**操作系统**

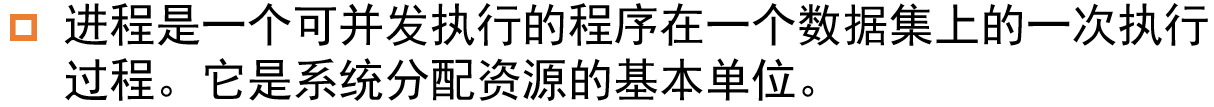
1. **概念回顾**

**操作系统的五大功能:**

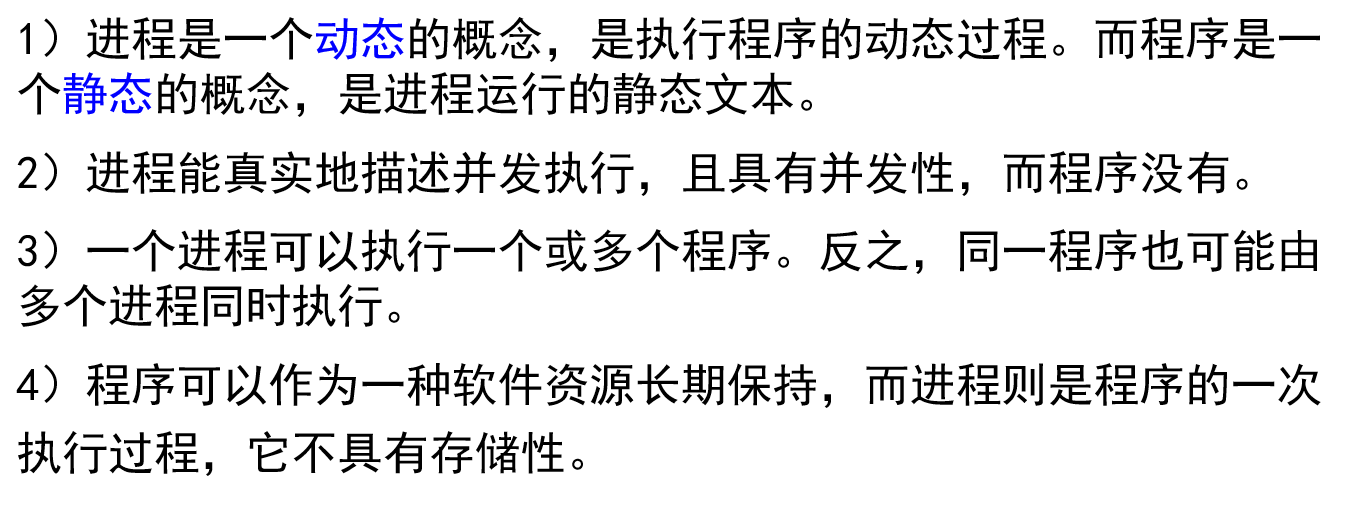
1. **进程管理（处理器管理）**
2. **存储管理**
3. **设备管理**
4. **文件管理**
5. **用户接口**

**1.1 处理器管理**

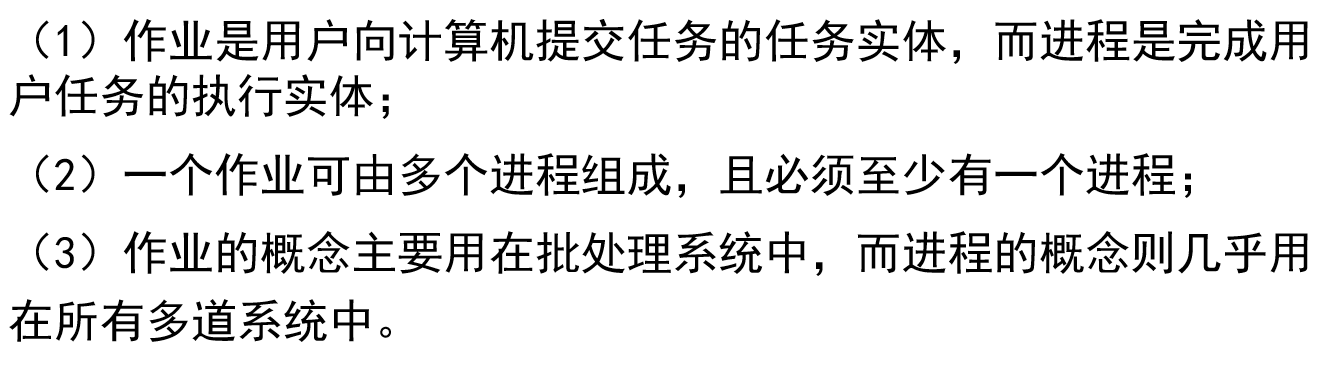
1. **程序、作业、进程**

****

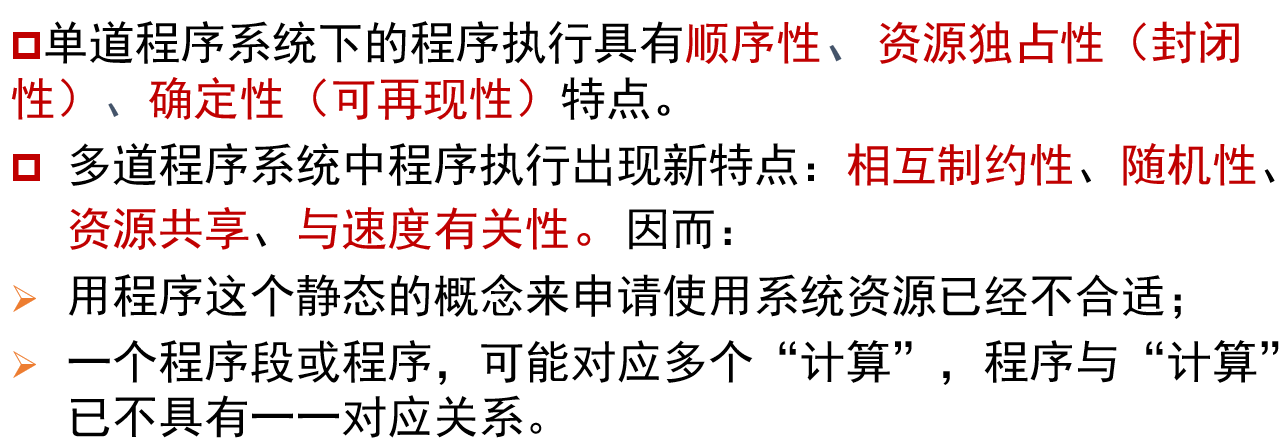
**进程与程序的区别**

****

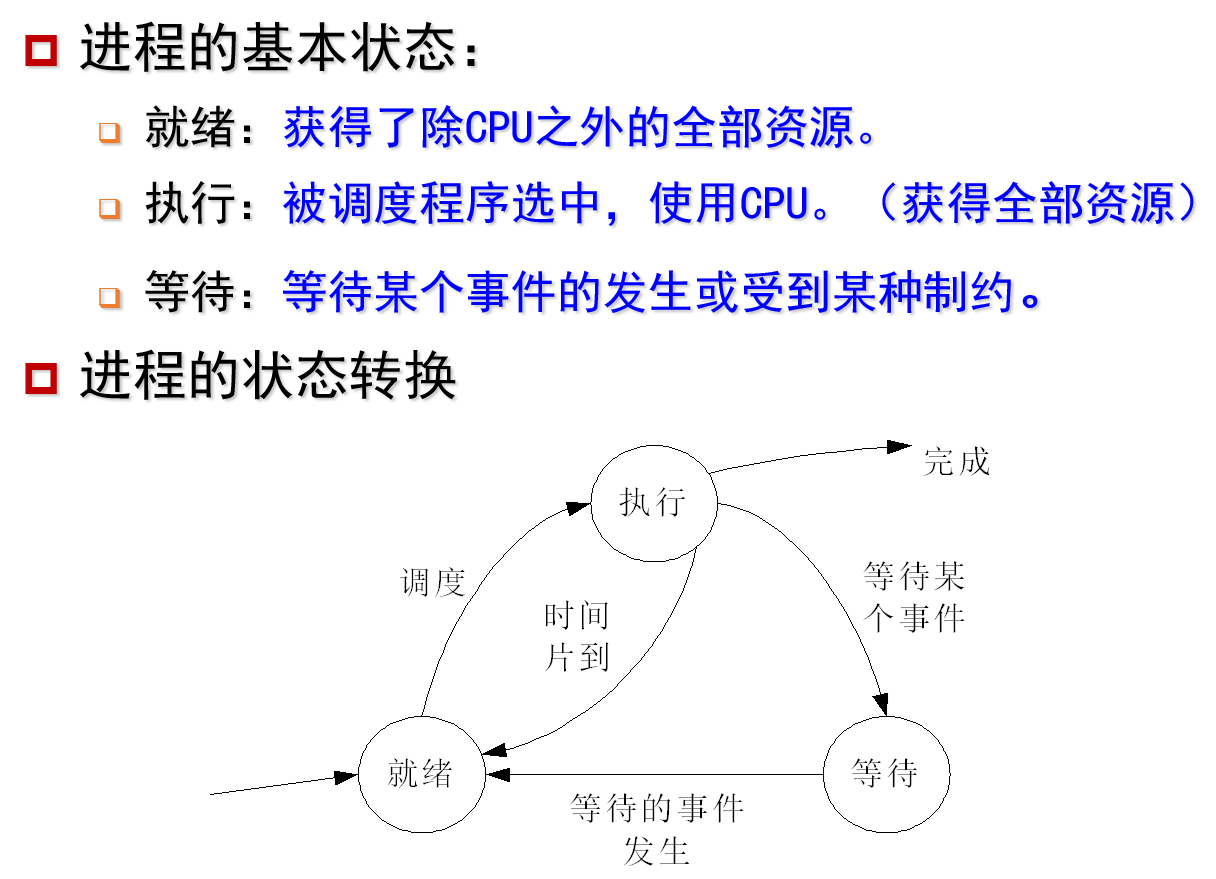
**作业与进程的关系**

****

1. **多道程序运行环境（在内存中允许同事存放若干道相互独立的程序，当一道程序的输入、输出由CPU交给通道执行后，CPU则去运行其他程序，因此各道程序“交替执行”，共享各种软硬件资源）**

****

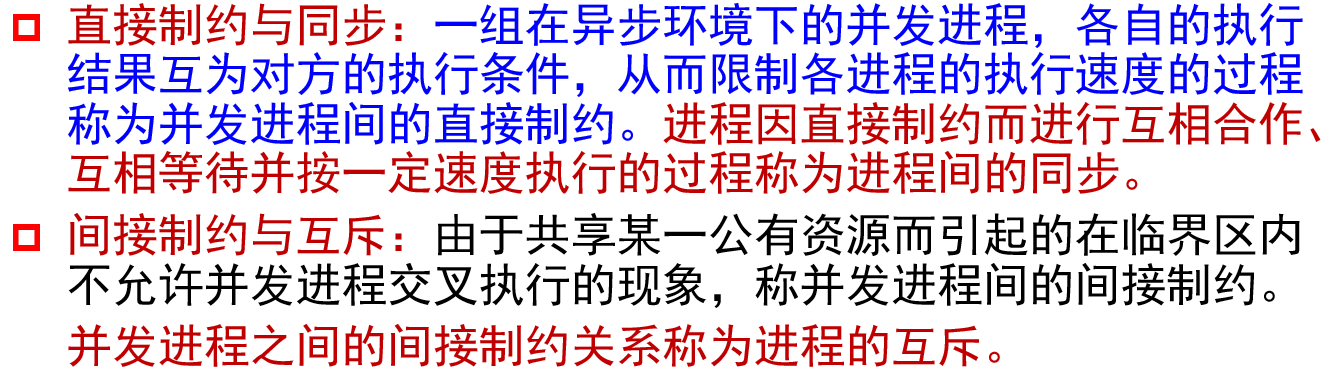
1. **进程的三种基本状态：等待、运行、就绪，三种状态之间的转换条件**

****

1. **进程的互斥与同步：**

**临界资源：一次只允许一个进程使用的资源称为临界资源**

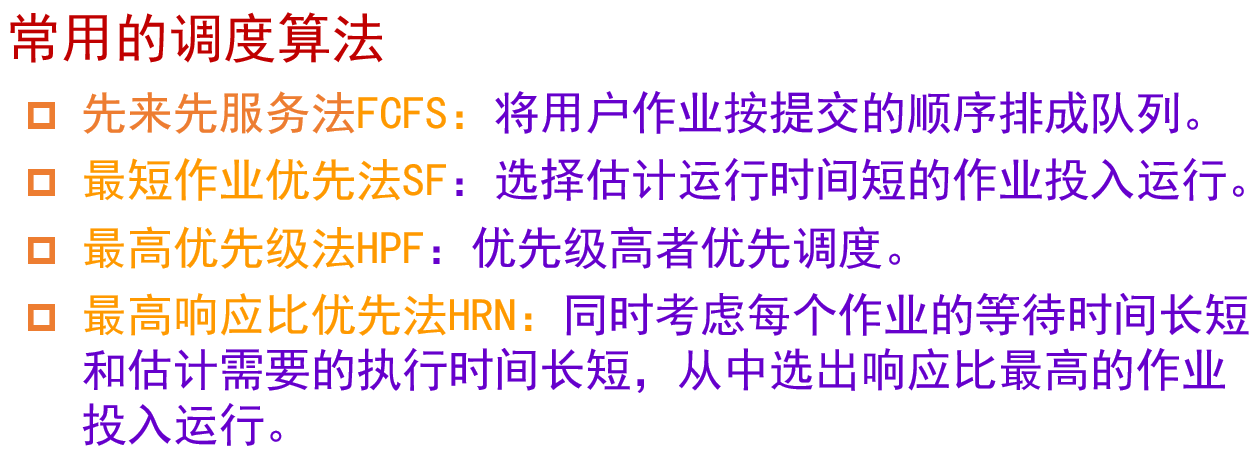
**临界区：在进程中访问临界资源的那段代码称为临界区**

****

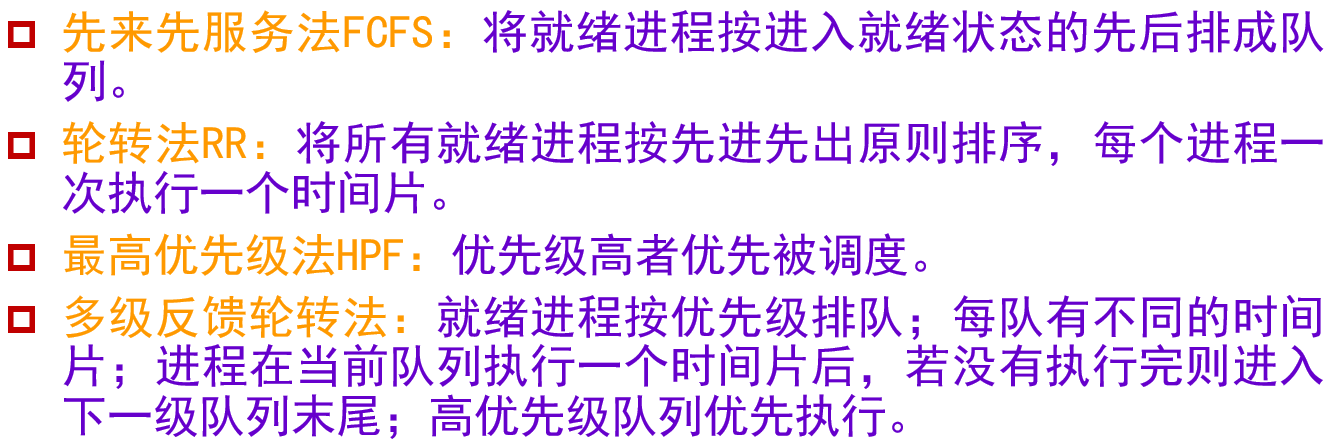
1. **处理器调度：**

**高级调度（作业调度）、中级调度（交换调度）、低级调度（进程调度）**

**高级调度常用调度算法**

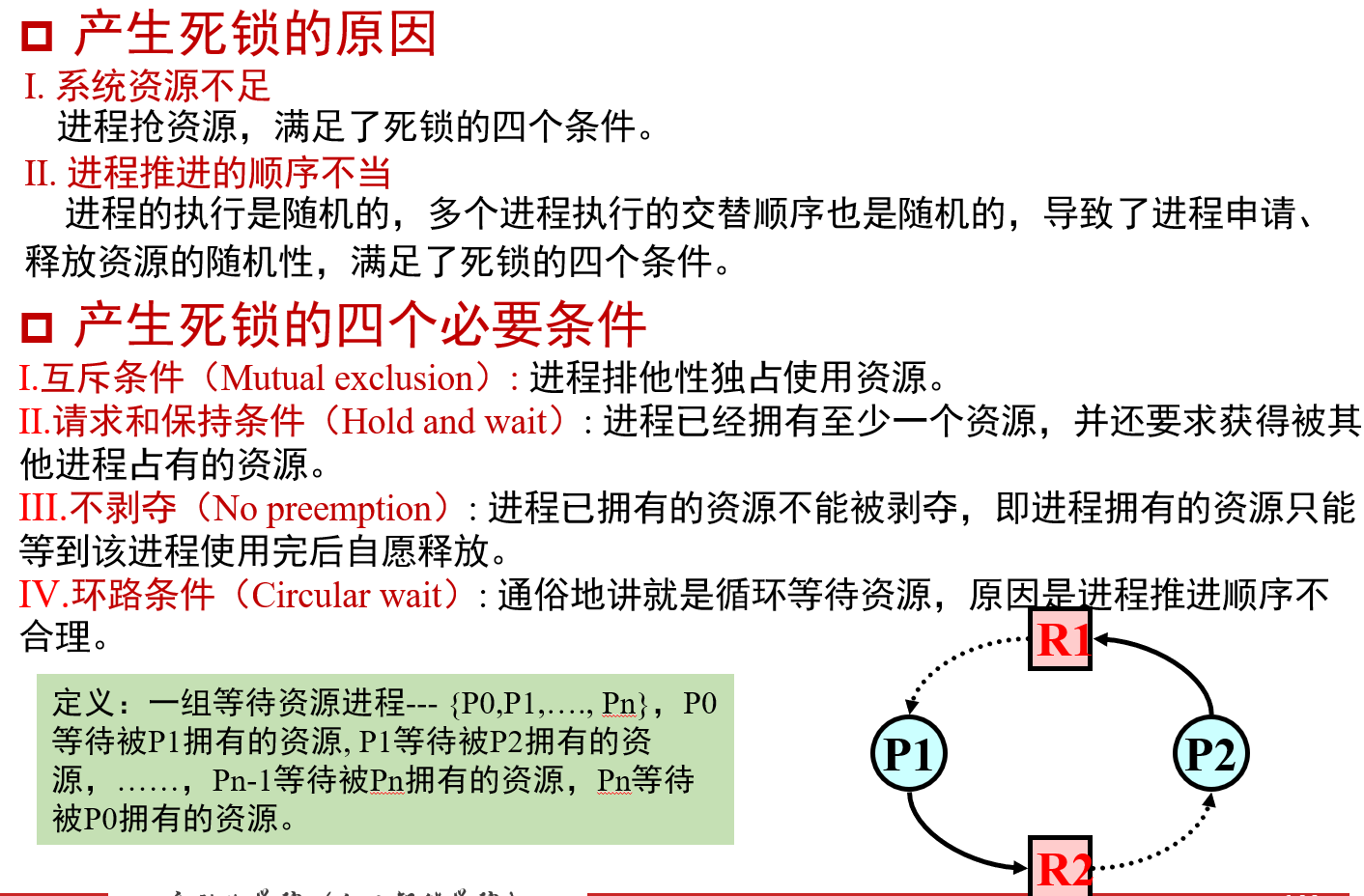
****

**低级调度常用调度算法**

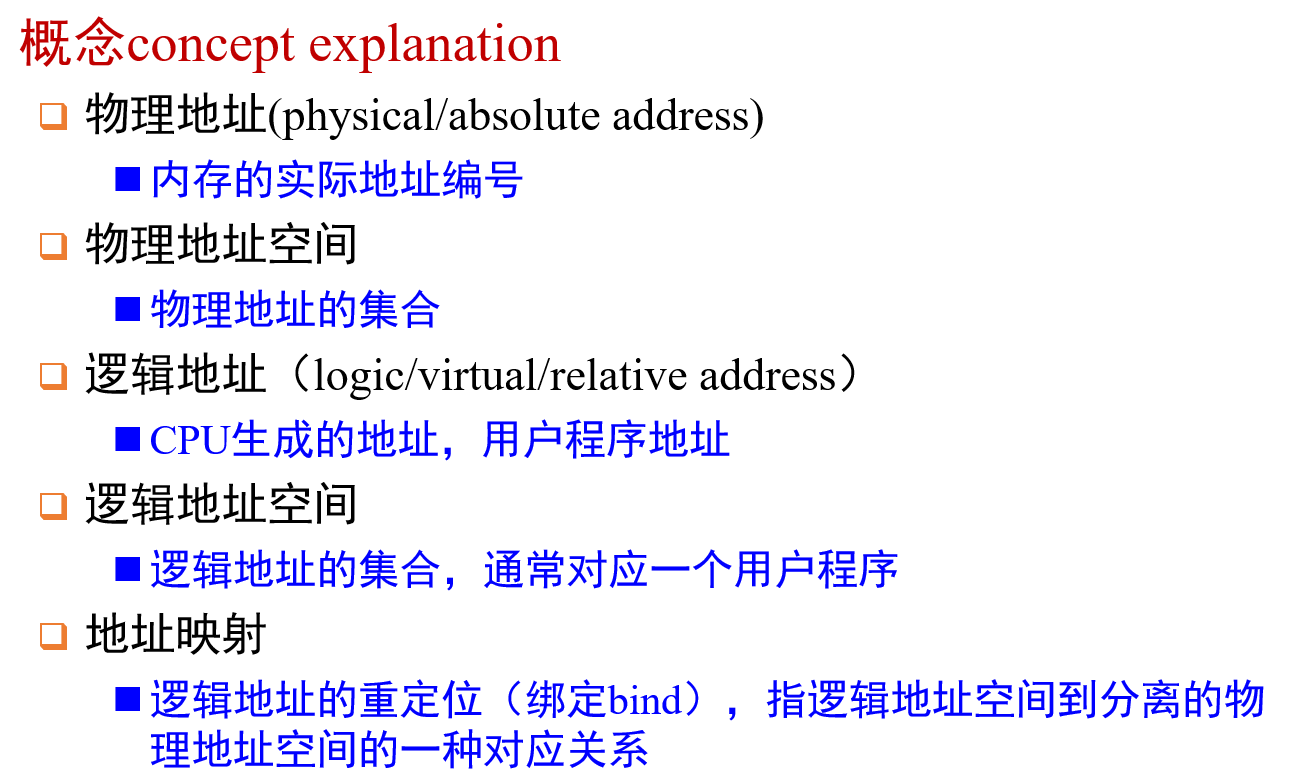
****

1. **死锁**

**死锁产生的原因、必要条件、排除方法**

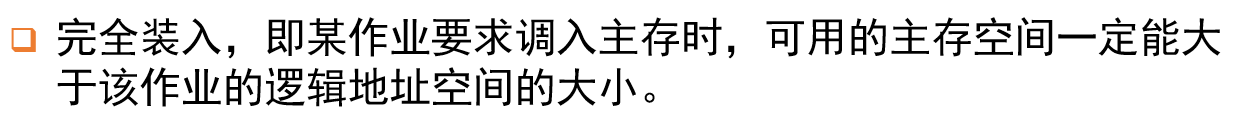
****

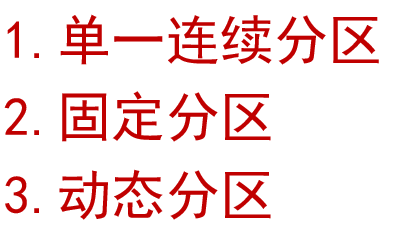
**1.2 存储管理**

****

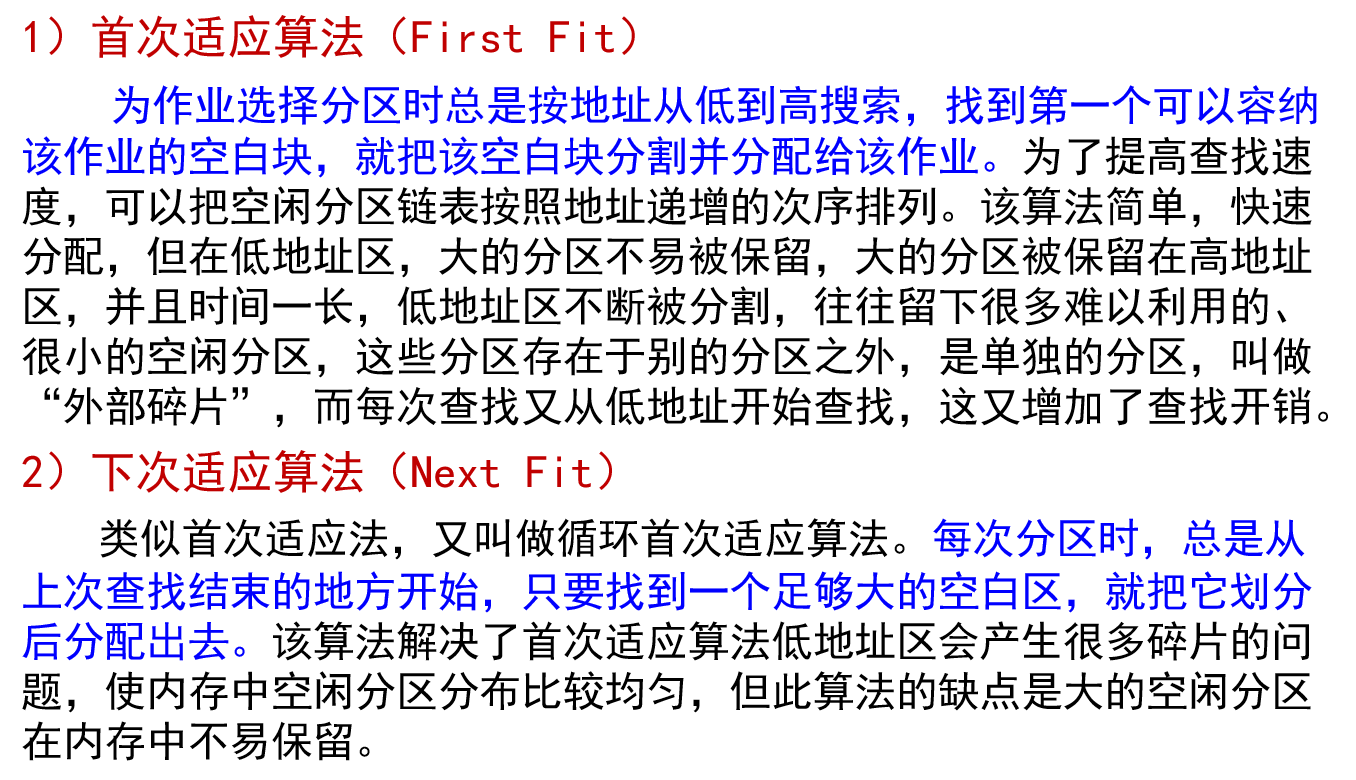
**内部碎片、外部碎片**

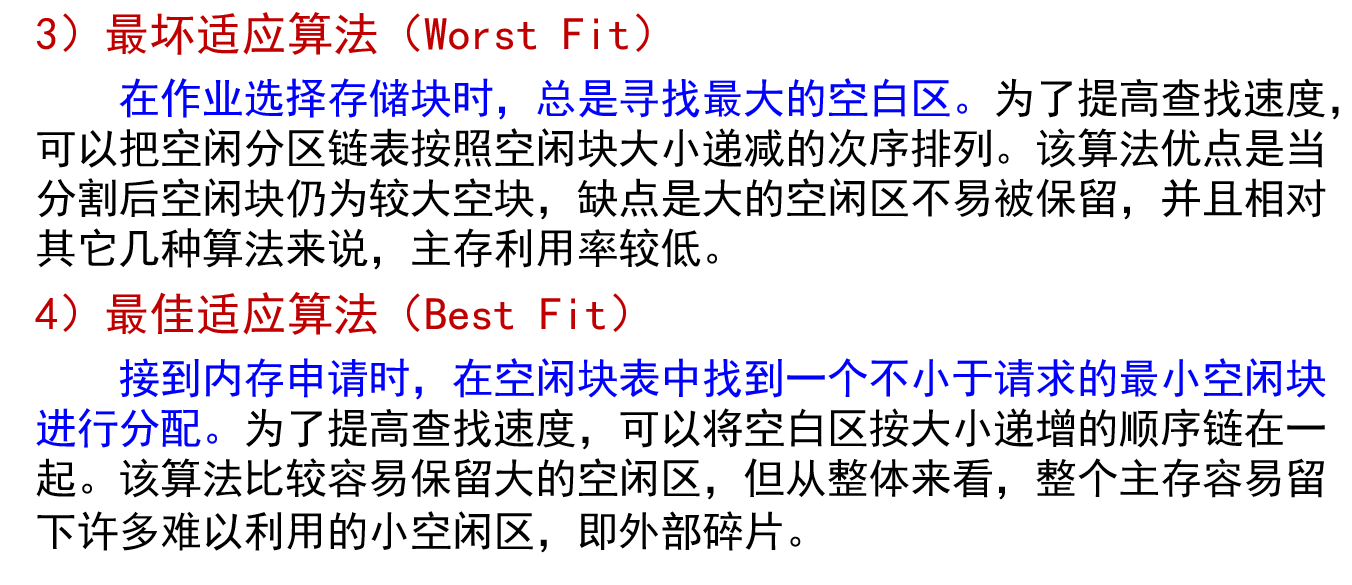
1. **实存储管理**

****

****

1. **主存空闲分区分配算法**

****

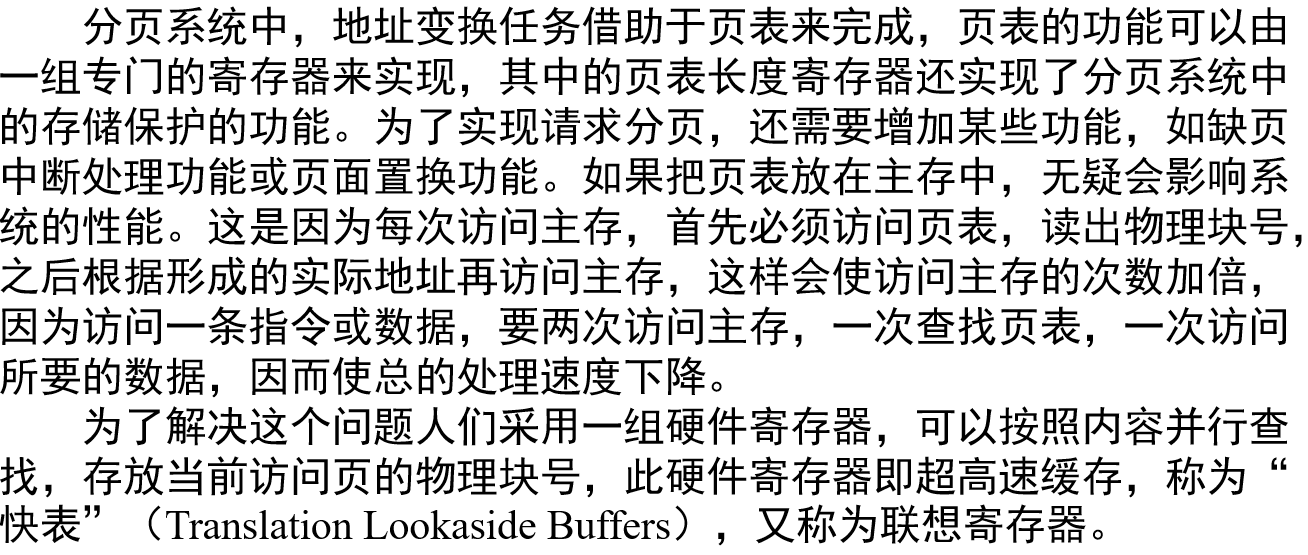
****

**动态分区便于动态申请内存、共享内存和动态链接，但缺点是有碎片问题(外部碎片)。**

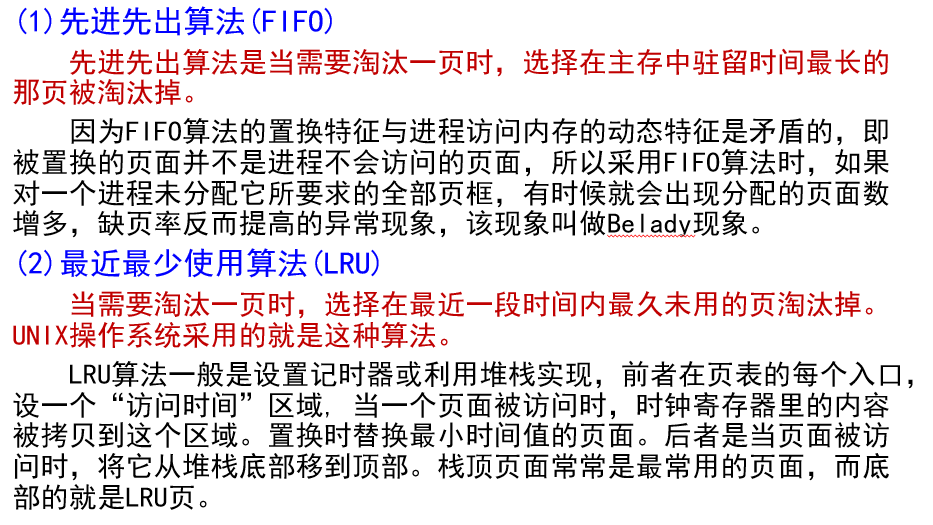
1. **虚拟存储管理**

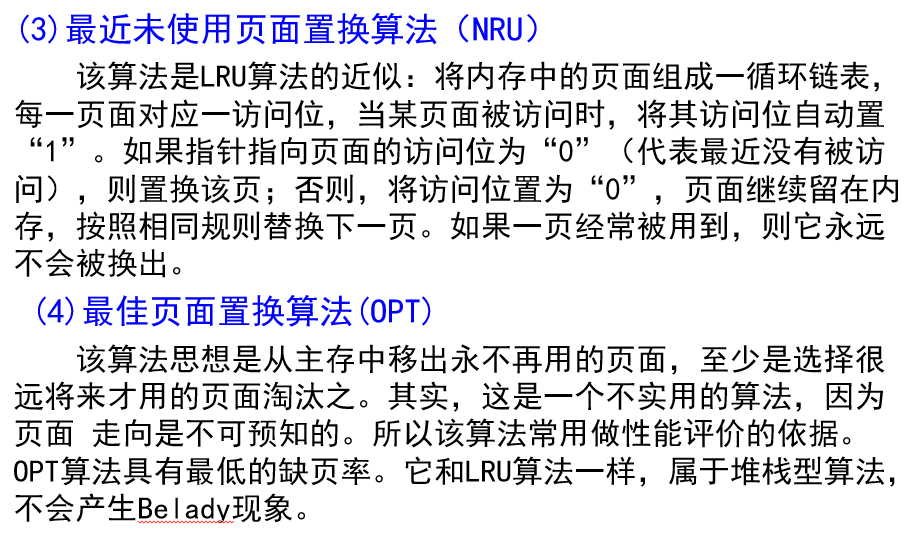
**1）请求分页管理**

**地址变换机构：页表，采用快表（即联想寄存器）进行加速**

****

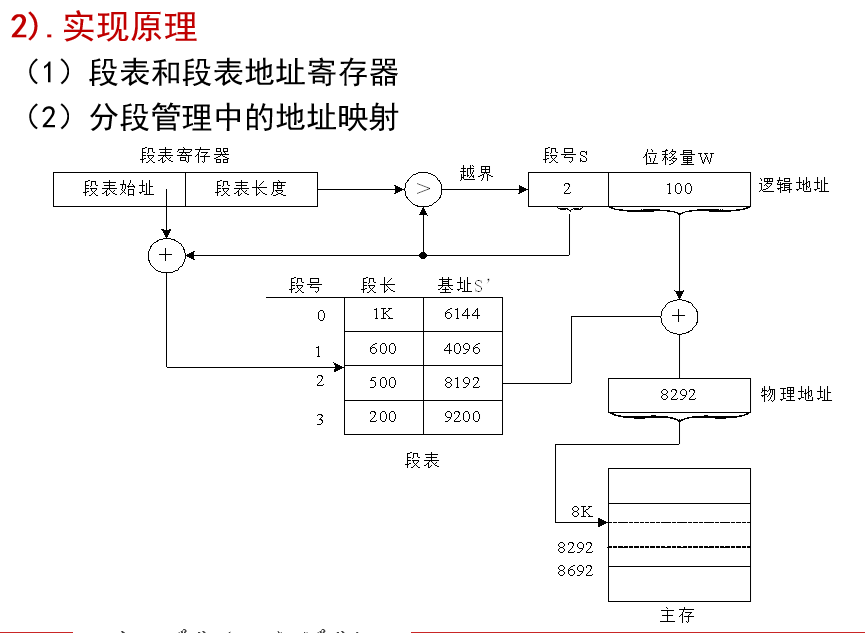
**页面置换算法：FIFO：Belady、LRU：UNIX、NRU、OPT**

****

****

**2）请求分段管理**

**实现原理**

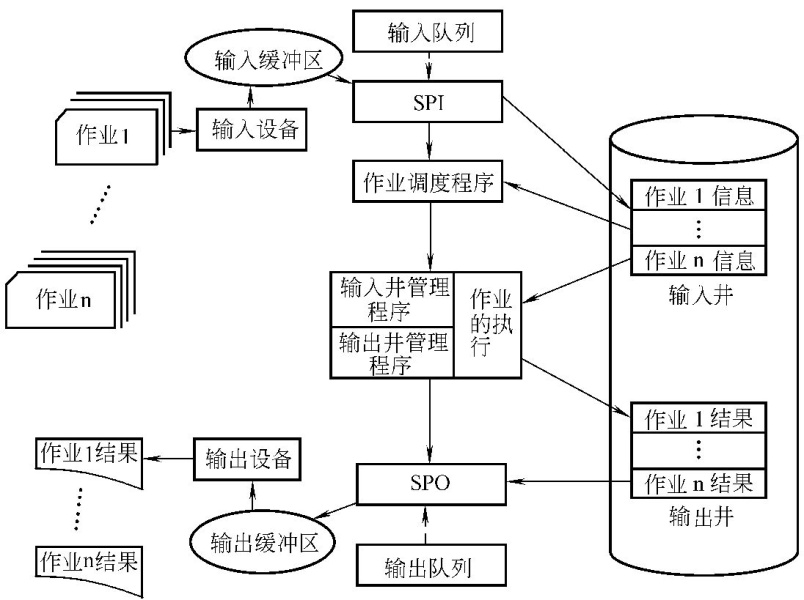
****

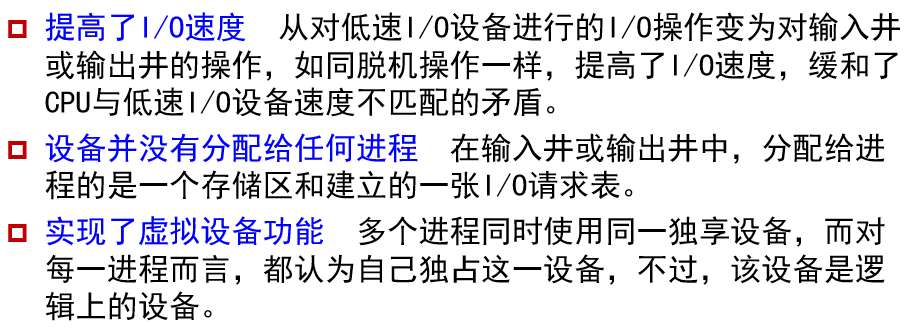
**1.3 设备管理**

1. **设备I/O请求的检测与控制：**
2. **循环测试I/O方式：原理**
3. **中断驱动方式：原理、优缺点**
4. **DMA：原理、与中断驱动方式的不同之处**
5. **I/O通道：原理**
6. **缓冲技术**
7. **设备管理程序：设备独立性/设备无关性**

**物理设备、逻辑设备**

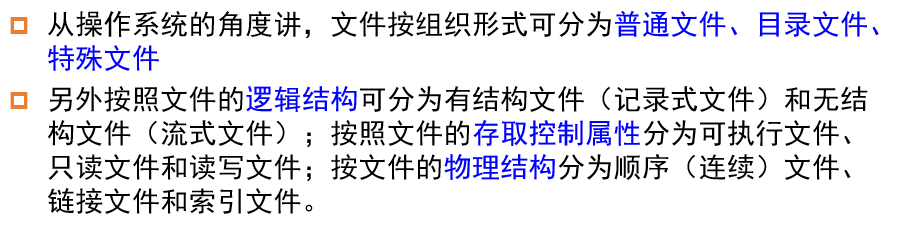
1. **虚拟设备技术**
2. **SPOOLing技术（假脱机技术）：使用辅助存储器作为缓冲存储器**
3. **系统组成、特点**

****

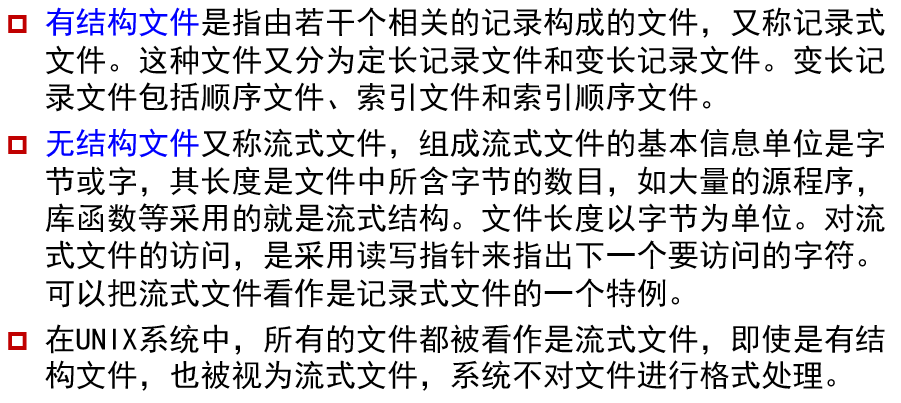
****

**1.4 文件管理**

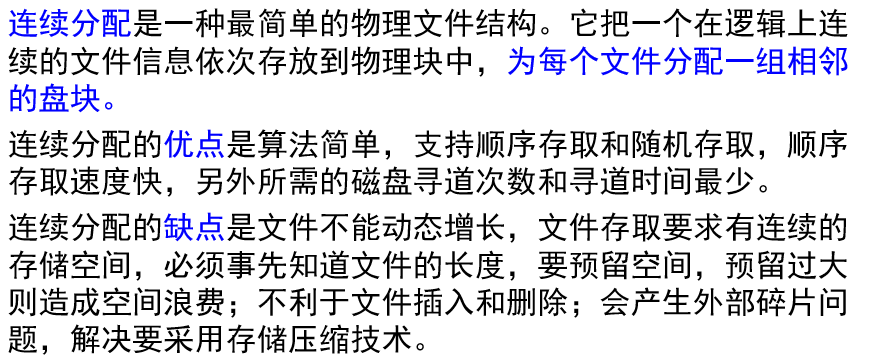
1. **文件分类**

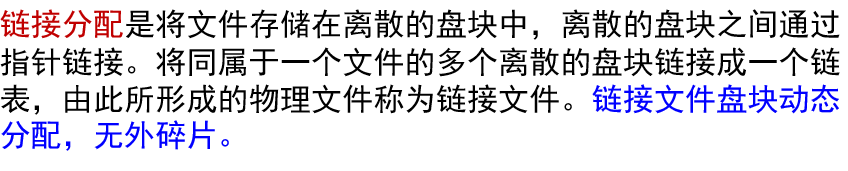
****

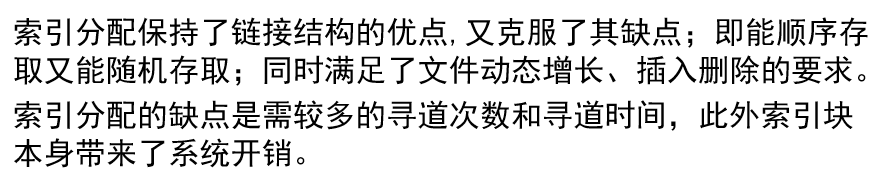
1. **文件存取方式：顺序、随机、按键**
2. **文件的逻辑结构：有结构文件/记录式文件、无结构文件/流式文件（UNIX系统）**

****

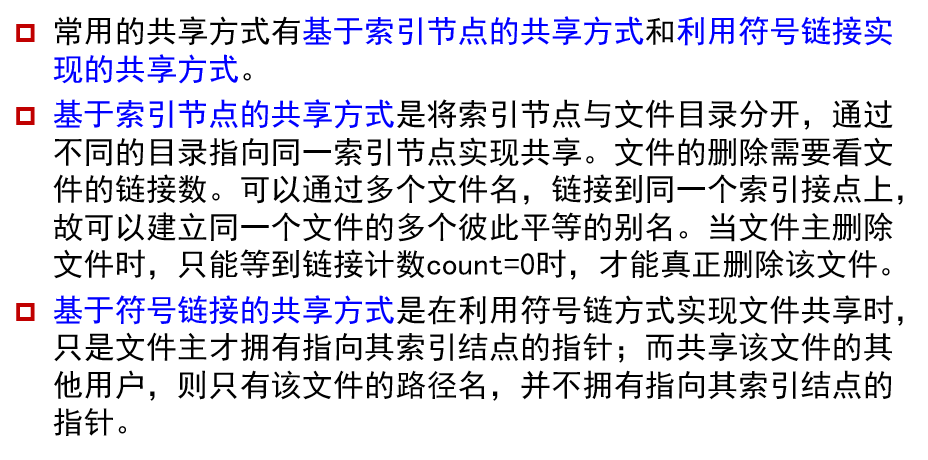
1. **文件的物理结构：连续分配、链接分配、索引分配（各种方法的原理和优缺点）**

****

****

****

1. **文件目录（了解概念）：文件控制块、文件的目录结构及特点**
2. **文件的共享**

****

**2、考查题型**

**选择、填空、简答**

**数据结构**

**1、概念**

**1.1数据结构三要素：数据的逻辑结构、数据的存储结构、数据的操作或运算**

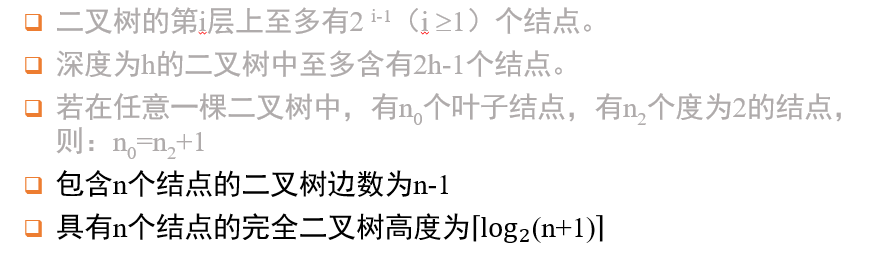
* 1. **数据的逻辑结构：集合、线性、树形、图状**
  2. **数据的存储结构：顺序、链式、索引、散列**
  3. **三者之间的关系**

**1.2线性结构**

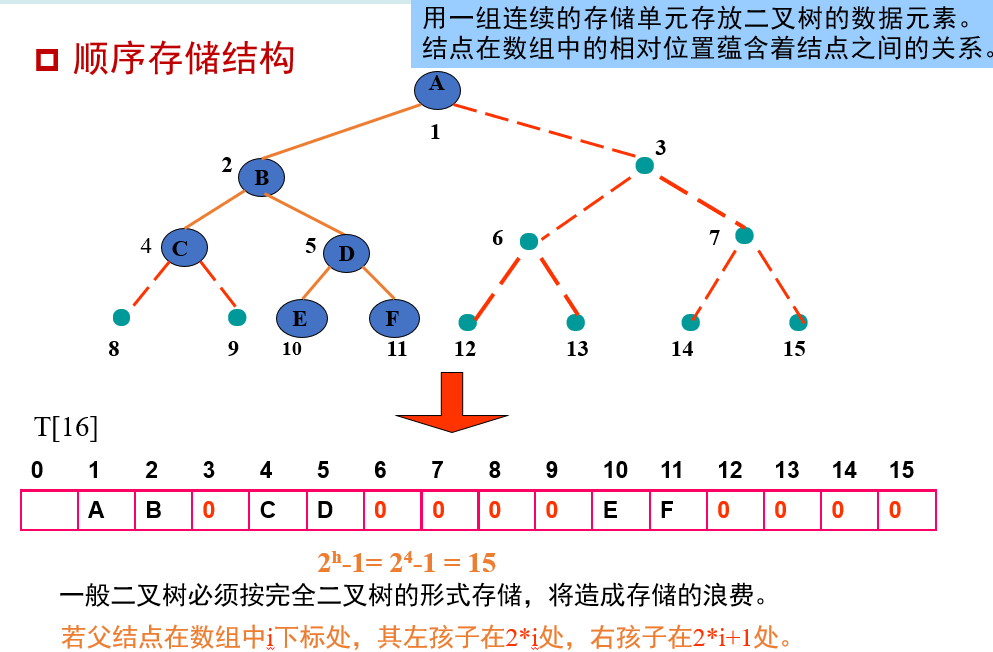
1. **线性表：顺序表、链表**
2. **顺序表的存储特点**
3. **链表的存储特点**
4. **链表结构体的定义**
5. **链表节点的定义、删除、插入的C语言实现**
6. **单循环链表节点的定义、删除、插入的C语言实现**
7. **双向链表（循环链表）的优点，双向链表节点定义、删除的C语言实现**
8. **栈：顺序栈、链栈**
9. **栈的特点：FILO**
10. **栈结构体的定义**
11. **队列：顺序存储队列（一般顺序队列、循环队列）、链队列**
12. **队列的特点：FIFO**
13. **队列结构体的定义**
14. **输入受限的双端队列**

**1.3 树结构**

1. **树的基本术语**
2. **二叉树的性质**

****

1. **二叉树的存储方式**
2. **顺序存储（适用于什么情况下的二叉树？）**

****

1. **链式存储**

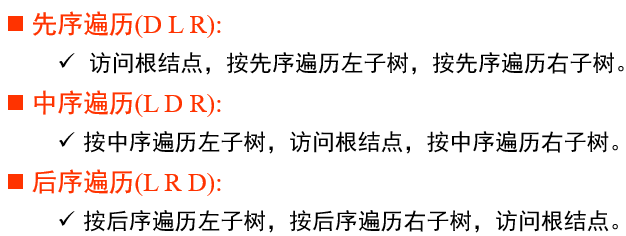
**链式存储结构的描述：**

**struct bitree {**

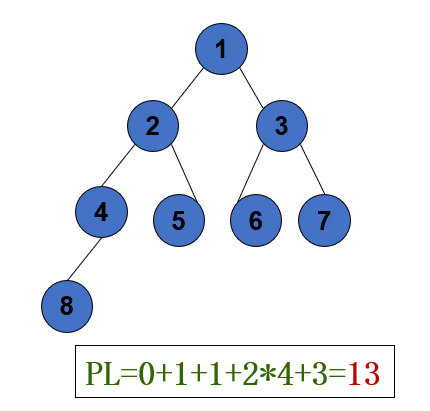
**elemtype data;**

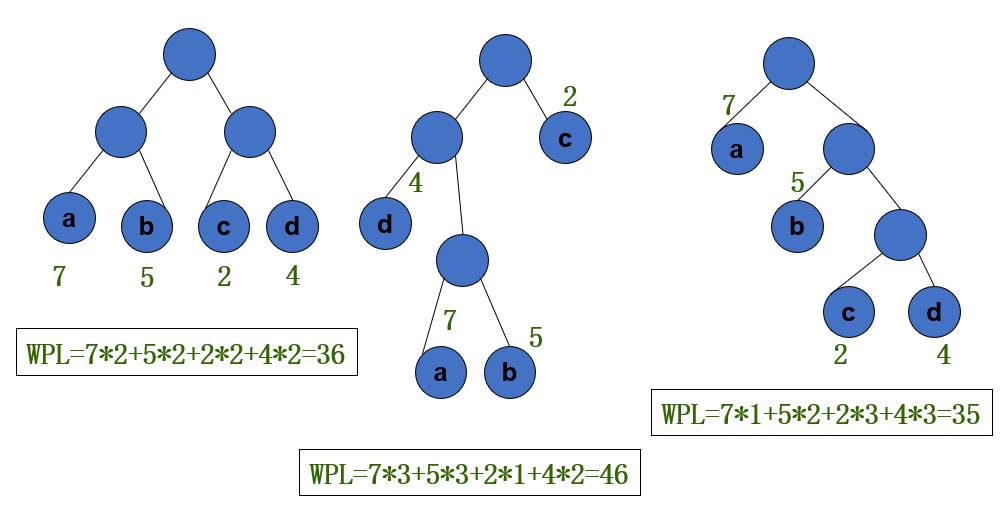
**bitree \*lchild, \*rchild;};**

1. **二叉树的遍历**

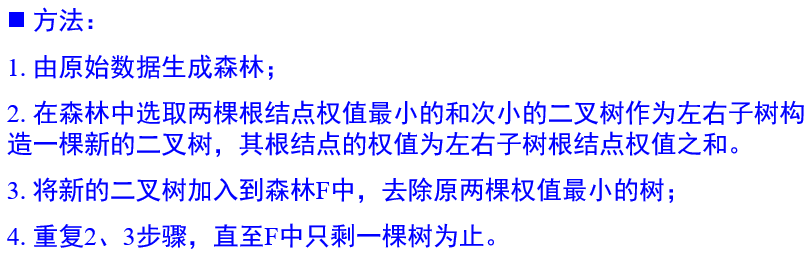
****

1. **二叉树的恢复**
2. **哈夫曼树：树的带权路径长度最小的二叉树**
   1. **相关名词：树的路径长度（PL）、树的带权路径长度（会计算WPL）**

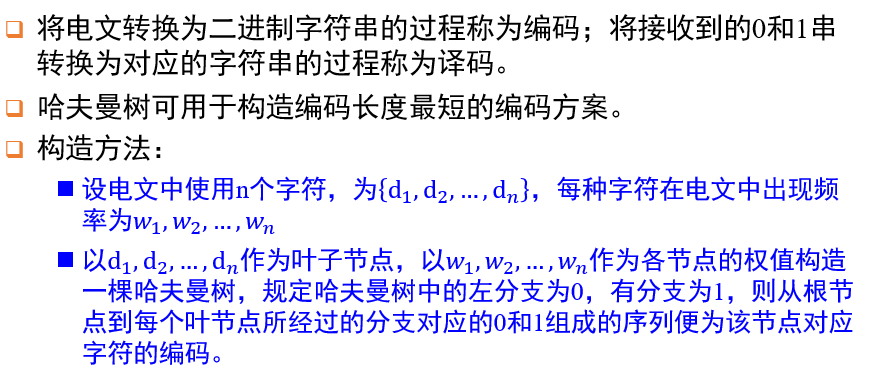
****

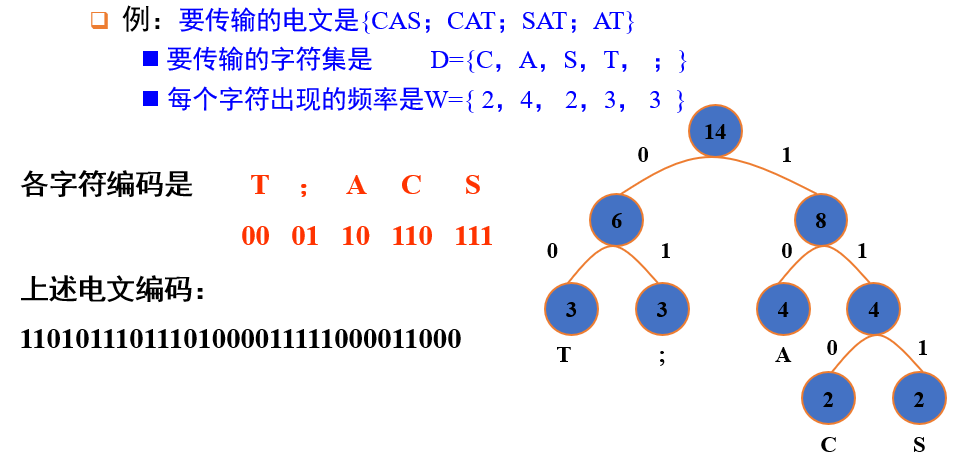
****

* 1. **哈夫曼树（Huffman Tree）的构造：**

****

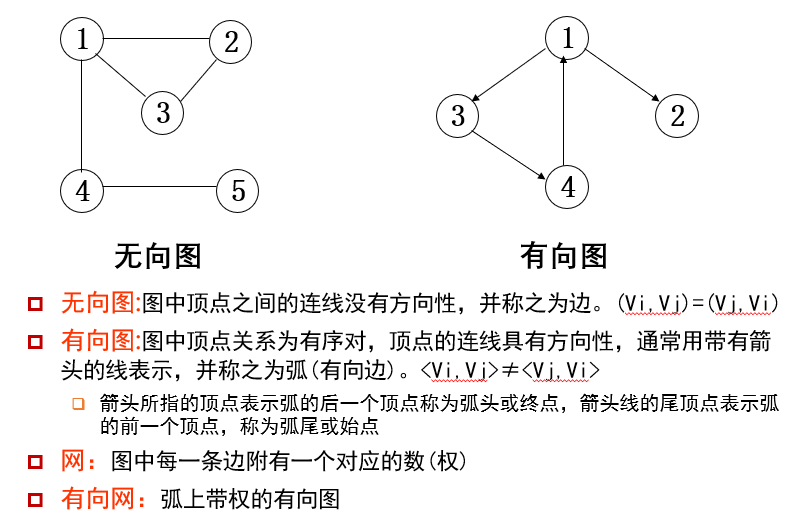
* 1. **哈夫曼编码（Huffman Coding）**

****

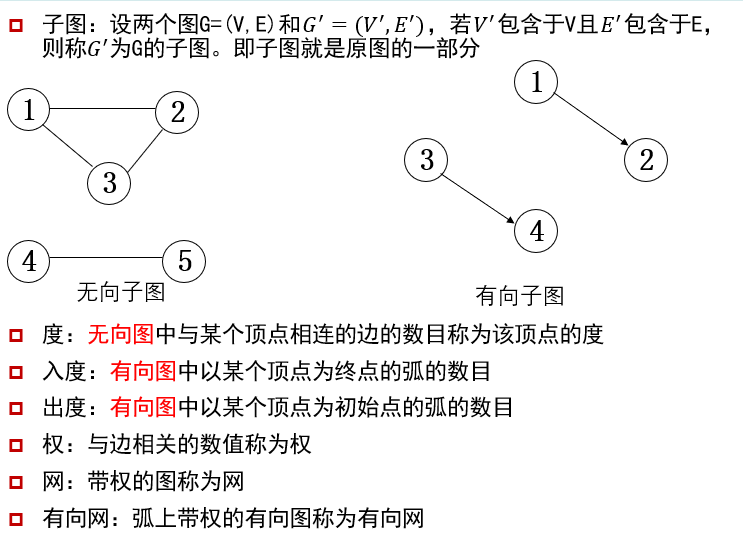
****

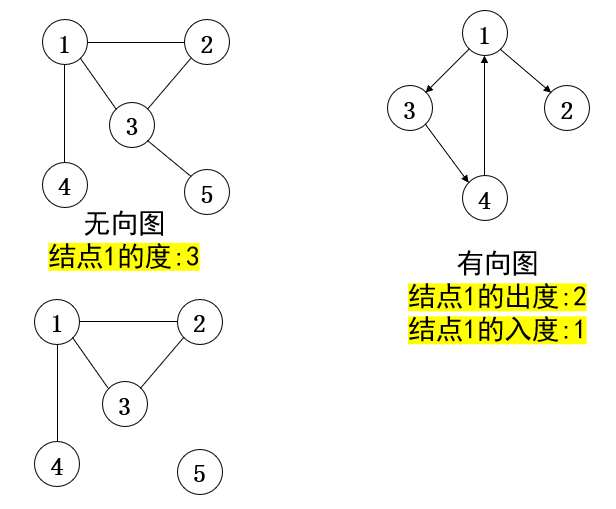
**1.4 图形结构**

1. **图的基本术语：度、入度、出度、无向图、有向图**

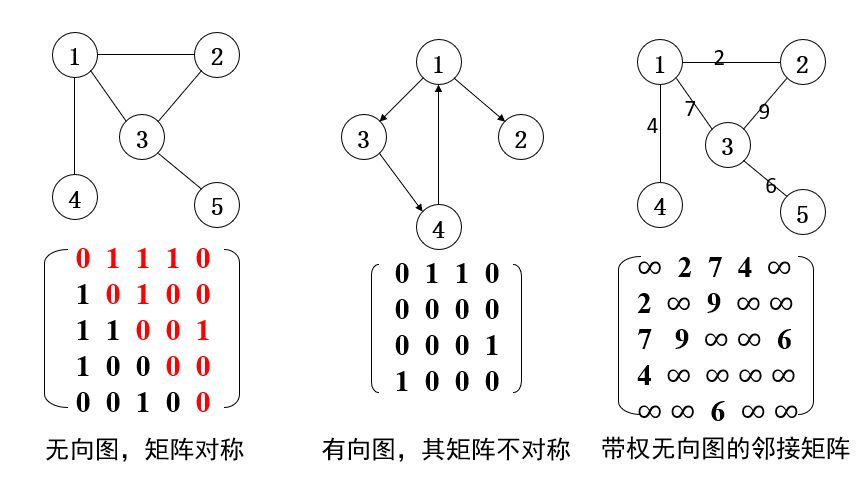
****

**注意无向图的表示方法：G=(V, E)，顶点集合V，边集合E**

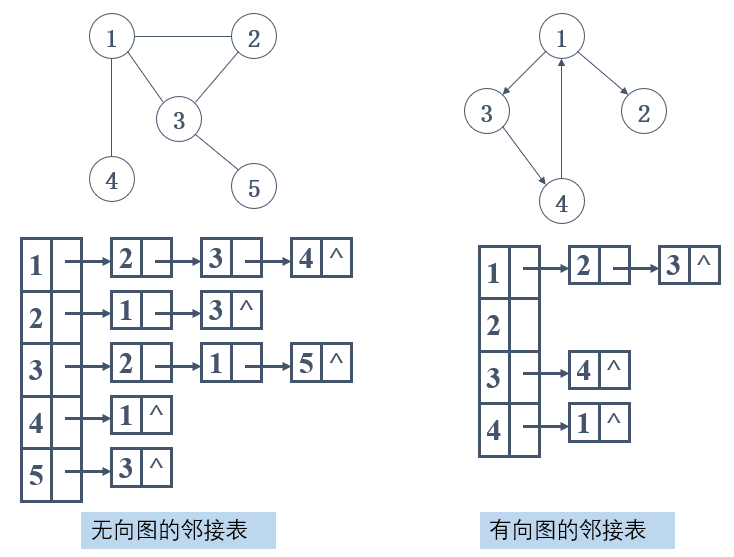
****

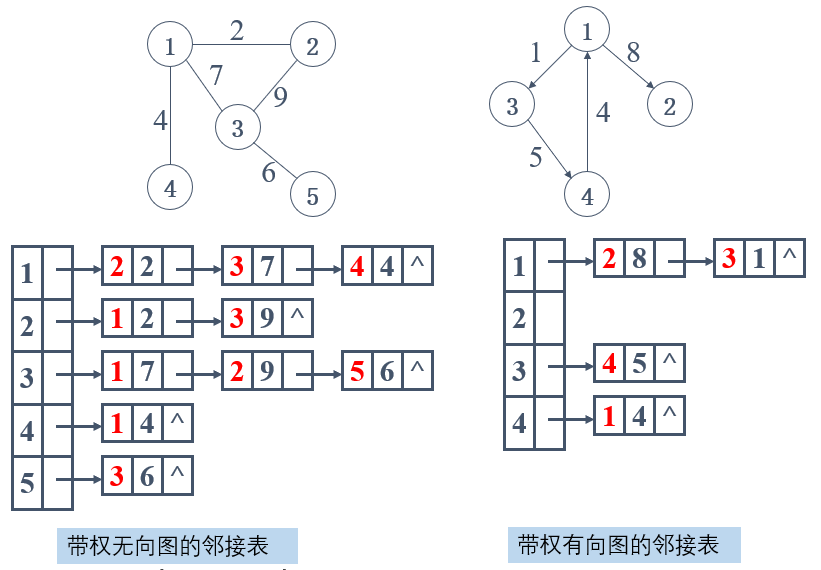
****

1. **图的存储结构：**
2. **邻接矩阵**

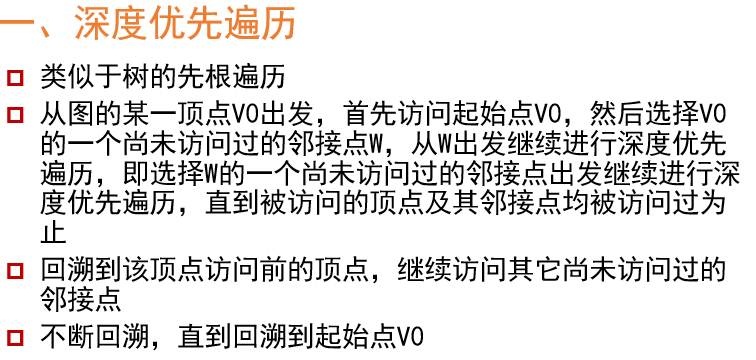
****

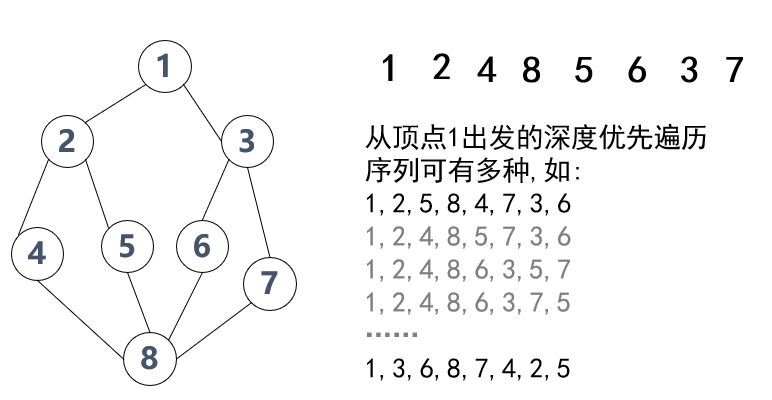
1. **邻接表**

****

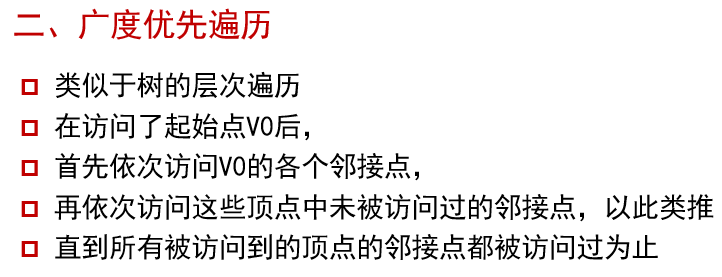
****

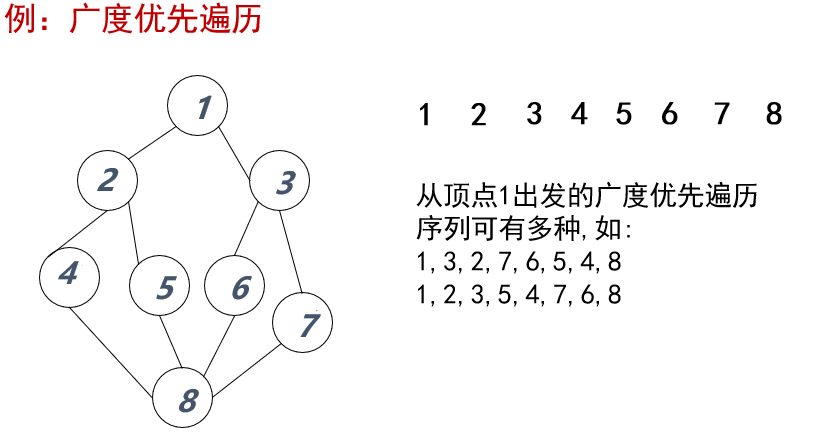
1. **深度优先遍历**

****

****

1. **广度优先遍历**

****

****

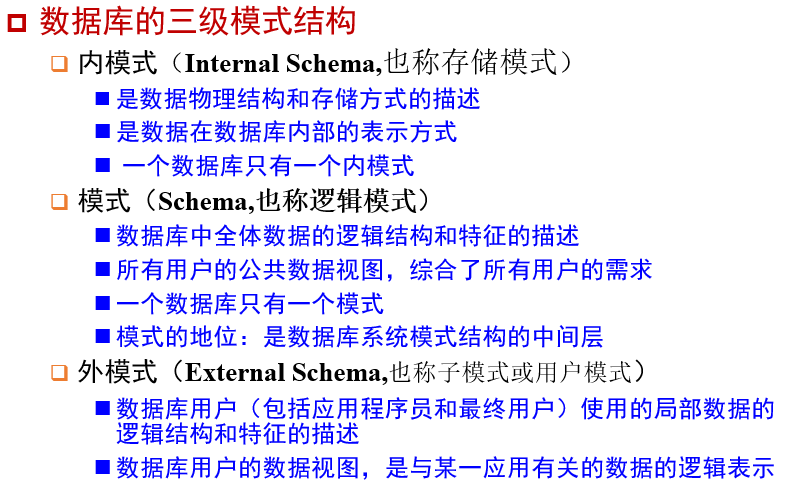
**2、题型**

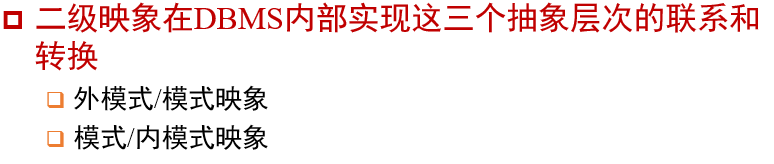
1. **单项选择题**
2. **简答题**

**数据库**

**1、概念回顾**

1. **数据、数据库、数据库管理系统、数据库系统**
   * 1. **概念**
     2. **特点**
     3. **典型的DBMS系统**
2. **数据库的三级模式、二级映像**

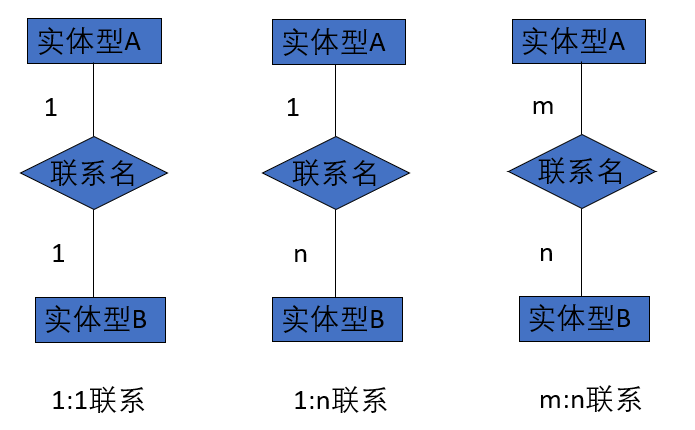
****

****

1. **数据模型**

**1）概念模型**

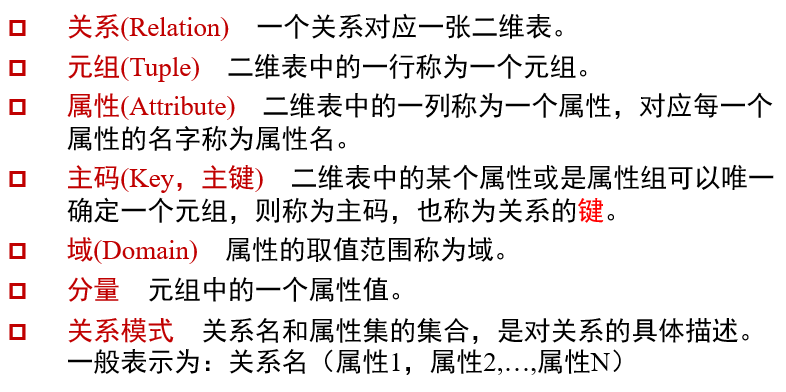
* **实体、属性、码、联系**
* **实体之间的联系类型**

****

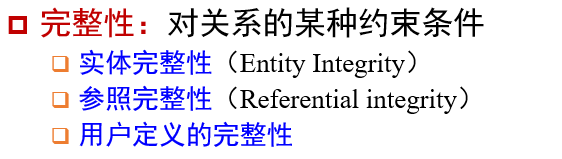
* **E-R图的表示形式**

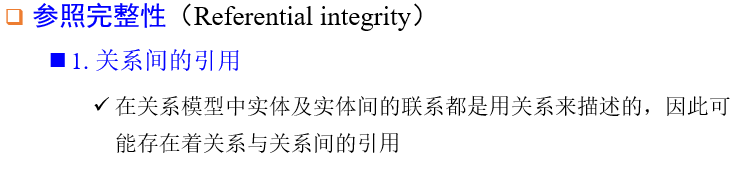
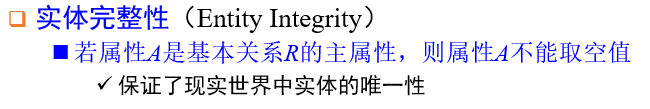
**2）逻辑数据模型**

* **层次模型**
* **网状模型**
* **关系模型**

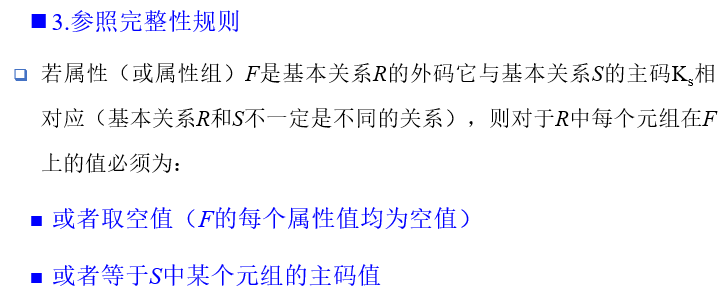
****

1. **关系数据库**

****

****

****

****

1. **SQL数据查询**

**Select [ALL][Distinct]**

**<Result Column Expression>**

**[, <Result Column Expression>]**

**……**

**From <Table Name or View Name>**

**[, <Table Name or View Name>]**

**[Where <Condition Expression>]**

**[Group By <Result Column Expression>**

**[Hanving] <Condition Expression>] ]**

**[Order By <Result Column Expression>[ASC|DESC]]**

|  |  |
| --- | --- |
| **查询条件** | **谓 词** |
| **大小比较** | **=,>,<,>=,<=,<> 以及Not+上述运算符** |
| **确定范围** | **Between… And……,以及Not+** |
| **确定集合** | **IN，Not+** |
| **字符匹配** | **Like, Not+** |
| **空值判断** | **Is Null, Is Not Null或Not+** |
| **多重条件** | **And , OR, NOT** |
| **谓词** | **EXISTS(存在量词)，ALL，SOME，UNIQUE ,LIKE** |
| **聚合函数** | **AVG(平均值)，MIN(最小值)，MAX(最大值)，SUM(和)，COUNT（计数）** |

**要求掌握：**

1. **会利用带搜索条件（WHERE子句）进行查询**
2. **根据字符匹配条件查询属性值与“匹配串”相匹配或不匹配的元组**

**字符匹配条件　通常用于查询属性值与“匹配串”相匹配或不匹配的元组，可以使用谓词LIKE或NOT　LIKE实现。其基本格式为：**

**[Not] Like ‘<匹配串>’ [Escape ‘<换码字符>’]**

**通配符：**

**％（百分号）：代表任意长度的字符串；**

**\_ （下划线）：代表任意单个字符。**

**e.g. Select \* From Course Where CName Like ‘微机%’**

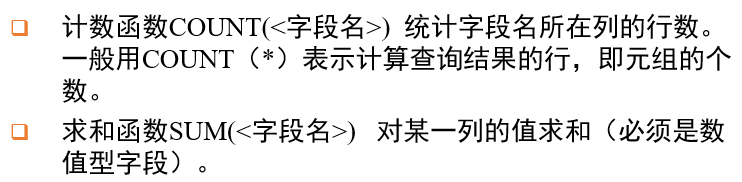
**e.g. Select \* From Course Where CNo Like ‘\_\_B%’**

**换码字符：匹配串中紧跟换码字符后的一个字符为普通字符，即不具有通配含义**

**e.g. Select CName from Course**

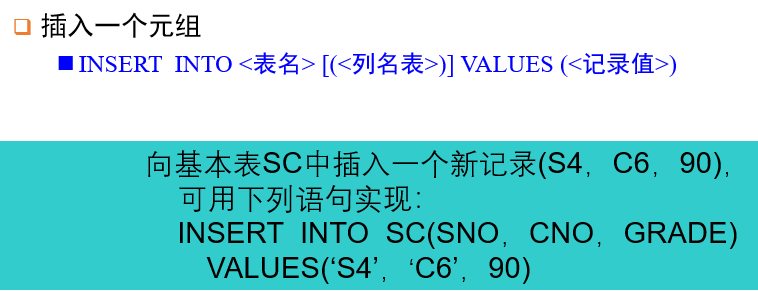
**Where SName Like ‘DB\\_%i\_\_’ Escape ‘\’**

1. **会在SELECT语句中使用集合函数对查询结果进行统计**

****

|  |  |
| --- | --- |
| **集合函数** | **功 能** |
| **COUNT(\*)** | **选定的行数** |
| **SUM(<EXP>)** | **表达式中所有值的和** |
| **AVG (<EXP>)** | **数字表达式中所有值的平均值** |
| **MAX (<EXP>)** | **表达式中的最高值** |
| **MIN (<EXP>)** | **表达式中的最低值** |
| **STDEV (<EXP>)** | **(\*)表达式中所有值的统计标准偏差** |
| **STDEVP (<EXP>)** | **(\*)表达式中所有值的填充统计标准偏差** |
| **VAR (<EXP>)** | **(\*)表达式中所有值的统计方差** |
| **VARP (<EXP>)** | **(\*)表达式中所有值的填充统计方差** |
| **(\*)并非所有SQL都支持，但大部分支持。** | |

1. **对查询结果排序**
   * **可按一个或多个列排序: Desc（递减）、Asc(递增)（缺省）；**
   * **每列可以指定自己的排序方法；**
   * **按排序列的先后顺序优先排序；**
   * **排序的列必须出现在查询结果列中；**
2. **会插入新数据**

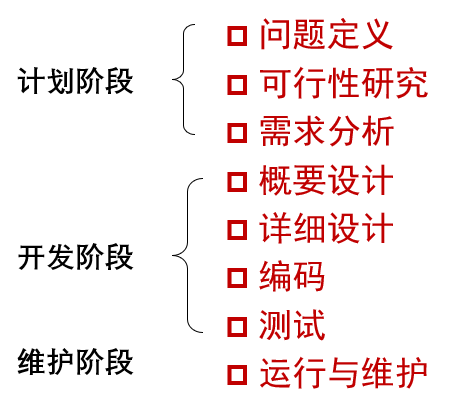
****

**2、考查题型**

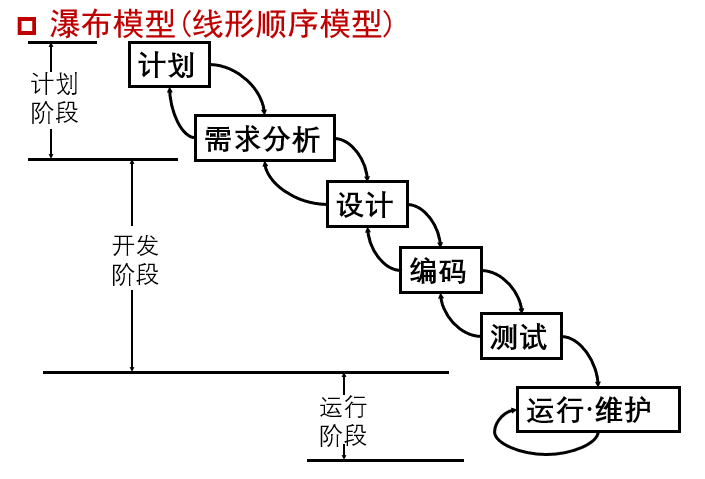
1. **选择题**
2. **问答题**

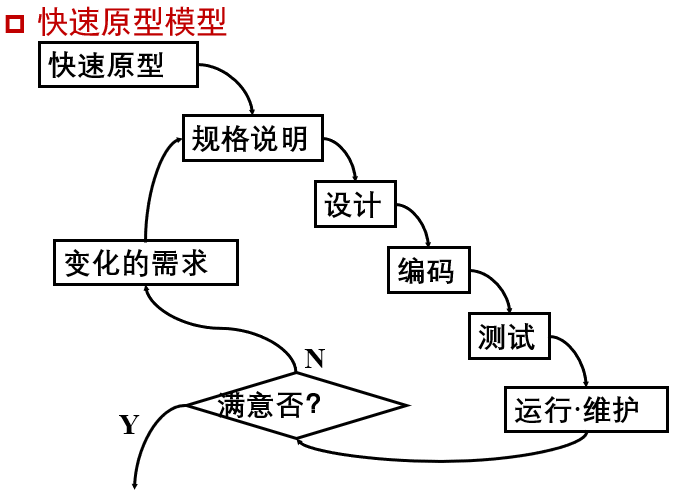
**软件工程**

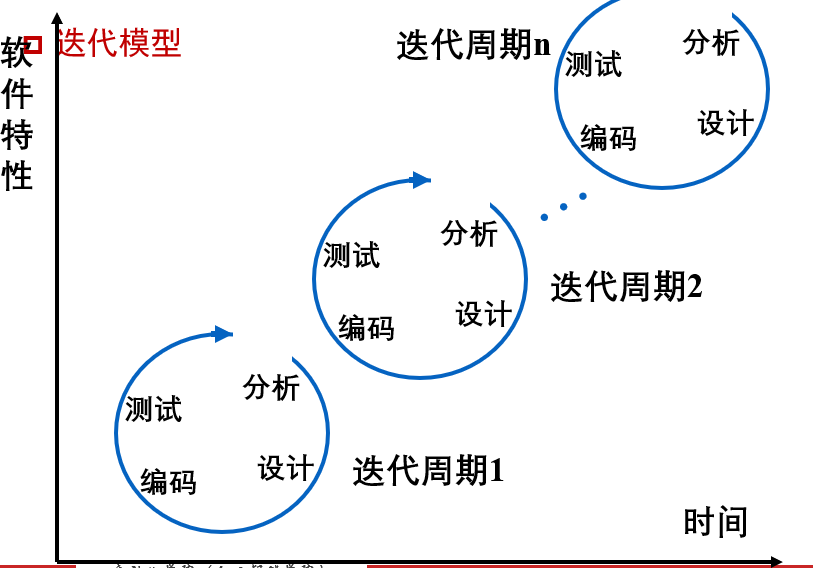
1. **软件工程出现的原因——软件危机**
2. **软件的生命周期**

****

1. **软件的生命周期模型**

****

****

****

1. **了解结构化分析和结构化设计方法**
2. **常见的维护手段和所占比重**

**题型及分值分布**

1. **操作系统（共30分）**
2. **选择题（1分/题，共8分）**
3. **填空题（2分/题，共8分）**
4. **简答题（4分，5分，5分，共14分）**
5. **数据结构（共35分）**
6. **选择题（1分/题，共5分）**
7. **简答题（5分/题，共10分）**
8. **设计题（10分/题，共20分）**
9. **数据库（共30分）**
10. **选择题（1分/题，共6分）**
11. **数据库问答题（3分/题，共24分）**
12. **软件工程（共5分）**
13. **选择题（1分/题，共5分）**

**练习题**

1、已知一段报文中各字符{A,B,C,D,E,F,G}的出现频率分别为W={8,2,5,3,2,17,4}，尝试构造该报文的哈夫曼编码（Huffman coding），写出各字符的编码，并计算最终的哈夫曼编码长度（Huffman coding的位长度，即哈夫曼树的WPL）

2、已知无向图的邻接表，写出从顶点V0 出发分别进行深度优先和广度优先搜索遍历得到的顶点序列。



3、请写出下图的邻接矩阵和邻接表

根据上述邻接表从顶点1开始分别按照深度优先和广度优先遍历该图（要求：在满足遍历算法的条件下按数字从小到大排序）

****

4、设有数据表Students，Course，Enrolls 如下表所示，分别写出符合下面要求的SQL查询语句。

(1) 列出姓名以“Zhang”开头的学生的学号、姓名和年龄。

(2) 找出数学课成绩在85分以上的学生姓名和学号。

(3) 列出所有学生选修的课程名称及成绩。

(4) 统计年龄小于等于22岁的学生人数。

|  |  |
| --- | --- |
| **Course** | |
| **Sno** | **Subject** |
| 0001 | Math |
| 0002 | English |
| 0003 | Chinese |

|  |  |  |
| --- | --- | --- |
| **Enrolls** | | |
| **Sno** | **Number** | **Grade** |
| 0001 | 980400 | 89 |
| 0002 | 980400 | 85 |
| 0003 | 980400 | 78 |
| 0001 | 980401 | 90 |
| 0002 | 980403 | 76 |

|  |  |  |
| --- | --- | --- |
| Student | | |
| **Number** | **Name** | **Age** |
| 980400 | Zhang San | 19 |
| 980401 | Li Si | 21 |
| 980402 | Wang Hao | 20 |

(1) Select \* from Students where Name like 'Zhang%'

(2) Select name From Students Where Sno in (Select Sno From Enrolls Where Grade>85)

(3) Select Subject, Grade From Students, Enrolls Where Students.Sno ＝ Enrolls.Sno

(4) Select Count(\*) From Students Where Age<＝22

5. 针对教学管理数据库的3个基本表

学生S(SNO,SNAME,AGE,SEX)

选课SC(SNO,CNO,GRADE)

课程C(CNO,CNAME,TEACHER)

试用SQL语句表达下列查询

* + 1. 检索LIU老师所授课程的课程号和课程名

SELECT CNO,CNAME FROM C

WHERE TEACHER=’LIU’

* + 1. 检索年龄大于23岁的男学生的学号和姓名

SELECT SNO,SNAME FROM S

WHERE AGE>23 AND SEX=’男’

* + 1. 检索至少选修LIU老师所授课程中的一门课程的女学生的姓名

1）连接查询

SELECT SNAME FROM S

WHERE S.SNO=SC.SNO AND SC.CNO=C.CNO AND TEACHER=’LIU’ AND SEX=’女’

2）嵌套查询

SELECT SNAME FROM S

WHERE SEX=’女’ AND SNO IN

(SELECT SNO FROM SC

WHERE CNO IN

(SELECT CNO FROM C

WHERE TEACHER=’LIU’ ))

* + 1. 检索WANG同学不学的课程的课程号

SELECT CNO FROM C

WHERE NOT EXISTS

(SELECT \* FROM S,SC

WHERE S.SNO=SC.SNO ANDSC.CNO=C.CNO AND SNAME=’WANG’)

* + 1. 统计有学生选修的课程门数

SELECT COUNT(DISTINCT CNO) FROM SC

* + 1. 求选修C4课程的学生的平均年龄

SELECT AVG(AGE) FROM S, SC

WHERE S.SNO=SC.SNO AND CNO=’C4’

* + 1. 求LIU老师所授课程的每门课程的学生平均成绩

SELECT C.CNO, AVG(GRADE) FROM S, SC

WHERE SC.CNO=C.CNO AND TEACHER=’LIU’

* + 1. 统计每门课程的学生选修人数（超过10人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

SELECT CNO, COUNT(SNO) FROM SC

GROUP BY CNO

HAVING COUNT (\*)>10

ORDER BY 2 DESC, 1

* + 1. 往基本表中插入一个学生元组(‘200912143’，‘张晶’，21，‘女’)

INSERT INTO S(SNO,SNAME,AGE,SEX)

VALUES(‘200912143’，‘张晶’，21，‘女’)