网络

OSI 七层模型

应用层->表示层->会话层->传输层->网络层->数据链路层->物理层

TCP/IP四层模型

应用层、传输层、网络层、网络接口层

为什么网络要分层

1、各层之间相互独立 2、提高了灵活性和可替换性 3、大问题化小

应用层有哪些常见的协议?

HTTP、SMTP、POP3/IMAP、FTP、Telet、SSH、RTP、DNS

传输层有哪些常见的协议?

TCP、UDP

网络层有哪些常见的协议?

IP、ARP、ICMP、NAT、OSPF、RIP、BGP

输入URL到页面展示发生了什么?

URL DNS->IP IP->TCP TCP->HTTP HTTP->HTTP HTTP->HTML... 关闭TCP或等待关闭

HTTP状态码:

1XX：接受的请求正在处理

200、204、206

301、302、304

400、401、403、404、

500、502、503

http1.0 1.1 2.0 3.0 https区别

http vs https 端口号、URL前缀、安全性和资源消耗、SEO

1.0 vs 1.1 连接方式、状态响应码、缓存机制、带宽、Host头部

1.1 vs 2.0 连接方式、二进制帧、头部压缩、服务器推送

2.0 vs 3.0 连接方式、响应时间、头部压缩、连接迁移、错误恢复、安全性

WebSocket和http区别

单/双向实时通信协议、协议前缀、扩展、数据包头部

GET和POST区别

语义、是否幂等、缓存、请求格式、安全性

ping的信息和流程

1. 目标主机IP、序列号、TTL、RTT、统计结果
2. 向目标主机->ICMP Echo Request，目标主机返回->ICMP Echo Reply

TCP/UDP区别

面向连接、可靠传输、有状态、传输效率、传输形式、首部开销、广播/多播服务

TCP三次握手

SYN（SEQ = x）SYN-SEND server收到--->半连接队列

SYN + ACK (SEQ = y, ACK = x + 1) SYN-RECV

ACK (ACK = y) ESTABLISHED(双方) server收到--->全连接队列

为什么三次握手

双方确认自己与对方的发送与接收是正常的1. 2. 3.

一旦完成前两次握手，第三次握手可以携带数据

TCP四次挥手

FIN（SEQ = x）FIN-WAIT-1

ACK (ACK = x + 1) CLOSE-WAIT FIN-WAIT-2

FIN（SEQ = y）FIN-WAIT-1 LAST-ACK

ACK (ACK = y + 1) TIME-WAIT CLOSE (2MSL)

IP 协议的作用是什么?

IP（Internet Protocol，网际协议） 是 TCP/IP 协议中最重要的协议之一，属于网络层的协议，主要作用是定义数据包的格式、对数据包进行路由和寻址，以便它们可以跨网络传播并到达正确的目的地。

操作系统

操作系统的功能

进程和线程的管理、存储管理、文件管理、设备管理、网络管理、安全管理

进程运行的级别

用户态和进程态

为什么分这两个? 1、安全考量 2、系统性能

用户态和内核态切换方式

1. 系统调用：文件管理、设备管理、进程管理、内存管理
   步骤?

- 用户态程序发起系统调用，Trap(一种中断)
- cpu执行程序中断跳转到中断程序 开始进入内核态
- 处理完成主动触发Trap，切换回用户态

2. 中断

3. 异常

进程和线程区别

1. 线程是进程划分的更小单位
2. 各进程运行基本独立
3. 线程开销小，不利于资料管理和保护

线程好处?
进程切换开销大、多cpu多核计算机线程并发效率高、线程轻量一个进程创建多个线程、同一个进程下线程共享资源

为什么使用多线程?

1. 总体：计算机底层、互联网发展趋势
2. 计算机底层：单核时代、多核时代

线程同步方式
互斥锁、读写锁、信号量、屏障、事件

PCB内容
进程的描述信息、调度信息、资源需求情况、打开文件信息、处理机状态信息

进程状态
创建、就绪、运行、阻塞、结束

进程间的通信方式：包括管道（匿名和有名）、信号、消息队列、共享内存、信号量和套接字等

进程的调度算法：如先来先服务（FCFS）、最短作业优先（SJF）、优先级调度和时间片轮转、多级反馈队列调度算法

僵尸进程：子终止PCB未被回收 **孤儿进程**：父终止子未被回收 (linux:top)

死锁产生的必要条件：互斥、占有并等待、非抢占、循环等待

死锁处理策略：

1. 预防：破坏互斥、占有并等待、非抢占、循环等待
2. 避免：银行家算法
3. 检测：资源分配图 有无环路-->是否资源类仅一个资源-->不阻塞不独立进程消除边
4. 接触：撤销挂起或者抢占资源

内存管理主要做了什么?
内存的分配与回收、地址转换、内存扩充、内存映射、内存优化和内存安全

什么是内存碎片?
内存碎片分为内部碎片和外部碎片，前者是已分配但未被使用的内存，后者是未分配且无法使用的内存。
-->连续内存管理、非连续内存管理

连续内存管理：块式管理、伙伴系统管理（内？SLAB）

非连续内存管理：段式管理、页式管理、段页式管理

虚拟内存能力:
隔离进程、提升物理内存利用率、简化内存管理、多个进程共享物理内存、提高内存使用安全性、提供更大的可使用内存空间

CPU：MMU（地址翻译/地址转换） ===> 虚拟地址----->物理地址

没有虚拟内存:
1、随意访问到系统内存 2、多个进程互用彼此内存 3、所有数据指令写入内存

MMU将虚拟地址翻译为物理地址的主要机制有 3 种:
分段机制、分页机制、段页机制

段内逻辑信息：主程序段 MAIN、子程序段 X、数据段 D 及栈段 S
段表：**段号、段内偏移量**、段长、段类型

段表项被删除（软件错误/恶意行为）、还没创建（内存不足/无法分配）

页表：页号、页内偏移量

具体的地址翻译过程如下:
1、MMU 首先解析得到虚拟地址中的虚拟页（段）号;
2、通过虚拟页（段）号去该应用程序的页（段）表中取出对应的物理页（段）号（找到对应的页（段）表项）;
3、用该物理页号对应的物理页起始地址（物理地址）加上虚拟地址中的页（段）内偏移量得到最终的物理地址。

单机页表-->多级页表

（MMU：TLB）转址旁路缓存（快表）

换页机制有什么用?
物理页的内容-->磁盘

页缺失: 硬性页缺失、软性页缺失

页面置换算法:
OPT、FIFO、LRU、LFU、Clock（LRU）

FIFO缺点
1、经常访问或者需要长期存在的页面会被频繁调入调出
2、存在 Belady 现象

在段页式机制下，地址翻译的过程分为两个步骤:
段式地址映射。
页式地址映射。

分页和分段机制区别:
共同点:
1、非连续内存管理
2、地址映射
区别:
1、单位：页、单位 大小（不）固定
2、物理单位、长度是2的幂，逻辑单位、取决程序所需
3、分段出现外部碎片，分页有内部碎片
4、页表（多级页表），段表

局部性:
时间局部性是指一个数据项或指令在一段时间内被反复使用的特点，空间局部性是指一个数据项或指令在一段时间内与其相邻的数据项或指令被反复使用的特点。

- 对于分页机制按照时间和空间局部性...

文件系统:
存储管理、文件管理、目录管理、文件访问管理

硬链接和软链接有什么区别?
1、硬：inode，软：路径
2、删除?
3、目录/不存在的文件，文件系统?
4、ln / ln -s

提升文件系统性能?

优化硬件、合适的文件系统选型、运用缓存、避免磁盘过度使用、合理分区

磁盘调度算法:

FCFS、SSTF、SCAN、C-SCAN、LOOK、C-LOOK