国家电网·必会考点
——计算机类

目录

知识点 1、进制转换

常用数制

数制	数码	运算规则	尾符
十进制	0—9	逢十进一	D 或10
二进制	0—1	逢二进一	B 或2
八进制	0—7	逢八进一	O 或8
十六进制	0—9，A—F	逢十六进一	H 或16

（一）非十进制数转换成十进制数方法

每位上的数码 × 基的位次方，然后求和。

（二）十进制数转换成非十进制数方法

整数部分：除N 取余数，直至商为0，余数倒输出。

小数部分：乘N 取整数，直至积为0（或满足精度），整数正输出。

（N 为进制数，N 可取2、8、16。）

（三）二/八/十六进制数的互相转换方法

由于一位八/十六进制数相当于三/四位二进制数，因此，要将八/十六进制数转换成二进制数时，只需以小数点为界，向左或向右每一位八/十六进制数用相应的三/四位二进制数取代即可。如果不足三/四位，可用零补足。反之，二进制数转换成相应的八/十六进制数，只是上述方法的逆过程，即以小数点为界，向左或向右，每三/四位二进制数用相应的一位八/十六进制数取代即可。

【习题演练】

1.二进制数1101.01 转换成十进制数是（ ）。

A.17.256

B.13.5

C.13.25

D.17.5

1.【答案】C。解析：$(1101.01)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 + 1 \times 2^{-2} = 13.25$。

2.下列数中最大的数是（ ）。

A.227O

B.1FFH

C.10100001B

D.1789D

2.【答案】D。解析：将所有数都统一转换为十进制进行比较，227O=151D，1FFH=511D，10100001B=161D，对比可知应选D选项。

知识点2、常见数据编码

（一）ASCII码

美国信息交换标准代码（American Strandard Code for Information Interchage，ASCII）是一种西

文机内码，是计算机中使用最广泛的字符编码，已经作为国际通用的信息交换标准代码。ASCII 码包括 0～9 十个数字，大小写英文字母和专用符号等 95 种可打印字符，以及33 种控制字符（如回车、换行等），通常采用一个字节编码，由7 位二进制编码组成，字节的最高位一般规定为0，或用作校验码，可表示 128 个不同的字符。

（二）汉字的编码

1. 国标码

《信息交换用汉字编码字符集・基本集》是我国于 1980 年制定的国家标准GB2312—80，简称国标码，是国家规定的用于汉字信息交换使用的代码的依据。GB2312—80 中规定了信息交换用的6763 个汉字和682 个非汉字图形符号（包括外文字母、数字和符号）的代码。国标码用两个字节表示一个汉字，每个字节只使用低7 位，最高位为0。

2.汉字机内码

汉字的机内码是供计算机系统内部进行存储、加工处理、传输统一使用的代码，又称为汉字内部码或汉字内码。汉字机内码是将国标码的两个字节的最高位分别置为 1 而得到的。其最大优点是机内码表示简单，同时也解决了中西文机内码存在的二义性的问题。

3.汉字输入码

汉字输入码是为了将汉字通过键盘输入计算机而设计的代码。汉字输入编码方案很多，其表示形式大多用字母、数字或符号。输入码的长度也不同，多数为四个字节。综合起来可分为流水码、音码、形码和音形码几大类。

4.汉字字形码

汉字字形码是汉字字库中存储的汉字字形的数字化信息，用于汉字的显示和打印。目前汉字字形的产生方式大多是数字式，即以点阵方式形成汉字。因此，汉字字形码主要是指汉字字形点阵的代码，

主要有 16×16 点阵、32×32 点阵、256×256 点阵等。

如 24×24 的点阵，每字需要 72 字节；32×32 的点阵，每字需要 128 字节。与每个汉字对应的这一串字节就是汉字的字形码。

（三）二-十进制编码

人们习惯于使用十进制数，而计算机内部多采用二进制数表示和处理数值数据，因此在计算机输入和输出数据时，就要进行由十进制到二进制和从二进制到十进制的转换处理，显然，这项事务性工作如果由人工来完成，势必造成大量时间浪费。因此，必须采用一种编码的方法，由计算机自己来完成这种识别和转换工作。

人们通常采用把十进制数的每一位分别写成二进制数形式的编码，称为二-十进制编码或 BCD 编码。BCD 编码方法很多，通常采用的是 8421 编码。其方法是用四位二进制数表示一位十进制数，自左

至右每一位对应的位权分别是8，4，2，1。值得注意的是，四位二进制数有0000～1111十六种状态，这里只取了0000～1001十种状态，而1010～1111六种状态在这种编码中没有意义。如十进制数864，其BCD进制编码为100001100100。

8　　6　　4

↓　　↓　　↓

1000　0110　0100

BCD码有压缩和非压缩两种。上述的编码方法属于压缩BCD码，特点是用1个字节表示2位BCD码；非压缩BCD码用1个字节的低4位表示1位BCD码，高4位为0（无意义）。如13的压缩BCD码为0001 0011，非压缩BCD码为00000001 00000011。

【习题演练】

1.已知字符0的ASCII码为48D，用ASCII码（7位）表示字符5和7是（　）。

A.0110010 和0110111

B.0100011 和0111011

C. 1000101 和1100011

D.0110101 和0110111

1.【答案】D。解析：ASCII 码表中字符5 和7 的码值分别为53 和55，转换为二进制表示为0110101 和0110111。

2.某数用压缩BCD 码表示为10010101，其真值为（ ）。

A.135

B.95

C.95H

D. 10010101B

2.【答案】B。解析：BCD 码是二-十进制编码，1001 和0101 分别对应十进制的9 和5，故选择B 选项。

3.一个16×16 点阵的汉字要占用（ ）。

A.24 个字节

B.32 个字节

C.48 个字节

D.256 个字节

3.【答案】B 。解析：16 × 16 点阵的汉字要用 16 × 16/8=32 个字节。

知识点 3 、机器数

对于数的符号“ + ”和“ – ”，计算机是无法识别的，需要把符号数码化。通常，约定二进制数的最高位为符号位，“0 ”表示正号，“1 ”表示负号。这种在计算机中使用的表示数的形式称为机器数，常见的机器数有原码、反码、补码等。

（ 一 ）原码

原码表示法是机器数的一种简单的表示法。其符号位用0 表示正号，用 1 表示负号，数值一般用

二进制形式表示。设有一数用X 表示真值，则原码表示可记作$[X]_{原}$，例如：

X1= + 1010110

X2= – 1001010

其原码记作：

$[X1]_{原}$=[+ 1010110]$_{原}$=01010110

$[X2]_{原}$=[– 1001010]$_{原}$= 11001010

在原码表示法中，对0 有两种表示形式：

[+ 0]$_{原}$=00000000

[– 0]$_{原}$=10000000

（二）反码

机器数的反码可由原码得到。如果机器数是正数，则该机器数的反码与原码一样；如果机器数是负数，则该机器数的反码是对它的原码（符号位除外）各位取反而得到的。假设有一数用X 表示真值，那么X 的反码表示记作$[X]_{反}$，例如：

X1= + 1010110

X2= – 1001010

那么

$[X1]_{原}$=01010110

$[X1]_{反}$=$[X1]_{原}$=01010110

$[X2]_{原}$=11001010

$[X2]_{反}$=10110101

在反码表示法中，对 0 也有两种表示形式：

$[+0]_{反}$=00000000

$[-0]_{反}$=11111111

反码通常作为求补过程的中间形式，即在一个负数的反码的末位上加 1，就得到该负数的补码。

（三）补码

机器数的补码可由原码得到。如果机器数是正数，则该机器数的补码与原码一样；如果机器数是负数，则该机器数的补码是对它的原码（除符号位外）各位取反，并在末位加 1 而得到的。设有一数用X 表示真值，则X 的补码表示记作$[X]_{补}$，例如：

X1= + 1010110

X2= – 1001010

那么

$[X1]_{原}$=01010110

$[X1]_{补}$= $[X1]_{原}$=01010110

$[X2]_{原}$=11001010

$[X2]_{补}$= 10110101+1= 10110110

在补码表示法中，0 只有一种表示形式：

$[+0]_{补}$=00000000

$[-0]_{补}$= 11111111 + 1=00000000

所以有$[+0]_{补}$=$[-0]_{补}$=00000000。

【习题演练】

1. 采用补码表示的 8 位二进制数真值范围是（ ）。

A.- 127 ~ +127

B.- 127 ~ +128

C.- 128 ~ +127

D.- 128 ~ +128

1.【答案】C。解析：补码表示的 8 位二进制数真值范围是- 128 ~ +127，原码和反码的范围是- 127 ~ +127。

2.补码25H 的真值是（ ）。

A.25

B.-25

C.-37

D.37

2.【答案】D。解析：正数的原码、反码与补码都相同。

3.在机器中，（ ）的零的表示形式是唯一的。

A.原码

B.补码

C.反码

D.原码和反码

3.【答案】B 。解析：补码零的表示形式唯一，而原码和反码不唯一。

知识点 4 、补码溢出

（ 一 ）补码溢出的产生

在补码加减运算中，有时会遇到这样的情况：两个正数相加，而结果的符号位却为 1（结果为负）；两个负数相加，而结果的符号位却为 0（结果为正）。发生这种错误的原因在于两数相加之和的数值已超过了机器允许的表示范围。字长为n+1 位的定点整数（其中一位为符号位），采用补码表示，当运算结果大于2^n-1 或小于-2^n 时，就产生溢出。

设参加运算的两数为X 、Y，做加法运算：

①若X 、Y 异号，不会溢出；

②若X 、Y 同号，运算结果为正且大于所能表示的最大正数或运算结果为负且小于所能表示的最小负数时，产生溢出。将两正数相加产生的溢出称为正溢；反之，两负数相加产生的溢出称为负溢。

（二）补码溢出的检测

设被操作数为$[X]_{补}=X_sX_1X_2\ldots X_n$，操作数为$[Y]_{补}=Y_sY_1Y_2\ldots Y_n$，其和（差）为$[S]_{补}=S_sS_1S_2\ldots S_n$，则判断溢出的方法有以下三种：

1.采用一个符号位

两正数相加，结果为负表明产生正溢；两负数相加，结果为正表明产生负溢。因此可得出采用一个符号位检测溢出的方法：

当$X_s=Y_s=0$，$S_s=1$ 时，产生正溢；

当$X_s=Y_s=1$，$S_s=0$ 时，产生负溢。

2.采用进位位

两数运算时，产生的进位为：$C_sC_1C_2\ldots C_n$，其中：C_s 为符号位产生的进位，C_1 为最高数值位产生的进位。

两正数相加，当最高有效位产生进位（$C_1=1$）而符号位不产生进位（$C_s=0$）时，发生正溢。

两负数相加，当最高有效位没有进位（C_1=0）而符号位产生进位（C_s= 1）时，发生负溢。

3.采用变形补码（双符号位补码）

在双符号位的情况下，把左边的符号位 S_{s1} 叫做真符，因为它代表了该数真正的符号，两个符号位都作为数的一部分参加运算。这种编码又称为变形补码。双符号位的含义如下：

$S_{s1}S_{s2}$=00，结果为正数，无溢出；

$S_{s1}S_{s2}$=01，结果正溢；

$S_{s1}S_{s2}$=10，结果负溢；

$S_{s1}S_{s2}$=11，结果为负数，无溢出。

【习题演练】

1.算式65H-3EH 的运算结果是否有溢出，结果的真值为（ ）。

A.有、39

B.有、27

C.无、39

D.无、27

1.【答案】C。解析：65H-3EH=101D-62D=39D，无溢出。

知识点 5、寻址方式

存储器既可用来存放操作数，又可用来存放指令。当某个操作数或某条指令存放在某个存储单元时，其存储单元的编号，就是该操作数或指令在存储器中的地址。因此，寻址方式可以分为数据寻址和指令寻址。寻找操作数的地址称为数据寻址，数据寻址方式较多，其最终目的都是寻找所需要的操作数。寻找下一条将要执行的指令地址称为指令寻址。

（一）指令寻址

指令寻址比较简单，它又可以细分为顺序寻址和跳跃寻址。顺序寻址可通过程序计数器PC 加 1，

自动形成下一条指令的地址；跳跃寻址是程序转移执行时的指令寻址方式，它通过转移类指令实现。

跳跃寻址的转移地址形成方式有三种：直接（绝对）、相对和间接寻址，它与数据寻址方式中的直接、相对和间接寻址是相同的，只不过寻找到的不是操作数的有效地址而是转移的有效地址而已。

（二）数据寻址

数据寻址方式种类较多，在指令字中必须设一字段来指明属于哪一种寻址方式。指令的地址码字段通常都不代表操作数的真实地址，把它称为形式地址，记作A。操作数的真实地址称为有效地址，记作EA，它是由寻址方式和形式地址共同来确定的。由此可得指令的格式应如下所示。

操作码OP	寻址特征#	形式地址A

寻址方式是根据指令中给出的地址码字段寻找真实操作数地址的方式。

指令中的形式地址A—（寻址方式）→有效地址EA

1.立即寻址

在取指令时，操作码和操作数被同时取出，不必再次访问存储器，从而提高了指令的执行速度，如下图所示。立即寻址的特点是操作数本身设在指令字内，即形式地址A 不是操作数的地址，而是操作数本身，也称立即数。由于操作数是指令的一部分，故立即数的大小将受到指令长度的限制。

立即寻址示意图

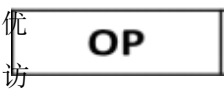
2.直接寻址

指令中地址码字段给出的地址A 就是操作数的有效地址：EA=A，如下图所示。直接寻址的缺点在于A 的位数限制了操作数的寻址范围，且必须修改A 的值，才能修改操作数的地址。

直接寻址示意图

3.间接寻址

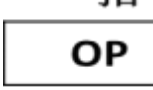
指令中给出的地址A 不是操作数的地址，而是存放操作数地址的地址：EA=（A），如下图所示。



间接寻址要比直接寻址灵活得多，它的主要优点为：一是扩大了寻址范围，可用指令的短地址访问大的主存空间，二是可将主存单元作为程序的地址指针，用以指示操作数在主存中的位置。当操作数的地址需要改变时，不必修改指令，只需修改存放有效地址的那个主存单元（间接地址单元）的内容就可以了。

一级间接寻址

二级间接寻址



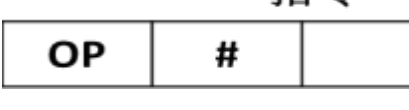
除去一级间接寻址外，还有多级间接寻址。多级间接寻址为取得操作数需要多次访问主存，即使在找到操作数有效地址后，还需再访问一次主存才可得到真正的操作数。

4.寄存器寻址

指令中地址码部分给出某一通用寄存器的编号，所指定的寄存器中存放着操作数，如下图所示。它有两个明显的优点：一是从寄存器存取数据比主存快得多；二是由于寄存器的数量较少，其地址码字段比主存单元地址字段短得多。

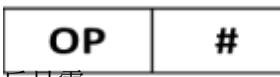
寄存器寻址示意图

5.寄存器间接寻址



寄存器间接寻址示意图

指令中的地址码给出某一通用寄存器的编号，被指定的寄存器中存放操作数的有效地址，而操作数则存放在主存单元中，如上图所示。



这种寻址方式的指令较短，并且在取指后只需一次访存便可得到操作数。

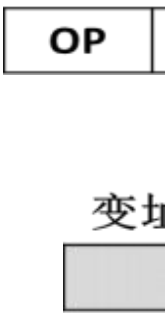
6.变址寻址

把指令给出的形式地址A 与变址寄存器Rx 的内容相加，形成操作数有效地址：EA = A + (Rx)，Rx 的内容为变址值，如下图所示。

变址寻址是一种广泛采用的寻址方式，通常指令中的形式地址作为基准地址，而Rx 的内容作为修改量。在遇到需要频繁修改地址时，无须修改指令，只要修改变址值就可以了。

变址寻址示意图

如要把一组连续存放在主存单元中的数据（首地址是A）依次传送到另一存储区（首地址为B）中去，则只需在指令中指明两个存储区的首地址A和B（形式地址），用同一变址寄存器提供修改量



K，即可实现（A + K）→B + K。变址寄存器的内容在每次传送之后自动地修改。

7.基址寻址

将基址寄存器Rb 的内容与形式地址A 相加，形成操作数有效地址：EA = (Rb) + A，基址寄存器的内容称为基址值，如下图所示。

基址寻址示意图

基址

基址寻址和变址寻址在形成有效地址时所用的算法是相同的，而且在一些计算机中，这两种寻址方式都是由同样的硬件来实现的。

但这两种寻址方式应用的场合不同，变址寻址是面向用户的，用于访问字符串、向量和数组等成批数据；而基址寻址面向系统，主要用于逻辑地址和物理地址的变换，用以解决程序在主存中的再定位和扩大寻址空间等问题。在某些大型机中，基址寄存器只能由特权指令来管理，用户指令无权操作和修改。

8.相对寻址

相对寻址是基址寻址的一种变通，由程序计数器PC 提供基准地址，即：

EA = (PC) + A，A 是操作数和现行指令之间的相对位置，如下图所示。

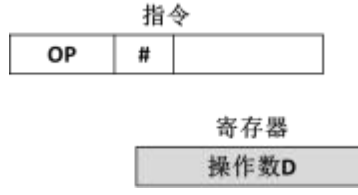

OP
#
程序计
（PC

相对寻址示意图

相对寻址方式的特点是：操作数的地址不是固定的，它随着PC 值的变化而变化，并且与指令地址之间总是相差一个固定值A。当指令地址改变时，由于其位移量不变，使得操作数与指令在可用的存储区内一起移动，所以仍能保证程序的正确执行。采用PC 相对寻址方式编写的程序可在主存中任意浮动，它放在主存的任何地方，所执行的效果都是一样的。指令中给出的位移量A 可正、可负，通常用补码表示，故对于指令地址而言，操作数地址可能在指令地址之前或之后。

9.隐含寻址

隐含寻址是指指令字中不明显地给出操作数的地址，其操作数的地址隐含在操作码或某个寄存器中。如一地址指令格式，只给出一个操作数的地址，另一个操作数隐含在累加器ACC 中，故累加器ACC 对一地址指令格式来说是隐含地址，如下图所示。



隐含寻址示意图

10.堆栈寻址

在堆栈寻址的指令字中没有形式地址码字段，它是一种零地址指令。堆栈寻址要求计算机中设有堆栈。堆栈既可用寄存器组来实现，也可利用主存的一部分空间作堆栈，前者称为硬堆栈，后者称为软堆栈。

【习题演练】

1. 对某个寄存器中操作数的寻址方式称为（ ）寻址。

A.直接

B.间接

C.寄存器

D.寄存器间接

1.【答案】C。解析：操作数在寄存器中的寻址方式为：“寄存器直接寻址”，也叫“寄存器寻址”。当操作数的有效地址在寄存器中时叫“寄存器间接寻址”。

2.操作数所处的位置，可以决定指令的寻址方式。操作数的地址在寄存器中，寻址方式为（ ）。

A.相对寻址

B.直接寻址

C.寄存器寻址

D.寄存器间接寻址

2.【答案】D。解析：操作数包含在指令中的寻址方式为立即寻址；操作数在寄存器中的寻址方式为寄存器寻址；操作数的地址在寄存器中的寻址方式为寄存器间接寻址。

3.变址寻址和基址寻址的有效地址形成方式类似，但（ ）。

A.变址寄存器的内容在程序执行过程巾是不能改变的

B.基址寄存器的内容在程序执行过程中是可以改变的

C.在程序执行过程中，变址寄存器的内容不能改变而基址寄存器的内容可变

D.在程序执行过程中，基址寄存器的内容不能改变而变址寄存器的内容可变

3.【答案】D。解析：基址寄存器的内容由操作系统确定，而变址寄存器的内容由用户确定，因此基址寄存器的内容不能改变而变址寄存器的内容可变。

知识点 6、存储芯片容量的计算

存储芯片的容量与地址线的位数、数据线的位数均有关。地址线和数据线的位数共同反映存储芯片的容量。其公式可写为：

存储芯片容量=$2^A \times D$ 位，其中，A 为地址线根数，D 为数据线根数。

如地址线为 10 根，数据线为4 根，则存储芯片容量为 $2^{10} \times 4 = 4K$ 位；又如地址线为 14 根，数据线为 1 根，则其容量为 $2^{14} \times 1 = 16K$ 位。

【习题演练】

1.某DRAM 芯片，其存储容量为512K × 8 位，该芯片的地址线和数据线数目为（ ）。

A.8，512

B.512，8

C.18，8

D.19，8

1.【答案】D 。解析：$512K \times 8 = 2^{19} \times 8$，所以地址线和数据线分别为 19 和8。

知识点 7、DRAM 的刷新

动态随机存储器刷新的过程实质上是先将原存信息读出，再由刷新放大器形成原信息并重新写入的再生过程。

DRAM 是靠电容来存储信息的，由于存储单元被访问是随机的，有可能某些存储单元长期得不到访问，不进行存储器的读/写操作，其存储单元内的原信息将会慢慢消失。为此，必须采用定时刷新的方法，它规定在一定的时间内，对DRAM 的全部基本单元电路必作一次刷新，一般取2ms（2000μs），这个时间称为刷新周期。刷新是一行行进行的，必须在刷新周期内，由专用的刷新电路来完成对基本电路的逐行刷新，才能保证DRAM 内的信息不会丢失。通常有三种方式刷新：集中刷新、分散刷新和异步刷新。

（一）集中刷新

集中刷新是在规定的一个刷新周期内，对全部存储单元集中一段时间逐行进行刷新，此刻必须停止读/写操作。如对 128 × 128 矩阵的存储芯片进行刷新，刷新的时间相当于 128 个读周期。若读/写周期为 0.5μs，则对 128 行集中刷新共需 128 × 0.5 = 64μs，其余的 1936μs（=2000 – 64）用来读/写或维持信息。由于在这 64μs 时间内不能进行读写操作，故称为“死时间”。

（二）分散刷新

分散刷新把对每行存储单元的刷新分散到每个存取周期内完成。其中，把机器的存取周期分成两段，前半段用来读/写或维持信息，后半段用来刷新。若读/写周期为 0.5μs，则存取周期为 1μs，那么每隔 128μs 就可将 128 行的存储芯片全部刷新一遍，但

这比允许的2ms 间隔要短得多，且存取周期长了，整个系统的速度降低。

（三）异步刷新

前两种方式的集合，既可缩短“死时间”，又充分利用最大刷新间隔为2ms 的特点。如可在2ms 内对128 行各刷新一遍，即每隔15.6μs（2000/128）刷新一行，每行的刷新时间仍为0.5μs，但对每行来说，刷新间隔仍为2ms，“死时间”缩短为0.5μs。

【习题演练】

1.动态RAM 的刷新是以（ ）为单位进行的。

A.存储矩阵

B.行

C.列

D.存储单元

1.【答案】B 。解析：动态RAM 的刷新以行为单位进行。

知识点 8、存储器与 CPU 的扩展连接

（一）存储容量的扩展

由于单片存储芯片的容量总是有限的，很难满足实际的需要，因此，必须将若干存储芯片连在一起才能组成足够容量的存储器，称为存储容量的扩展，通常有位扩展和字扩展。

1.位扩展

位扩展是指增加存储字长，只在位数方向扩展（加大字长）。位扩展的连接方式是将各存储芯片的地址线、片选线和读/写线相应地并联起来，而将各芯片的数据线单独列出。

位扩展组成

	容量	地址	数据

存储器	64K×8	16 位	8 位
存储芯片	64K×1	16 位	1 位

如用64K×1 的SRAM 芯片组成64K×8 的存储器，需要8 个芯片，如上表所示。

当CPU 访问该存储器时，其发出的地址和控制信号同时传给8 个芯片，选中每个芯片的同一单元，其单元的内容被同时读至数据总线的相应位，或将数据总线上的内容分别同时写入相应单元。

2.字扩展

字扩展是指增加存储器字的数量，它仅在字数方向扩展，而位数不变。字扩展将芯片的地址线、数据线、读/写线并联，由片选信号来区分各个芯片。

如用16K×8 的SRAM 组成64K×8 的存储器，需要4 个芯片，如下表所示。

字扩展组成

	容量	地址	数据

存储器	64K×8	16 位	8 位
存储芯片	16K×8	14 位	8 位

在同一时间内四个芯片中只能有一个芯片被选中。

3.字和位同时扩展

字和位同时扩展组成

	容量	地址	数据
存储器	64K×8	16 位	8 位
存储芯片	16K×4	14 位	4 位

字和位同时扩展是指既增加存储字的数量，又增加存储字长。当构成一个容量较大的存储器时，往往需要在字数方向和位数方向上同时扩展，这将是前两种扩展的组合，实现起来也较为容易。如用16K×4 的SRAM 组成64K×8 的存储器，需要8 个芯片，如上表所示。

（二）存储器与 CPU 的连接

存储芯片与CPU 芯片相连时，特别要注意片与片之间的地址线、数据线和控制线的连接。

1.地址线的连接

存储芯片的容量不同，其地址线数也不同，CPU 的地址线数往往比存储芯片的地址线数多。通常总是将CPU 地址线的低位与存储芯片的地址线相连。CPU 地址线的高位或在存储芯片扩充时用，或做其他用途，如片选信号等。

2.数据线的连接

CPU 的数据线数与存储芯片的数据线数也不一定相等。此时，必须对存储芯片扩位，使其数据位数与CPU 的数据线数相等。

3.读/写命令线的连接

CPU 读/写命令线一般可直接与存储芯片的读/写控制端相连，通常高电平为读，低电平为写。

4.片选线的连接

片选线的连接是CPU 与存储芯片正确工作的关键。存储器由许多存储芯片组成，哪一片被选中完全取决于该存储芯片的片选控制端**CS**是否能接收到来自CPU 的片选有效信号。

5.合理选择存储芯片

合理选择存储芯片主要是指存储芯片类型（RAM 或ROM）和数量的选择。通常选用ROM 存放系统程序、标准子程序和各类常数等。RAM 则是为用户编程而设置的。此外，在考虑芯片数量时，要尽量使连线简单方便。

【习题演练】

1.用存储容量为16K×1 位的存储器芯片来组成一个64K×8 位的存储器，则在字方向和位方向上分别扩展了（　）倍。

A.4 和2

B.8 和4

C.2 和4

D.4 和8

1.【答案】D 。解析：用64K 除以16K 等于4，用8 除以1 等于8。

2.RAM 芯片并联时可以使（ ）。

A.存储器存储字长增加

B.存储器地址范围增加

C.存储器速度增加

D.降低存储器的平均价格

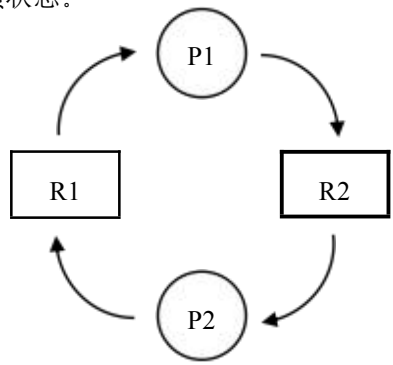
2.【答案】A。解析：并联时可以增加存储器的存储字长。

知识点 9、死锁

死锁是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。

如下图所示，有两个资源R1 和R2 供进程P1 和P2 共享，P1 已占用资源R1，P2 已占用资源R2，

此时若P2 继续要求R1,P1 要求R2，则P1 和P2 之间便会形成僵局，它们都在等待对方释放自己所需的资源，但同时又不释放自己已占用的资源，从而进入死锁状态。



死锁示例

（一）死锁产生的原因

产生死锁的原因可归结为如下两点：

1.竞争资源

当系统中供多个进程共享的资源如打印机、公用队列等，其数目不足以满足诸进程的需要时，会引起诸进程对资源的竞争而产生死锁。

2.进程间推进顺序非法

进程在运行过程中，请求和释放资源的顺序不当，也同样会导致产生进程死锁。

（二）死锁产生的必要条件

死锁的发生必须具备下列四个必要条件：

1.互斥条件

指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。如果此时还有其它进程请求该资源，则请求者只能等待，直至占有该资源的进程用毕释放。

2.请求和保持条件

指进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源又已被其它进程占有，此

时请求进程阻塞，但又对自己已获得的其它资源保持不放。

3.不剥夺条件

指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。

4.环路等待条件

指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P_0，P_1，P_2，…，P_n}中的P_0正在等待一个P_1占用的资源；P_1正在等待P_2占用的资源，……，P_n正在等待已被P_0占用的资源。

（三）死锁的处理

为保证系统中各进程的正常运行，应事先采取必要的措施，来预防发生死锁。系统已经出现死锁后，则应及时检测到死锁的发生，并采取适当措施来解除死锁。目前，处理死锁的方法可归结为以下四种:

1.预防死锁

这是一种较简单和直观的事先预防的方法。该方法是通过设置某些限制条件，破坏产生死锁的四个必要条件中的一个或几个条件，来预防发生死锁。

2.避免死锁

该方法同样是属于事先预防的策略，但它并不须事先采取各种限制措施去破坏产生死锁的四个必要条件，而是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。

3.检测死锁

这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区，而是允许系统在运行过程中发生死锁。但可通过系统所设置的检测机制，及时地检测出死锁的发生，并精确地确定与死锁有关的进程和资源；然后，采取适当措施，从系统中将已发生的死锁清除掉。

4.解除死锁

这是与检测死锁相配套的一种措施。当检测到系统中已发生死锁时，须将进程从死锁状态中解脱

出来。常用的实施方法是撤销或挂起一些进程，以便回收一些资源，再将这些资源分配给已处于阻塞状态的进程，使之转为就绪状态，以继续运行。

【习题演练】

1. 以下各种情况中，属于死锁现象的是（ ）。

A.某进程执行了死循环

B.某进程为了得到某资源，等待了很长的时间

C.某进程为了得到某资源，等待了无限的时间

D.操作系统故障

1.【答案】C。解析：死锁是指多个进程在运行过程中因争夺资源而造成的一种僵局，若无外力作用，它们都将无法再向前推进。由死锁概念可知，只有C 选项符合。

2.死锁预防是保证系统不进入死锁状态的静态策略，其解决方法是破坏产生死锁的四个必要条件之一。下列方法中破坏了“循环等待”条件的是（ ）。

A.银行家算法

B.一次性分配策略

C.剥夺资源法

D.资源有序分配法

2.【答案】D。解析：资源有序分配法可以破坏“循环等待”条件。

3.产生系统死锁的原因可能是由于（ ）。

A.进程释放资源

B.一个进程进入死循环

C.多个进程竞争资源，出现了循环等待

D.多个进程竞争共享型设备

3.【答案】C。解析：产生系统死锁的原因可能是多个进程竞争资源，从而出现了循环等待。

知识点 10、处理机调度的层次

（一）作业调度

作业调度又称高级调度，或宏观调度，根据某种算法，把外存上处于后备队列中的作业调入内存。

（二）中级调度

中级调度又称中程调度，将那些暂时不能运行的进程调至外存上去等待，此时的进程状态称为挂起状态。中级调度实际上就是存储器管理中的对换功能。

（三）进程调度

进程调度又称低级调度，或微观调度，它决定就绪队列中的哪个进程应获得处理机。

进程调度的方式有两种：非抢占方式和抢占方式。抢占方式基于的原则有三个：优先权原则、短作业/进程优先原则、时间片原则。

【习题演练】

1.操作系统中的高级调度是指（ ）。

A.线程调度

B.作业调度

C.进程内、外存交换调

D.进程调度

1.【答案】B。解析：高级调度又称为作业调度、宏观调度或者长程调度，其主要任务是按照一定的原则从外存上处于后备状态的作业中选择一个或者多个，给它们分配内存、输入输出设备等必要的资源，并建立相应的进程，以使作业具有获得竞争处理器的权利。

2.CPU 的调度分为高级、中级和低级三种，其中低级调度是指（ ）调度。

A.作业

B.交换

C.进程

D.线程

2.【答案】C 。解析：进程调度又称低级调度。

知识点 11 、调度算法

（ 一 ）先来先服务调度算法

先来先服务（First Come First Served，FCFS）调度算法是一种最简单的调度算法，该算法既可用于作业调度，也可用于进程调度。FCFS 算法比较有利于长作业（进程），而不利于短作业（进程）。

（ 二 ）短作业/ 进程优先调度算法

短作业/ 进程优先调度算法（Shortest Job/Process First，SJ/PF），是指对短作业或短进程

优先调度的算法。它们可以分别用于作业调度和进程调度。短作业优先（SJF）的调度算法是从后备队列中选择一个或若干个估计运行时间最短的作业，将它们调入内存运行。而短进程优先（SPF）调度算法则是从就绪队列中选出一个估计运行时间最短的进程，将处理机分配给它，使它立即执行并一直执行到完成，或发生某事件而被阻塞放弃处理机时再重新调度。

需要注意的是短进程优先调度算法中，每次选择的是已进入系统的、要求服务时间最短的进程。

SJ/PF 调度算法的缺点在于：该算法对长作业不利；完全未考虑作业的紧迫程度，因而不能保证紧迫性作业/进程会被及时处理；由于作业/进程的长短只是根据用户所提供的估计执行时间而定的，而用户又可能会有意或无意地缩短其作业的估计运行时间，致使该算法不一定能真正做到短作业优先调度。

（三）高优先权优先调度算法

作为作业调度算法，此算法常被用于批处理系统中，作为进程调度算法，还可用于实时系统中。对于最高优先权优先调度算法，其关键在于：它是使用静态优先权，还是用动态优先权，以及如何确定进程的优先权。

1.静态优先权

静态优先权是在创建进程时确定的，且在进程的整个运行期间保持不变。确定进程优先权的依据：进程类型、进程对资源的需求和用户需求。

2.动态优先权

动态优先权是指在创建进程时所赋予的优先权，是可以随进程的推进或随其等待时间的增加而改变的，以便获得更好的调度性能。

动态优先权的变化规律可描述为：

优先权=（等待时间+要求服务时间）/要求服务时间

= 响应时间/要求服务时间

= 响应比

这种算法即为高响应比优先调度算法（Highest Response-ratio Next，HRN），它既照顾了短作业，又考虑了作业到达的先后次序，不会使长作业长期得不到服务。因此，该算法实现了一种较好的折中。但每次要进行调度之前，都须先做响应比的计算，这会增加系统开销。

（四）基于时间片的轮转调度算法

1.时间片轮转法

其基本思想是为每一个进程分配一个时间段，该时间段被称为时间片，即允许该进程运行的时间。每个进程只能依次循环轮流运行，如果时间片结束时进程还在运行，CPU 将剥夺该进程的使用权转而将CPU 分配给另一个进程。如果进程在时间片结束之前阻塞或结束，CPU 当即进行切换。为了实现进程的循环执行，将每次被中止运行的进程存入就绪

队列的末尾，同时将CPU 分配给就绪队列中的队首进程。该算法是一种简单而又公平的算法，使用非常广泛。

2.多级反馈队列调度算法

前面介绍的各种用作进程调度的算法都有一定的局限性。如短进程优先的调度算法，仅照顾了短进程而忽略了长进程，而且如果并未指明进程的长度，则短进程优先和基于进程长度的抢占式调度算法都将无法使用。而多级反馈队列调度算法则不必事先知道各种进程所需的执行时间，而且还可以满足各种类型进程的需要，因而它是目前被公认的一种较好的进程调度算法。

【习题演练】

1.为了对紧急进程或重要进程进行调度，调度算法应采用（　）。

A.先进先出调度算法

B.优先数法

C.最短作业优先调度

D.定时轮转法

1.【答案】B。解析：为了对紧急进程或重要进程进行调度，可遵循优先权准则，采用优先数法。

2.按照作业到达的先后次序调度作业，排队等待时间最长的作业被优先调度，这是指（ ）调度算法。

A.先来先服务

B.最短作业优先

C.定时轮转法

D.优先数法

2.【答案】A。解析：先来先服务按作业进入的先后次序安排。优点是实现简单，缺点是不利于运行时间短的作业。

3.以下（ ）调度算法对CPU 繁忙型进程有利。

A.FCFS

B.时间片轮转

C.多级反馈队列

D.短进程优先

3.【答案】A。解析：CPU 繁忙型进程占用CPU时间比较多，更接近于长进程，故选A，FCFS 算法有利于长进程。

知识点 12、内存的连续分配

连续分配方式，是指为一个用户程序分配一个连续的内存空间。它又可进一步分为：单一连续分配、固定分区分配、动态分区分配以及动态重定位分区分配四种。

（一）单一连续分配

单一连续分配是最简单的一种存储管理方式，只能用于单用户、单任务的操作系统中。它将内存分为系统区和用户区两部分，系统区仅提供给OS使用，通常是放在内存的低址部分；用户区是指除

系统区以外的全部内存空间，提供给用户使用，其中仅能存放一道作业。

（二）固定分区分配

固定分区分配是最早的多道程序的存储管理方式，是指系统先把内存划分成若干个大小固定的分区，一旦划分好，在系统运行期间便不再重新划分。为了满足不同程序的存储需求，各分区的大小可以不等，也可以相等，但在每个分区中只装入一道作业。

采用固定分区分配方法存在以下缺点：

（1）由于预先规定了分区的大小，使得大作业无法装入，用户不得不采取其他技术加以补救，增加了用户的负担；

（2）内存利用率不高，作业很少能恰好填满分区；

（3）固定分区无法实现动态扩充内存空间的要求；

（4）由于分区个数在系统初启时确定，因此会限制多道运行的程序个数，特别不适于分时系统交互型用户及内存变化很大的情形。

（三）动态分区分配

动态分区分配算法根据进程的实际需要，动态地分配内存空间。在实现可变分区分配时，涉及到分区分配中所用的数据结构、分区分配算法和分区的分配与回收操作三个问题。

常用的分配算法主要有以下几种：首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法。

1.首次适应算法

该算法将空闲分区按起始地址递增的次序排列，每次分配均从空闲分区表或空闲分区链首开始顺序查找，直至找到第一个能满足要求的空闲分区为止，并按作业的大小从该分区中分割出一块内存空间分配给请求者，剩余的部分仍留在空闲分区表

或链中。该算法优先使用内存低址部分的空闲分区，从而能在高址部分保留较大的空闲分区，但它会在内存的低端留下很多难以利用的小空闲分区，而每次分配时，对空闲分区的查找都必须经过这些分区，从而会增加查找的开销。

2.循环首次适应算法

该算法由首次适应算法演变而成，每次分配均从上次分配的位置之后开始查找，并将第一个能满足要求的空闲分区分割并分配出去。其特点是内存中的空闲分区分布得更均匀，从而减少查找空闲分区的开销，但它会使存储器中缺乏大的空闲分区。

3.最佳适应算法

该算法将空闲分区按大小递增的次序排列，每次分配均从空闲分区表或空闲分区链首开始顺序查找，并将第一个能满足要求的空闲分区分割并分配出去。该算法尽管被称为“最佳”，但情况并非总是如此，因为每次分配后所切割下来的剩余部分总

是最小的，它会在内存中留下大量难以利用的小空闲分区。

4.最坏适应算法

该算法将空闲分区按大小递减的次序排列。当作业申请一个空闲区时，先检查空闲分区链（表）的第一个空闲区是否大于或等于所要求的内存长度，若该空闲区长度小于要求，则分配失败，否则分配相应的存储空间给用户，然后修改和调整空闲分区链（表）。该算法的优点是产生碎片的几率最小，缺点主要是使内存中缺乏大的空闲区。

（四）可重定位分区分配

在动态分区分配方式中，经过一段时间的分配和回收后，内存中会产生很多小的空闲分区。此时，可能有用户程序因找不到足够大的空闲分区而难以装入，但所有空闲分区容量的总和却足以满足该程序的要求。

上述这些不能被利用的空闲分区可采用以下办法加以解决：将内存中的所有作业进行移动，从而将原来分散的多个空闲分区移到同一处拼接成一个大的空闲分区，以装入用户的作业。这种技术被称为“拼接”或“紧凑”。

可重定位分区分配方式就是在动态分区分配方式的基础上增加紧凑功能，即在找不到足够大的空闲分区、而空闲分区总和却能满足用户的要求时，对内存空间进行紧凑。由于紧凑时，作业要在内存中移动位置，因此，它需要得到动态重定位技术的支持，这也是它被称为动态重定位分区分配的原因。

【习题演练】

1.分区存储管理中的首次适应算法，要求把空闲区按照（　）的次序登记在空闲区表中。

A.大小递减

B.大小递增

C.地址递减

D.地址递增

1.【答案】D。解析：首次适应算法要求对空闲分区按地址从小到大的顺序排列。

2.可重定位内存的分区分配目的为（ ）。

A.解决碎片问题

B.便于多作业共享内存

C.回收空白区方便

D.便于用户干预

2.【答案】A。解析：可重定位内存的分区增加了紧凑的功能，通常在找不到足够大的空闲分区来满足用户需求时使用，解决了碎片问题。

3.在以下存储管理方案中，不适用于多道程序设计系统的是（ ）。

A.单一连续分配

B.固定式分区分配

C.可变式分区分配

D.页式存储管理

3.【答案】A。解析：单一连续分配只能用于单用户、单任务的操作系统。

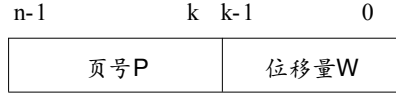
知识点 13、基本的分页存储管理

在分页存储管理方式中，如果不具备页面对换功能，则称为基本的分页存储管理方式，它不具有支持实现虚拟存储器的功能，它要求把每个作业全部装入内存后方能运行。

（一）页面与页表

基本的分页存储管理方式中，系统将一个进程的逻辑地址空间分成若干个大小相等的片，称为页面或页。相应地，将内存空间分成若干个与页面同样大小的块，称为物理块或页框。内存的分配以块为单位，并允许将一个进程的若干页分别装入到多个不相邻的物理块中。

为了地址映射的方便，页面的大小通常设置成2 的幂。如果页面的大小为2^k 字节，逻辑地址的长度为n 位，则分页系统的地址结构如下图所示，可将线性的逻辑地址分成两部分：右边的k 位为页内位移量（即页内地址）W，左边的n-k 位为页号P。

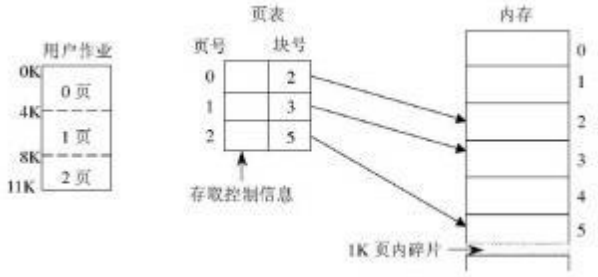


分页地址中的地址结构

在进程运行时，为了能在内存中找到每个页面对应的物理块，系统为每个进程建立了一张页面映射表，简称页表，进程的每个页占页表的一个表项，其中记录了相应页对应的内存块的块号，以及用于分页保护的存取控制信息。

下图给出了分页系统的一个内存分配实例，其中页面大小为4K，用户作业的大小为11K。由于进程的最后一页不足一块，因此造成了存放该页的物

理块中部分空间的浪费，这部分被浪费的空间被称为“页内碎片”。



分页系统中的内存分配

（二）地址变换机构

页式存储管理系统中，逻辑地址到物理地址的转换是在进程执行的过程中，由硬件地址变换机构借助于页表自动进行的。

1.基本的地址变换机构

通常将作业的页表存放在内存中，而在系统中只设置一个页表寄存器PTR（Page-Table Register），当一进程因CPU 调度而转入执行状态时，其页表的

内存始址和长度将从该进程的PCB 中装入页表寄存器。当进程要访问某个逻辑地址中的指令或数据时，地址变换机构自动地将逻辑地址分为页号和页内地址两部分，并将页号与页表寄存器中的页表长度进行比较，若页号大于或等于页表长度，便产生越界中断，否则便以页号为索引去检索页表，从中得到该页的物理块号，送入物理地址寄存器与页内地址拼接，形成对应的物理地址。

2.具有快表的地址变换机构

由于页表存放在内存中，故CPU 每存取一个指令或数据时都要两次访问内存：第一次是访问内存中的页表，以形成物理地址；第二次才根据物理地址存取指令或数据，这使得计算机的处理速度降低近1/2。

为了提高地址变换速度，可在地址变换机构中增设一个具有并行查寻能力的特殊高速缓冲寄存器，又称为“联想寄存器”或“快表”，用以存放当前访问的那些页表项。

在进行地址转换时，地址变换机构自动将逻辑地址中的页号并行地与快表中的所有页号进行比较，若其中有与此相匹配的页号，便可直接从快表中读出该页对应的物理块号，并送到物理地址寄存器中。如果在快表中未找到对应的页号，则仍需访问内存中的页表来进行地址转换，同时还必须将得到的页表项与页号一起装入到快表中，若快表已满，则还需根据置换算法淘汰某个快表项，以装入新的内容。

（三）多级页表

现代的大多数计算机系统都支持非常大的逻辑地址空间，此时，页表就变得非常大，要为它分配一大段连续的内存空间将变得十分困难。

对于该问题，可利用将页表进行分页，并离散地将各个页表存放到内存块中的办法加以解决，此时，必须为离散分配的页表再建立一张页表，称为外层页表，用来记录存放各页表页的内存块号，从

而形成了两级页表。在使用两级页表的分页系统中，每次访问一个数据需访问三次内存，故同样需增设快表来有效地提高访问速度。

如果外层页表仍十分大，则可以将它再进行分页，并离散地存放到内存中，然后再通过一张第二级的外层页表来记录存放各外层页表页的内存块号，这样，便形成了三级页表，并可进一步形成更多级的页表。

【习题演练】

1.碎片现象的存在使得（ ）。

A. 内存空间利用率降低

B. 内存空间利用率提高

C. 内存空间利用率得以改善

D. 内存空间利用率不影响

1.【答案】A。解析：碎片会占用计算机的内存，使部分内存空间不能满足程序运行的最低标准，从而不能被使用，降低内存的使用效率。

2.基本分页存储管理中，若没有引入快表，则每次从主存取指令或取操作数，要（ ）访问主存。

A.1 次

B.2 次

C.3 次

D.4 次

2.【答案】B 。解析：要两次访问内存：第一次是访问内存中的页表，以形成物理地址；第二次根据物理地址存取指令或操作数。

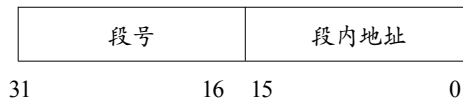
知识点 14 、基本的分段存储管理

用户通常喜欢将自己的作业按逻辑关系划分成若干段，然后通过段名和段内地址来访问相应的程序或数据，还希望能以段为单位对程序和数据进行共享和保护，并要求分段能动态增长。分页系统虽然能较好地解决动态分区的碎片问题，却难以满足

用户的上述要求，因此又引入了分段式存储管理方式。

在分段存储管理方式中，作业的地址空间被划分为若干个段，每个段定义了一组逻辑信息，如主程序段、子程序段、数据段等。每个段都有自己的名字，都从0开始编址，并采用一段连续的地址空间。段的长度由相应的逻辑信息组的长度决定，故各段长度不等。整个作业的地址空间由于是分成多个段，因而是二维的，即其逻辑地址由段号和段内地址所组成。

分段地址中的地址具有如下图的结构：



分段地址中的地址结构

在该地址结构中，允许一个作业最长有64K个段，每个段的最大长度为64KB。

（一）段表

在分段式存储管理系统中，为每个分段分配一个连续的分区，而进程中的各个段可以离散地放入内存不同的分区。系统中为每个进程建立一张段映射表，简称“段表”。每个段在表中占有一个表项，其中记录了该段在内存中的起始地址和段的长度。段表可以存放在一组寄存器中，但更常多的是放在内存中。

在配置了段表后，执行中的进程可通过查找段表找到每个段所对应的内存区。故段表是用于实现从逻辑段到物理内存区的映射。

（二）地址变换机构

为了实现从进程的逻辑地址到物理地址的变换功能，在系统中设置了一个段表寄存器，用于存放正在执行进程的段表始址和长度。在进行地址转换时，地址变换机构将逻辑地址中的段号与段表寄存

器中的段表长度进行比较，若段号不小于段表长度，便产生越界中断；否则便以段号为索引去检索段表，从中得到该段的基址和长度；接着检查段内地址是否超过段长，若超过，则发出越界中断信号；若未超过，则将该段的基址与段内地址相加，从而得到对应的物理地址。

（三）分页和分段的主要区别

分页和分段系统都采用离散分配方式，且都要通过地址映射机构来实现地址变换，这是它们的相同之处，其不同主要表现在以下三个方面：

①页是信息的物理单位，分页是为提高内存的利用率；段则是信息的逻辑单位，分段的目的是为了能更好地满足用户的需要。

②页的大小固定且由系统决定，在系统中只能有一种大小的页面；而段的长度却不固定，取决于用户所编写的程序，通常由编译程序在编译时，根据信息的性质来划分。

③分页的作业地址空间是一维的，即单一的线性地址空间，程序员只需利用一个记忆符，即可表示一个地址；而分段的作业地址空间则是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

（四）段页式存储管理方式

为了既能像分页系统那样有效地利用内存，又能像分段系统那样满足用户多方面的需要，操作系统中又引入了段页式存储管理方式。

段页式系统的基本原理，是先将用户程序分成若干个段，再把每个段分成若干个页。在段页式系统中，其地址结构由段号、段内页号及页内地址三部分所组成。

在段页式系统中，为获得一条指令或数据须三次访问内存。第一次访问是访问内存中的段表，从中取得页表始址；第二次访问内存中的页表，从中

取出该页所在的物理块号，并将该块号与页内地址一起形成物理地址；第三次访问是取指令或数据。

【习题演练】

1.页是（ ）单位，由（ ）划分，它的长度（ ）。

A.逻辑，系统，定长

B.逻辑，用户，不定长

C.物理，用户，不定长

D.物理，系统，定长

1.【答案】D。解析：页是物理单位，由系统划分，它的长度固定。

2.下列方法中，解决碎片问题最好的存储管理方法是（ ）。

A.基本页式存储管理

B.基本分段存储管理

C. 固定大小分区管理

D.不同大小分区管理

2.【答案】A。解析：离散分配方式将一个作业离散地存放到内存中，从而使系统无须紧凑便能很好地解决碎片问题，而离散分配方式中，分页是为解决碎片问题，提高内存的利用率，分段的目的则是为了能更好地满足用户的需要，故选A。

3.分段管理提供（ ）维的地址结构。

A.1

B.2

C.3

D.4

3.【答案】B。解析：分段存储管理的地址空间是二维的，标识一个地址时，既需给出段名，又需给出段内地址。

知识点 15 、虚拟存储器

（一）局部性原理

程序局部性原理是指程序在执行时将呈现出局部性规律，即在一较短时间内，程序的执行仅局限于某个部分；相应地，它所访问的存储空间也局限于某个区域。局限性还表现在下述两方面：

1.时间局限性

如果程序中的某条指令一旦执行，则不久以后该指令可能再次执行；如果某数据被访问过，则不久以后该数据可能再次被访问。产生时间局限性的典型原因是由于在程序中存在着大量的循环操作。

2.空间局限性

一旦程序访问了某个存储单元，在不久之后，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之内，其典型情况便是程序的顺序执行。

基于局部性原理产生了虚拟存储器。

（二）虚拟存储器的实现

虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。

虚拟存储器的实现，建立在离散分配的存储管理方式的基础上。目前，所有的虚拟存储器都是采用下述方式之一实现的：

1.请求分页系统

在分页系统的基础上，增加了请求调页功能和页面置换功能，便形成页式虚拟存储系统。它允许只装入少数页面的程序及数据，便启动运行。之后再通过调页功能及页面置换功能，陆续地把即将要运行的页面调入内存，同时把暂不运行的页面换出到外存上。

2.请求分段系统

在分段系统的基础上，增加了请求调段及分段置换功能后，便形成段式虚拟存储系统。它允许只装入少数而非所有段的用户程序和数据，即可启动运行。之后再通过调段功能和段的置换功能将暂不运行的段调出，同时调入即将运行的段。

（三）虚拟存储器的特征

虚拟存储器具有多次性、对换性和虚拟性三大主要特征。

1.多次性

多次性是指虚拟存储器将一个作业分成多次调入内存。多次性是虚拟存储器最重要的特征，任何其它的存储管理方式都不具有这一特征。

2.对换性

在作业运行期间，虚拟存储器允许将内存中暂时不能运行的进程或暂时不用的程序或数据，调出到外存上，以便腾出足够的内存空间，再把具备运

行条件的进程或进程所需要的程序和数据调入内存。

如果对换是以整个进程为单位的，便称之为“整体对换”或“进程对换”。这种对换被广泛地应用于分时系统中，其目的是用来解决内存紧张问题，并可进一步提高内存的利用率。而相应的，如果对换是以“页”或“段”为单位进行的，则分别称之为“页面对换”或“分段对换”，又统称为“部分对换”。它们是实现虚拟存储器的基础。

3.虚拟性

虚拟性是指能够从逻辑上扩充内存容量，使用户所看到的内存容量远大于实际内存容量。虚拟性是实现虚拟存储器的最重要的目标。

虚拟性是以多次性和对换性为基础的，仅当系统允许将作业分多次调入内存，并能将内存中暂时不运行的程序和数据换至盘上时，才有可能实现虚拟存储器；而多次性和对换性又必须建立在离散分配的基础上。

（四）请求分页存储管理方式

请求分页系统是建立在基本分页存储管理基础上的，增加了请求调页功能和页面置换功能。请求分页便成为目前最常用的一种实现虚拟存储器的方式。

为了确定所缺的页面调入内存的时机，可采取预调页策略或请求调页策略：

1.预调页策略

如果进程的许多页是存放在外存的一个连续区域中，则一次调入若干个相邻的页，会比一次调入一页更高效些。但如果调入的一批页面中的大多数都未被访问，则又是低效的。可采用一种以预测为基础的预调页策略，将那些预计在不久之后便会被访问的页面预先调入内存。这种策略主要用于进程的首次调入时，由程序员指出应该先调入哪些页。

2.请求调页策略

当进程在运行中需要访问某部分程序和数据时，若发现其所在的页面不在内存，便立即提出请求，由OS将其所需页面调入内存。由请求调页策略所确定调入的页，是一定会被访问的，再加之请求调页策略比较易于实现，故在目前的虚拟存储器中大多采用此策略。

【习题演练】

1.实现虚拟存储器的目的是（ ）。

A.实现存储保护

B.实现程序浮动

C.扩充辅存容量

D.扩充主存容量

1.【答案】D。解析：实现虚拟存储器的目的是为了扩充主存容量。

2.虚拟存储器的最大容量是由（ ）决定的。

A.计算机系统的地址结构和外存空间

B.页表长度

C. 内存空间

D.逻辑空间

2.【答案】A。解析：虚存容量不是无限的，最大容量受外存可利用的总容量和计算机总线地址结构的限制。

3.具有虚拟存储功能的管理方法包括（ ）。

A.可变分区存储管理

B.分页式存储管理

C.请求分段存储管理

D.段页式存储管理

3.【答案】C。解析：虚拟存储功能涉及内存的存储空间扩充问题。请求分段存储管理：作业可以装入不连续的存储空间中，且作业不要求全部装入内存就可以运行，因此具有空间扩充功能；段页式存储管理：作业虽然可以装入不连续的存储空间中，但是作业仍要求全部装入才可运行，因此不具备空间扩充功能。

知识点 16、页面置换算法

置换算法的好坏直接影响系统的性能。若采用的置换算法不合适，可能出现这样的现象：刚被换出的页，很快又被访问，为把它调入而换出另一页，之后又访问刚被换出的页……如此频繁地更换页面，以致系统的大部分时间花费在页面的调度和传输上。此时，系统看起来很忙，但实际效率却很低。这种现象称为“抖动”。

好的页面置换算法能够适当降低页面更换频率（减少缺页率），尽量避免系统“抖动”。为评价一个算法的优劣，可将该算法应用于一个特定的存储访问序列上，并且计算缺页数量，存储访问序列也叫页面走向。

（一）最佳（OPT）置换算法

最佳置换算法（Optimal，OPT）是选择不再访问的页面或者是在未来最长时间内不再被访问的页

面予以淘汰。最佳页面置换算法是在理论上提出的一种算法，具有最好的性能，但实现是困难的，因为它需要人们预先知道一个进程在整个运行过程中页面走向的全部情况。不过，这个算法可用来衡量其他算法的优劣。

（二）先进先出（FIFO）页面置换算法

FIFO 算法总是淘汰最先进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰，该算法的出发点是最早调入内存中的页面，其不再被使用的可能性会比较高。其实现简单、直观，对按线性顺序访问的程序比较合适，面对其他情况则效率不高，因为经常被访问的页面，往往在内存中因停留得最久而被淘汰。

一般而言，对于任一作业或进程，如果给它分配的内存页面数越接近于它所要求的页面数，则发生缺页的次数会越少。在极限情况下，这个推论是成立的，如果给一个进程分配了它所要求的全部页

面，则不会发生缺页现象。但是，使用FIFO 算法时，在给进程或作业分配满足它所要求的页面数时，有时会出现分配的页面数增多，缺页次数反而增加的奇怪现象，这种现象称为Belady 现象。

（三）最近最久未使用（LRU）置换算法

LRU 算法是根据页面调入内存后的使用情况进行决策的。由于无法预测各页面将来的使用情况，只能以“最近的过去”作为“最近的将来”的近似，选择在最近一段时间里最久没有使用过的页面予以置换。因此，LRU 算法是与每个页面最后使用的时间有关的，当必须置换一个页面时，LRU 算法选择过去一段时间内最久未使用的页面。

LRU 置换算法是一种比较好的算法，但需要硬件的支持：移位寄存器或栈。

关于确定最后使用时间的问题，对于移位寄存器而言，当进程访问某物理块时，要将相应寄存器的位置成“1”。此时，定时信号将每隔一定时间将

寄存器右移一位。如果把n 位寄存器的数看做是一个整数，那么，具有最小数值的寄存器所对应的页面，就是最近最久未使用的页面。

对于栈而言，每当访问一个页面时，便将它的页面号从栈中移出，压入栈顶。因此栈顶始终是最新被访问页面的编号，而栈底则是最近最久未使用页面的页面号。

（四）Clock 置换算法

LRU 算法是较好的一种算法，但它要求有较多的硬件支持，故在实际应用中大多采用LRU 的近似算法。Clock 算法就是用得较多的一种LRU 近似算法。

Clock 置换算法采用循环队列机制构造页面队列，形成类似于钟表面的环形表，队列指针相当于钟表面上的表针，指向可能要淘汰的页面。它在存储分块表的每一表项中增加一个访问位，操作系统定期地将它们置为0 。当某一页被访问时，由硬件

将该位置为 1 。过一段时间后，通过检查这些位可以确定哪些页使用过，哪些页自上次置0 后还未使用过，可把该位是0 的页淘汰出去，因为在最近一段时间里它未被访问过。由于该算法是循环地检查各页面的使用情况，故称为Clock 算法。但因该算法只有一位访问位，只能用它表示该页是否已经使用过，而置换时是将未使用过的页面换出去，故又把该算法称为最近未用算法（Not Recently Used，NRU）。

【习题演练】

1. 在请求调页中可采用多种置换算法，其中LRU 是（ ）置换算法。

A.最佳

B.最近最久未用

C.最近未用

D.最少使用

1.【答案】B 。解析：LRU 是最近最久未用置换算法。

2.Clock 置换算法又称为（ ）。

A.最久驻留置换算法

B.最近最久未使用置换算法

C.最近未使用置换算法

D.最少使用置换算法

2.【答案】C 。解析：最近未使用算法是Clock置换算法。

知识点 17 、线性表

（一）概念

线性表是最基本、最简单、也是最常用的一种数据结构。

线性表是具有n （n≥0）个类型相同的数据元素组成的有限序列。

线性表的长度：线性表中元素的个数。

空表：长度为0 的线性表。

（ 二 ）特点

线性表中一定存在唯一的“第一元素”。

线性表中一定存在唯一的“最后元素”。

除最后一个元素之外，均有唯一的后继（后件）。

除第一个元素之外，均有唯一的前驱（前件）。

（ 三 ）表示方式

1.线性表的顺序表示

线性表中的数据元素是用一组地址连续的存储单元依次存储的。

顺序表是在计算机内存中以数组的形式保存的线性表，是指用一组地址连续的存储单元依次存储数据元素的线性结构。线性表采用顺序存储的方式存储就称之为顺序表。顺序表是将表中的结点依次存放在计算机内存中一组地址连续的存储单元中。

2.线性表的链式表示

（1）线性链表

用一组任意的存储单元存储线性表的数据元素，存储单元不一定是连续的，可以连续，也可以不连续。

（2）循环链表

循环链表中最后一个结点的后继指针指向头结点，使整个链表形成一个环形。

（3）双向链表

在双向链表中，每个结点有两个指针域，一个指向它的直接后继，另一个指向它的直接前驱。

【习题演练】

1.下列有关线性表的叙述中，正确的是（ ）。

A.线性表中的元素之间是线性关系

B.线性表中至少有一个元素

C.线性表中任何一个元素有且仅有一个直接前驱

D.线性表中任何一个元素有且仅有一个直接后继

1.【答案】A。解析：线性表可以为空，并且线性表中的第一个结点是没有前驱结点的，最后一个结点没有后继结点。

知识点 18、栈和队列

（一）栈的定义

1.栈的定义

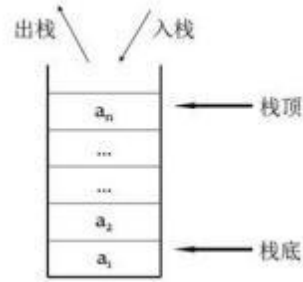
栈是一种只能在一端进行插入或删除操作的线性表。栈中的数据元素是线性关系。

栈顶：允许进行插入或删除操作的一端。

栈底：不允许进行插入和删除操作，固定不变的一端。

入栈：栈的插入操作。

出栈：栈的删除操作。



2.栈的特点

先进后出（First In Last Out，简称FILO）、后进先出（Last In First Out，简称LIFO）。

3.栈的存储结构

（1）顺序栈

使用顺序存储结构存储栈。

（2）链式栈

使用链式存储结构存储栈。

（二）队列的定义

1.队列的定义

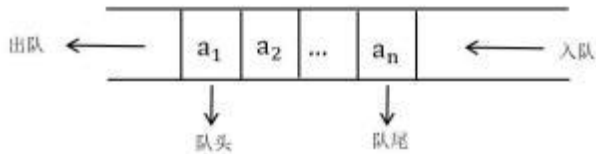
队列是一种运算受限制的线性表，元素的添加在表的一端进行，而元素的删除在表的另一端进行。

队头：允许删除元素的一端。

队尾：允许添加元素的一端。

入队：向队列添加元素。

出队：从队列中删除元素。



2. 队列的特点

先进先出（FIFO）（先入队的元素先出队，后入队的元素后出队）。

3.存储结构

顺序队：使用顺序存储结构的队列。

链队：使用链式存储结构的队列。

【习题演练】

1. 下述有关栈和队列的区别，说法错误的是（ ）。

A.栈是限定只能在表的一端进行插入和删除操作

B. 队列是限定只能在表的一端进行插入和在另一端进行删除操作

C.栈和队列都属于线性表

D.栈的插入操作时间复杂度都是O（1），队列的插入操作时间复杂度是O（n）

1.【答案】D。解析：栈的插入操作时间复杂度都是O（1），队列的插入操作时间复杂度是O（1）。因为都是在端点处进行的操作。

2.为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是（ ）。

A.栈

B.队列

C.树

D.图

2.【答案】B。解析：主机将数据依次写入，打印机依次取出，应该属于先进先出的特点，既使用的逻辑结构是队列。

知识点 19、树

（一）定义

树形结构是一种重要的非线性结构，树是n 个结点的有限集合，在任一棵非空树中：

（1）有且仅有一个称为根的结点。

（2）其余结点可分为m 个互不相交的集合，而且这些集合中的每一集合都本身又是一棵树，称为根的子树，因此树是递归结构。

（二）基本术语

结点：包含一个数据元素及若干指向其子树的分支。

结点的度数：结点的非空子树个数。

树的度：树中各节点中度的最大值。

分支结点：度不为0 的结点。

叶子结点：度为0 的结点。

孩子：结点的子树的根。

双亲：结点的直接前驱。

兄弟：同一个双亲的孩子之间互为兄弟。

堂兄弟：双亲在同一层的结点互为堂兄弟。

祖先：从根到某结点的路径上的所有结点，都是这个结点的祖先。

子孙：以某结点为根的子树中的所有结点，都是该结点的子孙。

层次：从根开始，根为第一层，根的孩子为第二层，以此类推。

树的深度（或者高度）：树中结点的最大层数。

结点的深度和高度：结点的深度是从根结点算起的，根结点的深度为1；结点高度是从最底层的叶子结点算起的，最底层叶子结点的高度为1。

有序树：子树按照一定的次序从左向右排列，相对次序不能随意变换。

无序树：子树无一定的次序排列，相对次序可以随意变换。

丰满树：即理想平衡树，要求除最底层外，其他层都是满的。

森林：是由零个或多个不相交的树所组成的集合。

【习题演练】

1.如果在数据结构中每个数据元素只可能有一个直接前驱，但可有多个直接后继，则该结构是（ ）。

A.栈

B. 队列

C.树

D. 图

1.【答案】C。解析：树结构的每个结点都有一个前驱，但可以有多个后继。

知识点 20 、二叉树

（一）定义

二叉树是一个连通的无环图，并且每一个顶点的度不大于 2 。有根二叉树还要满足根结点的度不大于 2 。有了根结点之后，每个顶点定义了唯一的父结点，和最多 2 个子结点。

（二）主要性质

①二叉树的第 i 层上至多有 2^{i-1} 个结点。

②深度为 k 的二叉树至多有 $2^{k}-1$ 个结点。

（补充概念：

a.满二叉树：深度为k，有2^k-1 个结点。

b.完全二叉树：给满二叉树的结点编号，从上至下，从左至右，n 个结点的完全二叉树中结点在对应满二叉树中的编号正好是从1 到n。

③叶子结点n0，度为2 的结点为n2，则n0 = n2+1。

④n 个结点的完全二叉树深度为 lpA g2n,+1。

⑤n 个结点的完全二叉树，结点按层次编号。

a.i 的双亲是 1 n/2，如果i=l 时为根（无双亲）。

b.i 的左孩子是2i，如果2i>n，则无左孩子。

c.i 的右孩子是2i + 1，如果2i + 1>n 则无右孩子。

（三）存储结构

1.顺序存储结构

顺序存储结构即用一个数组来存储一棵二叉树，这种存储方式最适合于完全二叉树，用于存储

一般二叉树则会浪费大量的存储空间。将完全二叉树中的结点值按编号依次存入一个一维数组中，即完成了一棵二叉树的顺序存储。例：

结点		A	B	C	D	E
数组下标	0	1	2	3	4	5

2.链式存储结构

二叉树中的每个结点用一个链结点来存放，结点结构如下：

lchild	data	rchild

其中，data 表示结点数据域，用于存储对应的数据元素；lchild 和rchild 分别表示左指针域和右指针域，分别用于存储左孩子结点和右孩子结点的位置。

（四）遍历算法

若以L、D、R 分别表示遍历左子树、遍历根结点和遍历右子树，则有六种遍历方案：DLR、LDR、LRD.DRL 、RDL 、RLD 。若规定先左后右，则只有

前三种情况，分别是：DLR--先（根）序遍历；LDR--中（根）序遍历；LRD--后（根）序遍历。

【习题演练】

1.设非空二叉树中度数为0 的结点数为n0，度数为1 的结点数为n1，度数为2 的结点数为n2，则下列等式成立的是（ ）。

A.n0=n1+n2

B.n0=2n1+1

C.n0=n2+1

D.n0=n1+1

1.【答案】C。解析：二叉树中度为0 的结点个数等于度为2 的结点个数加一。

2.一棵树高为k 的完全二叉树至少有（ ）个结点。

A.2^k-1

B.$2^{k-1}-1$

C.2^{k-1}

D.2k

2.【答案】C。解析：完全二叉树层次最小为k时，前k-1层都是满的，第k层有一个结点。

3.深度为5的完全二叉树的结点数不可能是（ ）。

A.15

B.16

C.17

D.18

3.【答案】A。解析：深度为5的完全二叉树结点最多为二的5次方减一个，即31个，最少为16个。

4.二叉树的先序遍历和中序遍历如下：先序遍历：EFHIGJK，中序遍历：HFIEJKG，则二叉树根结点为（ ）。

A.E

B.F

C.G

D.H

4.【答案】A 。解析：先序遍历顺序为根左右，所以树根为E。

知识点 21 、图

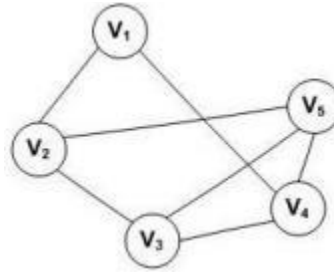
（ 一 ）图的定义

图是由顶点集V 和集合E（边的集合）组成的，可以定义为G=（V，E）。V 是顶点的非空有穷集合，E 是可空的边的有穷集合。

（ 二 ）图的术语

1.无向图

对于一个图，若每条边都是没有方向的，则称该图为无向图。如下图：



对于此无向图，(Vi，Vj)和(Vj，Vi)表示的是同一条边。

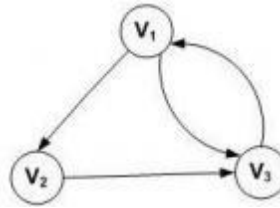
无向图的顶点集和边集分别表示为：

V(G)={V1，V2，V3，V4，V5}

E(G)={(V1，V2)，(V1，V4)，(V2，V3)，(V2，V5)，(V3，V4)，(V3，V5)，(V4，V5)}

2.有向图

对于一个图，若每条边都是有方向的，则称该图为有向图。如下图：



在此有向图中，<Vi，Vj>和<Vj，Vi>是两条不同的有向边。有向边又称为弧。终端点也称为弧头，初始点也称为弧尾。有向图的顶点集和边集分别表示为：

V(G)={V1，V2，V3}

E(G)={<V1，V2>，<V2，V3>，<V3，V1>，<V1，V3>}

3.无向完全图和有向完全图

具有n(n-1)/2 条边的无向图为无向完全图。

具有n(n-1)条边的有向图称为有向完全图。

4.稀疏图

有很少条边或弧的图。

5.稠密图

有很多条边或弧的图

6.顶点

图中的数据元素通常称为顶点。

7.顶点的度

无向图中，顶点的度表示以该顶点作为端点的边的数目。

有向图中，顶点的度分为入度和出度。入度表示以该顶点为终点的边的数目，出度是以该顶点为起点的边的数目，该顶点的度等于其入度、出度之和。

8.回路、环

指一条路径的起点和终点为一个顶点。

9.连通图

在无向图中，如果从顶点v1 到顶点v2 有路径，则称v1 和v2 是连通的。如果图中任意两个顶点都是连通的，则称图是连通图。

10.连通分量

指的是无向图中的极大连通子图。

11.强连通图

在有向图G 中，对任意一对顶点vi 和vj（vi ≠vj），若从vi 到vj 和从vj 到vi 都存在路径，则称G 是强连通图。

12.强连通分量

有向图中的极大强连通子图称为有向图的强连通分量。

（三）遍历算法

从图的某一顶点出发，访问图中其余顶点，且使每个顶点仅被访问一次，这一过程称为图的遍历。图的遍历分为深度优先搜索遍历和广度优先搜索遍历。

1.广度优先

广度优先搜索类似于树的按层次遍历。选取图中任意一个顶点V_i作为出发点，按照下列步骤遍历图。

（1）首先访问出发点V_i。

（2）接着依次访问V_i的所有未被访问过的邻接点V_{i1}，V_{i2}，V_{i3}，…，V_{it}并均标记为已访问过。

（3）然后再按照V_{i1}，V_{i2}，…，V_{it}的次序，访问每一个顶点的所有未被访问过的顶点，并均标记为已访问过。

（4）依此类推，直到图中所有和初始出发点V_i有路径相通的顶点都被访问过为止。

2.深度优先

深度优先搜索（DFS）类似于树的先序遍历。

【习题演练】

1.具有n 个结点的连通图至少有（　）条边。

A.n-1

B.n

C.n（n-1）/2

D.2n

1.【答案】A。解析：具有n 个结点的连通图至少有n-1 条边。

2. 在有向图中每个顶点的度等于该顶点的（ ）。

A.入度

B. 出度

C.入度与出度之和

D.入度与出度之差

2.【答案】C 。解析：有向图的某个顶点v，把以v 为终点的边的数目称为v 的入度；以v 为始点的边的数目称为v 的出度；v 的度则定义为该顶点的入度和出度之和。

3.设连通图G 中的边集E={（a，b），（a，e），（a，c），（b，e），（e，d），（d，f），（f，c）}，则从顶点a 出发可以得到一种深度优先遍历的顶点序列为（ ）。

A．abedfc

B．acfebd

C．aebdfc

D．aedfcb

3.【答案】A。图的深度优先遍历类似于树的前序遍历。采用的搜索方法的特点是尽可能先对纵深方向进行搜索。

知识点 22、数据模型

（一）数据模型的概念

数据模型是对现实世界数据特征的抽象。也就是说，数据模型是用来描述数据、组织数据和对数据进行操作的。

（二）两类数据模型

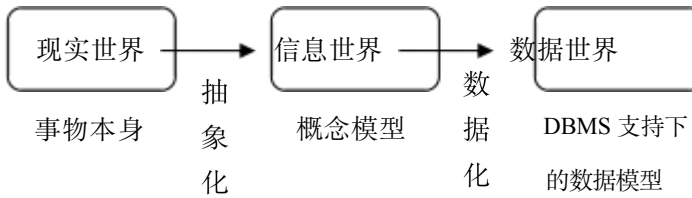
在数据库中模型主要分为两大类：一类为概念模型；另一类为逻辑模型和物理模型。

概念模型也称为信息模型，在数据库设计阶段，由设计员按照用户观点建模实现对现实世界的概念抽象。

逻辑模型包括网状、层次、关系和对象模型等，由设计人员按照计算机观点建模实现概念模型到适应某DBMS 的逻辑模型的转变。再由DBMS 完成逻辑到物理模型的转变。

物理模型是数据最底层的抽象，描述数据的存储方式和方法。

（三）现实世界的数据化过程



现实世界的数据化过程

【习题演练】

1.常见的逻辑模型有3 种，它们是（ ）。

A.字段名、字段类型和记录

B.层次、关系和网状

C.环状、层次和关系

D. 网状、关系和语义

1.【答案】B。解析：常见的数据模型有3 种，即按图论理论建立的层次结构模型和网状结构模型以及按关系理论建立的关系结构模型。

知识点 23、概念模型

概念模型用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有利工具，也是数据库设计人员和用户之间进行交流的语言。

（一）常用术语

1.实体

客观存在并且可以区分的具体事物或者抽象概念。例如，一个学生、一个宿舍、一个操作流程等等。

2.属性

属性是对客观事物特征的一种反映，是实体具有的某个特征。例如，学生实体中有学号属性、姓名属性、性别属性，年龄属性、所在系属性等。属性有属性名称和属性值之分，例如，姓名为属性名，则“李白”就是这个属性的值。

3.码

在现实生活中，实体是可以互相区分的，所以没有两个完全相同的实体，即不能够有两个实体在各自对应属性上的属性值都是相同的。

4.域

一个属性的取值范围被称为域。例如，性别属性的属性值只能为“男”或者只能为“女”。

5.实体型

实体集的名称及其所有属性名的集合称为实体型。例如，学生（学号，姓名，性别，年龄，所在系）就是学生实体集的实体型。

6.实体集

所有属性名完全相同的实体集合在一起，称为实体集。比如说学生实体集，教师实体集等。

（二）概念模型的表示方法：实体-联系图

E-R 图中文称为实体-联系图，它是概念模型中的一种表示形式。E-R 图主要包含三个概念，分别是：实体集、联系集和属性。

1.实体-联系图的表示方法

实体型：用矩形表示，矩形框内写明实体名。

属性：用椭圆表示，并用无向边将其对应的实体型连接起来。

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型（1:1、1:N 或者M：N）。

2.联系

两个实体集之间的联系可归纳为以下三类：

（1）一对一联系（1:1）

如果对于实体集A 中的每一个实体，实体集B 中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集A 与实体集B 具有一对一关系，记为 1：1 。例如，一个学校只能有一个校长；一个账号只能提供给一个人使用等。

（2）一对多联系（1：N）和多对一联系（N：1)

如果对于实体集A 中的每一个实体，实体集B 中有N 个实体（N>=0）与之联系，反之，对于实体集B 中的每一个实体，实体集A 中至多只有一个实体与之联系，则称实体集A 与实体集B 有一对多的联系，记为 1：n 。例如，一个工厂可以有若干个员工，而一个员工只能属于一个工厂；一个导师可以有若干个学员，而一个学员只能属于一个导师等。

（3）多对多联系（M：N）

如果对于实体集A 中的每一个实体，实体集B中有N 个实体（N>=0）与之联系，反之，对于实体集B 中的每一个实体，实体集A 中也有M 个实体（M>=0）与之联系，则称实体集A 与实体集B具有多对多的联系，记为M：N 。例如，一门课程可以被多个学生选择，一个学生可以选择多门课程等。

【习题演练】

1.下列实体类型的联系中，属于一对一联系的是（ ）。

A.教研室对教师的所属联系

B.父亲对孩子的联系

C.省对省会的所属联系

D.供应商与工程项目的供货联系

1.【答案】C 。解析：实体之间的联系分为：一对一、一对多和多对多。因为每一个省都只有一个省会，而一个省会只属于一个省，所以省和省会之

间是一对一的联系。而A 选项属于一对多的关系；B 选项属于一对多的关系；D 选项属于多对多的关系。

2.在E-R 图中，用长方形表示（ ），用椭圆表示（ ）。

A.联系、属性

B.属性、实体

C.实体、属性

D.什么也不代表、实体

2.【答案】C 。解析：矩形表示实体，椭圆表示属性。

3.关系模型中，候选码（ ）。

A.可由多个任意属性组成

B.至多由一个属性组成

C.可由一个或多个其值能惟一标识该关系模式中任何元组的属性组成

D. 以上都不是

3.【答案】C。解析：候选码可以是一个也可以是多个，但是必须能唯一标识元组。

知识点 24、关系模型

（一）简介

关系数据模型是目前最重要的一种数据模型。关系数据库系统采用关系数据模型作为数据的组织方式。有层次数据模型和网状模型相比，关系模型概念简单、清晰，并且具有严格的数据基础，形成了关系数据理论，操作也直观、容易，因此易学易用。支持关系数据模型的DBMS 称为关系型数据库管理系统RDBMS 。与其他数据模型相同，关系数据模型也是由数据结构、数据操作和完整性约束三个部分组成。

（二）关系模型的性质

列是同质的，即同一列的分量值应该出自相同的域。

列名是唯一的，即在同一个关系中不能出现完全相同的两个属性名称。

行的顺序无关，即元组与元组之间互换位置，不相互影响。

列的顺序无关，即列于列之间互换位置，不相互影响。

任何两行不能完全相同，即不能出现两个完全相同的元组。

分量必须是原子量，即是不可分的基本数据项。

【习题演练】

1.下列叙述正确的为（ ）。

A.关系中元组没有先后顺序，属性有先后顺序

B.关系中元组有先后顺序，属性没有先后顺序

C.关系中元组没有先后顺序，属性也没有先后顺序

D.关系中元组有先后顺序，属性也有先后顺序

1.【答案】C。解析：关系中元组的顺序无关，属性的顺序也无关。

知识点25、关系操作和完整性约束

（一）基本的关系操作

1.查询操作

关系的查询表达能力很强，是关系操作最主要的部分。

查询操作又可以分为：选择、投影、连接、除、并、差、交、笛卡尔积等。

2.更新操作

更新操作又可以分为：插入、删除、修改。

3.基本操作

关系的基本操作有五种，分别是：选择、投影、并、差、笛卡尔积。

其他操作是可以用基本操作来定义和导出的。就像乘法可以用加法定义和导出一样。

4.关系操作的特点

关系操作的特点是集合操作方式，即操作的对象和结果都是集合。这种操作方式也称为一次一集合的方式。相应地，非关系数据模型的数据操作方式则为一次一记录的方式。

（二）关系的三类完整性

关系模型中有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系系统自动支持。用户定义的完整性是应用领域需要遵循的约束条件，体现了具体领域中的语义约束。

1.实体完整性

（1）规则

若属性（指一个或者一组属性）A 是基本关系 R 的主属性，则A 不能取空值。

按照实体完整性规则的规定基本关系的主码都不能取空值。如果主码由若干属性组成，则所有这些主属性都不能取空值。

（2）具体说明

①实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。

②现实世界中的实体是可以区分的，即它们具有某种唯一性标识。

③相应地，关系模型中以主码作为唯一标识。

④主码中的属性即主属性不能取空值。如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与②点相矛盾，因此这个规则称为实体完整性。

2.参照完整性

若属性（或属性组）F 是基本关系R 的外码，它与基本关系S 的主码K_s相对应（基本关系R 和S 不一定是不同的关系），则对于R 中每个元组在F 上的值必须为：

①或者取空值（F 的每个属性值均为空值）；

②或者等于S 中某个元组的主码值。

3.用户自定义完整性

任何关系数据库系统都应该支持实体完整性和参照完整性。这是关系模型所要求的。

除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。例如某个属性必须取值唯一、某个非主属性也不能取空值等。

【习题演练】

1.若属性A 是关系R 的主属性，则A 不能为空，该规则称为（ ）。

A.实体完整性规则

B.属性完整性规则

C.参照完整性规则

D.用户定义完整性规则

1.【答案】A。解析：数据完整性包括：实体完整性、参照完整型和用户定义完整性。实体完整性即规定主键不能为空。

2.关系模型有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。定义外键实现的是（ ）完整性。

A.实体完整性

B.参照完整性

C.用户定义的完整性

D.实体完整性、参照完整性和用户定义的完整性

2.【答案】B。解析：定义外键实现的是参照完整性。

知识点 26、SQL 常用语句

SQL（Structured Query Language），即结构化查询语言，是关系数据库的标准语言。当前，几乎所有的关系数据库管理系统软件都支持 SQL，许多软件厂商对 SQL 的基本命令集还进行了不同程度的扩充和修改。

SQL 集数据查询、数据操纵、数据定义和数据控制功能于一体，是一个综合的、通用的、功能极强、简洁易学的语言。

（一）数据查询

1.语句格式

SELECT [ALL|DISTINCT] column_list

FROM table_list

[WHERE search_condition]

[GROUP BY group_by_list]

[HAVING search_condition]

[ORDER BY order_list[ASC|DESC]]

2.运算符

算术比较运算符：=，<，<=，>，>=，<>，!>，!<

逻辑运算符：AND、OR、NOT

所属集合运算符：IN、NOT IN

谓词：EXISTS（存在)、BETWEEN…AND（范围）、LIKE（匹配）、IS NULL（空值）

3.统计函数

常见的统计函数

统计函数	描述
COUNT（*）	计算记录的个数
COUNT（[DISTINCT]列名）	对一列中的值计算个数
SUM（[DISTINCT]列名）	求某一数值型列的总和
AVG（[DISTINCT]列名）	求某一数值型列的平均值

统计函数	描述
MAX（[DISTINCT]列名）	求某一列的最大值
MIN（[DISTINCT]列名）	求某一列的最小值

4.连接查询

在查询的数据涉及到多个表时，必须用连接条件将这些表连接起来。

5.嵌套查询

在SQL 语句中，可以将一个查询嵌入在另一个查询的WHERE 子句中，这类查询称为嵌套查询。一般内层的查询称为子查询，将外层的查询称为父查询。嵌套的SELECT 查询使得SQL 可以实现各种复杂的查询，子查询必须用括号括起来。

6.示例

a.查询计算机系年龄在20 岁以下的学生姓名：

```
SELECT Sname
FROM Student
WHERE Sdept='CS' AND Sage<20;
```

b.查询选修了3 门以上课程的学生学号：

SELECT Sno

FROM SC

GROUP BY Sno

HAVING COUNT(*)>3;

（二）数据更新

1.INSERT 语句

（1）语句格式

INSERT INTO<表名>[（<属性名清单>）]

VALUES（<常量清单>）；

（2）实例：在Employee 表中插入一职工记录。

INSERT INTO Employee

VALUES（’2032’，’张杉’，’男’，28，’工程师’，’01’）；

2.UPDATE 语句

（1）语句格式

UPDATE<表名>

SET　　<列名>=<表达式>[,<列名>=<表达式>]

[WHERE<条件>];

（2）实例：在工资表中，将所有职工的基本工资都增加500。

UPDATE　　Salary

SET　　Basepay=Basepay+500;

3.DELETE 语句

（1）语句格式

DELETE　　FROM <表名>

[WHERE<条件>]

（2）实例：从职工表中删除Eno（职工号）为1003 的记录。

DELETE　　FROM　Employee

WHERE　　Eno='1003';

【习题演练】

1.SQL 语言的标准库函数 COUNT 、SUM 、AVG、MAX 和MIN 等，不允许出现在下列哪个子句中（ ）。

A.SELECT

B.HAVING

C.WHERE

D.GROUP，HAVING

1.【答案】C 。解析：SQL 语言的标准库函数 COUNT 、SUM 、AVG 、MAX 和MIN 等，不允许出现在WHERE 子句中。

2.在SQL 语句中，与X BETWEEN 20 AND 30 等价的表达式是（ ）。

A.X>=20 AND X<30

B.X>=20 AND X<=30

C.X>20 AND X<=30

D.X>20 AND X<30

2.【答案】B 。解析：BETWEEN A AND B 是指在 A-B 之间的范围，且包括A 和B 。故与X

BETWEEN 20 AND 30 等价的表达式是 X>=20 AND X<=30。

3.一个查询的结果成为另一个查询的条件，这种查询被称为（ ）。

A. 内查询

B.连接查询

C. 自查询

D.子查询

3.【答案】D。解析：一个查询的结果成为另一个查询的条件，这种查询被称为子查询，也叫嵌套查询。

4.在考试表中，要将58 分、59 分的分数调整到60 分，下列SQL 语句中能实现的是（ ）。

A.UPDATE 考试表 SET 分数 = 60 HAVING 分数>=58 AND 分数<60

B.UPDATE 考试表 SET 分数 = 60 WHERE 分数>=58 AND 分数<60

C.UPDATE 考试表 SET 分数 = 60 WHERE 分数 IN(58,59)

D.UPDATE 考试表 SET 分数 = 60 WHERE 分数 BETWEEN 58 AND 59

4.【答案】C 。解析：由题知至包含58 分和59 分，符合题意的只有C 选项。

知识点 27 、计算机网络分类

（ 一 ）按作用范围

1.局域网 （LAN）

局域网 （Local Area Network， LAN）是指在某一区域内由多台计算机互联成的计算机组。一般是方圆几千米以内。局域网可以实现文件管理、应用软件共享、打印机共享、工作组内的日程安排、电子邮件和传真通信服务等功能。局域网是封闭型的，

可以由办公室内的两台计算机组成，也可以由一个公司内的上千台计算机组成。

2.城域网（MAN）

城域网（Metropolitan Area Network）是在一个城市范围内所建立的计算机通信网，简称MAN。属于宽带局域网。由于采用具有有源交换元件的局域网技术，网中传输时延较小，它的传输媒介主要采用光缆，传输速率在100兆比特/秒以上。

3.广域网（WAN）

广域网（WAN，Wide Area Network）也称远程网（Long Haul Network）。通常跨接很大的物理范围，所覆盖的范围从几十公里到几千公里，它能连接多个城市或国家，或横跨几个洲并能提供远距离通信，形成国际性的远程网络。

（二）按拓扑结构

1.星型拓扑结构

星型拓扑结构是一种以中央结点为中心，把若干外围结点连接起来的辐射式互联结构，各结点与中央结点通过点与点方式连接，中央结点执行集中式通信控制策略，因此中央结点相当复杂，负担也重。

2.树型拓扑结构

树型拓扑结构是一种层次结构，结点按层次连接，信息交换主要在上下结点之间进行，相邻结点或同层结点之间一般不进行数据交换。

3.总线型拓扑结构

总线型拓扑结构是采用单根传输作为共用的传输介质，将网络中所有的计算机通过相应的硬件接口和电缆直接连接到这根共享的总线上。使用总线型拓扑结构需解决的是确保用户使用媒体发送数据时不能冲突。

在点到点的链路配置时，如果链路是半双工操作，只需使用简单的机制便可保证两个用户轮流工作。在一点到多点方式中，对线路的访问依靠控制

端的探询来确定。采取分布式访问控制策略来协调网络上计算机数据的发送。

4.环型拓扑结构

环型拓扑结构是使用公共电缆组成一个封闭的环，各结点直接连到环上，信息沿着环按一定方向从一个结点传送到另一个结点。环接口一般由发送器、接收器、控制器、线控制器和线接收器组成。

5.网状拓扑结构

网状拓扑结构，这种拓扑结构主要指各结点通过传输线相互连接起来，并且每一个结点至少与其他两个结点相连。网状拓扑结构具有较高的可靠性，但其结构复杂，实现起来费用较高，不易管理和维护，不常用于局域网。

（三）按使用范围

1.公用网

一般是国家的邮电部门建造的网络。“公用”的意思就是从所有愿意按邮电部门规定交纳费用的

人都可以使用。因此，公用网也可以称为公众网，例如CHINANET 、CERNET 等。

2.专用网

“专用网”是某个部门为本单位的特殊工作的需要而建立的网络。专用网指专用于一些保密性要求较高的部门的网络，比如企业内部专用网、军队专用网，尤其是涉及国家机密的部门。这种网络不向本单位以外的人提供服务。例如，军队、铁路、电力等系统均有本系统的专用网。

【习题演练】

1.局域网常用的基本拓扑结构有（ ）、环型和星型。

A.层次型

B.总线型

C.交换型

D.分组型

1.【答案】B。解析：局域网常用的拓扑结构有星型、环型、树型、总线型。网状型通常应用于广域网。

知识点 28 、网络地址

（一）物理地址

网卡物理地址存储器中存储单元对应实际地址称物理地址。

MAC（介质访问控制）地址是识别LAN（局域网）结点的标识。网卡的物理地址通常是由网卡生产厂家烧入网卡的EPROM（一种闪存芯片，通常可以通过程序擦写），它存储的是传输数据时真正赖以标识发出数据的电脑和接收数据的主机的地址。

物理地址一般记作 06-2E-14-F9-5A-23，48 比特的不同组合。

（二）IP 地址

是为网络每台计算机分配的唯一标识，为了使连入网络的众多计算机主机在通信时能够互相识别，网络中的每一台主机都有唯一的32 位地址。

IP 地址一般用小数点隔开的十进制数，即点分十进制表示，如202.120.70.23。

以点分十进制的形式表示IP 地址时，用小数点将IP 地址分为四部分，每个部分的范围为0～255。

每一类地址都由两个固定长度的字段组成，其中一个字段是网络号 net-id，它标志主机（或路由器）所连接到的网络，而另一个字段则是主机号 host-id，它标志该主机（或路由器）。

两级的IP 地址可以记为：IP 地址：：＝｛＜网络号>，<主机号>｝

（三）IP 地址分类汇总

IP 地址分类

类型	IP 范围	私有IP
A	1-126	10.0.0.1-10.255.255.255
B	128-191	172.16.0.1-172.31.255.255
C	192-223	192.168.0.1-192.168.255.255
D	224-239	
E	240-254	

（四）特殊的 IP 地址

主机号全为 1 表示本网段的内部广播地址，不能分给计算机使用。

主机号全为0 表示本网络的网段，计算机所在网络。

127 开头的本地回环地址，主要测试TCP/IP 协议正确性。

D 类地址为组播地址。主要是路由协议OSPF 等使用。

E 类地址用于研究使用。

169.254.*.* 的地址是当计算机自动获取IP 地址失败后的标识，但不能用于计算机之间的通信。

（五）IPv6 地址

IPv6 是InternetProtocoLVersion 6 的缩写，是下一代IP 地址，用于替代现行版本IP 协议（IPv4）的下一代IP 协议，目的是为解决IP 地址紧缺的问题。扩大地址的可用空间，满足计算机网路增长的需求。

IPv6 将地址32 位（IPv4）增大到了128 位二进制，使得地址数量变为原来的2 的96 次方倍，地址数量更多，适应未来网络扩展的需求。

【习题演练】

1. 以下IP 地址中，属于A 类地址的是（ ）。

A.52.213. 12.23

B.210.123.23. 12

C.223.123.213.23

D.156.123.32. 12

1.【答案】A。解析：A 类地址的第一个部分范围是1～126，符合的只有A 选项。

2.IPv4 地址和IPv6 地址的位数分别为（ ）。

A.4，6

B.8，16

C.16，24

D.32，128

2.【答案】D 。解析：IPv4 地址和Ipv6 地址的位数分别为32，128。

知识点 29 、子网掩码

（一）概念

子网掩码是一种用来指明一个IP 地址的网络位和主机位，知道了IP 地址的网络位，就可以确定

计算机IP 地址所在的网络，知道了主机位就可以确定该IP 地址所在网络的位置。

（二）功能

子网掩码不能单独存在，它必须结合IP 地址一起使用。

子网掩码是一个32 位地址，一般在书写的时候也是点分十进制表示。

在子网掩码中，1 表示网络位，0 表示主机位。

（三）默认子网掩码

A 类网络的子网掩码为255.0.0.0，B 类网络的子网掩码为255.255.0.0，C 类网络的子网掩码为255.255.255.0。

（四）求网络号

主机号全为0 的地址表示网络号。

第一步把IP 地址转化为二进制。

第二步把子网掩码转化为二进制。

第三步保留子网掩码中 1 所对应的部分，主机位用0 填充。

第四步转为点分十进制。

（五） 求主机号

第一步把IP 地址转化为二进制。

第二步把子网掩码转化为二进制。

第三步保留子网掩码中0 所对应的部分。

（六） 求广播号

主机号全为 1 的地址表示广播号。

第一步把IP 地址转化为二进制。

第二步把子网掩码转化为二进制。

第三步保留子网掩码中 1 所对应的部分，主机位用 1 填充。

第四步转为点分十进制。

（七） 子网划分计算

形成子网的数量计算：从主机位借位当网络位来使用。

借位数	子网个数	借位数	子网个数
1	$2^1=2$	4	$2^4=16$
2	$2^2=4$	5	$2^5=32$
3	$2^3=8$	6	$2^6=64$

【习题演练】

1. 以下关于子网掩码的作用，错误的是（ ）。

A.标识IP 地址的网络部分和主机部分

B.主机部分对应的子网掩码部分全为“0”

C. 网络部分对应的子网掩码部分全为“1”

D.通过子网掩码标识网络的类型

1.【答案】D。解析：子网掩码是一种用来指明一个IP地址的哪些位标识的是主机所在的子网，以及哪些位标识的是主机位。其中主机部分对应的子网掩码部分全为0，网络部分全为1。

2.一个子网掩码是255.255.240.0，这个子网能拥有的最大主机数是（ ）。

A.240

B.255

C.4094

D.65534

2.【答案】C。解析：255.255.240.0，将子网掩码中的240转换为二进制，240=11110000，所以进行子网划分的时候借走了4位主机位充当网络位，能够形成2^4=16个子网，每个子网中的IP数量2^12=4096，但是能够分配给计算机使用的只有4096-2=4094。

知识点 30、网络设备

（一）局域网设备

1. 中继器

中继器（Repeater）是局域网互连的最简单设备，工作在OSI 体系结构的物理层，它接收并识别网络信号，然后再生信号并将其发送到网络的其他分支上。

信号在传输线路上传输的时候，受到距离、噪音、电阻等影响，使得信号传输的距离不能是长到无限远，在传输一定距离后，信号需要加强。

如果在线路中间插入放大器，则伴随信号的放大，噪音也被放大了。中继器的功能是对接收信号进行再生和发送。

2.集线器

集线器属于物理层设备。

集线器是有多个端口的中继器，简称HUB。

集线器在创建网络时，一般是以星型或树型拓扑结构为主。

集线器没有MAC 地址表，采用广播方式发送，当网络中的主机数量过多的时候，就会在网络生成广播风暴。以集线器为主要网络设备的网络被称为共享式以太网。

3. 网卡

网络接口卡NIC（Network Interface Card）又称网卡或网络适配器。

网卡是工作在链路层的网络组件，是局域网中连接计算机和传输介质的接口，不仅能实现与局域网传输介质之间的物理连接和电信号匹配，还涉及帧的发送与接收、帧的封装与拆封、介质访问控制、数据的编码与解码以及数据缓存的功能等。

4. 网桥

在数据链路层扩展局域网是使用网桥。

网桥工作在数据链路层，它根据MAC 帧的目的地址对收到的帧进行转发。

网桥不隔绝广播风暴。

5.交换机

以太网交换机的每个接口都直接与主机相连，并且一般都工作在全双工方式。

交换机在同一时刻可进行多个端口对之间的数据传输。每一端口都可视为独立的网段，连接在其上的网络设备独自享有全部的带宽，无须同其他设备竞争使用。

（二）广域网设备

1.网关

网关（Gateway）又称网间连接器、协议转换器。网关在网络层以上实现网络互连，是最复杂的网络互连设备，仅用于两个高层协议不同的网络互连。

2.路由器

路由器（Router），是连接因特网中各局域网、广域网的设备，它会根据信道的情况自动选择和设定路由，以最佳路径，按前后顺序发送信号。

3.调制解调器

调制解调器（modem）是一种计算机硬件，它能把计算机的数字信号翻译成可沿普通电话线传送的模拟信号，是家用电话拨号上网的必不可上少的设备。

调制：将数字信号转化为模拟信号。

解调：将模拟信号转化为数字信号。

【习题演练】

1.集线器是工作在（　）的设备。

A.物理层

B.链路层

C.网络层

D.运输层

1.【答案】A。解析：中继器和集线器就是物理层的设备。

2.在下列网间连接器中，（ ）在数据链路层实现网络互连。

A. 中继器

B. 网桥

C. 路由器

D. 网关

2.【答案】B。解析：网桥是数据链路层的设备，中继器是物理层设备，路由器是网络层设备，网关在传输层及其以上实现网络互联。

知识点 31 、信息加密技术

（一）信息加密

信息加密是指使有用的信息变为看上去似为无用的乱码，使攻击者无法读懂信息的内容从而保护信息。

任何一个加密系统至少由下面四个部分组成：

1.未加密的信息，也称明文。

2.加密后的信息，也称密文。

3.加密解密设备或算法。

4.加密解密的密钥。

发送方用加密密钥，通过加密设备或算法，将信息加密后发送出去。接收方在收到密文后，用解密密钥将密文解密，恢复为明文。如果传输中有人窃取，他只能得到无法理解的密文，从而对信息起到保密作用。

其中，密钥是唯一能够控制明文与密文之间变换的关键，通常是一随机字符串，在计算机上实现的数据加密算法，其加密或解密变换是由一个密钥来控制的。

（二）对称加密

1.概念

对称密码体制是指如果一个加密系统的加密密钥和解密密钥相同，或者虽然不相同，但是由其中的任意一个可以很容易地推导出另一个，即密钥是双方共享的，则该系统所采用的就是对称密码体制。

2.加密过程

3.对称加密算法特点

对称加密算法的特点是算法公开、计算量小、加密速度快、加密效率高。

不足之处是，交易双方都使用同样钥匙，安全性得不到保证。每对用户每次使用对称加密算法时，都需要使用其他人不知道的唯一钥匙，这会使得发收信双方所拥有的钥匙数量呈几何级数增长，密钥管理成为用户的负担。对称加密算法在分布式网络系统上使用较为困难，主要是因为密钥管理困难，使用成本较高。

常见的对称加密算法：DES 算法，3DES 算法，TDEA 算法，Blowfish 算法，RC5 算法，IDEA 算法。

（三）非对称密钥加密

1.概念

公钥密码体制指一个加密系统的加密密钥和解密密钥是不一样的，或者说不能由一个推导出另一

个。其中一个称为公钥用于加密，是公开的，另一个称为私钥用于解密，是保密的。

公钥密码体制解决了密钥的管理和发布问题，每个用户都可以把自己的公开密钥进行公开，如发布到一个公钥数据库中。

2.工作原理

A 要给B 发送信息时，A 用B 的公钥加密信息，因为A 知道B 的公钥。

A 将这个消息发给B。

B 收到这个消息后，B 用自己的私钥解密A 的消息。其他所有收到这个报文的人都无法解密，因为只有B 才有B 的私钥。

3.RSA 算法

RSA 是Rivest、Shamire 和Adleman 于 1978 年在美国麻省理工学院研制出来的，它是一种比较典型的公开密钥加密算法。

【习题演练】

1.在数据加密技术中，将待加密的报文称为（ ）。

A.密文

B.正文

C.短文

D.明文

1.【答案】D。解析：待加密的报文称为明文。

2.在公钥密码体制中，公开的是（ ）。

A.公钥和私钥

B.公钥和算法

C.明文和密文

D.加密密钥和解密密钥

2.【答案】B。解析：公钥加密算法是计算机网络中经常使用的算法，能跟好的保证网络安全。其中有两个秘钥，公钥和算法是公开的，可以在服务器上查找到，这样方便用于加密。私钥是用户自己保存的一种密钥。

3. 以下关于对称与非对称加密算法叙述中正确的是（ ）。

A.对称加密比非对称加密的安全性好

B.对称加密比非对称加密的解密速度慢

C.非对称加密算法中公钥是公开的，算法是保密的

D.非对称加密与对称加密相比，用户需要保管的密钥数量少

3.【答案】D。解析：非对称加密使用了一对密钥，即公钥与私钥，公钥是公开的，私钥个人持有，不公开，故D 选择正确。

知识点 32 、计算机病毒

（ 一 ）计算机病毒的概念

广义上，能够引起计算机故障、破坏计算机数据的程序都可称为计算机病毒；

狭义上，计算机病毒是指编制者在计算机程序中插入的破坏计算机或者毁坏数据，影响计算机的使用，并能自我复制的一组计算机指令或者程序代码。

（ 二 ）计算机病毒的特性

目前，计算机病毒有数十万种，各有其不同的特征，但总的说来，计算机病毒又有明显的共性。计算机病毒主要有以下几种特征：

1.传染性

传染性是计算机病毒的基本特征。计算机病毒能通过自我复制来感染正常的文件，达到破坏计算

机系统正常运行的目的。但传染性是有条件的，只有病毒程序被执行之后才具有传染性，才能感染其他文件。

2.破坏性

任何计算机病毒只要侵入计算机系统，都会对系统及应用程序产生不同程度的影响和破坏，轻则降低计算机的工作效率，占用系统资源，重则破坏数据，删除文件，甚至导致系统崩溃，给用户造成不可挽回的损失。

3.寄生性

虽然计算机病毒是一种程序，但这种程序通常不是以独立文件的形式存在的，它寄生在合法的程序之中。这些合法的程序可以是系统引导程序、可执行程序、一般应用程序等。现在的某些病毒本身就是一个完整的程序，如广义病毒中的网络蠕虫。

4.欺骗性

黑客常常会把带有病毒程序的名字起成一些用户比较关心的程序名字。

5.隐蔽性和潜伏性

计算机病毒要有效的传染和传播，就应该尽量在用户能够觉察的范围之外进行。

6.衍生性

既然计算机病毒是一段特殊的程序，了解病毒程序的人就可以根据其个人意图随意改动，从而衍生出另一种不同于原版病毒的新病毒。

7.可触发性

编制计算机病毒的人，一般都为病毒程序设定了一些触发条件，例如系统时钟的某个时间或日期、系统运行了某些程序等。一旦条件满足，计算机病毒就会“发作”，使系统遭到破坏。

（三）计算机病毒的传播方式

计算机病毒的传播方式主要有：

（1）移动存储设备传播；

（2）计算机网络传播；

（3）电子邮件传播。

【习题演练】

1.计算机病毒是可以造成计算机故障的（ ）。

A.一种微生物

B.一种特殊的程序

C.一块特殊芯片

D.一个程序逻辑错误

1.【答案】B。解析：计算机病毒指人为编制的一段具有破坏性的，能够自我复制的程序或者代码。

2.计算机病毒具有破坏性、（ ）、潜伏性和传染性等特点。

A.必然性

B.再生性

C.隐蔽性

D.易读性

2.【答案】C。解析：计算机病毒具有寄生性、传染性、潜伏性、隐蔽性。

3.根据统计，当前计算机病毒扩散最快的途径是（ ）。

A.软件复制

B.网络传播

C.磁盘拷贝

D.运行游戏软件

3.【答案】B。解析：计算机病毒扩散最快的是网络传播，需要提前安装使用杀毒软件来做好防范。