

2.2 进程和线程

进程：进程包含很多线程，包含逻辑内存，文件/网络句柄，进程只是容器，是调度和分配资源的基本单位。

线程：线程是内存的组成部分，操作系统真正运行的是线程。是分配资源的基本单位，但不是调度资源的基本单位。

栈：堆栈

PC：下一条执行的指令地址，指令放在内存中。

TLS：线程局部存储，它主要是为了避免多个线程同时访存同一全局变量或者静态变量时所导致的冲突，尤其是多个线程同时需要修改这一变量时。为了解决这个问题，我们可以通过 TLS 机制，为每一个使用该全局变量的线程都提供一个变量值的副本，每一个线程均可以独立地改变自己的副本，而不会和其它线程的副本冲突。

2.3 存储和寻址

存储：硬盘，长久存放，容量大

内存，关机就消失，快速随机访问

CPU-缓存

寄存器

以上从上到下，访问速度变快，价格也越贵。

寻址空间：每个进程里面的指针，可以取到的地址，每个进程的寻址空间相互独立。跟内存大小无关，跟位数相关。

32 位->4G

64 位->10¹⁹ 字节

寻址 `int n = *p; -> MOV EAX,[EBX]`

在物理内存中

指针 `p`--->逻辑内存，进程独立 2³² 或 2⁶⁴----->物理内存----->寄存器

^

|

分页

虚拟内存（通过特定算法，交换物理内存中不常用的数据，类

似缓存）

2.4 进程间通信

线程之间是共享内存的，相互通信很方便，只要防止线程访问安全性问题（同步）。

但是进程间如何通信？

如果面试官在问线程和进程的不同点，你侧重说内存的话，就可能会问进程间通信方式

能说出多少说多少通信方法，重点是比较。

1.通过文件

2.信号，是一个数字，代表特定含义(Signal)

3.消息队列

4.管道(单向)/命名管道(双向), `cat test.log | grep -e "ERROR" --color | wc`,这里的|就是管道，`cat,grep,wc` 是进程

5.共享内存

6.同步机制，如信号量

7.Socket(最重要的，一定要说，这个可以用于不同机器间的通信，上面的只能在一台机器)

2.5 操作系统的问题

1.进程与线程区别？

2.简要说说寻址的过程？

3.32 位操作系统和 64 位操作系统区别是什么？

4.说出你知道的进程间通信机制？

5.简要说一下中断的概念和流程？

三、网络

3.1 网络基础

应用层

表示层(忽略)

会话层(忽略)

传输层

网络层

数据链路层

物理层

不可靠：丢包、重复包

出错

乱序

不安全：中间人攻击

窃取

篡改

3.2 不可靠性的解决：

滑动窗口协议：

TCP 协议中使用

维持发送方/接收方缓冲区

主要分成四种情况，

1.已发送，已 Ack

2.已发送，未 Ack，在窗口中

3.待发送，未 Ack，在窗口中

4.未发送，未 Ack

丢 Ack 情况：

窗口都是已发送，未 Ack，窗口满。

解决办法是：超时重传机制

3.4 TCP 问题

1.谈谈 TCP 协议要达成什么目的？

2.滑动窗口协议是怎么运行的？解决了什么问题？

3.TCP 协议如何做到流量控制(滑动窗口)？如何做到拥塞控制？

3.5 TCP 链接建立(三次握手)和断开(四次挥手)

客户端

服务器

LISTEN

客户端发 SYN
 SYN_SEND
 服务端发 SYN,ACK
 ESTABLISHED
 客户端发 ACK
 ESTABLISHED
 客户端发数据

发起方 接收方
 ESTABLISHED ESTABLISHED
 发送方发 FIN
 FIN_WAIT_1
 CLOSE_WAIT(接收方还没准备好断开)
 接收方发 ACK
 FIN_WAIT_2
 接收方发 FIN
 LAST_ACK(等发送方的最后一个 ACK)
 TIME_WAIT
 发送方发 ACK
 CLOSED
 接收方等待一段时间(怕接收方没收到前一个 ACK)
 CLOSED

常见问题

- 1.为什么要三次握手？流程是怎样的？
- 2.断开是为什么要四次挥手？流程是怎样的？
- 3.断开连接的发起方经历了哪些状态？
- 4.为何断开连接发起方最后要进入超时等待 TIME_WAIT 状态？

四、数据库

4.1 数据库概述

关系型数据库

- 1.基于关系代数理论
- 2.缺点：表结构不直观，实现复杂，速度慢
- 3.优点：健壮性高，社区庞大

4.2 JOIN 和 GROUP BY

内连接

```
SELECT * FROM 'product' p JOIN 'category' c
```

```
ON p.'categoryId' = c.'categoryId'
```

```
GROUP BY p.'categoryId'
```

外链接

```
SELECT * FROM 'product' p join (
```

```
SELECT p.'categoryId','categoryName',MIN(p.'price') as min_price FROM 'product' p LEFT JOIN 'category' c
```

```
ON p.'categoryId' = c.'categoryId'
```

```
GROUP BY p.'categoryId','categoryName') AS cat_min
```

```
ON p.'categoryId' = cat_min.categoryId
```

```
WHERE p.'price' = cat_min.min_price
```

括号里面是子查询

4.3 事务和乐观锁

事务：要符合 ACID 特性

1.Atomicity:原子性

2.Consistency:一致性

3.Isolation:独立性，隔离性(关键)

4.Durability:持久性

事务的隔离级别

1.Read uncommitted

2.Read Committed

3.Repeatable Reads

4.Serializable

事务：

```
BEGIN;
```

```
SET AUTOCOMMIT = 0;
```

```
SELECT count FROM 'product' WHERE 'productId' = 2 FOR UPDATE;
```

(FOR UPDATE:加锁)

乐观锁(版本保护)(用于冲突几率多的时候)

锁太耗费资源，乐观锁以一种逻辑上的实现。

1.读取数据，记录 version

2.修改数据

3.检查和提交数据，必须放在一行中

```
UPDATE 'product' SET 'count' = 45 WHERE 'productId' = 2 AND 'count' = 46;
```

4.4 数据库问题

1.介绍一下关系型数据库的理论基础？(1-3 范式)

2.给定一个场景，请设计表结构？(要做到 3 范式)

3.给定表结构，请按要求写出 SQL 语句？

4.什么是事务的 ACID 属性？事务有哪些隔离级别？

5.事务的性能太慢怎么办？(用乐观锁)

4.5 数据库的索引

4.6 索引的实现

B 树

B+树

二叉树优化比较次数，而 B/B+树优化磁盘读写次数(有多个子节点)

数据库索引的作用

1.加快查找速度

2.约束数据的值，如 UNIQUE INDEX

数据库索引的分类

1.Clustered Index，每个表至多一个

2.Non-clustered Index

B 树和 B+树的区别

B 树和二叉搜索树的区别

怎样选择给那些字段加上索引

4.7 数据库的连接池

普通模式存在的问题：

1.建立连接要花很多时间

2.每个用户建立连接造成浪费

连接池的模式：先建立好很多连接，用户请求的时候再分配，不用的时候回收。

8.设计模式

8.1 设计模式简介

SignleTon 单例模式优缺点

确保全局至多只有一个对象

用于：构造缓慢的对象，需要统一管理的资源

缺点：很多全局状态，线程安全性

如何创建单例

8.2 State 模式（状态模式）

变继承关系变成组合关系