2014/4/23

腾讯实习生面试

一面题目：

1. TCP三步握手、四步挥手

|  |
| --- |
| http://www.tcpipguide.com/free/diagrams/tcpopen3way.png |
| 细节：http://www.tcpipguide.com/free/t\_TCPBasicOperationConnectionEstablishmentManagement.htm |

1. http协议简介，状态码

Hyper Text Transfer Protocol, http是用于从WWW服务器传输超文本到本地浏览器的传送协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。

|  | 已定义范围 | 分类 |
| --- | --- | --- |
| 1XX | 100-101 | 信息提示 |
| 2XX | 200-206 | 成功 |
| 3XX | 300-305 | 重定向 |
| 4XX | 400-415 | 客户端错误 |
| 5XX | 500-505 | 服务器错误 |

|  |
| --- |
| 200 OK 服务器成功处理了请求（这个是我们见到最多的） |
| 301/302 Moved Permanently（重定向）请求的URL已移走。Response中应该包含一个Location URL, 说明资源现在所处的位置 |
| 304 Not Modified（未修改）客户的缓存资源是最新的， 要客户端使用缓存 |
| 404 Not Found 未找到资源 |
| 501 Internal Server Error服务器遇到一个错误，使其无法对请求提供服务 |

详解

<http://www.cnblogs.com/TankXiao/archive/2013/01/08/2818542.html>

<http://www.360doc.com/content/10/0930/17/3668821_57590979.shtml>

1. IO复用模型

IO的方式有阻塞IO、非阻塞IO模型、IO复用、信号驱动、异步IO等

阻塞IO是最通用的IO类型，使用这种模型进行数据接收的时候，在数据没有到之前程序会一直等待。



当把套接字设置成非阻塞的IO，则对每次请求，内核都不会阻塞，会立即返回；当没有数据的时候，会返回一个错误。

 使用IO复用模型可以在等待的时候加入超时的时间，当超时时间没有到达的时候与阻塞的情况一致，而当超时时间到达仍然没有数据接收到，系统会返回，不再等待。

select()函数按照一定的超时时间轮询，直到需要等待的套接字有数据到来，利用recvfrom()函数，将数据复制到应用层。

 信号驱动的IO在进程开始的时候注册一个信号处理的回调函数，进程继续执行，当信号发生时，即有了IO的时间，这里即有数据到来，利用注册的回调函数将到来的函数用recvfrom()接收到。

 异步IO与前面的信号驱动IO相似，其区别在于信号驱动IO当数据到来的时候，使用信号通知注册的信号处理函数，而异步IO则在数据复制完成的时候才发送信号通知注册的信号处理函数。



函数select()和pselect()用于IO复用，除了使用select()进行文件描述符的监视，还有一组函数也可以完成相似的功能，即函数poll()和函数ppoll()。

详解：http://wenku.baidu.com/link?url=1gpMuhsHVun3POm42jG\_oniHAydwvavAoQ4xwdIqCI6uW7bBHwxehJyrDjF5t\_tEiD-JBTiInCLmLJJtVSe1tvMJ-xzBA\_hLxxlrzPxa44C###

1. 多线程举例

线程，进程中某个单一顺序的控制流，进程的实体

一个进程可有多个线程，一个线程必有一个父进程

线程与父进程共享进程所有资源

线程本身无系统资源

不同线程可运行在不同处理器中

进程是资源管理的最小单位，线程是程序执行的最小单位

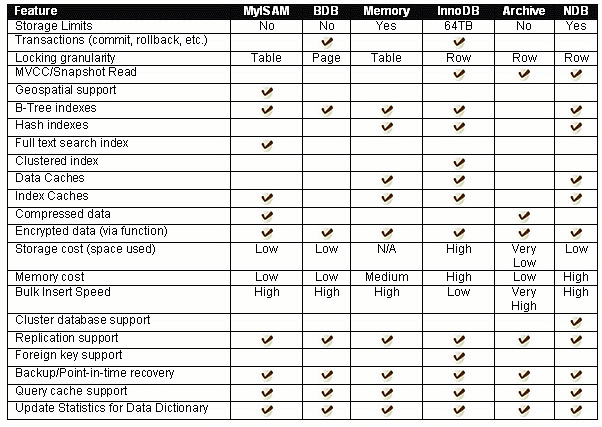
各个线程使用相同的地址空间，不同线程共享绝大部分数据，启动线程开销小

1. 数据库MySQL几种引擎

与MySQL一起提供的各种存储引擎在设计时考虑了不同的使用情况。为了更有效地使用插件式存储体系结构，最好了解各种存储引擎的优点和缺点。

在下面的表格中，概要介绍了与MySQL一起提供的存储引擎：

**图14.2：存储引擎比较**



下述存储引擎是最常用的：

·         MyISAM：默认的MySQL插件式存储引擎，它是在Web、数据仓储和其他应用环境下最常使用的存储引擎之一。注意，通过更改STORAGE\_ENGINE配置变量，能够方便地更改MySQL服务器的默认存储引擎。

·         InnoDB：用于事务处理应用程序，具有众多特性，包括ACID事务支持。

·         BDB：可替代InnoDB的事务引擎，支持COMMIT、ROLLBACK和其他事务特性。

·         Memory：将所有数据保存在RAM中，在需要快速查找引用和其他类似数据的环境下，可提供极快的访问。

·         Merge：允许MySQL DBA或开发人员将一系列等同的MyISAM表以逻辑方式组合在一起，并作为1个对象引用它们。对于诸如数据仓储等VLDB环境十分适合。

·         Archive：为大量很少引用的历史、归档、或安全审计信息的存储和检索提供了完美的解决方案。

·         Federated：能够将多个分离的MySQL服务器链接起来，从多个物理服务器创建一个逻辑数据库。十分适合于分布式环境或数据集市环境。

·         Cluster/NDB：MySQL的簇式数据库引擎，尤其适合于具有高性能查找要求的应用程序，这类查找需求还要求具有最高的正常工作时间和可用性。

·         Other：其他存储引擎包括CSV（引用由逗号隔开的用作数据库表的文件），Blackhole（用于临时禁止对数据库的应用程序输入），以及Example引擎（可为快速创建定制的插件式存储引擎提供帮助）。

请记住，对于整个服务器或方案，你并不一定要使用相同的存储引擎，你可以为方案中的每个表使用不同的存储引擎，这点很重要。