逻辑控制

3. 流水线

流水线是指在程序执行时多条指令重叠进行操作的一种准并行处理实现技术。即可以同时为多条指令的不同部分进行工作，以提高各部件的利用率和指令的平均执行速度。

(1) 流水线指令执行时间

标准算法： $T = \text{第一条指令执行所需时间} + (\text{指令条数} - 1) \times \text{流水线周期}$

关于流水线的周期，我们需要知道的是，流水线周期为指令执行阶段中执行时间最长的一段。

例如指令流水线把一条指令分为取指令、分析和执行 3 个部分，且 3 个部分的时间分别是取指令 2ns、分析 2ns 及执行 1ns。那么最长的是 2ns，因此 100 条指令全部执行完毕所需要的时间就是 $(2\text{ns} + 2\text{ns} + 1\text{ns}) + (100 - 1) \times 2\text{ns} = 203\text{ns}$ 。

(2) 流水线的技术指标

吞吐率：指的是计算机中的流水线在特定的时间内可以处理的任务数量。 $TP = n / T_k$ (n 为指令条数， T_k 为流水线方式时间)，其中理论上的最大吞吐率是： $1 / \text{流水线周期}$ 。

加速比：完成一批任务，不使用流水线所用的时间与使用流水线所用的时间之比称为流水线的加速比。 $S = T_s / T_k$ (T_s 为顺序执行时间， T_k 为流水线方式时间)。

流水线的效率指的是流水线的设备利用率。

1.2.5 存储系统

1. 存储方式

存储器中数据常用的存取方式有顺序存取、直接存取、随机存取和相联存取等四种。

顺序存取：

存储器的数据是以记录的形式进行组织，对数据的访问必须按特定的线性顺序进行。磁带存储器的存取方式就是顺序存取。

直接存取：

共享读写装置，但是每个记录都有一个唯一的地址标识，共享的读写装置可以直接移动到目的数据块所在位置进行访问。访问时间跟数据位置有关。磁盘存储器采用的这种方式。

随机存取：

存储器的每一个可寻址单元都具有唯一地址和读写装置，系统可以在相同的时间内对任意一个存储单元的数据进行访问，而与先前的访问序列无关。主存储器采用的是这种方式。

相联存取：

也是一种随机存取的形式，但是选择某一单元进行读写是取决于其内容而不是其地址。Cache 采用该方法进行访问。相联存储器是 Cache 一部分，Cache 中有按内容寻址的相联存储器，用于存放与 Cache 中数据相对应的主存地址，可以快速检索、判断 CPU 读取的某个字当前是否存在于 Cache 中。

2. 存储设备

传统意义上存储器分为 RAM 和 ROM。

(1) RAM 和 ROM

RAM 是随机存储器，数据可读可写，一旦掉电，数据将消失。ROM 是只读存储器，掉电后数据依然保存。

RAM 有静态和动态两种：

静态 RAM 只要上电后信息不丢失，无须刷新电路过程，消耗较多功率，价格也较高。常作为芯片中的 Cache 使用。

最常用的动态 RAM 需要上电后，再定时刷新电路才能保持数据，而动态 RAM 集成度高、存储密度高、成本低，功耗低，适于作大容量存储器。常用在内存中。

(2) Cache

在计算机执行时，需要从主存中读取指令和数据，需要将外存的数据读入内存中，这些读取的过程都是造成计算机性能下降的瓶颈，为了尽可能减少速度慢的设备对速度快的设备的约束，可以利用高速缓存 Cache 技术。

(3) 磁盘

与计算机技术一样，存储技术也在不断发展，在现代计算机中，最常见的存储介质包括机械硬盘、光盘、磁带，固态硬盘 SSD 等。

3. 存储计算

实际的存储器总是由一片或多片存储芯片+控制电路构成的。芯片数量 \geq 存储器容量/存储芯片容量。

如果存储器有 256 个存储单元，那么它的地址编码为 0~255，对应的二进制数是 00000000~11111111，需要用 8 位二进制来表示，也就是地址宽度为 8 位，需要 8 根地址线。存储器中所有存储单元的总和称为这个存储器的存储容量，存储容量的单位是 B、KB、MB、GB 和 TB 等。

例如：按某存储器字节编址，地址从 A4000H 到 CBFFFH，则表示有 (A4000—CBFFF)+1 个字节，即 28000H 个字节，转换为十进制是 160KB。若用 16K \times 4bit 的存储器芯片构成该内存，共需 160K \times 8/16K \times 4=20 片。

1.2.6 系统可靠性

计算机系统是一个复杂的系统，而且影响其可靠性的因素也非常繁复，很难直接对其进行可靠性分析。但通过建立适当的数学模型，把大系统分割成若干子系统，可以简化其分析过程。

1. 串联系统

假设一个系统由 n 个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作，这种系统称为串联系统，如图 1-2 所示。

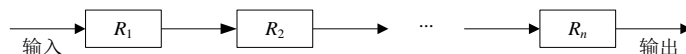


图 1-2 串联系统

设系统各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠性 $R = R_1 \times R_2 \times \dots \times R_n$ 。

2. 并联系统

假如一个系统由 n 个子系统组成，只要有一个子系统能够正常工作，系统就能正常工作，如图 1-3 所示。

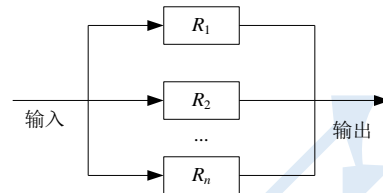


图 1-3 并联系统

设系统各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠性： $R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$ 。

第二章 操作系统

2.1 本章重点

知识模块	知识点分布	重要程度
进程管理	• 进程状态	• ★★★
	• 死锁问题	• ★★★
	• PV 操作	• ★
存储管理	• 页面置换算法	• ★
文件管理	• 绝对路径和相对路径	• ★★★
设备管理	• IO 控制方式	• ★★★

2.2 考点精讲

2.2.1 进程的状态

操作系统为了便于管理进程，按进程在执行过程中的不同状况，至少定义 3 种不同的进程状态。

运行态：占有处理器正在运行。

就绪态：具备运行条件，等待系统分配处理器以便运行。

等待态（阻塞态）：不具备运行条件，正在等待某个事件的完成。

一个进程在创建后将处于就绪状态。在执行过程中，每个进程任一时刻只会处于这 3 种状态之一。同时，在一个进程执行过程中，它的状态将会发生改变。图 2-1 表示进程的状态转换。

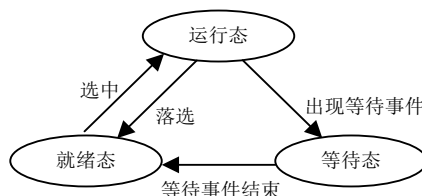


图 2-1 进程三态模型及其状态转换

运行态→等待态：处于运行状态的进程在运行的过程中需要等待某一事件发生后，才能继续运行，于是该进程由运行状态变成等待状态。例如等待 I/O 完成。

等待态→就绪态：处于等待状态的进程，假如其等待的事件已经发生结束。于是进程由等待状态变成就绪状态。

就绪态→运行态：当处于就绪状态的进程被进程调度程序选中后，就分配处理器来运行，进程由就绪状态变成运行状态。

运行态→就绪态：处于运行状态的进程在运行的过程中，因分给它的处理器时间片已用完而不得不让出处理器，于是进程由运行状态变成就绪状态。

2.2.2 进程的死锁

进程管理是操作系统的核心，如果设计不当，就会出现死锁的问题。如果一个进程在等待一个不可能发生的事件，则其将死锁；如果一个或多个进程产生死锁，则会造成系统死锁。

如果在一个系统中以下四个条件同时成立，那么就能引起死锁：

互斥：至少有一个资源必须处于非共享模式，即一次只有一个进程可使用。如果另一进程申请该资源，那么申请进程应等到该资源释放为止。

占有并等待：一个进程应占有至少一个资源，并等待另一个资源，而该资源为其他进程所占有。

非抢占：资源不能被抢占，即资源只能被进程在完成任务后自愿释放。

循环等待：有一组等待进程 $\{P_0, P_1, \dots, P_n\}$ ， P_0 等待的资源为 P_1 占有， P_1 等待的资源为 P_2 占有， \dots ， P_{n-1} 等待的资源为 P_n 占有， P_n 等待的资源为 P_0 占有。

要想防止死锁的发生，其根本方法就是使得上述的必要条件之一不存在，换言之，就是破坏其必要条件使之永不成立。

解决死锁的策略包括死锁预防、死锁避免、死锁检测和死锁解除。

1.死锁预防：例如，要求用户申请资源时一次性申请所需要的全部资源，这样就破坏了保持和等待条件；将资源分层，得到上一层资源后，才能够申请下一层资源，它破坏了环路等待条件。预防通常会降低系统的效率。

2.死锁避免：避免是指进程在每次申请资源时判断这些操作是否安全，典型算法是银行家算法。但这种算法会增加系统的开销。

所谓银行家算法，是指在分配资源之前，先看清楚，如果资源分配下去后，是否会导致系统死锁。如果会死锁，则不分配，否则就分配。具体来说，银行家算法分配资源的原则总结如下：

(1) 当一个进程对资源的最大需求量不超过系统中的资源数时可以接纳该进程。

(2) 进程可以分期请求资源，但请求的总数不能超过最大需求量。

(3) 当系统现有的资源不能满足进程尚需资源数时，对进程的请求可以推迟分配，但总能使进程在有限的时间里得到资源。

注意：

如果系统中有 N 个并发进程，若规定每个进程需要申请 R 个某类资源，则当系统提供 $K=N*(R-1)+1$ 个同类资源时，无论采用何种方式申请使用，一定不会发生死锁。

3.死锁检测：前两者是事前措施，而死锁的检测则是判断系统是否处于死锁状态，如果是，则执行死锁解除策略。

4.死锁解除：这是与死锁检测结合使用的，它使用的方式就是剥夺。即将某进程所拥有的资源强行收回，分配给其他的进程。

2.2.3 进程的同步和互斥

计算机有了操作系统后性能大幅度提升，其根本原因就在于实现了进程的并发运行。多个并发的进程彼此之间围绕着紧俏的资源产生了两种关系，同步或互斥。而可以通过 PV 操作结合信号量解决进程间的同步和互斥问题。

1. 同步

进程同步也是进程之间直接的制约关系，是为完成某种任务而建立的两个或多个线程，这个线程需要在某些位置上协调他们的工作次序而等待、传递信息所产生的制约关系。进程间的直接制约关系来源于他们之间的合作。

比如说进程 A 需要从缓冲区读取进程 B 产生的信息，当缓冲区为空时，进程 B 因为读取不到信息而被阻塞。而当进程 A 产生信息放入缓冲区时，进程 B 才会被唤醒。

2. 互斥

进程互斥是进程之间的间接制约关系。当一个进程进入临界区使用临界资源时，另一个进程必须等待。只有当使用临界资源的进程退出临界区后，这个进程才会解除阻塞状态。

比如进程 B 需要访问打印机，但此时进程 A 占有了打印机，进程 B 会被阻塞，直到进程 A 释放了打印机资源，进程 B 才可以继续执行。

3. 信号量

信号量 S 可以直接理解成计数器，是一个整数。信号量的值仅能由 PV 操作来改变。通过 PV 操作控制信号量来实现进程的同步和互斥。

4. PV 操作

PV 操作：解决互斥和同步的问题。

PV 操作是分开来看的：

P 操作：使 $S=S-1$ ，若 $S \geq 0$ ，则该进程继续执行，否则该进程排入等待队列。

V 操作：使 $S=S+1$ ，若 $S \leq 0$ ，唤醒等待队列中的一个进程。

在资源使用之前将会执行 P 操作，之后将会执行 V 操作。在互斥关系中 PV 操作在一个进程中成对出现，而在同步关系中则一定在两个或多个进程中成对出现。

例如：

生产者-消费者问题是一个经典的进程同步和互斥问题，生产者进程生产物品，然后将物品放置在一个空缓冲区中供消费者进程消费。消费者进程从缓冲区中获得物品，然后释放缓冲区。当生产者进程生产物品时，如果没有空缓冲区可用，那么生产者进程必须等待消费者进程释放出一个空缓冲区。当消费者进程消费物品时，如果没有满的缓冲区，那么消费者进程将被阻塞，直到新的物品被生产出来。其 PV 操作过程如图 2-2 所示。

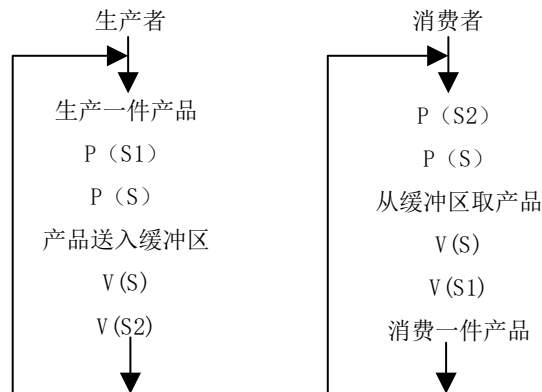


图 2-2 PV 操作

其中 $S1$ 的初值为缓冲区的空间大小（一开始缓冲区为空，站在生产者的角度来看可以存放 N 件产品，那么 $S1$ 初值为 N ）， $S2$ 为 0 （一开始缓冲区为空，站在消费者的角度来看，无产品可取，所以初值为 0 ）， S 属于互斥量，初值为 1 （生产者和消费者必须互斥的访问缓冲区）。

2.2.4 页式存储

页式存储管理是通过引入进程的逻辑地址，把进程地址空间与实际物理存储位置分离，从而增加存储管理的灵活性。我们把逻辑地址空间划分为一些相等的片，这些片称为页或页面。同样，物理地址空间也被划分为同样大小的片，称为块。这样用户程序进入内存时，通过页表就可以将一页对应存入到一个块中。这些物理块不必连续。所以内存利用率可以大大提高。

在页式系统中，指令所给出的逻辑地址分为两部分：逻辑页号和页内地址。其中页号与页内地址所占多少位，与主存的最大容量、页面的大小有关。

CPU 中的内存管理单元按逻辑页号查找页表（操作系统为每一个进程维护了一个从虚拟地址到物理地址的映射关系的数据结构，页表的内容就是该进程的虚拟地址到物理地址的一个映射）得到物理页号，将物理页号与页内地址相加形成物理地址。

2.2.5 页面置换算法

当程序的存储空间要求大于实际的内存空间时，就使得程序难以运行了。虚拟存储技术就是利用实际内存空间和相对大得多的外部存储器存储空间相结合构成一个远远大于实际内存空间的虚拟存储空间，程序就运行在这个虚拟存储空间中，能够实现虚拟存储的依据是程序的局部性原理，即程序在运行过程中经常体现出运行在某个局部范围之内。即在一段时间内，整个程序的执行仅限于程序中的某一部分。

虚拟存储是把一个程序所需要的存储空间分成若干页，程序运行用到的页就放在内存里，暂时不用就放在外存中，当用到外存中的页时，就把它们调到内存，反之就把它们送到外存中。由于所有的进程页面不是一次性地全部调入内存，而是部分页面装入。

这就有可能出现下面的情况：要访问的页面不在内存，这时系统产生缺页中断。操作系统在处理缺页中断时，要把所需页面从外存调入到内存中。如果这时内存中有空闲块，就可以直接调入该页面；如果这时内存中没有空闲块，就必须先淘汰一个已经在内存中的页面，腾出空间，再把所需的页面装入，即进行页面置换。

常用页面置换算法有：先进先出法（FIFO）、最佳置换法（OPT）和最近最少使用置换法（LRU）。

1. 先进先出法（FIFO）

FIFO 算法认为最早调入内存的页不再被使用的可能性要大于刚调入内存的页，因此，先进先出法总是淘汰在内存中停留时间最长的一页，即先进入内存的页，先被换出。

2. 最佳置换法（OPT）

最佳置换算法（OPT）在为调入新页面而必须预先淘汰某个老页面时，所选择的老页面应该以后不被使用，或者是在最远的以后时间才被访问。采用这种算法，能保证有最小缺页率。

3. 最近最少使用置换法（LRU）

最近最少使用置换法（LRU）是选择在最近一段时间里最久没有使用过的页面予以淘汰。

2.2.6 文件管理概念

如图 2-5 所示的就是一个树型目录结构，其中方框代表目录，圆形代表文件。

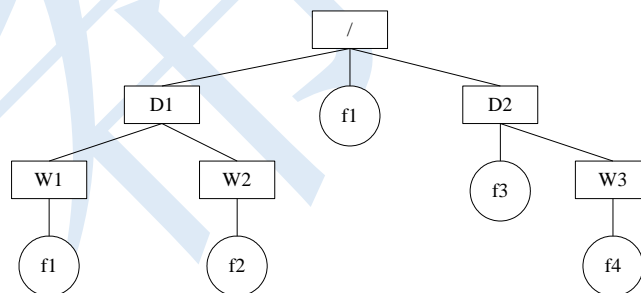


图 2-5 树型文件结构

在树型目录结构中，树的根结点为根目录，数据文件作为树叶，其他所有目录均作为树的结点。系统在建立每一个目录时，都会自动为它设定两个目录文件，一个是“.”，代表该目录自己；另一个是“..”，代表该目录的父目录，也就是上级目录。

从逻辑上讲，用户在登录到系统之后，每时每刻都处在某个目录之中，此目录被称作工作目录或当前目录。工作目录是可以随时改变的。

对文件进行访问时，需要用到路径的概念。路径是指从树型目录中的某个目录层次到某个文件的一条道路。在树型目录结构中，从根目录到任何数据文件之间，只有一条唯一的通路，从树根开始，把全部目录文件名与数据文件名依次用“/”连接起来，构成该数据文件的路径名，且每个数据文件的路径名是唯一的。这样，可以解决文件重名问题，不同路径下

的同名文件不一定是相同的文件。例如，在图 2-5 中，根目录下的文件 f1 和/D1/W1 目录下的文件 f1 可能是相同的文件，也可能是不相同的文件。

2.2.7 绝对路径和相对路径

用户在对文件进行访问时，要给出文件所在的路径。路径又分相对路径和绝对路径。绝对路径是指从根目录开始的路径，也称为完全路径；相对路径是指从用户工作目录开始的路径。应该注意到，在树型目录结构中到某一确定文件的绝对路径和相对路径均只有一条。绝对路径是确定不变的，而相对路径则随着用户工作目录的变化而不断变化。

用户要访问一个文件时，可以通过路径名来引用。例如，在图 2-5 中，如果当前路径是 D1，则访问文件 f2 的绝对路径是/D1/W2/f2，相对路径是 W2/f2。如果当前路径是 W1，则访问文件 f2 的绝对路径仍然是/D1/W2/f2，但相对路径变为../W2/f2。

“../”来表示上一级目录，“../..”表示上上级的目录，以此类推。

2.2.8 传送控制方式

设备管理的主要任务之一是控制设备和内存或 CPU 之间的数据传送，常用的数据传送控制方式一般分为五种：程序查询方式、程序中断方式、DMA 方式、I/O 通道控制方式和输入输出处理机方式。

1. 程序查询方式

最简单的 I/O 控制方式是程序查询方式，要求 CPU 不断使用指令检测方法来获取外设工作状态。由于 CPU 的速度远远高于 I/O 设备，导致 CPU 的绝大部分时间都处于等待 I/O 设备过程中，造成 CPU 的运行效率极低。CPU 和外围设备只能串行工作。但是它管理简单，在要求不高的场合可以被采用。

2. 程序中断方式

某一外设的数据准备就绪后，它“主动”向 CPU 发出中断请求信号，请求 CPU 暂时中断目前正在执行的程序转而进行数据交换；当 CPU 响应这个中断时，便暂停运行主程序，自动转去执行该设备的中断服务程序；当中断服务程序执行完毕（数据交换结束）后，CPU 又回到原来的主程序继续执行。

程序中断方式虽然大大提高了主机的利用率，但是它以字（节）为单位进行数据传送，每完成一个字（节）的传送，控制器便要向 CPU 请求一次中断（做保存现场信息，恢复现场等工作），仍然占用了 CPU 的许多时间。这种方式对于高速的块设备的 I/O 控制显然不适合。

3. DMA 方式

DMA 存取方式，是一种完全由硬件执行 I/O 数据交换的工作方式。它既考虑到中断的响应，同时又要节约中断开销。此时，DMA 控制器代替 CPU 完全接管对总线的控制，数据

交换不经过 CPU，直接在内存和外围设备之间成批进行。优点：速度快，CPU 不参加传送操作，省去了 CPU 取指令、取数、送数等操作，也没有保存现场、恢复现场之类的工作。

缺点：批量数据传送前的准备工作，以及传送结束后的处理工作，仍由 CPU 通过执行管理程序来承担，DMA 控制器只负责具体的数据传送工作。CPU 仍然摆脱不了管理和控制外设的沉重负担，难以充分发挥高速运算的能力。

4. I/O 通道控制方式

通道是一个特殊功能的处理器，代替 CPU 管理控制外设的独立部件。有自己的指令和程序，专门负责数据输入输出的传输控制，而 CPU 在将“传输控制”功能下放给通道后，只负责“数据处理”功能。通道与 CPU 分时使用主存，实现了 CPU 内部运算与 I/O 设备的并行工作。

5. 输入输出处理机方式

采用专用的小型通用计算机，可完成 I/O 通道所完成的 I/O 控制，还可完成码制转换、格式处理、检错纠错等操作，具有相应的运算处理部件、缓冲部件，还可形成 I/O 程序锁必需的程序转移手段。输入输出处理机基本独立于主机工作。在多数系统中，设置多台外围处理机，分别承担 I/O 控制、通信、维护等任务。

目前单片机、微型机多采用程序查询、程序中断和 DMA 方式。通道方式和输入输出处理机方式一般用在大中型计算机中。

第三章 系统开发和项目管理

3.1 本章重点

知识模块	知识点分布	重要程度
软件生命周期	<ul style="list-style-type: none"> 软件生命周期概念 	<ul style="list-style-type: none"> ★
软件开发模型	<ul style="list-style-type: none"> 瀑布模型 演化、增量模型 喷泉模型 V 模型 	<ul style="list-style-type: none"> ★★ ★★ ★ ★
软件设计概念	<ul style="list-style-type: none"> 概要设计 详细设计 	<ul style="list-style-type: none"> ★★ ★
软件测试概念	<ul style="list-style-type: none"> 动态测试 静态测试 	<ul style="list-style-type: none"> ★★ ★
进度管理	<ul style="list-style-type: none"> 甘特图 计划评审图 	<ul style="list-style-type: none"> ★★ ★★★★

3.2 考点精讲

3.2.1 程序设计语言

1. 机器语言

每种型号的计算机都有自己的指令系统，也叫机器语言，每条指令都对应一串二进制代码。机器语言是计算机唯一能够识别并直接执行的语言，所以与其他程序设计语言相比，其执行效率较高。

2. 汇编语言

为了方便地使用计算机，20 世纪 50 年代初，出现了汇编语言。汇编语言不再使用难以记忆的二进制代码编程，而是使用比较容易识别、记忆的助记符号，所以汇编语言又叫符号语言。汇编语言只是将一条机器语言用符号表示而已，也是面向机器的一种低级语言，或者说，汇编语言是符号化了的机器语言。

用汇编语言编写出来的程序称为汇编语言源程序，计算机不能直接识别、执行它。必须先把汇编语言源程序翻译成机器语言程序（称目标程序），然后才能执行。这个翻译过程是

由事先存放在机器里的“汇编程序”完成的，叫作汇编过程。

3. 高级语言

低级语言是对计算机硬件直接进行操作的语言，包括机器语言和汇编语言，这种语言编写程序对程序员的要求比较高，必须了解计算机内部结构。所以现在一般用高级语言编写程序。高级语言是一种用表达各种意义的“词”和“数学公式”按照一定的“语法规则”编写程序的语言，也称高级程序设计语言或算法语言。这里的“高级”是指这种语言与自然语言和数学式子相当接近，而且不依赖于计算机的型号，通用性好。

高级语言的使用，大大提高了编写程序的效率，改善了程序的可读性。同样，用高级语言编写的程序称为高级语言源程序，计算机是不能直接识别和执行的，也要用翻译的方法把高级语言源程序翻译成等价的机器语言程序（称为目标程序）才能执行。

把高级语言源程序翻译成机器语言程序的方法有“解释”和“编译”两种。

编译方式是当用户将高级语言编写的源程序输入计算机后，编译程序便把源程序整个的翻译成用机器语言表示的与之等价的的目标程序，然后计算机再执行该目标程序，以完成源程序要处理的运算并取得结果。比如将高级语言（如 C++）源程序作为输入，进行编译转换，产生出机器语言的目标程序，然后再让计算机去执行这个目标程序，得到计算结果。

解释（翻译）方式是指源程序进入计算机后，解释程序边扫描边解释，逐句输入逐句翻译，计算机一句句执行，并不产生目标程序。比如将源语言（如 BASIC）书写的源程序作为输入，解释一句后就提交计算机执行一句，并不形成目标程序。但解释程序执行速度很慢，例如源程序中出现循环，则解释程序也重复地解释并提交执行这一组语句，这就造成很大浪费。编译程序与解释程序最大的区别之一在于前者生成目标代码，而后者不生成；此外，前者产生的目标代码的执行速度比解释程序的执行速度要快；后者人机交互好，适于初学者使用。

3.2.2 软件生命周期阶段

系统开发的生命周期讲的是一个系统历经计划、分析、设计、编程、测试、维护直至淘汰的整个过程。

计划时期：包括问题定义和可行性研究。

需求分析：需求分析的任务不是具体地解决问题，而是确定软件系统的功能、性能、数据、界面等要求，从而来确定系统的逻辑模型。

总体设计：在总体设计阶段，开发人员需要将各项功能需求转换成相应的体系结构，也就是每个模块都和某些功能需求相对应。

详细设计：详细设计的主要任务就是对每个模块完成的功能进行具体描述，也就是要知道每个模块的控制结构是怎么样的，先做什么，后做什么，有什么样的条件判定等，用相应的工具把这些控制结构表示出来。

编码：把每个模块的控制结构用程序代码进行表示。

测试：测试是保证软件质量的重要手段，其主要方式是在设计测试用例的基础上检查软件的各个组成部分。

维护：软件维护是软件生命周期中时间最长的阶段。软件交付给用户使用之后，就进入

软件维护阶段，它可以持续几年甚至几十年。

3.2.3 软件开发模型

瀑布模型：瀑布模型也称为生命周期法，是生命周期法中最常用的开发模型，它把软件开发的过程分为软件计划、需求分析、软件设计、程序编码、软件测试和运行维护 6 个阶段，规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。采用瀑布模型的软件过程如图 3-1 所示。

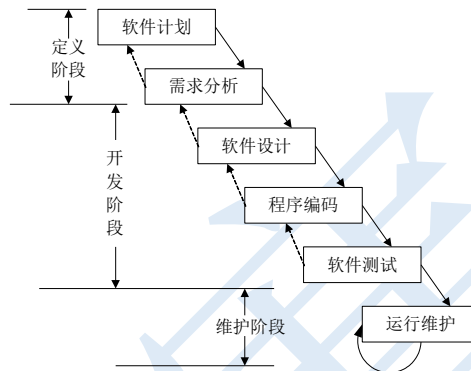


图 3-1 瀑布模型

瀑布模型是最早出现的软件开发模型，在软件工程中占有重要的地位，它提供了软件开发的基本框架。瀑布模型的本质是“一次通过”，即每个活动只做一次，最后得到软件产品，过程就是利用本次活动应完成的内容，给出该项活动的工作成果，作为输出传给下一次活动；对该项活动实施的工作进行评审，如果工作得到确认，那么就继续下一项活动，否则返回前项，甚至更前项的活动进行返工。

瀑布模型有利于大型软件开发过程中人员的组织与管理，有利于软件开发方法和工具的研究与使用，从而提高了大型软件项目开发的质量和效率。然而软件开发的实践表明，上述各项活动之间并非完全是自上而下的，而是呈线性图式，因此，瀑布模型存在严重的缺陷。

(1) 由于开发模型呈线性，所以当开发成果尚未经过测试时，用户无法看到软件的效果。这样，软件与用户见面的时间间隔较长，也增加了一定的风险。

(2) 在软件开发前期未发现的错误传到后面的开发活动中时，可能会扩散，进而可能会导致整个软件项目开发失败。

(3) 最突出的一点是围绕需求分析的，通常客户一开始并不知道他们需要的是什么，而需求是在整个项目进程中通过双向交互不断明确的，而瀑布模型是强调一开始精准捕获需求和设计。

演化模型的主要步骤是首先开发系统的一个核心功能，使得客户可以与开发人员一同确认该功能，这样开发人员将会得到第一手的经验，再根据客户的反馈进一步开发其他功能或进一步扩充该功能，直到建立一个完整的系统为止。演化模型的特点基本上与增量模型一致，但对于演化模型的管理是一个主要的难点，也就是说，我们很难确认整个系统的里程碑、成本和时间基线。

增量模型与原型实现模型和其他演化方法一样，本质上是迭代的，但与演化不一样的是

其强调每一个增量均发布一个可操作产品。采用增量模型的优点是人员分配灵活，刚开始不用投入大量人力资源。如果核心产品很受欢迎，则可增加人力实现下一个增量。当配备的人员不能在设定的期限内完成产品时，它提供了一种先推出核心产品的途径。这样即可先发布部分功能给客户，对客户起到镇静剂的作用。此外，增量能够有计划地管理技术风险。

喷泉模型不像瀑布模型一样，需要分析活动结束后才开始设计活动，设计活动结束后才开始编码活动。喷泉模型强调的是无间隙，也就是各个阶段没有明显的界限，开发人员可以同步进行开发。其优点是可以提高软件项目开发效率，节省开发时间，适应于面向对象的软件开发过程。由于喷泉模型在各个开发阶段存在重叠现象，需要大量的开发人员，难以进行项目管理。

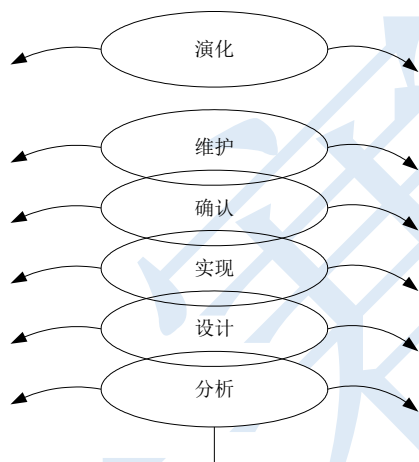


图 3-2 喷泉模型

V 模型主要是说明测试活动是如何与分析和设计相联系的，在开发模型中，测试常常作为亡羊补牢的事后行为，但也有以测试为中心的开发模型，那就是 V 模型。V 模型只得到软件业内比较模糊的认可。

如图 3-3 所示。

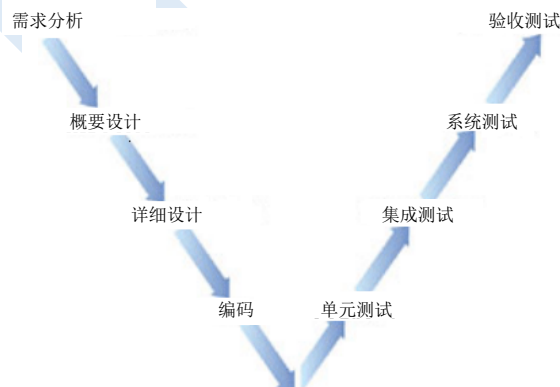


图 3-3 V 模型

(1) 单元测试的主要目的是针对编码过程中可能存在的各种错误。例如，用户输入验证过程中边界值的错误，对应的是详细设计。

(2) 集成测试的主要目的是针对详细设计中可能存在的问题，尤其是检查各单元与其他程序部分之间的接口上可能存在的错误，对应的是概要设计。

(3) 系统测试主要针对概要设计，检查系统作为一个整体是否有效地得到运行。例如，在产品设置中是否达到了预期的高性能，对应的是需求分析。

(4) 验收测试通常由业务专家或用户进行，以确认产品能真正符合用户业务上的需要。

螺旋模型将瀑布模型和演化模型相结合，它综合了两者的优点，并增加了风险分析，如图 3-4 所示。每轮循环包含如下六个步骤：1.确定目标，可选项，以及强制条件。2.识别并化解风险。3.评估可选项。4.开发并测试当前阶段。5.规划下一阶段。6.确定进入下一阶段的方法步骤。

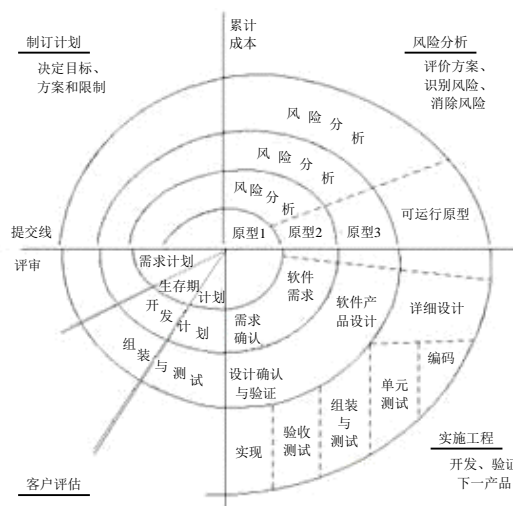


图 3-4 螺旋模型

3.2.4 软件设计概念

概要设计也称为总体设计，将软件需求转化为数据结构和软件的系统结构。

在结构化方法中，模块化是一个很重要的概念，它是将一个待开发的软件分解成若干个小的简单部分——模块，每个模块可以独立地开发、测试。目的是使程序的结构清晰，比较容易进行测试和修改。模块独立是指每个模块完成一个相对独立的子功能，并且与其他模块之间的联系最简单。保持模块的高度独立性，也是设计时的一个很重要的原则。通常我们用耦合（模块之间联系的紧密程度）和内聚（模块内部各元素之间联系的紧密程度）两个标准来衡量，我们的目标是高内聚、低耦合。

详细设计也称为低层设计，将对结构表示进行细化，得到详细的数据结构与算法。同样的，如果采用结构化设计，则详细设计的任务就是为每个模块进行设计。

详细设计确定应该如何具体地实现所要求的系统，得出对目标系统的精确描述。它采用自顶向下、逐步求精的设计方式和单入口单出口的控制结构。

总的来说，在整个软件设计过程中，需完成以下工作任务：

- (1) 制定规范，作为设计的共同标准。
- (2) 完成软件系统结构的总体设计，将复杂系统按功能划分为模块的层次结构，然后确定模块的功能，以及模块间的调用关系、模块间的组成关系。
- (3) 设计处理方式，包括算法、性能、周转时间、响应时间、吞吐量、精度等。
- (4) 设计数据结构。

(5) 可靠性设计。

(6) 编写设计文档，包括概要设计说明书、详细设计说明书、数据库设计说明书、用户手册、初步的测试计划等。

(7) 设计评审，主要是对设计文档进行评审。

在设计阶段，必须根据要解决的问题，做出设计的选择。例如，对于半结构化决策问题就适合于交互式计算机软件来解决。

3.2.5 软件测试概念

动态测试是指通过运行程序发现错误，分为黑盒测试法、白盒测试法和灰盒测试法。不管是哪一种测试，都不能做到穷尽测试，只能选取少量最有代表性的输入数据，以期用较少的代价暴露出较多的程序错误。这些被选取出来的数据就是测试用例（一个完整的测试用例应该包括输入数据和期望的输出结果）。

(1) 黑盒法。把被测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求。因此，黑盒测试又称为功能测试或数据驱动测试。常用的黑盒测试用例的设计方法有等价类划分、边值分析、错误猜测、因果图和功能图等。

(2) 白盒法。把测试对象看作一个打开的盒子，测试人员需了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。由于白盒测试是结构测试，所以被测对象基本上是源程序，以程序的内部逻辑为基础设计测试用例。常用的白盒测试用例设计方法有基本路径测试、循环覆盖测试、逻辑覆盖测试。

(3) 灰盒法。灰盒测试是一种介于白盒测试与黑盒测试之间的测试，它关注输出对于输入的正确性。同时也关注内部表现，但这种关注不像白盒测试那样详细且完整，而只是通过一些表征性的现象、事件及标志来判断程序内部的运行状态。

静态测试是指被测试程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。静态分析中进行人工测试的主要方法有桌前检查（程序员自查）、代码审查和代码走查。经验表明，使用这种方法能够有效地发现 30%~70% 的逻辑设计和编码错误。

3.2.6 进度管理

项目管理中的进度管理是指在项目实施过程中，对各阶段的进展程度和项目最终完成的期限所进行的管理。项目的进度计划和工作的实际进展情况，通常表现为各项任务之间的进度依赖关系，因而通常使用图表的方式来说明。

1. 甘特图

甘特图（Gantt 图）使用水平线段表示任务的工作阶段，线段的起点和终点分别对应着任务的开工时间和完成时间，线段的长度表示完成任务所需的时间。而跟踪甘特图则是在甘特图的基础上，加上一个表示现在时间的纵线，可以直观地看出进度是否延误。甘特图的优

点在于标明了各任务的计划进度和当前进度，能动态地反映项目进展；其缺点在于难以反映多个任务之间存在的复杂逻辑关系。

2. 计划评审图

PERT 图是一种网络模型，描述一个项目任务之间的关系。可以明确表达任务之间的依赖关系，即哪些任务完成后才能开始另一些任务，以及如期完成整个工程的关键路径，但是不能清晰地描述各个任务之间的并行关系。

在网络图中的某些活动可以并行地进行，所以完成工程的最少时间是从开始顶点到结束顶点的最长路径长度，称从开始顶点到结束顶点的最长（工作时间之和最大）路径为关键路径（临界路径），关键路径上的活动为关键活动。在一条路径中，每个工作的时间之和等于工程工期，这条路径就是关键路径。

关键路径法 CPM 是借助网络图和各活动所需的时间（估计值），计算每一活动的最早或最迟开始和结束时间。

松弛时间（slack time）指不影响完工前提下可能被推迟完成的最大时间（松弛时间=关键路径的时间-包含某活动最长路径所需要的时间），在关键路径上的任务的松弛时间为 0。

例如，在图 3-5 中，一共有 3 条路径，分别是 ABEG、ACFG 和 ABDFG，其路径长度分别为 16、17 和 21。因此，图 3-5 的关键路径为 ABDFG。如果图 3-5 是代表某个项目的网络计划图，则该项目的工期为 21 天。

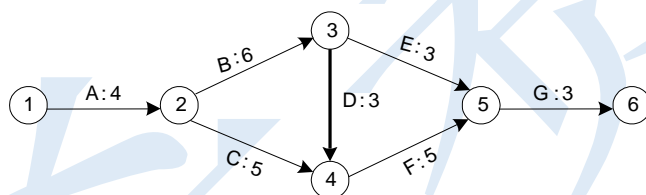


图 3-5 某项目的网络计划图

第四章 知识产权

4.1 本章重点

知识模块	知识点分布	重要程度
著作权	<ul style="list-style-type: none"> • 著作权的保护期限 • 产权人的确定 • 侵权判定 	<ul style="list-style-type: none"> • ★★ • ★★★★★ • ★★★★★
商标权	<ul style="list-style-type: none"> • 商标权的归属 	<ul style="list-style-type: none"> • ★★
专利权	<ul style="list-style-type: none"> • 专利权的归属 	<ul style="list-style-type: none"> • ★★

4.2 考点精讲

依照我国《中华人民共和国著作权法》规定，自作品完成创作之日起即拥有著作权。著作权也称版权，是指作者及其他权利人对文学、艺术和科学作品享有的人身权和财产权的总称。

著作权人对其创作的产品，享有以下几种权利：

- （一）发表权，即决定作品是否公之于众的权利；
- （二）署名权，即表明作者身份，在作品上署名的权利；
- （三）修改权，即修改或者授权他人修改作品的权利；
- （四）保护作品完整权，即保护作品不受歪曲、篡改的权利；
- （五）复制权，即以印刷、复印、拓印、录音、录像、翻录、翻拍等方式将作品制作一份或者多份的权利；
- （六）发行权，即以出售或者赠与方式向公众提供作品的原件或者复制件的权利；
- （七）出租权，即有偿许可他人临时使用电影作品和以类似摄制电影的方法创作的作品、计算机软件的权利，计算机软件不是出租的主要标的的除外；
- （八）展览权，即公开陈列美术作品、摄影作品的原件或者复制件的权利；
- （九）表演权，即公开表演作品，以及用各种手段公开播送作品的表演的权利；
- （十）放映权，即通过放映机、幻灯机等技术设备公开再现美术、摄影、电影和以类似摄制电影的方法创作的作品等的权利；
- （十一）广播权，即以无线方式公开广播或者传播作品，以有线传播或者转播的方式向公众传播广播的作品，以及通过扩音器或者其他传送符号、声音、图像的类似工具向公众传播广播的作品的权利；
- （十二）信息网络传播权，即以有线或者无线方式向公众提供作品，使公众可以在其个人选定的时间和地点获得作品的权利；

- (十三) 摄制权，即以摄制电影或者以类似摄制电影的方法将作品固定在载体上的权利；
- (十四) 改编权，即改变作品，创作出具有独创性的新作品的权利；
- (十五) 翻译权，即将作品从一种语言文字转换成另一种语言文字的权利；
- (十六) 汇编权，即将作品或者作品的片段通过选择或者编排，汇集成新作品的权利；
- (十七) 应当由著作权人享有的其他权利。

4.2.1 著作权保护期限

根据著作权法相关规定，著作权的保护是有一定期限的。

1. 著作权属于公民

署名权、修改权、保护作品完整权的保护期没有任何限制，永远属于保护范围。而发表权、使用权和获得报酬权的保护期为作者终生及其死亡后的 50 年(第 50 年的 12 月 31 日)。作者死亡后，著作权依照继承法进行转移。

2. 著作权属于单位

发表权、使用权和获得报酬权的保护期为 50 年(首次发表后的第 50 年的 12 月 31 日)，若 50 年内未发表的，不予保护。但单位变更、终止后，其著作权由承受其权利义务的单位享有。

4.2.2 著作权人的确定

《中华人民共和国著作权法》第十一条规定：“著作权属于作者，本法另有规定的除外。创作作品的公民是作者。由法人或者其他组织主持，代表法人或者其他组织意志创作，并由法人或者其他组织承担责任的作品，法人或者其他组织视为作者。”具体如表 4-1 所示：

表 4-1 著作权归属

情况说明		判断说明	归 属
作品	职务作品	利用单位的物质技术条件进行创作并由单位承担责任的	除署名权外其他著作权归单位
		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
作品	职务作品	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和技术条件，并由单位或组织承担责任的软件	单位享有著作权
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	只进行组织并提供咨询意见、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

除此之外，还有一个小知识点是如果遇到作者不明的情况，那么作品原件的所有人可以行使除署名权以外的著作权，直到作者身份明确。

4.2.3 侵权判定

对是否侵犯了知识产权的判断通常也是显而易见的，但是如下比较特殊的情况考生容易混淆和出错。

(1) 口述作品（包括即兴演说、授课和法庭辩论等以口头语言形式表现的作品）、摄影作品及示意图受著作权保护。

(2) 对于作品而言，以下行为不侵权：即个人学习、介绍或评论时引用；在各种形式的新闻报道中引用；学校教学与研究及图书馆陈列用的少量复制；执行公务使用；免费表演已发表作品；将汉字作品翻译成为少数民族文字或改为盲文出版。

(3) 对于作品而言，公开表演及播放需要另外授权。例如，在商场公开播放正版的音乐及 VCD 也是侵权行为。而且版权人对作品还享有保护作品完整权，这一点也不容忽视。

(4) 对于软件产品而言，要注意保护只是针对计算机软件和文档，并不包括开发软件所用的思想、处理过程、操作方法或数学概念等；另外以学习和研究为目的所做的少量复制与修改，为保护合法获得的产品所做的少量复制也不侵权。

(5) 若国家出现紧急状态或者非常情况，可以为了公共利益强制实施发明和实用新型专利的许可。

最后还要提醒考生在侵权判断的题目中，如果给出的条件没有明确说明双方的约定情况，并且答案中出现“是否侵权，应根据甲乙双方协商情况而定”时，通常这才是正确答案。

如果了解更多的内容，请考生阅读《中华人民共和国著作权法》《计算机软件保护条例》《中华人民共和国专利法》《中华人民共和国商标法》《中华人民共和国反不正当竞争法》，以及相应的《实施细则》。

4.2.4 商标权

商标权是指商标主管机关依法授予商标所有人对其注册商标受国家法律保护的专有权。商标是用以区别商品和服务不同来源的商业性标志，由文字、图形、字母、数字、三维标志、颜色组合或者上述要素的组合构成。

根据《商标法》规定，商标权有效期 10 年，自核准注册之日起计算，期满前 12 个月内申请续展，在此期间内未能申请的，可再给予 6 个月的宽展期。续展可无限重复进行，每次续展注册的有效期为 10 年。自该商标上一届有效期满次日起计算。期满未办理续展手续的，注销其注册商标。

关于商标权的归属：

1. 谁先申请谁拥有（除知名商标的非法抢注）；
2. 同时申请，则根据谁先使用（需提供证据）；
3. 无法提供证据，协商归属，无效时使用抽签。

4.2.5 专利权

专利权（Patent Right），简称“专利”，是发明创造人或其权利受让人对特定的发明创造在一定期限内依法享有的独占实施权，是知识产权的一种。

执行本单位的任务或者主要是利用本单位的物质技术条件所完成的发明创造为职务发明创造。职务发明创造申请专利的权利属于该单位；申请被批准后，该单位为专利权人。非职务发明创造，申请专利的权利属于发明人或者设计人；申请被批准后，该发明人或者设计人为专利权人。利用本单位的物质技术条件所完成的发明创造，单位与发明人或者设计人订有合同，对申请专利的权利和专利权的归属作出约定的，从其约定。

两个或两个以上的申请人分别就同样的发明创造申请专利的，专利权授予原则是谁先申请谁拥有，否则就需要协商归属。

专利权解决的办法一般有两种：

一种是两申请人作为一件专利申请的共同申请人。

另一种是其中一方放弃权利并从另一方得到适当的补偿。

2020

希赛 | 专注在线教育19年

软考通关直播班



软考领航专家讲师团

课程专属

- ▶ 全套直播精讲，配内部课程讲义、章节练习，取证快人一步
- ▶ 社群交流互动，适应状态、提升学习效率，你不是一个人在战斗！
- ▶ 模考冲刺、历年真题+解析，查漏补缺，提分更有效
- ▶ 助教督学，班主任贴心提醒，告别懒惰，每个人跟得上，听得懂



长按扫码咨询