桩模块

比如对函数 A 做**单元测试**时，被测的函数单元下还包括了一个函数 B，为了更好的错误，定位错误，就要为函数 B 写桩，来模拟函数 B 的功能，保证其正确。

白盒测试

白盒测试(White-box **Testing**，又称逻辑驱动测试,结构测试),它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

对开发语言的支持：白盒**测试工具**是对源代码进行的测试，测试的主要内容包括词法分析与语法分析、静态错误分析、动态检测等。目前测试工具主要支持的开发语言包括：标准 C、C＋＋、Visual C＋＋、Java、Visual J＋+等。

静态测试

动态通过评审文档、阅读代码等方式测试软件称为静态测试,通过运行程序测试软件称为测试.在动态测试中,通常使用白盒测试和**黑盒测试**从不同的角度设计测试用例,查找软件代码中的错误.

回归测试

回归测试的目的是在程序有修改的情况下，保证原有功能正常的一种测试策略和方法。
说白了就是，我们测试人员在对程序进行测试时发现 bug，然后返还程序员修改，程序员修改后发布新的软件包或新的软件补丁包给我们测试人员，我们就要重新对这个程序测试，已保证程序在修正了以前 bug 的情况下，正常运行，且不会带来新的错误的这样一个过程。一般情况下是不需要全面测试的，而是根据修改的情况进行有效的测试。

白盒测试有哪几种方法？

白盒测试也称结构测试或逻辑驱动测试，它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。
“白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。“白盒”法是穷举路径测试。

软件的缺陷等级应如何划分？

1．致命错误，可能导致本模块以及其他相关模块异常，死机等问题；
2．严重错误，问题局限在本模块，导致模块功能失效或异常退出
3．一般错误，模块功能部分失效；
4．建议问题，由问题提出人对测试对象的改进意见；

如果能够执行完美的黑盒测试，还需要进行白盒测试吗？(白盒与黑盒的区别)

任何工程产品（注意是任何工程产品）都可以使用以下两种方法之一进行测试。
黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。
白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。
软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

1、是否有不正确或遗漏的功能？

2、在接口上，输入是否能正确的接受？能否输出正确的结果？

3、是否有数据结构错误或外部信息（例如数据文件）访问错误？

4、性能上是否能够满足要求？

5、是否有初始化或终止性错误？

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查：

1、对程序模块的所有独立的执行路径至少测试一遍。

2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性，等等。

以上事实说明，**软件测试**有一个致命的缺陷，即测试的不完全、不彻底性。由于任何程序只能进行少量（相对于穷举的巨大数量而言）的有限的测试，在未发现错误时，不能说明程序中没有错误。

软件测试应该划分几个阶段?简述各个阶段应重点测试的点?各个阶段的含义？

大体上来说可分为单元测试,集成测试,系统测试,验收测试,每个阶段又分为以下五个步骤:

测试计划，测试设计，用例设计，执行结果，测试报告

初始测试集中在每个模块上，保证源代码的正确性，该阶段成为单元测试，主要用白盒测试方法。
接下来是模块集成和集成以便组成完整的软件包。集成测试集中在证实和程序构成问题上。主要采用黑盒测试方法，辅之以白盒测试方法。
软件集成后，需要完成确认和系统测试。确认测试提供软件满足所有功能、性能需求的最后保证。确认测试仅仅应用黑盒测试方法。

单元测试
单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。
集成测试
集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。
系统测试
系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。
验收测试
验收测试旨在向软件的购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集.
回归测试
回归测试是在软件维护阶段，对软件进行修改之后进行的测试。其目的是检验对软件进行的修改是否正确。

针对缺陷采取怎样的管理措施？

1. 要更好的管理缺陷，必须引入缺陷管理工具，商用的或者开源的都可。
2. 根据缺陷的生命周期，考虑缺陷提交的管理、缺陷状态的管理和缺陷分析的管理。
3. 所有发现的缺陷（不管是测试发现的还是走读代码发现的）都必须全部即时的、准确的提交到缺陷管理工具中，这是缺陷提交的管理。
4. 缺陷提交后，需要即时的指派给相应的开发人员，提交缺陷的人需要密切注意缺陷的状态，帮助缺陷的尽快解决。缺陷解决后需要即时对缺陷的修复进行验证。这样的目的有两个：一个是让缺陷尽快解决；二是方便后面缺陷的分析（保证缺陷相关的信息准确，如龄期等），这是缺陷状态的管理。
5. 为了更好的改进开发过程和测试过程，需要对缺陷进行分析，总结如缺陷的类别、缺陷的龄期分布等信息，这是缺陷分析的管理。

单元测试、集成测试、系统测试的侧重点是什么？

单元测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试，测试重点是系统的模块，包括子程序的正确性验证等。

集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求，组装成为子系统或系统，进行集成测试。实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。程序在某些局部反映不出来的问题，在全局上很可能暴露出来，影响功能的实现。测试重点是模块间的衔接以及参数的传递等。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。测试重点是整个系统的运行以及与其他软件的兼容性。

设计用例的方法、依据有那些？

白盒测试用例设计有如下方法:基本路径测试\等价类划分\边界值分析\覆盖测试\循环测试\数据流测试\程序插桩测试\变异测试.这时候依据就是详细设计说明书及其代码结构

黑盒测试用例设计方法:基于用户需求的测试\功能图分析方法\等价类划分方法\边界值分析方法\错误推测方法\因果图方法\判定表驱动分析方法\正交实验设计方法.依据是用户需求规格说明书,详细设计说明书。

测试用例通常包括那些内容？着重阐述编制测试用例的具体做法不同结构的用例包括的不一样（版本、编号、项目、设计人员、设计日期、输入、预期输出……）

软件测试用例的基本要素包括测试用例编号、测试标题、重要级别、测试输入、操作步骤、预期结果。

用例编号：测试用例的编号有一定的规则，比如系统测试用例的编号这样定义规则：PROJECT1-ST-001 ，命名规则是项目名称＋测试阶段类型（系统测试阶段）＋编号。定义测试用例编号，便于查找测试用例，便于测试用例的跟踪。

测试标题：对测试用例的描述，测试用例标题应该清楚表达测试用例的用途。比如“ 测试用户登录时输入错误密码时，软件的响应情况 ”.重要级别：定义测试用例的优先级别，可以笼统的分为“ 高 ”和“ 低 ”两个级别。一般来说，如果软件需求的优先级为“ 高 ”，那么针对该需求的测试用例优先级也为“ 高 ”；反之亦然，测试输入：提供测试执行中的各种输入条件。根据需求中的输入条件，确定测试用例的输入。测试用例的输入对软件需求当中的输入有很大的依赖性，如果软件需求中没有很好的定义需求的输入，那么测试用例设计中会遇到很大的障碍。

操作步骤：提供测试执行过程的步骤。对于复杂的测试用例，测试用例的输入需要分为几个步骤完成，这部分内容在操作步骤中详细列出。

预期结果：提供测试执行的预期结果，预期结果应该根据软件需求中的输出得出。如果在实际测试过程中，得到的实际测试结果与预期结果不符，那么测试不通过；反之则测试通过。

描述使用 bugzilla 缺陷管理工具对软件缺陷（BUG）跟踪的管理的流程

1) 测试人员或开发人员发现 bug 后，判断属于哪个模块的问题，填写 bug 报告后，系统会自动通过 Email 通知项目组长或直接通知开发者。

2) 经验证无误后，修改状态为 VERIFIED.待整个产品发布后，修改为 CLOSED.

3) 还有问题，REOPENED，状态重新变为“New"，并发邮件通知。

4) 项目组长根据具体情况，重新 reassigned 分配给 bug 所属的开发者。

5) 若是，进行处理，resolved 并给出解决方法。（可创建补丁附件及补充说明）

6) 开发者收到 Email 信息后，判断是否为自己的修改范围。

7) 若不是，重新 reassigned 分配给项目组长或应该分配的开发者。

8) 测试人员查询开发者已修改的 bug，进行重新测试。（可创建 test case 附件）

一、判断题

1．软件测试的目的是尽可能多的找出软件的缺陷。（Y）

2．Beta 测试是验收测试的一种。（Y）

3．验收测试是由最终用户来实施的。（N）

4．项目立项前测试人员不需要提交任何工件。（Y）

5．单元测试能发现约 80%的软件缺陷。（Y）

6．代码评审是检查源代码是否达到模块设计的要求。（N）

7．自底向上集成需要测试员编写驱动程序。（Y）

8．负载测试是验证要检验的系统的能力最高能达到什么程度。（N）

9．测试人员要坚持原则，缺陷未修复完坚决不予通过。（N）

10．代码评审员一般由测试员担任。（N）

11．我们可以人为的使得软件不存在配置问题。（N）

12．集成测试计划在需求分析阶段末提交。（N）

二、选折

1．软件验收测试的合格通过准则是：（ABCD）

A． 软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。

B． 所有测试项没有残余一级、二级和三级错误。

C． 立项审批表、需求分析文档、设计文档和编码实现一致。

D． 验收测试工件齐全。

2．软件测试计划评审会需要哪些人员参加？（ABCD）

A．项目经理

B．SQA 负责人

C．配置负责人

D．测试组

3．下列关于 alpha 测试的描述中正确的是：（AD）

A．alpha 测试需要用户代表参加

B．alpha 测试不需要用户代表参加

C．alpha 测试是系统测试的一种

D．alpha 测试是验收测试的一种

4．测试设计员的职责有：（BC）

A．制定测试计划

B．设计测试用例

C．设计测试过程、脚本

D．评估测试活动

5．软件实施活动的进入准则是：（ABC）

A．需求工件已经被基线化

B．详细设计工件已经被基线化

C．构架工件已经被基线化

D．项目阶段成果已经被基线化

三、添空

1.软件验收测试包括：正式验收测试，alpha 测试，beta 测试。

2.系统测试的策略有：功能测试，性能测试，可靠性测试，负载测试，易用性测试，强度测试，安全测试，配置测试，安装测试，卸载测试，文挡测试，故障恢复测试，界面测试，容量测试，兼容性测试，分布测试，可用性测试，（有的可以合在一起，分开写只要写出 15 就满分哦）

3.设计系统测试计划需要参考的项目文挡有：软件测试计划，软件需求工件和迭代计划。

4.对面向过程的系统采用的集成策略有：自顶向下，自底向上两种。

5.（这题出的有问题哦，详细的 5 步骤为~~）通过画因果图来写测试用例的步骤为：

（1）分析软件规格说明描述中，哪些是原因（即输入条件或输入条件的等价类），哪些是结果（即输出条件），并给每个原因和结果赋予一个标识符。

（2）分析软件规格说明描述中的语义，找出原因与结果之间，原因与原因之间对应的是什么关系？根据这些关系，画出因果图。

（3）由于语法或环境限制，有些原因与原因之间，原因与结果之间的组合情况不可能出现。为表明这些特殊情况，在因果图上用一些记号标明约束或限制条件。

（4）把因果图转换成判定表。

（5）把判定表的每一列拿出来作为依据，设计测试用例。

四、简答（资料是搜集整理的，感谢前辈的解题）无

1.区别阶段评审的与同行评审

同行评审目的:发现小规模工作产品的错误,只要是找错误;

阶段评审目的:评审模块 阶段作品的正确性 可行性 及完整性

同行评审人数:3-7 人 人员必须经过同行评审会议的培训,由 SQA 指导

阶段评审人数:5 人左右 评审人必须是专家 具有系统评审资格

同行评审内容:内容小 一般文档 < 40 页, 代码 < 500 行

阶段评审内容: 内容多,主要看重点

同行评审时间:一小部分工作产品完成

阶段评审时间: 通常是设置在关键路径的时间点上!

2.什么是软件测试

为了发现程序中的错误而执行程序的过程

3 简述集成测试的过程

系统集成测试主要包括以下过程:

1. 构建的确认过程。

2. 补丁的确认过程。

3. 系统集成测试测试组提交过程。

4. 测试用例设计过程。

5. 测试代码编写过程。

6. Bug 的报告过程。

7. 每周/每两周的构建过程。

8. 点对点的测试过程。

9. 组内培训过程。

4 怎么做好文档测试

仔细阅读,跟随每个步骤,检查每个图形,尝试每个示例。P142

检查文档的编写是否满足文档编写的目的

内容是否齐全，正确

内容是否完善

标记是否正确

5 白盒测试有几种方法

总体上分为静态方法和动态方法两大类。

静态：关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义

动态：语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。

6 系统测试计划是否需要同行审批，为什么

需要，系统测试计划属于项目阶段性关键文档，因此需要评审。

7Alpha 测试与 beta 的区别

Alpha 测试 在系统开发接近完成时对应用系统的测试；测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。

Beta 测试 当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。

8 比较负载测试，容量测试和强度测试的区别

负载测试：在一定的工作负荷下，系统的负荷及响应时间。

强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。

容量测试：容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。

华为 C/C++笔试题(附答案)
发布于:软件开发网 来源:互联网 作者:西城网络整理 时间:2008-12-13 点击: 1335

1.写出判断 ABCD 四个表达式的是否正确, 若正确, 写出经过表达式中 a 的值(3 分)
int a = 4;
(A)a += (a++); (B) a += (++a) ;(C) (a++) += a;(D) (++a) += (a++);
关键词：C 语言面试题 C++面试题 华为面试题
本文地址：http://www.teecool.com/post/2007081104.html
内容正文：

1.写出判断 ABCD 四个表达式的是否正确, 若正确, 写出经过表达式中 a 的值(3 分)
int a = 4;
(A)a += (a++); (B) a += (++a) ;(C) (a++) += a;(D) (++a) += (a++);
a = ?
答：C 错误，左侧不是一个有效变量，不能赋值，可改为(++a) += a;
改后答案依次为 9,10,10,11

2.某 32 位系统下, C++程序，请计算 sizeof 的值(5 分).
char str[] = “http://www.ibegroup.com/”
char *p = str ;
int n = 10;
请计算

```
sizeof (str ) = ? （1）
sizeof ( p ) = ? （2）
sizeof ( n ) = ? （3）
void Foo ( char str[100]){
请计算
sizeof( str ) = ? （4）
}
void *p = malloc( 100 );
请计算
sizeof ( p ) = ? （5）
```
答：（1）17 （2）4 （3） 4 （4）4 （5）4

3. 回答下面的问题. (4 分)
(1).头文件中的 ifndef/define/endif 干什么用？预处理
答：防止头文件被重复引用
(2). #i nclude 和 #i nclude “filename.h” 有什么区别？
答：前者用来包含开发环境提供的库头文件，后者用来包含自己编写的头文件。
(3).在 C++ 程序中调用被 C 编译器编译后的函数，为什么要加 extern “C”声明？
答：函数和变量被 C++编译后在符号库中的名字与 C 语言的不同，被 extern "C"修饰的变
量和函数是按照 C 语言方式编译和连接的。由于编译后的名字不同，C++程序不能直接调
用 C 函数。C++提供了一个 C 连接交换指定符号 extern“C”来解决这个问题。
(4). switch()中不允许的数据类型是?
答：实型

4. 回答下面的问题(6 分)
```
(1).Void GetMemory(char **p, int num){
*p = (char *)malloc(num);
}
void Test(void){
char *str = NULL;
GetMemory(&str, 100);
strcpy(str, "hello");
printf(str);
}
```
请问运行 Test 函数会有什么样的结果？
答：输出“hello”
```
(2). void Test(void){
char *str = (char *) malloc(100);
strcpy(str, “hello”);
free(str);
if(str != NULL){
```

```
strcpy(str, “world”);
printf(str);
}
}
```
请问运行 Test 函数会有什么样的结果？
答：输出“world”
(3). char *GetMemory(void){
```
char p[] = "hello world";
return p;
}
void Test(void){
char *str = NULL;
str = GetMemory();
printf(str);
}
```
请问运行 Test 函数会有什么样的结果？
答：无效的指针，输出不确定

5. 编写 strcat 函数(6 分)
已知 strcat 函数的原型是 char *strcat (char *strDest, const char *strSrc);
其中 strDest 是目的字符串，strSrc 是源字符串。
（1）不调用 C++/C 的字符串库函数，请编写函数 strcat
答：
VC 源码：
```
char * __cdecl strcat (char * dst, const char * src)
{
char * cp = dst;
while( *cp )
cp++; /* find end of dst */
while( *cp++ = *src++ ) ; /* Copy src to end of dst */
return( dst ); /* return dst */
}
```
（2）strcat 能把 strSrc 的内容连接到 strDest，为什么还要 char* 类型的返回值？
答：方便赋值给其他变量

6.MFC 中 CString 是类型安全类么？
答：不是，其它数据类型转换到 CString 可以使用 CString 的成员函数 Format 来转换

7.C++中为什么用模板类。
答：(1)可用来创建动态增长和减小的数据结构
（2）它是类型无关的，因此具有很高的可复用性。
（3）它在编译时而不是运行时检查数据类型，保证了类型安全

（4）它是平台无关的，可移植性
（5）可用于基本数据类型

8.CSingleLock 是干什么的。
答：同步多个线程对一个数据类的同时访问

9.NEWTEXTMETRIC 是什么。
答：物理字体结构，用来设置字体的高宽大小

10.程序什么时候应该使用线程，什么时候单线程效率高。
答：1. 耗时的操作使用线程，提高应用程序响应
2. 并行操作时使用线程，如 C/S 架构的服务器端并发线程响应用户的请求。
3. 多 CPU 系统中，使用线程提高 CPU 利用率
4. 改善程序结构。一个既长又复杂的进程可以考虑分为多个线程，成为几个独立或半独
立的运行部分，这样的程序会利于理解和修改。
其他情况都使用单线程。

11.Windows 是内核级线程么。
答：见下一题

12.Linux 有内核级线程么。
答：线程通常被定义为一个进程中代码的不同执行路线。从实现方式上划分，线程有两
种类型：“用户级线程”和“内核级线程”。 用户线程指不需要内核支持而在用户程序
中实现的线程，其不依赖于操作系统核心，应用进程利用线程库提供创建、同步、调度
和管理线程的函数来控制用户线程。这种线程甚至在象 DOS 这样的操作系统中也可实现
，但线程的调度需要用户程序完成，这有些类似 Windows 3.x 的协作式多任务。另外一
种则需要内核的参与，由内核完成线程的调度。其依赖于操作系统核心，由内核的内部
需求进行创建和撤销，这两种模型各有其好处和缺点。用户线程不需要额外的内核开支
，并且用户态线程的实现方式可以被定制或修改以适应特殊应用的要求，但是当一个线
程因 I/O 而处于等待状态时，整个进程就会被调度程序切换为等待状态，其他线程得不
到运行的机会；而内核线程则没有各个限制，有利于发挥多处理器的并发优势，但却占
用了更多的系统开支。
Windows NT 和 OS/2 支持内核线程。Linux 支持内核级的多线程

13.C++中什么数据分配在栈或堆中，New 分配数据是在近堆还是远堆中？
答：栈：存放局部变量，函数调用参数,函数返回值，函数返回地址。由系统管理
堆：程序运行时动态申请，new 和 malloc 申请的内存就在堆上

14.使用线程是如何防止出现大的波峰。
答：意思是如何防止同时产生大量的线程，方法是使用线程池，线程池具有可以同时提
高调度效率和限制资源使用的好处，线程池中的线程达到最大数时，其他线程就会排队
等候。

15 函数模板与类模板有什么区别？
答：函数模板的实例化是由编译程序在处理函数调用时自动完成的，而类模板的实例化
必须由程序员在程序中显式地指定。

16 一般数据库若出现日志满了，会出现什么情况，是否还能使用？
答：只能执行查询等读操作，不能执行更改，备份等写操作，原因是任何写操作都要记
录日志。也就是说基本上处于不能使用的状态。

17 SQL Server 是否支持行级锁，有什么好处？
答：支持，设立封锁机制主要是为了对并发操作进行控制，对干扰进行封锁，保证数据
的一致性和准确性,行级封锁确保在用户取得被更新的行到该行进行更新这段时间内不
被其它用户所修改。因而行级锁即可保证数据的一致性又能提高数据操作的迸发性。

18 如果数据库满了会出现什么情况，是否还能使用？
答：见 16

19 关于内存对齐的问题以及 sizof()的输出
答：编译器自动对齐的原因：为了提高程序的性能，数据结构（尤其是栈）应该尽可能
地在自然边界上对齐。原因在于，为了访问未对齐的内存，处理器需要作两次内存访问
；然而，对齐的内存访问仅需要一次访问。

20 int i=10, j=10, k=3; k*=i+j; k 最后的值是？
答：60，此题考察优先级，实际写成： k*=(i+j);，赋值运算符优先级最低

21.对数据库的一张表进行操作,同时要对另一张表进行操作,如何实现?
答：将操作多个表的操作放入到事务中进行处理

22.TCP/IP 建立连接的过程?(3-way shake)
答：在 TCP/IP 协议中，TCP 协议提供可靠的连接服务，采用三次握手建立一个连接。
第一次握手：建立连接时，客户端发送 syn 包(syn=j)到服务器，并进入 SYN_SEND 状
态，等待服务器确认；
第二次握手：服务器收到 syn 包，必须确认客户的 SYN（ack=j+1），同时自己也发送一个
SYN 包（syn=k），即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态；
第三次握手：客户端收到服务器的 SYN＋ACK 包，向服务器发送确认包 ACK(ack=k+1)
，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

23.ICMP 是什么协议,处于哪一层?
答：Internet 控制报文协议，处于网络层（IP 层）

24.触发器怎么工作的?
答：触发器主要是通过事件进行触发而被执行的，当对某一表进行诸如 UPDATE、INSERT
、 DELETE 这些操作时，数据库就会自动执行触发器所定义的 SQL 语句，从而确保对数
据的处理必须符合由这些 SQL 语句所定义的规则。

25.winsock 建立连接的主要实现步骤?
答：服务器端：socker()建立套接字，绑定（bind）并监听（listen），用 accept（）
等待客户端连接。
客户端：socker()建立套接字，连接（connect）服务器，连接上后使用 send()和 recv（
），在套接字上写读数据，直至数据交换完毕，closesocket()关闭套接字。
服务器端：accept（）发现有客户端连接，建立一个新的套接字，自身重新开始等待连
接。该新产生的套接字使用 send()和 recv（）写读数据，直至数据交换完毕，closesock
et()关闭套接字。

26.动态连接库的两种方式?
答：调用一个 DLL 中的函数有两种方法：
1．载入时动态链接（load-time dynamic linking），模块非常明确调用某个导出函数
，使得他们就像本地函数一样。这需要链接时链接那些函数所在 DLL 的导入库，导入库向
系统提供了载入 DLL 时所需的信息及 DLL 函数定位。
2.运行时动态链接(run-time dynamic linking)，运行时可以通过 LoadLibrary 或 Loa

dLibraryEx 函数载入 DLL。DLL 载入后，模块可以通过调用 GetProcAddress 获取 DLL 函数的
出口地址，然后就可以通过返回的函数指针调用 DLL 函数了。如此即可避免导入库文件了
。

27.IP 组播有那些好处?
答：Internet 上产生的许多新的应用，特别是高带宽的多媒体应用，带来了带宽的急剧
消耗和网络拥挤问题。组播是一种允许一个或多个发送者（组播源）发送单一的数据包
到多个接收者（一次的，同时的）的网络技术。组播可以大大的节省网络带宽，因为无
论有多少个目标地址，在整个网络的任何一条链路上只传送单一的数据包。所以说组播
技术的核心就是针对如何节约网络资源的前提下保证服务质量。