

数据库

1. 数据库保护(访问)的内容有哪些?

- ① 利用权限机制和完整性约束防止非法数据进入数据库;
- ② 提供故障恢复能力;
- ③ 提供并发访问控制。

2. DBA 的职责是什么?

DBA 负责全面管理和控制数据库系统. 职责有:

- ① 决定数据库中的信息内容和结构:数据库中要存放哪些信息
- ② 决定数据库的存储结构和存取策略:获得较高的存取效率和存储空间利用率
- ③定义数据的安全性要求和完整性约束条件:负责确定各个用户对数据库的存取权限数据的保密级别和完整性约束条件
- ④监控数据库的使用和运行:转储数据、维护日志文件、故障恢复
- ⑤数据库的改进和重组重构:对运行情况进行记录、统计分析,以此来不断改进数据库设计

3. 数据库系统和文件系统相比有什么优点?

文件系统	数据库管理系统
某一应用	现实世界
共享性差, 冗余度大	共享性高, 冗余度小
记录内有结构, 整体无结构	整体结构化, 用数据模型描述
应用程序自己控制	由数据库管理系统提供数据安全性, 完整性, 并发控制和恢复能力
独立性差	具有高度的物理独立性和一定的逻辑独立性

4. 什么是完整性约束

完整性约束可确保数据库中的数据正确性和相容性, 主要包括: 实体完整性、参照完整性、用户自定义完整性。

5. DBMS 支持那几种数据模型

常用的是层次模型, 网状模型和关系模型(最重要)。新兴的是面向对象数据模型和对象关系数据模型。

层次模型: 层次模型是指用树行结构表示实体及其之间的联系, 树中每一个节点代表一个记录类型, 树状结构表示实体型之间的联系。

在一个层次模型中的限制条件是: 有且仅有一个节点, 无父节点, 此节点为树的根; 其他节点有且仅有一个父节点。

网状模型的数据结构主要有以下两个特征:

(1) 允许一个以上的节点无双亲。

(2) 一个节点可以有多个双亲。

网状模型：用有向图结构表示实体类型及实体间联系的数据结构模型称为网状模型

关系实际上就是关系模式在某一时刻的状态或内容。也就是说，关系模式是型，关系是它的值。关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的，因为关系操作在不断地更新着数据库中的数据。但在实际当中，常常把关系模式和关系统称为关系

现实世界的实体以及实体间的各种联系均用关系来表示，从用户角度看，关系模型中数据的逻辑结构是一张二维表。

6. SQL (Structured Query Language) 的四个组成部分？

- ① 数据定义语言 (Data Definition Language) ；
- ② 查询语言 (Query Language) ；
- ③ 数据操纵语言 (Data Manipulation Language) ；
- ④ 数据控制语言 (Data Control Language) 。

7. 数据库操纵语言举例

数据操纵语言 DML (Data Manipulation Language) ，用户通过它可以实现对数据库的基本操作。例如，对表中数据的查询 (select) 、插入 (insert) 、删除 (delete) 和修改 (update) 。

8. 介绍下有哪些应用数据库？

大概知道的有：

桌面型：Access、Foxpro、dBase

企业型：SQL Server 系列、MySQL、Oracle、Sybase

9. 什么是数据独立性。

数据独立性表示应用程序与数据库中存储的数据不存在依赖关系，包括逻辑数据独立性和物理数据独立性。

逻辑数据独立性是指局部逻辑数据结构 (外视图即用户的逻辑文件) 与全局逻辑数据结构 (概念视图) 之间的独立性。当数据库的全局逻辑数据结构 (概念视图) 发生变化 (数据定义的修改、数据之间联系的变更或增加新的数据类型等) 时，它不影响某些局部的逻辑结构的性质，应用程序不必修改。

物理数据独立性是指数据的存储结构与存取方法 (内视图) 改变时，对数据库的全局逻辑结构 (概念视图) 和应用程序不必作修改的一种特性，也就是说，数据库数据的存储结构与存取方法独立。

10. 1NF 是啥，数据模式

第一范式 是数据库最基本的要求，即每一个分量必须是不可分的数据项；

第二范式 是数据库非主属性对码的部分函数依赖，即每一个非主键完全函数依赖于主键；

第三范式 是非主属性对码的传递依赖，即不包含已在其他表中已包含的非主键信息，防止数据冗余；

数据模式是基于选定的数据模型对数据进行的“型”的方面的刻画，而相应的“实例”则是对数据“值”的方面的描述。先有数据模型，才能据其讨论相应数据模式，有了数据模式，就能依据该模式得到相应的实例。

11. 数据库的关系操作有哪些？各有什么作用？

关系模型中常用的关系操作包括：选择（Select）、投影（Project）、连接（Join）、除（Divide）、并（Union）、交（Intersection）、差（Difference）、笛卡尔积等查询（Query）操作和增加（Insert）、删除（Delete）、修改（Update）操作两大部分。查询的表达能力是其中最主要的部分。

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（set-at-a-time）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（record-at-a-time）的方式

12. DB 并发操作通常带来哪三类问题？

丢失修改（Lost Update）：当一个事务修改了数据，并且这种修改还没有提交到数据库中时，另外一个事务又对同样的数据进行了修改，并且把这种修改提交到了数据库中。这样，数据库中没有出现第一个事务修改数据的结果，好像这种数据修改丢失了一样。

脏读（Dirty Read）：当一个事务正在访问数据，并对数据进行了修改，而这种修改还没有提交到数据库中，这时，另一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。

不可重复读（Non-Repeatable Read）：在一个事务内，多次读同一数据。在这个事务还没有结束时，另一个事务也访问该同一数据，那么，在第一个事务中的两次读数据之间，由于第二个事务的修改，第一个事务两次读到的数据可能是不一样的。

13. 两段锁协议

所谓两段锁协议是指所有事务必须分两个阶段对数据项加锁和解锁：

1. 在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁；
2. 而且在释放一个封锁之后，事务不再申请和获得任何其他封锁。

所谓“两段”锁的含义是，事务分为两个阶段，第一阶段是获得封锁，也称为扩展阶段。在这阶段，事务可以申请获得任何数据项上的任何类型的锁，但是不能释放任何锁。第二阶段是释放封锁，也称为收缩阶段。在这阶段，事务可以释放任何数据项上的任何类型的锁，但是不能再申请任何锁。

例如事务 T1 遵守两段锁协议，其封锁序列是：（如右）

又如事务 T2 不遵守两段锁协议，其封锁序列是：

Slock A ... Unlock A ... Slock B ... Xlock C ... Unlock C ... Unlock B;

可以证明，若并发执行的所有事务均遵守两段锁协议，则对这些事务的任何并发调度策略都是可串行化的。

另外要注意两段锁协议和防止死锁的一次封锁法的异同之处。一次封锁法要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行，因此一次封锁法遵守两段锁协议；但是两段锁协议并不要求事务必须一次将所有要使用的数据全部加锁，因此遵守两段锁协议的事务可能发生死锁。

14. 事务的四个特点

事务的概念：用户定义的一个数据库操作序列，这些操作要么全做要么全不做，是一个不可分割的工作单位。

1. 一致性（Consistency）
2. 原子性（Atomicity）
3. 隔离性（Isolation）
4. 持续性（Durability）

15. 数据库故障的种类？

1、事务内部的故障 2、系统故障 3. 介质故障 4. 计算机病毒

16. 数据库恢复策略有哪几种？

1. 数据转储（数据冗余） 2. 登记日志文件

17. 数据库的三级模式

1、模式又称概念模式或逻辑模式，对应于概念级，是对数据库中全部数据的逻辑结构和特征的总体描述，是所有用户的公共数据视图(全局视图)，反映了数据库系统的整体观。

2、外模式又称子模式或用户模式，对应于用户级。它是某个或某几个用户所看到的数据库的数据视图，是与某一应用有关的数据的逻辑表示。外模式是从模式导出的一个子集，包含模式中允许特定用户使用的那部分数据。外模式反映了数据库的用户观。

3、内模式又称存储模式，对应于物理级，它是数据库中全体数据的内部表示或底层描述，是数据库最低一级的逻辑描述，它描述了数据在存储介质上的存储方式和物理结构，对应着实际存储在外存储介质上的数据库。内模式是数据库的存储观。

在一个数据库系统中，只有唯一的数据库，因而作为定义、描述数据库存储结构的内模式和定义、描述数据库逻辑结构的模式，也是唯一的，但建立在数据库系统之上的应用则是非常广泛、多样的，所以对应的外模式不是唯一的，也不可能是唯一的。

18. 什么是表？什么是视图？两者的区别和联系是什么？

表是实实在在得保存数据的实体，写入的数据都保存在表中，而视图是不保存数据的，

也没有数据。视图就是一条语句，实际上视图从表中取数据。只是给我们的感觉好像直接从表中取得一样。表可以建立各种触发器，可以建立索引，可以建立主键，约束等。但是视图不能建立这些对象(视图可以建立替代触发器)。表和视图可以更新，但是视图的更新受到约束。比如，`groupby` 和表连接生成的视图不能更新表是实实在在得保存数据的实体，写入的数据都保存在表中，而视图是不保存数据的，也没有数据。联系：可以理解为视图的数据都是从表中取出来，视图就是你定下的一个框框而已，内容都是在表中。

19. 1NF 有哪些缺点？

1. 数据冗余太大（每一个系的系主任名字重复出现）
2. 更新异常（某个系更换系主任之后，必须将该系学生有关的每一个元组）
3. 插入异常（如果一个系刚成立，尚无学生，就无法将老师存入到数据库中）
4. 删除异常（如果学生都毕业了，在删除学生时，把这个系的系主任的所有信息都删除了）

20. 数据字典通常包含五个部分

1. 数据项 2. 数据结构 3. 数据流 4. 数据存储 5. 处理过程

21. 数据库中视图的优点

- ① 化用户操作；
- ② 使用户能以多种角度看待统一数据；
- ③ 对重构数据库提供了一定程度的逻辑独立性；
- ④ 对机密数据提供安全保护；
- ⑤ 可以更清晰的表达查询。

22. 在做数据库开发中使用视图的优点有：

1. 可以访问表中列的子集。在表中有些列是比较敏感的数据不想用户看到的，如用户密码，员工工资等，用视图可以隐藏这些列。
2. 可以访问表中行的子集。有时不想让用户看到与 **TA** 无关的数据时，可以在 **where** 条件中过滤，如企业中子公司的员工只能看到 **TA** 所在子公司的同事资料，而不想其它子公司员工资料也显示出来，用过滤后查询出来的数据集组成视图。
3. 可以重命名列名。在表中有些列名定义没有代表意义，如列“**abc**”，为了让用户看到列名就可以知该列是什么数据，可以在组建视图时重命名列名。
4. 可以快速访问两表或多表连接所组成的数据。有时要访问表间连接所组成的数据集，可以把查询出来的数据集定义成视图，可以帮助快速访问所需的数据。
5. 可以快速读取用聚合函数运算所返回的数据集。有时用户要读取经过用聚合函数复杂运算后的结果集，每次读取都是很耗时，这时可以把这运算后的数据集组成视图，就可以避免每次读取都要运算的消耗。总的来讲，使用视图在提高安全性的同时也加快查询的速度。

23. 数据库的三要素

数据模型是一组严格定义的概念的集合，这些概念精确地描述了系统的静态特征（数据结构）、动态特征（数据操作）和完整性约束条件，这就是数据模型的三要素。

24. 数据库索引

目的：提供多种存储路径，加快查找速度。建立索引需要考虑的问题：1。没有查询、统计的需要则不建 2。数据增删改频繁，系统会花费许多时间来维护索引，从而降低了查询效率。

25. 哪些视图是可以更新、哪些是不可以更新的，请举例说明、

行列子集视图可以更新；

- ① 由两个以上基本表导出的视图，不可以更新；
- ② 字段来自于字段表达式或常数的视图，不允许(Insert、Update)，但是可以 Delete；
- ③ 字段来自聚集函数的视图不可以更新；
- ④ 定义中含有 Group By 字句的视图，不可以更新；
- ⑤ 定义中含有 DISTINCT 字句的视图，不可以更新；
- ⑥ 定义中有嵌套查询，并且内层查询的 From 字句中涉及的表也是导出该视图的基本表的视图，不可以更新；
- ⑦ 不允许更新的视图上定义的视图也不允许更新。

26. 日志文件的定义与作用。

日志文件是用来记录事务对数据库的更新操作的文件。

作用：在数据库恢复中起着重要的作用。并协助后备副本进行介质故障恢复。

日志文件主要包含：事务标识、操作的类型、操作对象、更新前数据的旧值、更新后数据的新值。

27. 数据库完整性与安全性的区别？

完整性和安全性是两个不同的概念。前者是为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出造成的无效操作和错误结果而后者是防止数据库被恶意的破坏和非法的存取。当然，完整性和安全性是密切相关的。特别是从系统实现的方法来看，某一种机制常常既可以用于安全保护亦可用于完整性保证。

28. 数据库有哪几种备份策略，大概的操作

完全备份、事务日志备份、差异备份、文件备份。

29. DBMS 控制管理功能

- ① 数据库定义和创建；
- ② 数据组织、存储和管理；
- ③ 数据存取；
- ④ 数据库事务管理和运行管理；

⑤ 数据库的建立和维护。

30. 数据库采用什么机制来保证数据库并发操作的正确性？

并发控制：封锁、时间戳、乐观控制法。

31. 数据库安全系统和计算机安全系统的人关系

安全性问题不是数据库系统所独有的，所有计算机系统都有这个问题。只是在数据库系统中大量数据集中存放，而且为许多最终用户直接共享，从而使安全性问题更为突出。系统安全保护措施是否有效是数据库系统的主要指标之一。数据库的安全性和计算机系统的安全性，包括操作系统，网络系统的安全性是紧密联系，相互支持的。

32. 函数模板和类模板的区别

-- 实例化不同。

33. 数据库管理系统的完整性要完成的功能

-- 防止不合语意的数据，防止无效操作和错误结果

34. 数据库的两种独立性

物理独立性指的是应用程序与数据存放在相互独立的磁盘地址，内模式与模式映像保证了其物理独立特性；

逻辑独立性指的数据与程序逻辑结构上的独立特性，数据或应用程序的逻辑结构发生变化性都不影响对方，外模式与模式映像保证了其逻辑独立性。

35. DBMS 的控制操作有哪些

①数据的安全性保护；②数据的完整性检查；③并发控制；④数据库恢复。

36. 什么是数据库控制语言,试举例说明

DCL 是用来设置或更改数据库用户角色权限的语句，主要包括 GRANT（授权）、DENY（拒绝）、REVOKE（删除）三类。

37. 数据库两级模式的作用？

三模式->①外模式/模式映像；②模式/内模式映像；

保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

38. 笛卡尔积

在数学中，两个集合 X 和 Y 的笛卡儿积（Cartesian product），又称直积，表示为 $X \times Y$ ，是其第一个对象是 X 的成员而第二个对象是 Y 的一个成员的所有可能的有序对。

39. 数据库中 primary key 约束和 unique 约束有什么区别？

主键约束：有一列或列的组合，其值能唯一的标识表中的每一行。

唯一约束：不允许数据库表在指定列上具有相同的值，但允许有空值。

区别：一个表可以定义多个唯一约束，但只能定义一个主键约束；空值的列上可以定义唯一约束，不能定义主键约束。

网络

1. 比较 TCP 与 UDP

UDP:①无连接;②尽最大努力交付;③面向报文;④无拥塞机制;⑤可以一对一,一对多,多对一,多对多;⑥首部开销小。

TCP:①面向连接;②点对点;③可靠交付;④全双工通信;⑤面向字节流。

2. 网络协议的三个核心要素及概念,各起什么作用?

语法:定义了数据与控制信息的格式;

语义:定义了需要发出何种控制信息,完成何种响应动作以及作出何种响应;

同步:定义了事件实现顺序的详细说明;

3. 解释一下网络体系结构,它得实现和理论有什么区别?

网络体系结构是指通信系统的整体设计,它为网络硬件、软件、协议、存取控制和拓扑提供标准。网络体系结构采用分层结构,各层之间相互独立、较易维护、灵活性好。

国际标准化组织制定了 OSI/RM 标准,该标准采用了七层结构应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。七层协议体系结构既复杂又不实用,但其概念清楚,体系结构理论较完整。而 TCP/IP 却成为了事实上的标准,它采用了四层结构即应用层、传输层、网络层和网络接口层。

4. 波特和比特的区别?

波特是码元传输的速率单位,说明每秒传多少个码元。码元传输速率也称为调制速率、波形速率或符号速率。

比特是信息量的单位,与码元的传输速率“波特”是两个完全不同的概念。

但是,信息的传输速率“比特/秒”与码元的传输速率“波特”在数量上却有一定的关系。

5. 什么是网络延时

时延(delay 或 latency)是指一个报文或分组从一个网络(或一条链路)的一端传送到另一端所需的时间。

6. 什么是码元?什么是码元长度?

在数字通信中常常用时间间隔相同的符号来表示一位二进制数字这样的时间间隔内的信号称为二进制码元,而这个间隔被称为码元长度。

7. 结合 Internet,说说有连接服务和无连接的服务?

面向连接服务具有连接建立、数据传输和连接释放这三个阶段。面向连接服务是在数据交换之前,必须先建立连接。当数据交换结束后,则必须终止这个连接。在传送数据时是按序传送的,是可靠交付。面向连接服务比较适合于在一定期间内要向同一目的地发送许多报文的情况。

无连接服务,两个实体之间的通信不需要先建立好一个连接,因此其下层的有关资源不需要事先进行预定保留。这些资源将在数据传输时动态地进行分配。无连接服务的优点是灵活方便和比较迅速。但无连接服务不能防止报文的丢失、重复或失序。是一种不可靠的服务。这种服务常被描述为“尽最大努力交付”。

8. 点对点和端到端工作在哪层?工作机制?

点对点协议(Point to Point Protocol)的缩写为PPP,是TCP/IP网络协议包的一个成员。PPP是TCP/IP的扩展,它增加了两个额外的功能组:

- (1)它可以通过串行接口传输TCP/IP包;
- (2)它可以安全登录。

数据传输的可靠性是通过数据链路层和网络层的点对点和传输层的端对端保证的。点对点是基于MAC地址或者IP地址,是指一个设备发数据给另外一个设备,这些设备是指直连设备包括网卡,路由器,交换机。端对端是网络连接,应用程序之间的远程通信。端对端不需要知道底层是如何传输的,是一条逻辑链路。

端到端与点到点是针对网络中传输的两端设备间的关系而言的。端到端传输指的是在数据传输前,经过各种各样的交换设备,在两端设备间建立一条链路,就像它们是直接相连的一样,链路建立后,发送端就可以发送数据,直至数据发送完毕,接收端确认接收成功。点到点系统指的是发送端把数据传给与它直接相连的设备,这台设备在合适的时候又把数据传给与之直接相连的下一台设备,通过一台一台直接相连的设备,把数据传到接收端。端到端传输的优点是链路建立后,发送端知道接收设备一定能收到,而且经过中间交换设备时不需要进行存储转发,因此传输延迟小。端到端传输的缺点是直到接收端收到数据为止,发送端的设备一直要参与传输。如果整个传输的延迟很长,那么对发送端的设备造成很大的浪费。端到端传输的另一个缺点是如果接收设备关机或故障,那么端到端传输不可能实现。点到点传输的优点是发送端设备送出数据后,它的任务已经完成,不需要参与整个传输过程,这样不会浪费发送端设备的资源。另外,即使接收端设备关机或故障,点到点传输也可以采用存储转发技术进行缓冲。点到点传输的缺点是发送端发出数据后,不知道接收端能否收到或何时能收到数据。在一个网络系统的不同分层中,可能用到端到端传输,也可能用到点到点传输。如Internet网,IP及以下各层采用点到点传输,IP层以上采用端到端传输。

端对端,点对点,只是称为问题,本质区别很小

端对端,主要服务于Application Layer,是说两台主机(终端),跨过网络直接连接
点对点,是说两台主机(终端)在局域网中传输。

9. 网络时延又拿几部分组成?各产生于何处?

网络时延主要由发送时延,传播时延,处理时延组成。

发送时延是指结点在发送数据时使数据块从结点进入到传输媒体所需的时间,也就是从数据块的第一个比特开始发送算起,到最后一个比特发送完毕所需的时间。发送时延又称为传输时延,它的计算公式是:

发送时延=数据块长度/信道带宽

信道带宽就是数据在信道上的发送速率,它也常称为数据在信道上的传输速率。

传播时延是指从发送端发送数据开始,到接收端收到数据(或者从接收端发送确认帧,到发送端收到确认帧),总共经历的时间。

传播时延 = d/s

d = 物理链路的长度

s = 介质的信号传播速度 ($\sim 2 \times 10^8$ m/sec)

处理时延是指计算机处理数据所需的时间, 与计算机 CPU 的性能有关。

10. 实体表示任何可发送或接收信息的硬件或软件进程。

协议是控制两个对等实体或多个实体进行通信的规则集合。

在协议的控制下, 两个对等实体间的通信使得本层能够向上一层提供服务。要实现本层协议, 还需要使用下面一层所提供的服务。协议是水平的, 服务是垂直的

11. TCP/IP 网络协议的核心是什么, 如何引出 “over everything” 和 “everything over?”

TCP/IP 协议的核心是 TCP、UDP 和 IP 协议

分层次画出具体的协议来表示 TCP/IP 协议族, 它的特点是上下两头大而中间小: 应用层和网络接口都有很多协议, 而中间的 IP 层很小, 上层的各种协议都向下汇聚到一个 IP 协议中。这种很像沙漏计时器形状的 TCP/IP 协议族表明: TCP/IP 协议可以为各种各样的应用提供服务 (everything over ip) 同时 TCP/IP 协议也允许 IP 协议在各种各样的网络构成的互联网上运行 (IP over everything)。

12. 用白军和蓝军解释下没有 100%可靠的通信

占据东边和西边两个山顶的蓝军与驻扎在这两个山之间的山谷的白军作战. 其力量对比是: 一个山顶上的蓝军打不过白军, 但两个山顶的蓝军同时协同作战则可战胜白军. 东边蓝军打算在第二天正午向白军发起攻击. 于是用计算机发送电文给西边的友军. 由于通信线路很不好, 电文出错或丢失的可能性较大 (没有电话使用). 因此要求收到电文的友军必须送回一个确认电文. 但是确认电文也可能丢失或者出错. 试问能否设计出一种协议能使协同作战实现从而一定 (100%而不是 99.999...%) 取得胜利?

东边蓝军先发送: “拟于明日正午向白军发起攻击. 请协同作战, 并确认.”

西边蓝军收到电文并加以确认, 回答: “同意.”

然而两边都不敢贸然下决心, 因为西边蓝军还要等待东边蓝军发送: “我已经收到你的确认了.”

然后东边蓝军还要等收到西边蓝军的 “我收到了你的 “我已经收到你的确认了”. ”

如此反复...

这样一直等对方确认的确认, 两边都无法确定自己发出去的电文是否对方一定收到了.

因此, 没有一种协议能够使两边的蓝军 100%地确定一定会共同进攻.

所以也设计不出来 100%可靠的协议.

13. 电路与分组交换的区别

答：（1）电路交换电路交换就是计算机终端之间通信时，一方发起呼叫，独占一条物理线路。当交换机完成接续，对方收到发起端的信号，双方即可进行通信。在整个通信过程中双方一直占用该电路。它的特点是实时性强，时延小，交换设备成本较低。但同时也带来线路利用率低，电路接续时间长，通信效率低，不同类型终端用户之间不能通信等缺点。电路交换比较适用于信息量大、长报文，经常使用的固定用户之间的通信。

（2）报文交换将用户的报文存储在交换机的存储器中。当所需要的输出电路空闲时，再将该报文发向接收交换机或终端，它以“存储——转发”方式在网内传输数据。报文交换的优点是中继电路利用率高，可以多个用户同时在一条线路上传送，可实现不同速率、不同规程的终端间互通。但它的缺点也是显而易见的。以报文为单位进行存储转发，网络传输时延大，且占用大量的交换机内存和外存，不能满足对实时性要求高的用户。报文交换适用于传输的报文较短、实时性要求较低的网络用户之间的通信，如公用电话网。

（3）分组交换分组交换实质上是在“存储——转发”基础上发展起来的。它兼有电路交换和报文交换的优点。分组交换在线路上采用动态复用技术传送按一定长度分割为许多小段的数据——分组。每个分组标识后，在一条物理线路上采用动态复用的技术，同时传送多个数据分组。把来自用户端的数据暂存在交换机的存储器内，接着在网内转发。到达接收端，再去掉分组头将各数据字段按顺序重新装配成完整的报文。分组交换比电路交换的电路利用率高，比报文交换的传输时延小，交互性好。

14. DNS

DNS 是计算机域名系统 (Domain Name System 或 Domain Name Service) 的缩写，它是由解析器和域名服务器组成的。域名服务器是指保存有该网络中所有主机的域名和对应 IP 地址，并具有将域名转换为 IP 地址功能的服务器。其中域名必须对应一个 IP 地址，而 IP 地址不一定有域名。域名系统采用类似目录树的等级结构。域名服务器为客户机/服务器模式中的服务器方，它主要有两种形式：主服务器和转发服务器。将域名映射为 IP 地址的过程就称为“域名解析”。在 Internet 上域名与 IP 地址之间是一一对一（或者多对一）的，也可采用 DNS 轮循实现一对多，域名虽然便于人们记忆，但机器之间只认 IP 地址，它们之间的转换工作称为域名解析，域名解析需要由专门的域名解析服务器来完成，DNS 就是进行域名解析的服务器。DNS 命名用于 Internet 的 TCP/IP 网络中，通过用户友好的名称查找计算机和服务。当用户在应用程序中输入 DNS 名称时，DNS 服务可以将此名称解析为与之相关的其他信息，如 IP 地址。因为，你在上网时输入的网址，是通过域名解析系统解析找到了相对应的 IP 地址，这样才能上网。其实，域名的最终指向是 IP。

15. 集线器交换机路由器各工作在哪一层？

1. 集线器：物理层设备，用于信号的放大和连接多个终端。
2. 交换机：数据链路层设备，有多个冲突域和广播域，有多个端口以用于连接各个主机，使用物理地址（MAC 地址），转发数据较快。
3. 路由器：网络层设备，阻止广播，安全性高，使用逻辑地址（IP 地址），转发数据较慢。

16. NAT

网络地址转换(NAT, Network Address Translation)属接入广域网(WAN)技术，是一种将私有（保留）地址转化为合法 IP 地址的转换技术，它被广泛应用于各种类型 Internet

接入方式和各种类型的网络中。原因很简单,NAT 不仅完美地解决了 IP 地址不足的问题,而且还能够有效地避免来自网络外部的攻击,隐藏并保护网络内部的计算机。

NAT (Network Address Translation, 网络地址转换) 是将 IP 数据报报头中的 IP 地址转换为另一个 IP 地址的过程。在实际应用中, NAT 主要用于实现私有网络访问公共网络的功能。这种通过使用少量的公有 IP 地址代表较多的私有 IP 地址的方式,将有助于减缓可用 IP 地址空间的枯竭。

17. 数据链路层协议分类

面向字符的链路层协议:

ISO 的 IS1745, 基本型传输控制规程及其扩充部分 (BM 和 XBM)

IBM 的二进制同步通信规程 (BSC)

DEC 的数字数据通信报文协议 (DDCMP)

PPP

面向比特的链路层协议:

IBM 的 SNA 使用的数据链路协议 SDLC(Synchronous Data Link Control protocol);

ANSI 修改 SDLC, 提出 ADCCP (Advanced Data Communication Control Procedure);

ISO 修改 SDLC, 提出 HDLC (High-level Data Link Control);

CCITT 修改 HDLC, 提出 LAP (Link Access Procedure) 作为 X.25 网络接口标准的一部分, 后来改为 LAPB。

18. 路由器的组成

路由器由 RAM、ROM、FLASH 和 NVRAM 这 4 部分组成。

RAM: 随机访问存储器。RAM 中运行着 Cisco IOS 的镜像文件以及 running-config 文件。

ROM: 只读内存。ROM 中保存着最基本功能的代码 (最小的 IOS 代码), 它用于引导路由器。FLASH: 闪存。FLASH 中容纳了 IOS 软件的镜像。

NVRAM: 非易失性随机访问存储器。NVRAM 用来存储 startup-config 文件。当切断电源时, NVRAM 用一个电池来维护其中的数据。□

可以使用 show file systems 命令来查看路由器的文件系统,

路由器有多种不同的接口, 有的路由器只提供固定接口, 有的可通过模块改变接口的种类和数量。局域网接口、广域网接口、控制台接口和辅助接口用来连接配置路由器的设备。控制台接口 (Console 接口) 可直接连接计算机终端。辅助接口 (AUI 接口) 可通过 Modem 使远程终端与路由器通信, 实现路由器的远程管理和配置。

IOS (路由器操作系统)

配置文件: 运行配置文件 (Running-Configuration) 启动配置文件
(Startup-Configuration)

19. IPV4 和 IPV6 的区别

(1) IPV6 地址长度为 128 比特, 地址增大了 296 倍;

(2) 灵活的 IP 报文头部格式。使用一系列固定格式的扩展头部取代了 IPV4 中可变长度的选项字段。IPV6 中选项部分的出现方式也有所变化, 使路由器可以简单路过选项而不做任何处理, 加快了报文处理速度。

(3) IPV6 简化了报文头部格式, 字段只有 7 个, 加快报文转发, 提高了吞吐量;

(4) 提高安全性。身份认证和隐私权是 IPV6 的关键特性。

(5) 支持更多的服务类型;

(6) 允许协议继续演变, 增加新的功能, 使之适应未来技术的发展。

20. TCP 的拥塞控制与流量控制的功能和区别?

拥塞控制: 防止过多的数据注入到网络中, 这样可以使网络中的路由器或链路不致过载。

拥塞控制所要做的都有一个前提: 网络能够承受现有的网络负荷。拥塞控制是一个全局性的过程, 涉及到所有的主机、路由器, 以及与降低网络传输性能有关的所有因素。

流量控制: 指点对点通信量的控制, 是端到端的问题。流量控制所要做的就是抑制发送端发送数据的速率, 以便使接收端来得及接收。

21. PPP 协议?

即是点对点协议, 就是用户计算机和 ISP(InternetServiceProvider)进行通信时所使用的数据链路层协议。HDLC (高级数据链路控制) 也是数据链路层的协议。

22. 集线器, 路由器和交换机有什么区别?

集线器工作在第一层 (即物理层), 它没有智能处理能力, 对它来说, 数据只是电流而已, 当一个端口的电流传到集线器中时, 它只是简单地将电流传送到其他端口, 至于其他端口连接的计算机接收不接收这些数据, 它就不管了。

交换机工作在第二层 (即数据链路层), 它要比集线器智能一些, 对它来说, 网络上的数据就是 MAC 地址的集合, 它能分辨出帧中的源 MAC 地址和目的 MAC 地址, 因此可以在任意两个端口间建立联系, 但是交换机并不懂得 IP 地址, 它只知道 MAC 地址。

路由器工作在第三层 (即网络层), 它比交换机还要 “聪明” 一些, 它能理解数据中的 IP 地址, 如果它接收到一个数据包, 就检查其中的 IP 地址, 如果目标地址是本地网络的就理会, 如果是其他网络的, 就将数据包转发出本地网络。

23. P2P 网络编程的特点

P2P (对等网络, 是一种有别于传统 C/S 客户/服务器式的分布式网络) 直接将人们联系起来, 让人们通过互联网直接交互。P2P 使得网络上的沟通变得容易、更直接共享和交互, 真正地消除中间商。P2P 就是人可以直接连接到其他用户的计算机、交换文件, 而不是像过去那样连接到服务器去浏览与下载。

在所有的 P2P 应用中, 对等节点首先必须能够彼此发现对方, 一旦能够找到提供 P2P 服务的计算机节点, 就可以直接与它通信。例如, 计算机 A 要下载某个 MP3 文件 x, 首先需要发现拥有文件 x 的 P2P 计算机节点, 而后直接和该计算机节点通信, 完成文件传输。

P2P 应用程序应该包括三个阶段：发现，连接和通信。发现阶段负责动态定位对等节点的网络位置连接阶段负责在对等节点之间建立网络连接而通信阶段负责在对等节点之间传输数据。

24. DNS 的递归查询与迭代查询

递归查询:一般客户机和服务器之间属递归查询,即当客户机向 DNS 服务器发出请求后,若 DNS 服务器本身不能解析,则会向另外的 DNS 服务器发出查询请求,得到结果后转交给客户机;

迭代查询(反复查询):一般 DNS 服务器之间属迭代查询,如:若 DNS2 不能响应 DNS1 的请求,则它会将 DNS3 的 IP 给 DNS2,以便其再向 DNS3 发出请求;举例:比如学生问老师一个问题,王老师告诉他答案这之间的叫递归查询。这期间也许王老师也不会,这时王老师问张老师,这之间的查询叫迭代查询!

25. ARP 协议的过程

ARP 协议是“AddressResolutionProtocol”(地址解析协议)的缩写。在局域网中,网络中实际传输的是“帧”,帧里面是有目标主机的 MAC 地址的。在以太网中,一个主机要和另一个主机进行直接通信,必须要知道目标主机的 MAC 地址。但这个目标 MAC 地址是如何获得的呢?它就是通过地址解析协议获得的。所谓“地址解析”就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程ARP 协议的基本功能就是通过目标设备的 IP 地址,查询目标设备的 MAC 地址,以保证通信的顺利进行。

26. 计算机网络的接入类型都有哪些?

局域网、城域网、广域网和互联网四种

27. 中继器,集线器,交换机,网桥,网关,路由器的功能作用,区别到底是什么?

中继器:物理层,适用于完全相同的两类网络的互连,主要功能是通过数据信号的重新发送或者转发,来扩大网络传输的距离。

集线器:物理层,基本功能信息分发,它把一个端口接收的所有信号向所有端口分发出去。一些集线器在分发之前将弱信号重新生成。

中继器与集线器的区别:区别在于集线器能够提供多端口服务,也称为多口中继器。

网桥:数据链路层,网桥(Bridge)像一个聪明的中继器,网桥是一种对帧进行转发的技术,根据 MAC 分区块,可隔离碰撞。网桥将网络的多个网段在数据链路层连接起来。

28. DNS 迭代和递归区别是什么

迭代:本地域名服务器和相关服务器都有信息交互;

递归:各域名服务器之间单线信息交互(现场举例说明最佳)

29. IP 和 MAC 特点

MAC 是数据链路层和物理层使用的地址;

IP 是网络层及以上各层使用的地址,是一种逻辑地址;

IP 地址存放在 IP 数据报的首部,MAC 地址存放在 MAC 帧首部,IP 数据报是 MAC 帧

的数据。

30. 多路复用与多路分解

多路复用：从源主机的不同套接字中收集数据块，并为每个数据块封装上首部信息（这将在多路分解时使用）从而生成报文段，然后将报文段传递到网络层的工作称为多路复用。

多路分解：将运输层报文段中的数据交付到正确的套接字的工作称为多路分解。

31. 计算机网络接入方式？举例说明。

ADSL—电话线；光纤同轴混合网（HFC）--有线电视网；FTTX—光纤到户。

32. 什么是虚电路网络，什么是数据报网络。

虚电路网络：逻辑信道构成的非实体电路---TCP；

数据报网络：单独发送，单独处理，无须虚电路链接---UDP。

33. 简述 C/S 和 P2P 混合结构的特点

网络上每台主机既是服务器端又是客户端。

34. 公司配置域名服务器，你是网络管理员该如何配置

-- 代理商注册域名，--域名解析--

域名申请成功之后首先需要做域名解析。点 DNS 解析管理，然后增加 IP，增加别名以及邮件 MX 记录。先增加 IP。如想要实现去掉 3W 的顶级域名亦可访问网站，除了要在空间里绑定不加 3W 的域名外，还要解析，主机名为空

35. 计算机网络：有哪几种校验算法

---- 循环冗余 CRC 。 奇偶校验

36. 子网掩码和默认网关是什么及作用？

-- 将某个 IP 地址划分成网络地址和主机地址两部分 --一个网络通向其他网络的 IP 地址，实现这两个网络之间的通信，则必须通过网关

37. 网络中数据的分片与重组发生什么时候

-- 大于其要转发的网络的 MTU 的数据报时

38. 如何实现 IPV4 和 IPV6 的互通；

IPV4（32）、IPV6（128）、MAC（48）

双协议栈、隧道技术。

39. UDP 和 TCP 套接字编程的区别

Socket 有两种主要的操作方式：面向连接的和无连接的。无连接的操作使用 UDP 数据报协议，这个操作不需要连接一个目的的 socket，它只是简单地投出数据报，快速高效，但缺少数据安全性。面向连接的操作使用 TCP 协议，一个这个模式的 socket 必须在发送数据之前与目的地的 socket 取得一个连接，一旦连接建立了，socket 就可以使用一个流接口：打开-读-写-关闭，所有的发送的信息都会在另一端以同样的顺序被接收，面向连接的操作比无连接的操作效率要低，但数据的安全性更高。基于 TCP 的

socket 编程是采用的流式套接字(SOCK_STREAM)。基于 UDP 采用的数据报套接字(SOCK_DGRAM)。

流式套接字的设计是针对面向连接的网络应用，在数据传输之前需要预先建立连接，在数据传输过程中需要维持连接，在数据传输结束后需要释放连接。由于采用校验和、确认与超时等差错控制手段，因此流式套接字可以保证数据传输的正确性。

数据报套接字(SOCK_DGRAM)提供无连接的、不可靠的数据传输服务，实际上它是基于 TCP/IP 协议族中的 UDP 协议实现的。数据报套接字提供无序、有差错与有重复的数据流服务。数据报套接字的设计是针对无连接的网络应用，在数据传输之前不需要预先建立连接。由于只采用很有限的差错控制手段，因此数据报套接字无法保证数据传输的正确性。

40. 发送序号和确认序号的作用

TCP 协议工作在 OSI 的传输层，是一种可靠的面向连接的数据流协议，TCP 之所以可靠，是因为它保证了传送数据包的顺序。顺序是用一个序列号来保证的。响应包内也包括一个序列号，表示接收方准备好这个序列号的包。在 TCP 传送一个数据包时，它会把这个数据包放入重发队列中，同时启动计时器，如果收到了关于这个包的确认信息，便将此数据包从队列中删除，如果在计时器超时的时候仍然没有收到确认信息，则需要重新发送该数据包。另外，TCP 通过数据分段中的序列号来保证所有传输的数据可以按照正常的顺序进行重组，从而保障数据传输的完整。

41. arp 协议过程

ARP (Address Resolution Protocol, 地址解析协议) 是一个位于 TCP/IP 协议栈中的底层协议，对应于数据链路层，负责将某个 IP 地址解析成对应的 MAC 地址。

ARP 协议的基本功能就是通过目标设备的 IP 地址，查询目标设备的 MAC 地址，以保证通信的进行。

ARP (AddressResolutionProtocol) 是地址解析协议，是一种将 IP 地址转化成物理地址的协议。从 IP 地址到物理地址的映射有两种方式：表格方式和非表格方式。ARP 具体说来就是将网络层 (IP 层，也就是相当于 OSI 的第三层) 地址解析为数据连接层 (MAC 层，也就是相当于 OSI 的第二层) 的 MAC 地址。

在局域网中，当主机或其它网络设备有数据要发送给另一个主机或设备时，它必须知道对方的网络层地址 (即 IP 地址)。但是仅仅有 IP 地址是不够的，因为 IP 数据报文必须封装成帧才能通过物理网络发送，因此发送站还必须有接收站的物理地址，所以需要有一个从 IP 地址到物理地址的映射。APR 就是实现这个功能的协议。

假设主机 A 和 B 在同一个网段，主机 A 要向主机 B 发送信息。具体的地址解析过程如下

(1) 主机 A 首先查看自己的 ARP 表，确定其中是否包含有主机 B 对应的 ARP 表项。如果找到了对应的 MAC 地址，则主机 A 直接利用 ARP 表中的 MAC 地址，对 IP 数据包

进行帧封装，并将数据包发送给主机 B。

(2)如果主机 A 在 ARP 表中找不到对应的 MAC 地址，则将缓存该数据报文，然后以广播方式发送一个 ARP 请求报文。ARP 请求报文中的发送端 IP 地址和发送端 MAC 地址为主机 A 的 IP 地址和 MAC 地址，目标 IP 地址和目标 MAC 地址为主机 B 的 IP 地址和全 0 的 MAC 地址。由于 ARP 请求报文以广播方式发送，该网段上的所有主机都可以接收到该请求，但只有被请求的主机（即主机 B）会对该请求进行处理。

(3)主机 B 比较自己的 IP 地址和 ARP 请求报文中的目标 IP 地址，当两者相同时进行如下处理：将 ARP 请求报文中的发送端（即主机 A）的 IP 地址和 MAC 地址存入自己的 ARP 表中。之后以单播方式发送 ARP 响应报文给主机 A，其中包含了自己的 MAC 地址。

(4)主机 A 收到 ARP 响应报文后，将主机 B 的 MAC 地址加入到自己的 ARP 表中以用于后续报文的转发，同时将 IP 数据包进行封装后发送出去。

当主机 A 和主机 B 不在同一网段时，主机 A 就会先向网关发出 ARP 请求，ARP 请求报文中的目标 IP 地址为网关的 IP 地址。当主机 A 从收到的响应报文中获得网关的 MAC 地址后，将报文封装并发给网关。如果网关没有主机 B 的 ARP 表项，网关会广播 ARP 请求，目标 IP 地址为主机 B 的 IP 地址，当网关从收到的响应报文中获得主机 B 的 MAC 地址后，就可以将报文发给主机 B；如果网关已经有主机 B 的 ARP 表项，网关直接把报文发给主机 B。

42. TCP 建立连接

TCP 建立连接，也就是我们常说的三次握手，它需要三步完成。在 TCP 的三次握手中，发送第一个 SYN 的一端执行的是主动打开。而接收这个 SYN 并发回下一个 SYN 的另一端执行的是被动打开。

第 1 步：客户端向服务器发送一个同步数据包请求建立连接，该数据包中，初始序列号 (ISN) 是客户端随机产生的一个值，确认号是 0；

第 2 步：服务器收到这个同步请求数据包后，会对客户端进行一个同步确认。这个数据包中，序列号 (ISN) 是服务器随机产生的一个值，确认号是客户端的初始序列号 + 1；

第 3 步：客户端收到这个同步确认数据包后，再对服务器进行一个确认。该数据包中，序列号是上一个同步请求数据包中的确认号值，确认号是服务器的初始序列号 + 1。

注意：因为一个 SYN 将占用一个序号，所以要加 1。

操作系统

1. 什么是程序局部性，为什么会有程序的空间局部性？

程序局部性是指程序在运行时呈现出局部性规律，在一段时间间隔内，程序的执行是局限在某个部份，所访问的存储空间也只局限在某个区域。空间局部性是指若一个存储单元被访问，那么它附近的单元也可能被访问，这是由于程序的顺序执行引起的。

2. 为了实现重定位需要哪些硬件？

最简单的方式是在系统中增设一个重定位寄存器，用来存放正在执行作业的内存地址，每次访问数据时，由硬件自动将相对地址与重定位寄存器中的起始地址相加，形成实际的物理地址。当然在分页式与分段式系统中，实现重定位需要地址变换机构，以及快表等硬件的支持。

3. 在交互式系统中，非剥夺是不是一个好的策略？为什么？

非剥夺方式：分派程序一旦把处理机分配给某进程后便让它一直运行下去，直到进程完成或发生某事件而阻塞时，才把处理机分配给另一个进程。

剥夺方式：当一个进程正在运行时，系统可以基于某种原则，剥夺已分配给它的处理机，将之分配给其它进程。剥夺原则有：优先权原则、短进程、优先原则、时间片原则。

在分时系统中不剥夺并不是一个好的策略。因为，在分时系统中，除了交互性以外，及时性是很重要的性能因素。当一个作业被阻塞后，CPU 就完全空闲了，别的用户的及时性就无法保证了，而完全可以把这些时间分配给别的作业运行。以提高整体的吞吐量。

4. 系统调用的定义

系统调用是 OS 与应用程序之间的接口，它是用户程序取得 OS 服务的惟一途径。它与一般的过程调用的区别：

运行在不同的系统状态。调用程序在运行在用户态，而被调用的程序运行在系统态，通过软中断机制，先由用户态转为系统态，经分析后，才能转向相应的系统调用处理子程序；

一般的过程调用返回后继续执行，但对系统调用，当调用的进程仍具有最高优先权时，才返回到调用进程继续处理；否则只能等被重新调度；

5. 死锁是什么？

指多个有关进程由于争夺资源而造成的一种僵局，在无外力的情况下这些进程都将无法再向前推进的状态。

6. CPU 不执行程序的时候干什么？

当没有被任何程序使用的时候，计算机的处理器被认为是空闲的。当然下面提到的空闲任务不在此列中。

当有程序利用 CPU 空闲时间的时候，就意味着它以较低的优先权运行着，以便不会影响那有正常优先权的程序运行。一般来讲，这会引入 CPU 消耗更多的电能，而大多数的现代 CPU 当它们空闲的时候是能够进入省电模式的。大多数操作系统都有个空闲任务，它是一个特殊的任务。仅当 CPU 无事可做的时候由操作系统调度器载入它。在现代的处理器中，

HLT 停机指令节省了大量的电能与热量，而空闲任务几乎总是由一个重复执行 HLT 停机指令的循环组成。

7. 举例解释一下同步和互斥

—— 同步 —— 协同完成， 互斥 —— 保证临界资源独占访问

同步表现为直接制约，如管道通信，一个进程写，一个进程读，它们是相互制约的。

互斥表现为间接制约，比如多个进程同时请求打印机(没使用 SPOOLing 技术)多个进程同时请求一张网卡发送数据包等。

8. 解释一下管程

管程是由一组局部变量对局部变量进行操作的一组过程和对局部变量进行初始化的语句序列组成。引入它的目的是因为 Wait/Signal 操作太过分散，对它的维护很麻烦且容易造成死锁管程的特点是：管程的过程只能访问管程的局部变量，管程的局部变量只能由其过程来访问；任何时刻只能有一个进程进入管程执行；进程只能通管程提供的过程入口进入管程；

9. 在可变分区管理中，需要哪些硬件机制

采用可变分区方式管理时，一般均采用动态重定位方式装入作业。地址变换要靠硬件支持，主要是两个寄存器：基址寄存器和限长寄存器，限长寄存器存放作业所占分区的长度，基址寄存器则存放作业所占分区的起始地址，这两个值确定了一个分区的位置和大小。转换时根据逻辑地址与限长值比较，如果不超过这个值，表示访问地址合法，再加上基址寄存器中的值就得到了绝对地址了，否则形成“地址越界”中断。

10. 中断和陷入有什么异同？

外中断时指来自处理机和内存外部的中断，如 I/O 中断、定时器中断、外部信号中断等。狭义上也叫中断；

内中断主要指在处理机和内存内部产生的中断，也称陷入，如校验错、页面失效、溢出、除数为零等；

中断和陷入的主要区别：

(1) 陷入通常由处理机正在执行的现行指令引起，而中断则是由与现行指令无关的中断源引起的。

(2) 陷入处理程序提供的服务为当前进程所用，而中断处理程序提供的服务则不是为了当前进程的。

(3) CPU 在执行完一条指令之后，下一条指令开始之前响应中断，而在一条指令执行中也可以响应陷阱。

11. 计算机网络和分布式计算机系统的区别？

两者在计算机硬件连接、系统拓扑结构和通信控制等方面基本都是一样的，它们都具有通信和资源共享的功能。

区别关键在于：分布式计算机系统是在分布式计算机操作系统支持下，进行分布式数据库处理的，也就是说各互联的计算机可以互相协调工作，共同完成一项任务，多台计算机上并行运行。且具有透明性，用户不知道数据、资源的具体位置，整个网络中所有计算机就像是一台计算机一样；而计算机网络却不具备这种功能，计算机网络系统中的各

计算机通常是各自独立进行工作的。

12. 为什么引入多道程序技术？

因为引入多道程序技术后，可以进一步提高了 CPU 利用率(阻塞)，提高内存和 I/O 设备利用率(小作业把内存浪费了)，增加系统吞吐量(两都提高后的必然)。

13. 什么是管态？什么是目态？它们与进程运行状态的关系是什么？

CPU 交替执行操作系统程序和用户程序。管态又叫特权态，系统态或核心态。CPU 在管态下可以执行指令系统的全集。通常，操作系统在管态下运行。

目态又叫常态或用户态。机器处于目态时，程序只能执行非特权指令。用户程序只能在目态下运行，如果用户程序在目态下执行特权指令，硬件将发生中断，由操作系统获得控制，特权指令执行被禁止，这样可以防止用户程序有意或无意的破坏系统。

14. n 个任务一个 cup，阻塞，运行，就绪的进程最多有多少个？

阻塞 n 个；运行 1 个；就绪 n-1 个；

15. 父子进程是否可以并发运行

可以并发运行，因为不管是父进程还是子进程都是系统分配资源的最小单位，都是独立的进程，是可以并发运行的，跟进程和线程的关系不一样。

16. 缓冲的定义，为什么引入

引入缓冲技术主要是为了解决设备速度差异引起的效率问题。如果两个设备速度相差很大，快速设备总是需要等待慢速设备，快速设备效率低下。在设备之间加上缓冲区，慢速设备将数据送入缓冲区时，快速设备不需等待，可以处理其他任务，直到缓冲区满时才通过中断通知快速设备处理缓冲区中的数据，将提高设备工作的独立性，提高快速设备的利用率。

(1) 缓和 CPU 与 IO 设备间速度不匹配的矛盾

(2) 减少对 CPU 的中断频率，放宽对 CPU 中断响应时间的限制

(3) 提高 CPU 和 IO 设备之间的并行性。

17. 进程调度算法

1 先来先服务 FCFS 2 短作业优先 (SJF) 3 高优先权优先调度算法

4 高响应比调度算法 5 时间片轮转法 6 多级反馈队列调度

18. 进程三个状态的转换

就绪态，当进程已分配到除 cpu 以外的所以必要资源后只要能再获得处理机便可立即执行这时的进程状态转为就绪模式。执行状态，指进程已获得处理机其程序正在执行。阻塞状态，进程因发生某件事如请求 io 而暂停执行的状态。

19. 设备驱动程序是否属于操作系统，他的作用是什么？

不是，驱动程序是另外安装的软件，是操作系统控制并且和硬件之间通讯的桥梁（程序）

20. 线程、进程、程序和任务的区别？

任务是最抽象的，是一个一般性的术语，指由软件完成的一个活动。一个任务既可以是一个进程，也可以是一个线程。简而言之，它指的是一系列共同达到某一目的的操作。例

如，读取数据并将数据放入内存中。这个任务可以作为一个进程来实现，也可以作为一个线程（或作为一个中断任务）来实现。

进程常常被定义为程序的执行。可以把一个进程看成是一个独立的程序，在内存中有其完备的数据空间和代码空间。一个进程所拥有的数据和变量只属于它自己。

线程则是某一进程中一路单独运行的程序。也就是说，线程存在于进程之中。一个进程由一个或多个线程构成，各线程共享相同的代码和全局数据，但各有其自己的堆栈。由于堆栈是每个线程一个，所以局部变量对每一线程来说是私有的。由于所有线程共享同样的代码和全局数据，它们比进程更紧密，比单独的进程间更趋向于相互作用，线程间的相互作用更容易些，因为它们本身就有某些供通信用的共享内存：进程的全局数据进程的全局数据进程的全局数据进程的全局数据

程序只是一组指令的有序集合，它本身没有任何运行的含义，它只是一个静态的实体。

21. 处理中断的过程

请求中断→响应中断→关闭中断→保留断点→中断源识别→保护现场→中断服务子程序→恢复现场→中断返回

1. 请求中断

当某一中断源需要 CPU 为其进行中断服务时，就输出中断请求信号，使中断控制系统的中断请求触发器置位，向 CPU 请求中断。系统要求中断请求信号一直保持到 CPU 对其进行中断响应为止。

2. 中断响应

CPU 对系统内部中断源提出的中断请求必须响应，而且自动取得中断服务子程序的入口地址，执行中断服务子程序。对于外部中断，CPU 在执行当前指令的最后一个时钟周期去查询 INTR 引脚，若查询到中断请求信号有效，同时在系统开中断即 $IF=1$ 的情况下，CPU 向发出中断请求的外设回送一个低电平有效的中断应答信号，作为对中断请求 INTR 的应答，系统自动进入中断响应周期。

3. 关闭中断

CPU 响应中断后，输出中断响应信号，自动将状态标志寄存器 FR 或 EFR 的内容压入堆栈保护起来，然后将 FR 或 EFR 中的中断标志位 IF 与陷阱标志位 TF 清零，从而自动关闭外部硬件中断。因为 CPU 刚进入中断时要保护现场，主要涉及堆栈操作，此时不能再响应中断，否则将造成系统混乱。

4. 保护断点

保护断点就是将 CS 和 IP/EIP 的当前内容压入堆栈保存，以便中断处理完毕后能返回被中断的原程序继续执行，这一过程也是由 CPU 自动完成。

5. 中断源识别

当系统中有多个中断源时，一旦有中断请求，CPU 必须确定是哪一个中断源提出的中断请求，并由中断控制器给出中断服务子程序的入口地址装入 CS 与 IP/EIP 两个寄

存器。**CPU** 转入相应的中断服务子程序开始执行。

6. 保护现场

主程序和中断服务子程序都要使用 **CPU** 内部寄存器等资源,为使中断处理程序不破坏主程序中寄存器的内容,应先将断点处各寄存器的内容压入堆栈保护起来,再进入的中断处理。现场保护是由用户使用 **PUSH** 指令来实现的。

7. 中断服务

中断服务是执行中断的主体部分,不同的中断请求,有各自不同的中断服务内容,需要根据中断源所要完成的功能,事先编写相应的中断服务子程序存入内存,等待中断请求响应后调用执行。

8. 恢复现场

当中断处理完毕后,用户通过 **POP** 指令将保存在堆栈中的各个寄存器的内容弹出,即恢复主程序断点处寄存器的原值。

9. 中断返回

在中断服务子程序的最后要安排一条中断返回指令 **IRET**,执行该指令,系统自动将堆栈内保存的 **IP/EIP** 和 **CS** 值弹出,从而恢复主程序断点处的地址值,同时还自动恢复标志寄存器 **FR** 或 **EFR** 的内容,使 **CPU** 转到被中断的程序中继续执行。

22. 分页、分段、断页式的特点,为什么要引入?

分页是为了提高内存的利用率,提高计算机性能,且分页通过硬件机制来实现,对用户完全透明。分段是为了方便编程,信息保护和共享、动态增长及动态链接等多方面的需要。断页式是两者的结合。

23. 计算机系统怎样实现存储保护?

1. 防止地址越界(对进程所产生的地址必须加以检查,发生越界时产生中断,由操作系统进行相应处理)
2. 防止操作越权(对属于自己区域的信息,可读可写;对公共区域中允许共享的信息或获得授权可使用的信息,可读而不可修改;对未授权使用信息,不可读,不可写)

24. MMU?

MMU 是 **MemoryManagementUnit** 的缩写中文名是内存管理单元它是中央处理器**CPU**)中用来管理虚拟存储器、物理存储器的控制线路,同时也负责虚拟地址映射为物理地址,以及提供硬件机制的内存访问授权。

25. 实时系统的分类

我们可以将通常人们所说的实时系统可分为两种类型:软实时系统和硬实时系统。在软实时系统中,系统的宗旨是使各个任务运行得越快越好,并不要求限定某一任务必须在多长时间内完成,这类系统并不具有真正实时操作的要求,其对操作系统的实时性要求较低,一般的实时操作系统均能满足要求。在硬实时系统中,各任务不仅要执行无误而且要做到准时,其对操作系统的实时性要求较严格,在系统设计时要非常注意实时性性能的满足,对实时操作系统的伪实时的特征应进行仔细分析。大多数实际应用的实时系统是以上二者的结合。

26. 调度的基本准则

1. CPU 利用率 2 系统吞吐量 3. 周转时间 4. 等待时间 5. 响应时间

27. 多线程，真的提高了效率吗？

多线程效率，我认为未必会高，而且有时候相反会低。多线程并不是为了提高效率，而是不必等待可以并行执行多条数据。可以这么想我们通过 xp 系统复制文件。你可以复制一份文件这叫单线程，但是你要等这个复制完了才能复制另一份文件，而且不能多复制。这样很难受，所以你可以选择多复制文件，这就是多线程。但复制多份文件用的时间未必会比一份一份文件所用时间少。只是它合理利用了时间进行了多个操作。如果是买票系统就会用到多线程。买票是同时进行的，如果一个用户一个用户等下去不是办法，所以可以多个用户同时买票，效率也就提高了。这里的效率不是执行的效率而是时间的合理利用，多线程同时进行。

28. 中断的作用

答：中断是计算机系统结构一个重要的组成部分。在中断机制中的硬件部分(中断装置)的作用就是在 CPU 每执行完一条指令后，判别是否有事件发生，如果没有事件发生，CPU 继续执行；若有事件发生，中断装置中断原先占用 CPU 的程序的执行，把被中断程序的断点保存起来，让操作系统的处理服务程序占用 CPU 对事件进行处理，处理完后，再让被中断的程序继续占用 CPU 执行下去。（所以中断装置的作用总的来说就是使操作系统可以控制各个程序的执行）

29. 磁盘调度算法

1. 先来先服务算法 2. 最短寻找时间优先算法 3. 扫描算法 4. 循环扫描算法

30. DMA 的优先级为什么比 CPU 的优先级高？

因为 DMA 请求得不到及时响应，I/O 传输数据可能会丢失

31. 虚拟内存容量由什么决定？

虚拟存储区的容量与物理主存大小无关，而受限于计算机的地址结构和可用磁盘容量。

32. RAID 的工作原理。

RAID（独立磁盘冗余阵列（Redundant Array of Independent Disks））通过条带化存储和奇偶校验两个措施来实现其冗余和容错的目标条带化存储意味着可以一次写入一个数据块的方式将文件写入多个磁盘。条带化存储技术将数据分开写入多个驱动器，从而提高数据传输速率并缩短磁盘处理总时间。这种系统非常适用于交易处理、但可靠性却很差，因为系统的可靠性等于最差的单个驱动器的可靠性。

奇偶校验通过在传输后对所有数据进行冗余校验可以确保数据的有效性利用奇偶校验，当 RAID 系统的一个磁盘发生故障时，其它磁盘能够重建该故障磁盘。在这两种情况中，这些功能对于操作系统都是透明的。由磁盘阵列控制器（DAC）进行条带化存储和奇偶校验控制。

33. 操作系统在发生缺页中断时，是否一定会执行页面淘汰算法

-- 不一定，若所缺页系统中有，则调入，缺页地址错误则结束进程。

34. 解决死锁的三个方法

预防 避免 检测

35. 文件关闭的过程

第一步：通过检索文件目录来找到指定文件；

第二步：调用“close（）”关闭文件，把该文件从打开文件表中的表目尚删除掉。

36. 在基于优先级调度的系统中，当一个进程正在执行时，这时来了一个高优先级的进程，是否会立即终止现行的进程？为什么？

不一定，首先非抢占式系统中，要等待当前进程执行完毕才可以抢占 CPU，其次在抢占式系统中，优先权检测也是在每条指令执行完毕才检测，所以不一定会立即终止。

37. 文件连续分配和离散分配的优缺点。

连续分配优点：顺序访问方便；顺序访问速度快；支持直接存取。

连续分配缺点：要求有连续的空间；必须事先知道文件长度。

离散分配优点：提高外存利用率，减少碎片产生；无须事先知道文件大小。

离散分配缺点：访问速度慢；索引分配可以直接存取，但是索引表占用额外空间；链接分配只能顺序访问。

38. 操作系统：文件的软链接和硬链接的概念，删除软链接或硬链接后文件是否还能被访问，WEB 链接属于什么链接。

软链接：符号连接（Symbolic Link），软链接文件有类似于 Windows 的快捷方式。它实际上是一个特殊的文件。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。删除后不影响文件访问。

硬链接：通过索引节点来链接；删除后不影响源文件访问

WEB 链接属于软链接。

39. 线程为什么能提高系统效率

线程可以减少并发执行的开销，并发性较进程进一步提高。线程是轻量级的进程，线程间的并发和切换，资源开销远远小于进程，因此可以提高系统效率。

40. 2. 生产者和消费者系统中，互斥和同步的含义

-- 互斥 间接制约 有产品才可以消费

同步 直接制约 有空间才可以生产

41. 独立磁盘冗余阵列的工作原理

-- 是把多个相对便宜的小磁盘组合起来，成为一个磁盘组

1、增强了速度

2、扩容了存储能力(以及更多的便利)

3、可高效恢复磁盘

42. 动态页面调度时，当发生缺页时是否立即调出一页也同时调入一页

不一定，系统分配的页数如若占用完毕，此时需要调出一页，但是如若页数有剩余的话，此时只需调入页面即可不涉及调出页面操作。

43. 操作系统是多线程技术是什么

操作系统中提出多线程技术主要是为了提高系统资源利用率，在单核计算机中，多线程就是在内存中同时存在对各线程并发执行，在微观上是顺序占用处理器执行；在多核计算机中，多线程可以实现多个处理器同时处理多个线程的操作，实现真正意义上的并发执行。

44. 操作系统：读者/写者问题在实际中有何应用

文件操作，过独木桥问题

45. 多线程的优点

提高资源利用率，程序响应快，代码易于设计。

46. CPU 工作在内核态与用户态有什么区别？

内核态：CPU 可以访问内存所有数据，包括外围设备，例如硬盘，网卡。CPU 也可以将自己从一个程序切换到另一个程序

用户态：只能受限的访问内存，且不允许访问外围设备。占用 CPU 的能力被剥夺，CPU 资源可以被其他程序获取

47. 段页式系统的原理及优点

段页式系统的基本原理，是基本分段存储管理方式和基本分页存储管理方式原理的结合，即先将用户程序分成若干个段，再把每个段分成若干个页，并为每一个段赋予一个段名。

优点：包含分页和分段的优点，既有分页系统的资源利用率高，碎片少的优点，又有分段系统方便用户、实现信息共享和保护的优点。

48. 什么是前台作业，什么是后台作业？有什么区别？

在批处理兼分时的系统中，往往把由分时系统控制的作业称“前台”作业，而批处理系统控制的作业称“后台”作业。所以用户以交互方式调试好的程序转向批处理自动控制执行的过程，实际上是把前台作业转换成后台作业的过程。前台作业优先权高，先处理前台作业。