

面试例题6：Please design a test plan and test case to test a simplified mobile phone: (写一个测试用例和测试方案来测试手机) [中国台湾著名杀毒软件公司Q2009年5月笔试题]

```
including following function (包括如下功能)
Calling
SMS
Address book
```

答案：

对于 Calling:

- 1 是否有拨打电话这个功能，能否接通电话
- 2 拨打正常号码
- 3 拨打不正常号码，是否有相应提示

对于 SMS:

- 1 是否有发送、接收短信功能
- 2 输入正常号码发送短信
- 3 输入位数不对等不正常号码是否有提示
- 4 发送短信能否保存在发件箱，能否保存草稿，短信能否删除等

对于 Address book:

- 1 是否有电话本这个功能
- 2 新建一个联系人，联系人信息为空是否有提示
- 3 新建联系人与已有联系人姓名或电话号码等信息重复是否有提示
- 4 删除联系人是否成功，删除时是否有提示信息

15.4 白盒测试

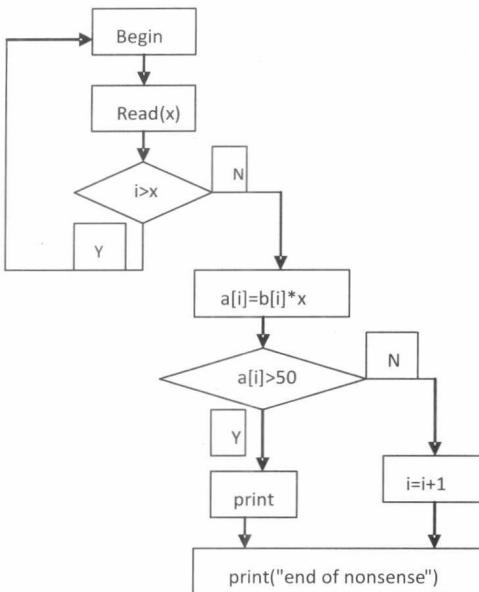
面试例题1：The following is a pseudocode fragment that has no redeeming feature except for the purpose of this question. (下面语句是一段伪代码。) [美国某软件公司2010年7月笔试题]

```
module nonsense[]
/*a[] and b[] are global variables */
begin
int i,x
i=1
read(x)
while(i <x)do begin
a[i]=b[i]*x
```

```
if a[i]>50 then
print("array a is over the limit")
else
print("ok")
i=i+1
end
print("end of nonsense")
end
```

Please list test cases to cover all branches in this fragment. (请设计一个测试用例涵盖该段代码各个分支。)

答案：白盒测试有几种测试方法：条件覆盖、路径覆盖、语句覆盖、分支覆盖。其中分支覆盖又称判定覆盖，使得程序中每个判断的取真分支和取假分支至少经历一次，即判断的真假均曾被满足。本题目的逻辑分支如下图所示。



根据判定覆盖准则，测试用例涵盖该段代码各个分支如下：

```

1 i>x
2 i<x && a[i]>50
3 i<x && a[i]<50
4 i=x && a[i]>50
5 i=x && a[i]<50
  
```

面试例题 2：下面是一道求闰年的 C 程序，这里要求你用基本路径法设计出测试用例。证明程序每一条可执行语句在测试过程中至少执行一次。[中国某互联网公司 2010 年 6 月笔试题]

```

int IsLeap(int year)
{
    if(year%4==0)          //编号 1
    {
        if(year%100==0)    //编号 2
            if(year%400==0) //编号 3
                leap=1;      //编号 5
            else            //编号 6
        else                //编号 7
    }
  
```

```

        leap=0;          //编号 8
    }
}
Else                      //编号 9
    leap=1;              //编号 10
}
else                      //编号 12
    leap=0;              //编号 13
return leap;             //编号 14
  
```

问题 1：该程序的控制流圆圈复杂度 $V(G)$ 是多少，独立线性路径数是多少？

问题 2：假设输入取值范围 $1000 < year < 2001$ ，请使用基本路径测试法为 $year$ 设计测试用例，满足基本路径覆盖要求。（画设计流图及设计路径时可以利用上面的编号）

解析：白盒测试的测试方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法、域测试、符号测试、Z 路径覆盖等方法，其中运用最为广泛的是基

本路径测试法。

基本路径测试法是在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例的方法。设计出的测试用例要保证在测试中程序的每个可执行语句至少执行一次。

在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例。包括以下4个步骤和一个工具方法。

(1) 程序的控制流图：描述程序控制流的一种图示方法。

(2) 程序圈复杂度：McCabe 复杂度量。从程序的环路复杂性可导出程序基本路径集合中的独立路径条数，这是确定程序中每个可执行语句至少执行一次所必须的测试用例数目的上界。

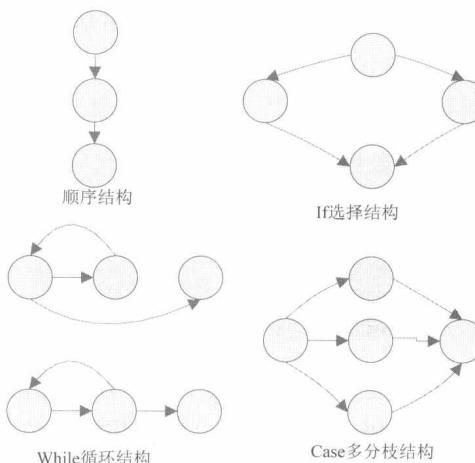
(3) 导出测试用例：根据圈复杂度和程序结构设计用例数据输入和预期结果。

(4) 准备测试用例：确保基本路径集中的每一条路径的执行。

工具方法：

- 图形矩阵：是在基本路径测试中起辅助作用的软件工具，利用它可以实现自动地确定一个基本路径集。
- 程序的控制流图：描述程序控制流的一种图示方法。

圆圈称为控制流图的一个节点，表示一个或多个无分支的语句或源程序语句，如下图所示。



例如：

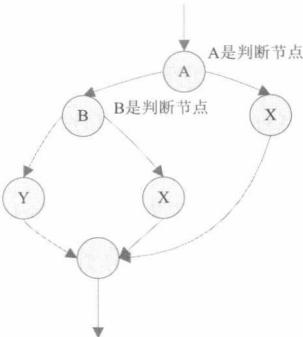
```
1 if A or B
```

```

2 X
3 else
4 Y

```

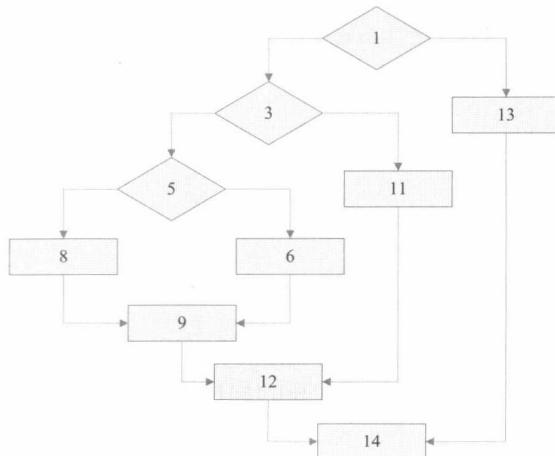
对应的逻辑如下图所示。



基本路径测试法的步骤：

第1步：画出控制流图

流程图用来描述程序控制结构。可将流程图映射到一个相应的流图（假设流程图的菱形决定框中不包含复合条件）。在流图中，每一个圆，称为流图的节点，代表一个或多个语句。一个处理方框序列和一个菱形决定框可被映射为一个节点，流图中的箭头，称为边或连接，代表控制流，类似于流程图中的箭头。一条边必须终止于一个节点，即使该节点并不代表任何语句（例如：if-else-then 结构）。由边和节点限定的范围称为区域。计算区域时应包括图外部的范围，如下图所示。



第2步：计算圈复杂度

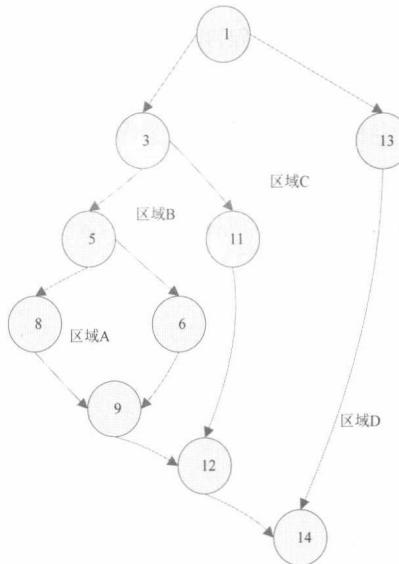
圈复杂度是一种为程序逻辑复杂性提供定量测度的软件度量，将该度量用于计算程序的基本的独立路径数目，为确保所有语句至少执行一次的测试数量的下界。独立路径必须包含一条在定义之前不曾用到的边。

有以下3种方法计算圈复杂度。

- 流图中区域的数量对应于环型的复杂性。
- 给定流图G的圈复杂度 $V(G)$ ，定义为 $V(G)=E-N+2$ ，E是流图中边的数量，N是流图中节点的数量。
- 给定流图G的圈复杂度 $V(G)$ ，定义为 $V(G)=P+1$ ，P是流图G中判定节点的数量。

对应本题图中的圈复杂度，如下图所示，计算如下：

- a. 流图中有4个区域。
- b. $V(G)=12$ 条边-10节点+2=4。
- c. $V(G)=3$ 个判定节点+1=4。



第3步：导出测试用例

根据上面的计算方法，可得出4个独立的路径（(一条独立路径是指和其他的独立路径相比，至少引入一个新处理语句或一个新判断的程序通路。 $V(G)$ 值正好等于该程序的独立路径的条数)。

路径1：1-13-14

路径2：1-3-11-12-14

路径3：1-3-5-6-9-12-14

路径4：1-3-5-8-9-12-14

根据上面的独立路径，设计输入数据，使程序分别执行到上面4条路径。[中国某互联网公司2010年6月笔试题]

答案：该程序的控制流圆圈复杂度 $V(G)$ 是 4，独立线性路径数是 4。

为了确保基本路径集中的每一条路径的执行，根据判断节点给出的条件，选择适当的数据以保证某一条路径可以被测试到，满足上面例子基本路径集的测试用例如下：

路径1：1-13-14

输入数据：1001

预期结果：leap=0；

路径2：1-3-11-12-14

输入数据：1004

预期结果：leap=1；

路径3：1-3-5-6-9-12-14

输入数据：1100

预期结果：leap=0；

路径4：1-3-5-8-9-12-14

输入数据：1600

预期结果：leap=1；

第4部分

操作系统、数据库和网络

Operating system, database, network

本部分主要介绍求职面试过程中出现的第三个重要的板块——操作
系统、数据库和网络知识。这些内容虽不是面试题目中的主流，
但仍然具有重要的意义。

第 16 章

操作 系 统

操作系统面试例题主要包括进程、线程、内存管理、垃圾回收，以及缓存等诸多方面。

16.1 进程

面试例题 1：试解释操作系统原理中的作业、进程、线程、管程各自的定义。[中国著名通讯公司 Z2009 年 10 月笔试题]

答案：作业：用户在一次解题或一个事务处理过程中要求计算机系统所做工作的集合。它包括用户程序、所需要的数据及控制命令等。作业是由一系列有序的步骤组成的。

进程：一个程序在一个数据集合上的一次运行过程。所以一个程序在不同数据集合上运行，乃至一个程序在同样数据集合上的多次运行都是不同的进程。

线程：线程是进程中的一个实体，是被系统独立调度和执行的基本单位。

管程：管程实际上是定义了一个数据结构和在该数据结构上的能为并发进程所执行的一组操作，这组操作能同步进程和改变管程中的数据。

面试例题 2：进程间的通信如何实现？[日本某著名家电/通信/IT 企业面试题]

答案：现在最常用的进程间通信的方式有信号、信号量、消息队列、共享内存。

所谓进程通信，就是不同进程之间进行一些“接触”。这种接触有简单，也有复杂。机制不同，复杂度也不一样。通信是一个广义上的意义，不仅仅指传递一些 message。它们的使用方法是基本相同的，所以只要掌握了一种使用方法，然后记住其他的使用方法就可以了。信号和信号量是不同的，它们虽然都可用来实现同步和互斥，但前者是使用信号处理器来进行的，后者是使用 P、V 操作来实现的。消息队列是比较高级的一种进程间通信方法，因为

它真的可以在进程间传送 message，连传送一个“*I seek you*”都可以。

一个消息队列可以被多个进程所共享（IPC 就是在这个基础上进行的）；如果一个进程的消息太多，一个消息队列放不下，也可以用多于一个的消息队列（不过可能管理会比较复杂）。共享消息队列的进程所发送的消息中除了 message 本身外还有一个标志，这个标志可以指明该消息将由哪个进程或者是哪类进程接受。每一个共享消息队列的进程针对这个队列也有自己的标志，可以用来声明自己的身份。

面试例题 3：在 Windows 编程中互斥器（mutex）的作用和临界区（critical section）类似，请说一下二者间的主要区别。[中国台湾某著名杀毒软件公司 2005 年面试题]

解析：多线程编程问题。

答案：两者的区别是 mutex 可以用于进程之间互斥，critical section 是线程之间的互斥。

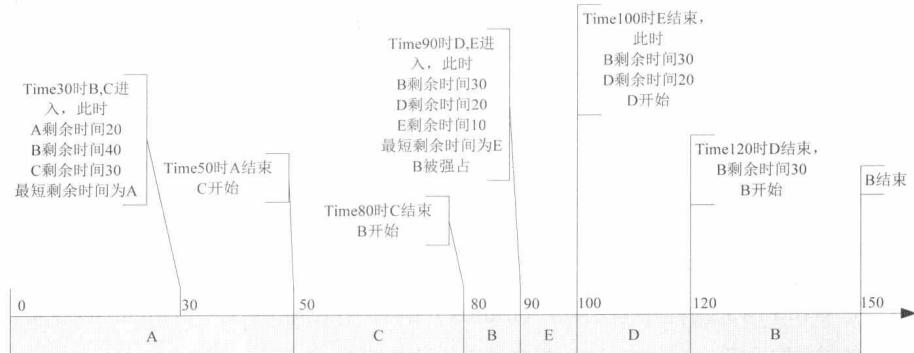
面试例题 4：At time 0, process A has arrived in the system, in that order; at time 30, both progress B and C have just arrived; at time 90, both progress D and E have also arrived. The quantum or timeslice is 10 units. (在 0 时刻，进程 A 进入系统，按照这个顺序，在 30 时刻，进程 B 和进程 C 也抵达；在 90 时刻，进程 D 和进程 E 也抵达。一个时间片是 10 个单元)

Process A requires 50 units of time in the CPU; Process B requires 40 units of time in the CPU; Process C requires 30 units of time in the CPU; Process D requires 20 units of time in the CPU; Process E requires 10 units of time in the CPU。

(进程 A 需要占用 CPU50 个单元；进程 B 需要占用 CPU40 个单元；进程 C 需要占用 CPU30 个单元；进程 D 需要占用 CPU20 个单元；进程 E 需要占用 CPU10 个单元。)

Which of the process will be the LAST to complete, if scheduling policy is preemptive SJF (Short Job First)? (如果按照短作业优先级的方法，哪个进程最后结束？) [美国著名软件公司 M2009 年 10 月笔试试题]

解析：牢记“短作业优先” = “最短剩余时间作业优先”。本题考的是可剥夺式处理机的调度问题。时间图如下所示。



答案：如果按短作业优先级的方法，进程 B 最后结束。

扩展知识

读者试着扩展思维，想想如果本题是不可剥夺式处理机的调度方法，哪个进程最后结束。

面试例题 5：For a multiple-processing OS, the way to deal with dead-lock is whether to prevent it from happening, or break it happens. Which of the following approaches belong to the way to prevent dead-lock from happening? (M) (多选：在多重处理系统中，处理死锁的办法有两种：一是防止其发生；二是发生后进行处理。下面的办法中属于防止其发生的是哪一个？) [中国台湾著名杀毒软件公司 Q2009 年 1 月笔试题]

- A. Destroy mutual condition (破坏互斥条件)
- B. Destroy non-preemptive condition (破坏不可剥夺条件)
- C. Destroy iterative waiting condition (破坏循环等待条件)
- D. To kill one process which involves in dead-lock (杀死某个激活死锁的进程)

解析：所谓 deadlocks (死锁) 是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁。

产生死锁的 4 个必要条件如下。

-互斥条件：一个资源每次只能被一个进程使用。
-请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。
-不剥夺条件：进程已获得的资源，在未使用完之前，不能强行剥夺。
-循环等待条件：若干进程之间形成一种头尾相接的循环等待资源关系。

这 4 个条件是死锁的必要条件，只要系统发生死锁，这些条件必然成立，而只要上述条件之一不满足，就不会发生死锁。

死锁的解除与预防方法如下：

理解了死锁的原因，尤其是产生死锁的 4 个必要条件，就可以最大可能地避免、预防和解除死锁。所以，在系统设计、进程调度等方面注意如何不让这 4 个必要条件成立，如何确定资源的合理分配算法，避免进程永久占据系统资源。此外，也要防止进程在处于等待状态的情况下占用资源，在系统运行过程中，对进程发出的每一个系统能够满足的资源申请进行动态检查，并根据检查结果决定是否分配资源，若分配后系统可能发生死锁，则不予分配，否则予以分配。因此，对资源的分配要给予合理的规划。

根据产生死锁的 4 个必要条件，只要使其中之一不能成立，死锁就不会出现。为此，可以采取下列 3 种预防措施：

- 采用资源静态分配策略，破坏“部分分配”条件。
- 允许进程剥夺使用其他进程占有的资源，从而破坏“不可剥夺”条件。
- 采用资源有序分配法，破坏“环路”条件。

这里注意一点：互斥条件无法被破坏。

死锁的避免不严格地限制死锁的必要条件的存在，而是系统在系统运行过程中小心地避免死锁的最终发生。避免死锁算法中最有代表性的算法是 Dijkstra E.W 于 1968 年提出的银行家算法，该算法需要检查申请者对资源的最大需求量，如果系统现存的各类资源可以满足申请者的请求，就满足申请者的请求。这样申请者就可很快完成其计算，然后释放它占用的资源，从而保证了系统中的所有进程都能完成，所以可避免死锁的发生。

在本题中选项 A 破坏互斥条件是无法做到的，B、C 正确，D 选项属于死锁事后处理操作，与题意不符。

答案：B，C。

面试例题 6：在 Linux 平台下运行 C 程序。如果 fork()函数不失败，下面哪个描述是正确的

```
#include <stdio.h>
#include <unistd.h>
int main(void)
{
    int i = 1;
    if(!fork()) i++;
}
```

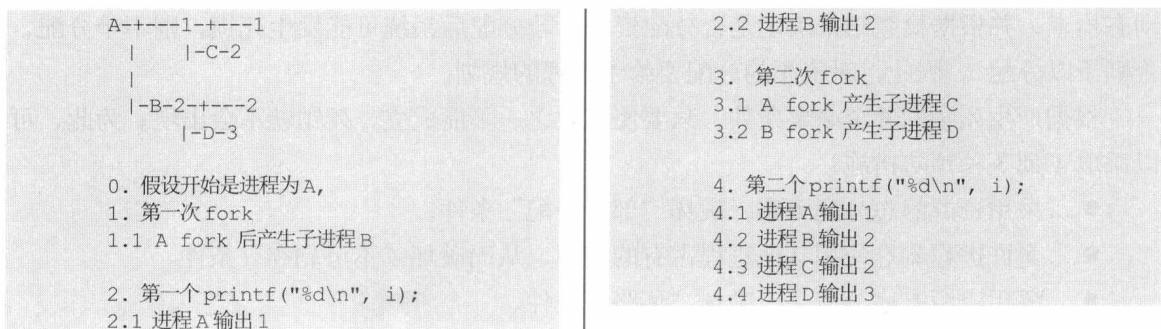
```
printf("%d\n", i);
if(!fork()) i++;
printf("%d\n", i);
return 0;
}
```

- A. The output will include number "1" exactly two times;
- B. The output will include number "2" exactly two times;
- C. Number "3" will always be last outputted;

D. It is possible that no number "3" in output, because "i++" is not an atomic operation.

解析：fork 英文是叉的意思。在这里的意思是进程从这里开始分叉，分成了两个进程：一个是父进程，一个子进程。子进程复制了父进程的绝大部分：栈、缓冲区等等。系统为子进程创建一个新的进程表项，其中进程 id 与父进程是不相同的，这也就是说父子进程是两个独立的进程，虽然父子进程共享代码空间，但是在涉及写数据时子进程有自己的数据空间，在有数据修改时，系统会为子进程申请新的页面。

在本题中，由于 if(!fork()) i++; 所以只有子进程才会执行 i++。执行顺序如下：



选项 B 不正确，2 会输出 3 次；选项 C 不一定，因为父子进程谁先运行不一定；选项 D 也不对，3 会出现。

答案：A

扩展知识：想一下，如下代码的输出结果是什么？

```

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    int i;
    for(i=0; i<2; i++){
        fork();

        printf("ppid=%d, pid=%d, i=%d \n", getppid(), getpid(), i);
    }
    sleep(10);
    return 0;
}

```

面试例题 7：请描述进程的三种基本状态。[中国某互联网软件公司 T2013 年面试题]

答案：进程在运行中不断地改变其运行状态。通常，一个运行进程必须具有以下三种基本状态。
1> 就绪(Ready)状态：当进程已分配到除 CPU 以外的所有必要的资源，只要获得处理机便可立即执行，这时的进程状态称为就绪状态。
2> 执行 (Running) 状态：当进程已获

得处理机，其程序正在处理机上执行，此时的进程状态称为执行状态。3> 阻塞(Blocked)状态：正在执行的进程，由于等待某个事件发生而无法执行时，便放弃处理机而处于阻塞状态。引起进程阻塞的事件可以有多种，例如，等待 I/O 完成、申请缓冲区不能满足、等待信件(信号)等。

面试例题 8：若系统中有五台绘图仪，有多个进程均需要使用两台，规定每个进程一次仅允许申请一台，则至多允许（ ）个进程参与竞争，而不会发生死锁。[中国台湾某 IT 公司 C2013 年 2 月笔试试题]

- A. 2 B. 3 C. 4 D. 5

解析：哲学家就餐问题：资源（餐叉）按照某种规则编号为 1 至 5，每一个工作单元（哲学家）总是先拿起左右两边编号较低的餐叉，再拿编号较高的。用完餐叉后，他总是先放下编号较高的餐叉，再放下编号较低的。在这种情况下，当 4 位哲学家同时拿起他们手边编号较低的餐叉时，只有编号最高的餐叉留在桌上，从而第 5 位哲学家就不能使用任何一只餐叉了。

答案：C

16.2 线程

面试例题 1：请描述进程和线程的差别。[美国某著名软件公司 2005 年面试题]

答案：进程是程序的一次执行。线程可以理解为进程中执行的一段程序片段。在一个多任务环境中下面的概念可以帮助我们理解两者间的差别。

进程间是独立的，这表现在内存空间、上下文环境上；线程运行在进程空间内。一般来讲（不使用特殊技术），进程无法突破进程边界存取其他进程内的存储空间；而线程由于处于进程空间内，所以同一进程所产生的线程共享同一内存空间。

同一进程中的两段代码不能够同时执行，除非引入线程。

线程是属于进程的，当进程退出时该进程所产生的线程都会被强制退出并清除。线程占用的资源要少于进程所占用的资源。进程和线程都可以有优先级。

进程间可以通过 IPC 通信，但线程不可以。

面试例题 2：下面哪个选项不是 PE 文件？

- A. EXE B. DLL C. COM D. DOC

解析：PE 文件被称为可移植的执行体是 Portable Execute 的全称，常见的 EXE、DLL、

OCX、SYS、COM 都是 PE 文件，PE 文件是微软 Windows 操作系统上的程序文件（可能是间接被执行，如 DLL）

答案：D

面试例题 3：Windows 将遵循下面的哪种搜索来定位 DLL？

- 1 进程的当前工作目录
 - 2 包含 EXE 文件的目录
 - 3 列在 Path 环境变量中的一系列目录
 - 4 Windows 系统目录
 - 5 Windows 目录
- A. 12453 B. 12543 C. 21453 D. 21345

解析：Windows 平台的大多数程序都使用各种动态链接库（DLL）来避免重复实现功能。操作系统为每个程序加载若干个 DLL，具体由程序的类型决定。当程序不指定 DLL 的绝对位置时，将使用默认的搜索顺序来找到它。默认情况下，操作系统所使用的搜索顺序为：

(1) 内存；(2) KnownDLLs；(3) 清单与 .local；(4) 应用程序目录；(5) 当前工作目录；(6) 系统目录；(7) 路径变量。

答案：C

面试例题 4：In user mode application development, sometimes we choose to use dynamic linked library instead of static linked library. That is because（动态链接库相对静态连接库的优点主要是）

- A. The dynamic link library can be upgraded without requiring applications to be re-linked or re-compiled.
- B. Runtime loading dynamic library will be faster than static library.
- C. Calling functions in dynamic library will be faster than static library.
- D. Dynamic library enable lots of memory sharing when multiple apps are using the same libraries at the same time. This is also true for saving disk space.
- E. Dynamic link library can be explicitly loaded/unloaded at runtime, this helps application provide optional features.

解析：目前以 lib 为后缀的库有两种，一种为静态链接库(Static Library)，另一种为动态链接库(DLL)的导入库 (Import Library，以下简称“导入库”)。虽然静态链接库和动态库的导入库都是.lib 文件，但是区别很大，它们实质是不一样的东西。静态库本身就包含了实际执行

代码、地址符号表等等，而对于导入库而言，其实际的执行代码位于动态库中，导入库只包含了地址符号表等，确保程序找到对应函数的一些基本地址信息。

静态链接库是一个或者多个 obj 文件的打包，所以有人干脆把从 obj 文件生成 lib 的过程称为 Archive，即合并到一起。比如你链接一个静态库，如果其中有错，它会准确地找到是哪个 obj 有错，即静态 lib 只是壳子。当我们的应用工程在使用静态链接库的时候，静态链接库要参与编译，在生成执行文件之前的链接过程中，将静态链接库的全部指令直接链接入可执行文件中，故而，在可执行文件生成以后，静态链接库.lib 文件即可以弃之不用。

动态链接库 (dll) 是作为共享函数库的可执行文件。动态链接提供了一种方法，使进程可以调用不属于其可执行代码的函数。函数的可执行代码位于一个.dll 文件中，该 dll 包含一个或多个已被编译、链接并与使用它们的进程分开存储的函数。dll 还有助于共享数据和资源。多个应用程序可同时访问内存中单个 dll 副本的内容。使用动态链接代替静态链接有若干优点。dll 节省内存，减少交换操作，节省磁盘空间，更易于升级(不需要重链接和重编译)，提供售后支持，提供扩展 MFC 库类的机制，支持多语言程序。

静态链接库与动态链接库都是共享代码的方式，如果采用静态链接库，lib 中的指令都全部被直接包含在最终生成的 exe 文件中了。但是若使用 dll (即动态链接库)，该 dll 不必被包含在最终 exe 文件中，exe 文件执行时可以“动态”地引用和卸载这个与 exe 独立的 dll 文件。静态链接库和动态链接库的另外一个区别在于静态链接库中不能再包含其他的动态链接库或者静态库，而在动态链接库中还可以再包含其他的动态或静态链接库。动态链接库与静态链接库的使用不同之处在于它允许可执行模块 (.dll 文件或.exe 文件) 仅包含在运行时定位 dll 函数的可执行代码所需的信息。在静态链接库的使用中，链接器从静态链接库获取所有被引用的函数，并将库同代码一起放到可执行文件中。

选项 B 和 C 不正确，静态链接库可能比动态链接库更快。

答案：A, D, E。

面试例题 5：假定我们有三个程序，每个程序花费 80% 的时间进行 I/O, 20% 的时间使用 CPU，每个程序的启动时间和其需要使用 CPU 进行计算机的分钟数如表所示。请问在多线程/进程环境下，系统总响应时间为多少？[中国某著名即时通讯软件公司 T2013 年面试题]

程序编号	启动时间	需要 CPU 时间(分钟)
1	00:00 (零点零分)	3.5
2	00:10	2
3	00:15	1.5

- A. 22.5 B. 23.5 C. 24.5 D. 25.5

解析：0 ~ 10 分钟， $10 * 0.8 = 8$ 10 * 0.2 = 2 A 还剩下 $3.5 - 2 = 1.5$ 分钟 CPU 需要跑。

10~15分钟，有两个进程，CPU利用率为 $1-0.8*0.8=0.36$ 。所以 $(15-10)*0.36=1.8$ 分钟； $1.8/2=0.9$ （两个进程均分CPU时间）这样A剩下 $1.5-0.9=0.6$ 分钟，B剩下 $2-0.9=1.1$ 分钟。

15分钟开始时，有三个进程 CPU利用率为 $1-0.8*0.8*0.8=0.488$ ，所以A在 $0.6*3/0.488=3.69$ 后也就是 $15+3.69=18.69$ 分完成，之后CPU利用率又为0.36，此时B剩下 $1.1-0.6=0.5$ 分钟，C剩下 $1.5-0.6=0.9$ 分钟之后，B在 $0.5*2/0.36=2.78$ 也就是 $2.78+18.69=21.46$ 分钟的时候B进程结束，之后C开始单跑 $0.9-0.5=0.4$ 分钟； $0.4/0.2=2$ 分钟，即2分钟之后C结束，也就是 $21.46+2=23.46 \approx 23.5$ 分钟之后，CPU跑完三个程序。

答案：A

面试例题6：创建两个线程模拟火车站两个窗口售票程序，窗口售票时间为1秒，两个窗口不能同时售票。[美国某著名软件公司I2013年面试题]

解析：进程是由两个部分构成的，一个是进程内核对象，另一个是地址空间。同样，线程也是由两个部分组成的：一个是线程的内核对象，操作系统用它来对线程实施管理。内核对象也是系统用来存放线程统计信息的地方。另一个是线程堆栈，它用于维护线程在执行代码时需要的所有函数参数和局部变量。

进程是不活泼的。进程从来不执行任何东西，它只是线程的容器。线程总是在某个进程环境中创建的，而且它的整个寿命期都在该进程中。这意味着线程在它的进程地址空间中执行代码，并且在进程的地址空间中对数据进行操作。因此，如果在单进程环境中，你有两个或多个线程正在运行，那么这两个线程将共享单个地址空间。这些线程能够执行相同的代码，对相同的数据进行操作。这些线程还能共享内核对象句柄，因为句柄表依赖于每个进程而不是每个线程的存在。

进程使用的系统资源比线程多得多，原因是它需要更多的地址空间。为进程创建一个虚拟地址空间需要许多系统资源。系统中要保留大量的记录，这要占用大量的内存。由于线程需要的开销比进程少，因此一般用增加线程来解决编程问题，而要避免创建新的进程。

每当进程被初始化时，系统就要创建一个主线程。该线程与C/C++运行期库的启动代码一道开始运行，启动代码则调用进入点函数，并且继续运行直到进入点函数返回并且C/C++运行期库的启动代码调用退出为止。对于许多应用程序来说，这个主线程是应用程序需要的唯一线程。不过，进程能够创建更多的线程来帮助执行它们的操作。

每个线程必须拥有一个进入点函数，线程从这个进入点开始运行。即main、wmain、WinMain或wWin Main。如果想要在你的进程中创建一个辅助线程，它必定也是一个进入点函数，类似下面的样子：

```
DWORD WINAPI ThreadFunc(PVOID pvParam)
```

```

{
    DWORD dwResult = 0;
    ...
    return(dwResult);
}

```

线程函数可以执行你想要它做的任何任务。最终，线程函数到达它的结尾处并且返回。这时，线程终止运行，该堆栈的内存被释放，同时，线程的内核对象的使用计数被递减。如果使用计数降为 0，线程的内核对象就被撤销。与进程内核对象的情况相同，线程内核对象的寿命至少可以达到它们相关联的线程那样长，不过，该对象的寿命可以远远超过线程本身的寿命。

答案：

```

#include <windows.h>
#include <iostream>
using namespace std;
//这是2个线程模拟卖火车票的小程序
DWORD WINAPI Fun1Proc(LPVOID lpParameter); //thread data
DWORD WINAPI Fun2Proc(LPVOID lpParameter); //thread data

int index=0;
int tickets=10;
HANDLE hMutex;
int main()
{
    HANDLE hThread1;
    HANDLE hThread2;
    //创建线程

    hThread1=CreateThread(NULL,0,Fun1Proc,NULL,0,NULL);
    hThread2=CreateThread(NULL,0,Fun2Proc,NULL,0,NULL);
    CloseHandle(hThread1);
    CloseHandle(hThread2);

    //创建互斥对象
    hMutex=CreateMutex(NULL,TRUE,"tickets");
    if (hMutex)
    {
        if (ERROR_ALREADY_EXISTS==GetLastError())
        {
            cout<<"only one instance can run!"<<endl;
            return 0
        }
    }
    WaitForSingleObject(hMutex,INFINITE);
    ReleaseMutex(hMutex);
    ReleaseMutex(hMutex);

    Sleep(4000);
    return 0;
}

```

```

//线程1 的入口函数
DWORD WINAPI Fun1Proc(LPVOID lpParameter)//thread data
{
    while (true)
    {
        ReleaseMutex(hMutex);
        WaitForSingleObject(hMutex, INFINITE);
        if (tickets>0)
        {
            Sleep(1);
            cout<<"thread1 sell ticket :"<<tickets--<<endl;
        }
        else
            break;
        ReleaseMutex(hMutex);
    }

    return 0;
}
//线程2 的入口函数
DWORD WINAPI Fun2Proc(LPVOID lpParameter)//thread data
{
    while (true)
    {
        ReleaseMutex(hMutex);
        WaitForSingleObject(hMutex, INFINITE);
        if (tickets>0)
        {
            Sleep(1);
            cout<<"thread2 sell ticket :"<<tickets--<<endl;
        }
        else
            break;
        ReleaseMutex(hMutex);
    }

    return 0;
}

```

16.3 内存管理

面试例题 1：简述 Windows 内存管理的几种方式和优缺点。[中国某著名互联网公司 B 面试题]

答案：Windows 内存管理方式主要分为：页式管理、段式管理、段页式管理。

页式管理的基本原理是将各进程的虚拟空间划分成若干个长度相等的页(page)；页式管理把内存空间按页的大小划分成片或者页面，然后把页式虚拟地址与内存地址建立一一对应的页表；并用相应的硬件地址变换机构来解决离散地址变换问题。页式管理采用请求调页或预调页技术来实现内外存存储器的统一管理。其优点是没有外碎片，每个内碎片不超过页

的大小。缺点：程序全部装入内存，要求有相应的硬件支持。例如地址变换机构缺页中断的产生和选择淘汰页面等都要求有相应的硬件支持。这增加了机器成本，增加了系统开销。

段式管理的基本思想就是把程序按内容或过程函数关系分成段，每段有自己的名字。一个用户作业或进程所包含的段对应一个二维线形虚拟空间，也就是一个二维虚拟存储器。段式管理程序以段为单位分配内存，然后通过地址影射机构把段式虚拟地址转换为实际内存物理地址。其优点是可以分别编写和编译，可以针对不同类型的段采取不同的保护，可以按段为单位来进行共享，包括通过动态链接进行代码共享。缺点是会产生碎片。

段页式管理：为了实现段页式管理，系统必须为每个作业或进程建立一张段表以管理内存分配与释放、缺段处理等。另外由于一个段又被划分成了若干页。每个段又必须建立一张页表以把段中的虚页转换成内存中的实际页面。显然与页式管理时相同，页表中也要有相应的实现缺页中断处理和页面保护等功能的表项。段页式管理是段式管理与页式管理方案结合而成的，所以具有它们两者的优点。但反过来说，由于管理软件的增加，复杂性和开销也就随之增加了。另外需要的硬件以及占用的内存也有所增加。使得执行速度下降。

面试例题 2：X64 和 X86 有何区别？

答案：Intel 曾用 8086、80286、80386 等作为其 PC 用 CPU 的型号表示法，x86 指 Intel 制造的普通 CPU（提出 x86 这个表示法时，个人电脑上以 32 位 Intel 的 CPU 为主），x64 是 x86_64 的缩写，指 x86 基础上的改进版（加入 64 位地址扩展等性能），而纯 64 位计算机架构用 IA64 表示，32 位兼容的 64 位架构用 amd64 表示（AMD 是这一架构的主要生产商）。由于 Intel 起步较早，影响较大，有时也把 amd64 架构的 CPU 称为 x86_64 架构。

面试例题 3：Belady's Anomaly 出现在哪？

- A. 内存管理算法
- B. 内存换页算法
- C. 预防死锁算法
- D. 磁盘调度算法

解析：所谓 Belady 现象是指：采用 FIFO 算法时，如果对一个进程未分配它所要求的全部页面，有时就会出现分配的页面数增多但缺页率反而提高的异常现象。Belady 现象的原因是 FIFO 算法的置换特征与进程访问内存的动态特征是矛盾的，即被置换的页面并不是进程不会访问的。这些页在 FIFO 算法下被反复调入和调出，并且有 Belady 现象。

答案：B

面试例题 4：什么是 Thrashing?

- A. 非常频繁的换页活动
- B. 非常高的 CPU 执行活动
- C. 一个极长的执行过程
- D. 一个极大的虚拟内存法

解析：内存抖动(Thrashing)一般是内存分配算法不好，内存太小或者程序的算法不佳引起的页面频繁地从内存调入/调出的行为。

答案：A

面试例题5：避免死锁的一个著名算法是？

- A. 先入先出发
- B. 银行家算法
- C. 优先级算法
- D. 资源有序分配法

解析：银行家算法是用来避免死锁的，该方法将系统的状态分为安全状态和不安全状态，只要使系统处于安全状态，便可避免死锁的发生。

答案：B

面试例题6：某主机安装了2GB内存，在其上运行的某支持MMU的32位Linux发行版中，一共运行了X、Y、Z三个进程，下面关于三个内存使用程序的方式，哪个是可行的？

- A. X、Y、Z的虚拟地址空间都映射到0~4G虚拟地址上
- B. X在堆上分配总大小为1GB的空间，Y在堆上分配200MB，Z在堆上分配500MB，并且内存映射访问一个1GB的磁盘文件。
- C. X在堆上分配1GB，Y在堆上分配800MB，Z在堆上分配400MB
- D. 以上的访问方式都是可行的

解析：虚拟存储器的基本思想是程序、数据、堆栈的总的大小可以超过物理存储器的大小，操作系统把当前使用的部分保留在内存中，而把其他未被使用的部分保存在磁盘上。

答案：D

面试例题7：某虚拟内存系统采用页式内存管理，使用LRU页面管理算法。考虑下面的页面访问地址流（每次访问在一个时间单位内完成）中一共有几次页面失败：

- 1, 8, 1, 7, 8, 2, 7, 2, 1, 8, 3, 8, 2, 1, 3, 1, 7, 1, 3, 7?
- A. 4
- B. 5
- C. 6
- D. 7

解析：LRU算法的提出，是基于这样一个事实：在前面几条指令中使用频繁的页面，很可能在后面的几条指令中频繁使用。反过来说，已经很久没有使用的页面，很可能在未来较长的一段时间内不会被用到。算法如下。

```
1, 8, 1, 7, 8, 2, 7, 2, 1, 8, 3, 8, 2, 1, 3, 1, 7, 1, 3, 7
1, 不在内存中, 1 次页面失效 内存中的页面为 1
8, 不在内存中, 1 次页面失效 内存中的页面为 1 8
1, 在内存中
7, 不在内存中, 1 次页面失效 内存中的页面为 1 8 7
8, 在内存中
```

2, 不在内存中, 1 次页面失效 内存中的页面为 1 8 7 2
7, 在内存中
2, 在内存中
1, 在内存中
8, 在内存中
3, 不在内存中, 1 次页面失效 内存中的页面为 1 8 3 2 根据 LRU 页面管理算法页面 7 最久未使用, 将它换出
8, 在内存中
2, 在内存中
1, 在内存中
3, 在内存中
1, 在内存中
7, 不在内存中, 1 次页面失效 内存中的页面为 1 7 3 2 根据 LRU 页面管理算法页面 8 最久未使用, 将它换出
1, 在内存中
3, 在内存中
7, 在内存中
一共 6 次页面失效

答案: C

第 17 章

数据库与 SQL 语言

数 据库面试题主要包括范式、事物、存储过程、SQL 语言，以及索引等诸方面。

17.1 数据库理论

面试例题 1：设有关系 R(S,D,M)，其函数依赖集 F={S→D,D→M}。则关系 R 至多满足_____。

[美国某著名搜索引擎公司面试题]

- A. 1NF
- B. 2NF
- C. 3NF
- D. BCNF

解析：数据库模式的 4 个范式问题。

1NF：第一范式。如果关系模式 R 的所有属性的值域中每一个值都是不可再分解的值，则称 R 属于第一范式模式。如果某个数据库模式都是第一范式的，则称该数据库模式属于第一范式的数据库模式。

第一范式的模式要求属性值不可再分裂成更小部分，即属性项不能是属性组合或由组属性组成。

2NF：第二范式。如果关系模式 R 为第一范式，并且 R 中每一个非主属性完全函数依赖于 R 的某个候选键，则称 R 为第二范式模式。如果某个数据库模式中每个关系模式都是第二范式的，则称该数据库模式属于第二范式的数据库模式。（注：如果 A 是关系模式 R 的候选键的一个属性，则称 A 是 R 的主属性，否则称 A 是 R 的非主属性。）

3NF：第三范式。如果关系模式 R 是第二范式，且每个非主属性都不传递依赖于 R 的候选键，则称 R 是第三范式的模式。如果某个数据库模式中的每个关系模式都是第三范式，则称 R 为 3NF 的数据库模式。

BCNF：BC范式。如果关系模式R是第一范式，且每个属性都不传递依赖于R的候选键，那么称R为BCNF的模式。

4NF：第四范式。设R是一个关系模式，D是R上的多值依赖集合。如果D中成立非平凡多值依赖 $X \rightarrow\!\!> Y$ 时，X必是R的超键，那么称R是第四范式的模式。

上题属于传递依赖，所以至多满足第二范式。

答案：B

面试例题2：存储过程和函数的区别是什么？[美国某著名搜索引擎公司面试题]

答案：存储过程是用户定义的一系列SQL语句的集合，涉及特定表或其他对象的任务，用户可以调用存储过程。而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值，并且不涉及特定用户表。

面试例题3：What is database transaction? (什么是数据库事务？)

答案：数据库事务是指作为单个逻辑工作单元执行的一系列操作，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。

事务的开始与结束可以由用户显式地控制。如果用户没有显式地定义事务，则由DBMS按默认规定自动划分事务。事务具有原子性、一致性、独立性及持久性等特点。

- 事务的原子性是指一个事务要么全部执行，要么不执行。也就是说一个事务不可能只执行了一半就停止了。比如你从银行取钱，这个事务可以分成两个步骤（1）存折减款，（2）拿到现金。不可能存折钱少了，而钱却没拿出来。这两步必须同时完成，要么就都不完成。
- 事务的一致性是指事务的运行并不改变数据库中数据的一致性。例如，完整性约束了 $a+b=10$ ，一个事务改变了a，那么b也应该随之改变。
- 事务的独立性是指两个以上的事务不会出现交错执行的状态。因为这样可能会导致数据不一致。
- 事务的持久性是指事务运行成功以后，对系统的更新是永久的。不会无缘无故的回滚。

面试例题4：游标的作用是什么，如何知道游标已经到了最后？[中国某著名计算机金融软件公司面试题]

答案：游标用于定位结果集的行。通过判断全局变量@@FETCH_STATUS可以判断其是否到了最后。通常此变量不等于0表示出错或到了最后。

面试例题 5：触发器分为事前触发和事后触发，这两种触发有何区别？语句级触发和行级触发有何区别？[美国某著名计算机软件公司面试题]

答案：事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

面试例题 6：什么叫做 SQL 注入式攻击，如何防范？[中国台湾某著名杀毒软件公司面试题]

答案：所谓 SQL 注入式攻击，就是攻击者把 SQL 命令插入到 Web 表单的输入域或页面请求的查询字符串中，欺骗服务器执行恶意的 SQL 命令。在某些表单中，用户输入的内容直接用来构造（或者影响）动态 SQL 命令，或作为存储过程的输入参数，这类表单特别容易受到 SQL 注入式攻击。

防范 SQL 注入式攻击闯入并不是一件特别困难的事情，只要在利用表单输入的内容构造 SQL 命令之前，把所有输入内容过滤一番就可以了。过滤输入内容可以按多种方式进行。

- 替换单引号，即把所有单独出现的单引号改成两个单引号，防止攻击者修改 SQL 命令的含义。
- 删除用户输入内容中的所有连字符，防止攻击者顺利获得访问权限。
- 对于用来执行查询的数据库账户，限制其权限。用不同的用户账户执行查询、插入、更新、删除操作。由于隔离了不同账户可执行的操作，因而也就防止了原本用于执行 SELECT 命令的地方却被用于执行 INSERT、UPDATE 或 DELETE 命令。
- 用存储过程来执行所有的查询。SQL 参数的传递方式将防止攻击者利用单引号和连字符实施攻击。此外，它还使得数据库权限可以被限制到只允许特定的存储过程执行，所有的用户输入必须遵从被调用的存储过程的安全上下文，这样就很难再发生注入式攻击了。
- 检查用户输入的合法性，确信输入的内容只包含合法的数据。数据检查应当在客户端和服务器端都执行。之所以要执行服务器端验证，是为了弥补客户端验证机制脆弱的安全性。在客户端，攻击者完全有可能获得网页的源代码，修改验证合法性的脚本（或者直接删除脚本），然后将非法内容通过修改后的表单提交给服务器。因此，要保证验证操作确实已经执行，唯一的办法就是在服务器端也执行验证。
- 将用户登录名称、密码等数据加密保存。加密用户输入的数据，然后再将它与数据库中保存的数据比较，这相当于对用户输入的数据进行了“消毒”处理。用户输入的数据不再对数据库有任何特殊的意义，从而也就防止了攻击者注入 SQL 命令。
- 检查提取数据的查询所返回的记录数量。如果程序只要求返回一个记录，但实际返回的记录却超过一行，那就当作出错处理。

面试例题 7: Explain the difference between clustered and non-clustered indexes. How does index affect the query? (解释聚集索引和非聚集索引之间的区别)

答案: 经典教科书对聚集索引的解释是：聚集索引的顺序就是数据的物理存储顺序，而对非聚集索引的解释是索引顺序与数据物理排列顺序无关。正是因为如此，所以一个表最多只能有一个聚集索引。

在 SQL Server 中，索引是通过二叉树的数据结构来描述的，我们可以这么理解聚集索引：索引的叶节点就是数据节点。而非聚集索引的叶节点仍然是索引节点，只不过有一个指针指向对应的数据块。

聚集索引确定表中数据的物理顺序。聚集索引类似于电话簿（电话簿按照字母簿排序），后者按姓氏排列数据。由于聚集索引规定数据在表中的物理存储顺序，因此一个表只能包含一个聚集索引。但该索引可以包含多个列（组合索引），就像电话簿按姓氏和名字进行组织一样。

聚集索引对于那些经常要搜索范围值的列特别有效。使用聚集索引找到包含第一个值的行后，便可以确保包含后续索引值的行在物理相邻。例如，如果应用程序执行的一个查询经常检索某一日期范围内的记录，则使用聚集索引可以迅速找到包含开始日期的行，然后检索表中所有相邻的行，直到到达结束日期。这样有助于提高此类查询的性能。同样，如果对从表中检索的数据进行排序时经常要用到某一列，则可以将该表在该列上聚集（物理排序），避免每次查询该列时都进行排序，从而节省成本。

使用非聚集索引，非聚集索引与课本中的索引类似。数据存储在一个地方，索引存储在另一个地方，索引带有指针指向数据的存储位置。索引中的项目按索引键值的顺序存储，而表中的信息按另一种顺序存储（这可以由聚集索引规定）。如果在表中未创建聚集索引，则无法保证这些行具有任何特定的顺序。

有索引就一定检索得快吗？答案是否定的。有些时候用索引还不如不用索引快。比如说我们要检索表中的所有 8000 条记录，如果不使用索引，需要访问 $8000 \text{ 条} \times 1000 \text{ 字节}/8\text{K 字节} = 1000$ 个页面，如果使用索引的话，首先检索索引，访问 $8000 \text{ 条} \times 10 \text{ 字节}/8\text{K 字节} = 10$ 个页面得到索引检索结果，再根据索引检索结果去对应数据页面，由于是检索所有数据，所以需要再访问 $8000 \text{ 条} \times 1000 \text{ 字节}/8\text{K 字节} = 1000$ 个页面将全部数据读取出来，一共访问了 1010 个页面，这显然不如不用索引快。

17.2 SQL语言

面试例题1：找出表ppp里面num最小的数，不能使用min函数。[中国某著名软件外包企业2004年面试题]

答案：

```
select * from ppp where num <= all(select num from ppp)
```

或者：

```
select top 1 num from ppp order by num
```

面试例题2：找出表ppp里面最小的数，可以使用min函数。[中国某著名软件外包企业2004年面试题]

答案：

```
select * from ppp where num = (select min(num) from ppp)
```

面试例题3：选择表ppp2中num重复的记录。[中国某著名软件外包企业2004年面试题]

答案：

```
select * from ppp2  
where num in(select num from ppp2 group by num having(count(num)>1))
```

面试例题4：写出复制表、复制表和四表联查的SQL语句。[中国某著名软件外包企业2004年面试题]

答案：

复制表（只复制结构，源表名：A，新表名：B）：

```
select * into B from A where 1=0
```

复制表（复制数据，源表名：A，新表名：B）：

```
select * into B from A
```

四表联查：

```
select * from A,B,C,D where 关联条件
```

面试例题5：在SQL Server中如何用SQL语句建立一张临时表？[中国某著名软件外包企业2004年面试题]

答案：

```
create table #Temp(字段1 类型,字段2 类型.....)
```

注意，临时表要在表名前面加“#”。

面试例题6：有数据表A，有一个字段LASTUPDATETIME，是最后更新的时间，如果要查最新更新过的记录，如何写SQL语句？[中国某著名软件外包企业2008年面试题]

答案：

```
SQL code
select *
from [数据表A]
where LASTUPDATETIME = (select max(LASTUPDATETIME) from [数据表A])
```

面试例题7：Let's say we have a database with 1 one-column table. It contains 1000 same records. Could you please give at least 1 solution to help get records between line 5 and 7. No line number, row id or index etc. (有一个数据库，只有一个表，包含着1000个记录，你能想出一种解决方案来把第5到第7行的记录取出来么？不要使用航标和索引。) [德国某著名软件咨询企业2005年面试题]

答案：

建立数据库：

```
declare @i int
set @i=1
creaté table #T(userid int)
while (@i<=10)
begin
insert into #T
select @i
set @i=@i+1
end

select userid from
(
    select top 3 userid from (select top 7 userid from #T order by
        userid)Ta order by userid desc
) TB order by userid
```

删除数据库：

```
drop table #T
```

提取数据：

```
select top 3 userid from T where userid not in (select top 4 userid from
    T order by userid) order by userid
```

或者：

```
select top 7 userid from T where userid > ANY (select top 4 userid from
    T order by userid) order by userid
```

或者：

```
select top 7 userid from T where userid > ALL (select top 4 userid from
T order by userid) order by userid
```

面试例题8：要查数据表中第30到40条记录，有字段ID，但是ID并不连续。如何写SQL语句？[德国某著名软件咨询企业2007年面试题]

答案：

SQL语句如下：

```
SQL codeselect * from tb a
where exists (select 1 from (
    select top 10 id from (select top 40 id from tb order by id desc) as c
) as b where b.id=a.id)
```

面试例题9：请问下列语句是否可以正确运行，为什么？[德国某著名软件咨询企业2007年面试题]

```
update table1
set name=(select name from table2 t1 inner join table1
t2 on t1.id=t2.id) where name is null
```

答案：不能。

```
select name from table2 t1 inner join table1 t2 on
t1.id =t2.id)
```

这一句返回的是一个表结果集，而update语句少了条件值。返回的是结果集，更新的结果并没有指定，会出现非一对一情况。

17.3 SQL语言客观题

面试例题1：Which statement shows the maximum salary paid in each job category of each department?（下面哪个SQL语句描述了每一个部门的每个工种的工资最大值？）[中国某著名计算机金融软件公司2005年面试题]

- A. select dept_id, job_cat,max(salary) from employees where salary > max (salary);
- B. select dept_id, job_cat,max(salary) from employees group by dept_id, job_cat;
- C. select dept_id, job_cat,max(salary) from employees;
- D. select dept_id, job_cat,max(salary) from employees group by dept_id;
- E. select dept_id, job_cat,max(salary) from employees group by dept_id, job_cat,salary;

答案：B

面试例题 2: Description of the students table (以下是学生表的字段描述):

sid_id	number
start_date	date
end_date	date

which two function are valid on the start_date column? (关于对 start_date 字段的使用, 以下哪两个函数是合法的?) [中国某著名计算机金融软件公司 2005 年面试题]

- A. sum(start_date)
- B. avg(start_date)
- C. count(start_date)
- D. avg(start_date,end_date)
- E. min(start_date)
- F. maximum(start_date)

答案: C, E。

面试例题 3: For which two constraints does the Oracle server implicitly create a unique index? (以下哪两种约束的情况下, Oracle 数据库会隐性创建一个唯一索引?) [中国某著名计算机金融软件公司 2005 年面试题]

- A. not null
- B. primary
- C. foreign key
- D. check
- E. unique

答案: B, E。

面试例题 4: In a select statement that includes a where clause, where is the group by clause placed in the select statement? (在 select 语句中包括一个 where 关键词, 请问 group by 关键词一般在 select 语句中什么位置?) [中国某著名计算机金融软件公司 2005 年面试题]

- A. immediately after the select clause (紧跟 select 关键词之后)
- B. before the where clause (在 where 关键词之前)
- C. before the from clause (在 from 关键词之前)
- D. after the order by clause (在 order by 关键词之后)
- E. after the where clause (在 where 关键词之后)

答案: E

面试例题 5: In a select statement that includes a where clause, where is the order by clause placed in the select statement? (在 select 语句中包括一个 where 关键词, 请问 order by 关键词一般在 select 语句中什么位置?) [中国某著名计算机金融软件公司 2005 年面试题]

- A. immediately after the select clause (紧跟 select 关键词之后)
- B. before the where clause (在 where 关键词之前)
- C. after all clause (在所有关键词之后)

- D. after the where clause (在 where 关键词之后)
- E. before the from clause (在 from 关键词之前)

答案：C

面试例题 6：Evaluate there two SQL statements. (对比下面两个 SQL 语句。) [中国某著名计算机金融软件公司 2005 年面试题]

Select last_name,salary from employees order by salary;

Select last_name,salary from employees order by 2 asc;

- A. the same result (相同的结果)
- B. different result (不同的结果)
- C. the second statement returns a syntax error (第二个结果会显示错误)

答案：A

面试例题 7：You would like to display the system date in the format “20051110 14: 44: 17”。Which select statement should you use? (如果你想把时间显示成像“20051110 14: 44: 17”这样的格式，下面哪个 select 语句应该被使用？) [中国某著名计算机金融软件公司 2005 年面试题]

- A. select to_date(sydate,'yearmmdd hh:mm:ss')from dual;
- B. select to_char(sydate,'yearmonthday hh:mi:ss')from dual;
- C. select to_date(sydate,'yyyymmdd hh24:mi:ss')from dual;
- D. select to_char(sydate,'yyyymmdd hh24:mi:ss')from dual;
- E. select to_char(sydate,'yy-mm-dd hh24:mi:ss')from dual;

答案：D

面试例题 8：Which select statement will the result ‘ello world’from the string‘Hello world’? (如果要从字符串“Hello world”中提取出“ello world”这样的结果，下面的哪条 SQL 语句适合？) [中国某著名计算机金融软件公司 2005 年面试题]

- A. select substr('Hello World',1)from dual;
- B. select substr(trim('Hello World',1,1))from dual;
- C. select lower(substr('Hello World',1))from dual;
- D. select lower(trim('H'from'Hello World'))from dual;

答案：D

面试例题 9：which are DML statements(choose all that apply)? (下面哪一个 DML (Data

Manipulation Language, 数据操纵语言) 的执行状态?) [中国某著名计算机金融软件公司 2005 年面试题]

- A. commit
- B. merge
- C. update
- D. delete
- E. creat
- F. drop

答案: C, D。

面试例题 10: Select 语句中用来连接字符串的符号是_____。[中国某著名计算机金融软件公司 2005 年面试题]

- A. +
- B. &
- C. ||
- D. |

答案: A

面试例题 11: Given the following CREATE TABLE statement (如下列表):

```
CREATE TABLE department
(deptid INTEGER,
 deptname CHAR(25),
 budget NUMERIC(12,2))
```

Which of the following statement prevents two departments from being assigned the same DEPTID, but allows null values? (下列哪个选项防止两个部门 (相同名称) 被分配在 DEPTID 字段, 但允许 NULL 值?)

- A. ALTER TABLE department ADD CONSTRAINT dpt_cst PRIMARY KEY(改变 department 表, 增加主键约束)
- B. CREATE INDEX dpt_idx ON department(deptid) (创建 department 索引)
- C. ALTER TABLE department ADD CONSTRAINT dpt_cst UNIQUE (创建唯一约束)
- D. CREATE UNIQUE INDEX dpt_idx ON department(deptid) -- (在 department 唯一索引)

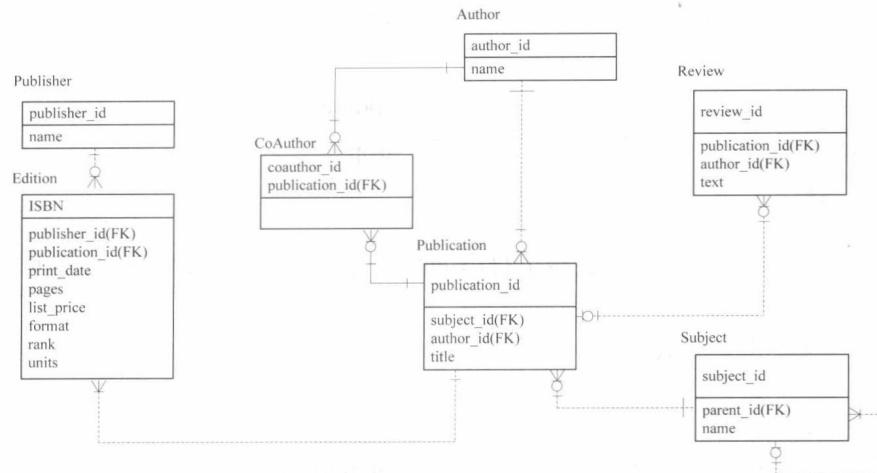
解析: A 选项有主键, 肯定不允许为 NULL。B、D 选项也显然不对。

unique 约束能够约束一列保证该列值唯一, 但允许该列有空值。故选 C。

答案: C

17.4 SQL 语言主观题

面试例题 1: Use the ERD to help you answer the following two questions (使用 ERD 图, 回答以下两个问题) [美国某数据库公司 2009 年 8 月笔试题]



问题 1: Find all ISBN values in the EDITION table, for all FORMAT='Hardcover' books. (在 EDITION 表中找到所有 ISBN 表，其 FORMAT 值为 Hardcover 的书。)

问题 2: Find the sum of all LIST_PRICE values in the EDITION table, for each publisher. (在 EDITION 表中找出 LIST_PRICE 值的总和，和每个 publisher 值。)

答案：上面两题的 SQL 语言表达分别是：

```

--1)
select ISBN from EDITION where FORMAT='Hardcover'
--2)
select publisher_id,sum(LIST_PRICE) from EDITION group by publisher_id
  
```

面试例题 2：一个表(表 a)，里面有 10 条数据，这 10 条数据都是相同的，现要删除其中 9 条。请问怎么写 SQL 语句？[中国著名杀毒软件公司 J 2008 年面试题]

解析：SQL 语言问题。本题十分简单，面试者可以写出多种答案来显示自己的水平。

答案：

可以实现的命令如下：

```

select top 1 * into # from a
go
truncate table a
go
insert a select * from #
  
```

或者：

```

select distinct * into tb_tmp from a
go
drop table a
go
  
```

```
exec sp_rename 'tb_tmp', 'a'
```

或者利用循环：

```
declare @i int
set @i=1
while(@i <9)
begin
    delete from dbo.表 where id=@i
    set @i=@i+1
end
```

面试例题3：一个表，表名为table，里面有n条数据。例如下表：

id	name	data
1	a	123
2	b	123
...		
...		
23	a	234
...		
32	b	345
...		
...		

name为a的数据共有n条，name为b的数据共有m条，请问怎么写SQL语句，使查询a时能够统计a的数量？结果类似于：

name	count
a	n

[中国著名杀毒软件公司J 2005年面试题]

解析：本题考查的是count语句的用法分析。

答案：

```
select name,count(*) as [count]
from 表a
where name='a'
group by name
```

面试例题4：

存表格式如下：

data表			
id	revtime(数据返回时间)	channel(通道号)	val(值)
1	2007-5-1	1	79
1	2007-5-1	2	46
1	2007-5-1	3	265
1	2007-5-2	1	80
1	2007-5-2	2	40
1	2007-5-2	3	266

channel表	
id	channel
1	1
1	2

1 3

请用 SQL 操作，最终显示为：

id	revtime	channel1data	channel2data	channel3data
1	2007-5-1	79	46	265
1	2007-5-2	80	40	266

//根据 channel 里相同 id 对应的通道数显示出对应通道的数据
 //例如：id 为 1 有 1\2\3 共 3 个通道
 //故应该显示 channel1\channel2\channel3 这 3 列，依此类推

[中国著名杀毒软件公司 J 2008 年面试题]

解析：Max 和 Sum 函数的用法。

答案：

```
select id,revtime,
max(case channel when 1 then val end) as channel1data,
max(case channel when 2 then val end) as channel2data,
max(case channel when 3 then val end) as channel3data
from [data]
group by id,revtime
```

或者使用 Sum 来实现：

```
select id,revtime ,
sum(case when channel=1 then val end)as data1,
sum(case when channel=2 then val end)as data2,
sum(case when channel=3 then val end)as data3
from t
group by id,revtime
```

面试例题 5：SQL Server 只能实现在本地备份，怎么才能方便地实施异地备份？试用 SQL 语言描述。[新加坡著名 ERP 公司 2008 年面试题]

答案：写成存储过程，建立作业定时备份。

代码如下：

```
--在 SQL 中做映射
exec master..xp_cmdshell 'net use z: \\yizhi\DS "密码"
                           /user:yizhi\administrator'
/*--说明：
exec master..xp_cmdshell 'net use z: \\xz\c$ "密码"
                           /user:xz\administrator'
z:是映射网络路径对应本机的盘符,与下面的备份对应
\\xz\c$是要映射的网络路径
xz\administrator
xz 是远程的计算机名
administrator 是登录的用户名
密码——指定的 administrator 用户的密码
--*/
--备份：with init 覆盖|noinit 添加
backup database 库名 to disk= 'E:\H_BACKUP.bak' with init
```

```
--COPY
exec master..xp_cmdshell 'copy E:\H_BACKUP.bak z: '
--删除
exec master..xp_cmdshell 'del E:\H_BACKUP.bak'
--完成后删除映射
exec master..xp_cmdshell 'net use z: /delete '
```

面试例题 6：我想把当前正在运行着的 MS SQL Server 中的一个数据库改名，请问应执行什么命令？（不能删除数据库）[新加坡著名 ERP 公司 2008 年面试题]

解析：

sp_renamedb 命令可以更改数据库的名称。

语法如下：

```
sp_renamedb[@dbname=] 'old_name' ,
[@newname=] 'new_name'
```

参数：

```
[@dbname=]'old_name'
```

是数据库的当前名称。old_name 为 sysname 类型，无默认值。

```
[@newname=]'new_name'
```

是数据库的新名称。new_name 必须遵循标识符规则。new_name 为 sysname 类型，无默认值。

返回代码值：0（成功）或非零数字（失败）

权限：

只有 sysadmin 和 dbcreator 固定服务器角色的成员才能执行 sp_renamedb。

答案：sp_renamedb 命令。

面试例题 7：两个表结构如下。

Customer（客户）的表结构：

```
CustomerID nchar(5) NOT NULL primary key,
CustomerName nvarchar(50) NOT NULL ,
CompanyName nvarchar(40) NOT NULL ,
ContactName nvarchar(30) NULL,
ContactTitle nvarchar(30) NULL,
Address nvarchar(60) NULL
```

Orders（订单）的表结构：

```
OrderID nvarchar(20) NOT NULL,
CustomerID nchar(5) NOT NULL ,
OrderDate datetime NULL,
RequiredDate datetime NULL,
ShippenDate datetime NULL,
```

```
primary key(OrderID,CustomerID)
```

Customer 与 Orders 的关系如下：

```
1          0..n  
Customer-----> Orders
```

写一个 SQL 语句，用 CustPrice 中的 Price 更新 Stock 的 Price。[新加坡著名 ERP 公司 2008 年面试题]

答案：

SQL 命令如下：

```
select * from Customer  
where not exists(select 1 from Orders where  
CustomerID=tmp.CustomerID)
```

面试例题 9：两个表结构如下。

Study 的表结构：

```
StudentID varchar(10) not null foreign key references  
        Student(StudentID)  
CourseID varchar(4) not null --课程代号
```

Student 的表结构：

```
StudentID varchar(10) not null,  
StudentName varchar(20) not null
```

写一个 SQL 语句，找出修 2~3 门的学生的名字。

[新加坡著名 ERP 公司 2008 年面试题]

答案：SQL 命令如下：

```
select * from Student as tmp  
where (select count(*) from Study where StudentID=  
tmp.StudentID) between 2 and 3
```

第 18 章

计算机网络及分布式系统



络面试例题主要包括局域网、广域网、IP 管理等诸方面。

18.1 网络结构

面试例题 1：在 OSI 参考模型中，物理层的作用是(1)。对等实体在一次交互作用中传送的信息单位称为(2)，它包括(3)两部分。上下层实体之间的接口称为服务访问点 (SAP)，网络层的服务访问点也称为(4)，通常分为(5)两部分。[中国某著名综合软件公司 2005 年面试题]

- | | |
|------------------|--------------|
| (1) A. 建立和释放连接 | B. 透明的传输比特流 |
| C. 在物理实体间传送数据帧 | D. 发送和接收用户数据 |
| (2) A. 接口数据单元 | B. 服务数据单元 |
| C. 协议数据单元 | D. 交互数据单元 |
| (3) A. 控制信息和用户数据 | B. 接口信息和用户数据 |
| C. 接口信息和控制信息 | D. 控制信息和校验信息 |
| (4) A. 用户地址 | B. 网络地址 |
| C. 端口地址 | D. 网卡地址 |
| (5) A. 网络号和端口号 | B. 网络号和主机地址 |
| C. 超网号和子网号 | D. 超网号和端口地址 |

解析：网络问题。

OSI 参考模型有 7 层，其分层原则如下：

- 根据不同层次的抽象分层。

- 每层应当有一个定义明确的功能。
- 每层功能的选择应该有助于制定网络协议的国际标准。
- 各层边界的选择应尽量节省跨过接口的通信量。
- 层数应足够多，以避免不同的功能混杂在同一层中，但也不能太多，否则体系结构会过于庞大。

根据以上标准，OSI 参考模型分为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

物理层涉及在信道上传输的原始比特流。

数据链路层的主要任务是加强物理层传输原始比特流的功能，使之对应的网络层显现为一条无错线路。发送包把输入数据封装在数据帧，按顺序传出去并处理接收方回送的确认帧。

网络层关系到子网的运行控制，其中一个关键问题是确认从源端到目的端如何选择路由。

传输层的基本功能是从会话层接收数据而且把其分成较小的单元传递给网络层。

会话层允许不同机器上的用户建立会话关系。

表示层用来完成某些特定的功能。

应用层包含着大量人们普遍需要的协议。

答案：B, C, A, B, B。

面试例题 2：TCP 和 UDP 有什么区别？[中国著名金融软件公司 S 2005 年面试题]

解析：举例说明两者间的区别。

TCP 连接就像打电话，两者之间必须有一条不间断的通路，数据不到达对方，对方就一直在等待，除非对方直接挂电话。先说的话先到，后说的话后到，有顺序。

UDP 就像寄一封信，发信者只管发，不管到。但是你的信封上必须写明对方的地址。发信者和收信者之间没有通路，靠邮电局联系。信发到时可能已经过了很久，也可能根本没有发到。先发的信未必先到，后发的也未必后到。

答案：TCP 是传输控制协议，提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前，必须先在双方之间建立一个 TCP 连接，之后才能传输数据。TCP 提供超时重发、丢弃重复数据、检验数据、流量控制等功能，保证数据能从一端传到另一端。

UDP 是用户数据报协议，是一个简单的面向数据报的运输层协议。UDP 不提供可靠性，它只是把应用程序传给 IP 层的数据报发送出去，但是并不保证它们能到达目的地。由于 UDP 在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输

速度很快。

18.2 网络协议问题

面试例题 1: If we divide the network 40.15.0.0 into two subnets, and the first one is 40.15.0.0/17, then the second subnet will be _____. (如果把一个网络 40.15.0.0 分为两个子网，第一个子网是 40.15.0.0/17，那么第二个子网将会是_____) [中国台湾某著名杀毒软件公司 2005 年 10 月面试题]

- A. 40.15.1.0/17
- B. 40.15.2.0/16
- C. 40.15.100.0/17
- D. 40.15.128.0/17

解析: 让主网分成两个网段，子网掩码分别是 0xff 0xff 0x80 0x00 和 0xff 0xff 0x00 0x00。

答案: D

面试例题 2: If a worm scans the hosts in Class A IP address space on a home PC, it is quite probably that the host will receive a lot of _____. (如果一个蠕虫病毒攻击了一个家用 PC 的 A 类地址主机的话，这个地址最有可能接收很多_____) [中国台湾某著名杀毒软件公司 2005 年 10 月面试题]

- A. HTTP response packet (HTTP 回应包)
- B. DNS response packet (DNS 回应包)
- C. ICMP destination unreachable packet (ICMP 目的无法抵达包)
- D. ARP response (ARP 回应)

解析: 大量发出 IP 请求，肯定很多不可达，返回不可达错误。

答案: C

面试例题 3: Before an IP datagram arrived at the destination, it _____. (在一个 IP 数据包到达目的地址之前，它_____) [中国台湾某著名杀毒软件公司 2005 年 10 月面试题]

- A. may be fragmented but never reassembled (可能成为碎片，而且不会重组)
- B. may be fragmented or reassembled (可能成为碎片，或者重组)
- C. can't be fragmented or reassembled (不能成为碎片，或者重组)
- D. can't be fragmented but may be reassembled (不能成为碎片，但是可能会重组)

解析: 网络问题，包未达到终点不可能重组，但可以分散成碎片。

答案: A

面试例题 4: In TCP/IP protocol stack, which of following is taken as an indication of congestion?