WXG 公众号小程序组 后台二面 2024.11.22

项目+八股 50min

- DragonOS
  - 介绍了 Netlink，比如属于广义上的本机内核空间和用户空间的IPC，基于 POSIX API，有哪些应用场景等（用户空间的设备管理，路由管理）。
  - Netlink 为什么不共享内存？胡乱说了个避免数据竞争，没有深入思考过

> TODO：查漏补缺：融会贯通深入思考现有项目和其他的知识，说到底还是在问共享内存在这个场景中的弊端。

- 共享内存有哪些应用？没答上来，只知道也是IPC的一种
- 介绍一下 fork()
  - 说了一下子进程是父进程的拷贝，子进程拷贝了父进程栈资源和内存资源（更具体不是很清楚），一开始是一模一样的，然后通过返回值区分父子进程，父进程返回子进程的pid，子进程返回0（面试的时候好像说反了）
  - 父子进程是用了几块内存？是完全拷贝吗？
    - 面试官说现实情况是总体的内存大小是小于两倍的，我没想到 COW。
  - 复用的有什么？
    - 只说了栈和内存
    - redis 就是用 fork() 拷贝内存，实际上内存空间并不是 x2，为什么会产生这种现象？
      - 答了一下一部分共享，一部分独占重新开辟内存空间。面试官只说了有一点的合理性。

> TODO：查漏补缺：COW的原理，redis为什么会用 fork()，fork()的工作原理，为什么会产生这种现象。

- 问 IBM 项目的情况
  - 部署？
    - 测试机上都用容器部署
  - 是否有用集群部署
    - 没用k8s
  - 没用k8s那服务之间是怎么通信的？
    - 直接用 restful api，服务都是手动用容器部署的，说了容器网络
  - 微服务场景下，A 调 B，A 如何知道 B 的地址，以及 B 的服务列表
    - 说了用 api 网关，追问有哪些实现形式
      - 说了 k8s 的 ingress 或者自己部署 nginx
  - 到了网关之后，网关如何决策转发到哪个服务

        - 说了负载均衡
        - 除了负载均衡，API 网关做转发的时候还有什么判断的原则和标准吗
            - 没说出来

> TODO：查漏补缺：微服务架构下的服务发现，服务注册，服务调用，负载均衡，API 网关的工作原理，以及如何实现服务的动态发现和负载均衡。

- 有没有了解过 2pc，3pc哪些分布式事务协议
- 说一下2pc的原理
    - 只知道是两阶段提交，第二个阶段是 commit
    - 两阶段提交解决了什么问题？为什么一阶段不行？
    - 说了回滚，但是面试官说一阶段也可以回滚，然后就说不出所以然了

> TODO：查漏补缺：2pc的原理，为什么需要两阶段提交，一阶段和二阶段的具体工作流程，以及2pc的缺点。其他分布式事务协议的原理和应用场景。

- 问了黑框框那个 socket 项目
    - 为什么要用双端链表维护好友信息？
    - 没想起来，乱答一通，面试官又该问双端链表的优势
        - 回答了插入和删除的时间复杂度都是 O(1)
        - 弊端是？
            - 多维护一个指针，空间占用大
        - 应用场景？
            - 说了 LRU，特征头部尾部有区别的场景
    - 连接的时候 server 端需要调用哪些系统调用？
        - 说了 socket，bind，listen，accept
        - listen 和 accept 的区别？
            - listen 是监听，accept 是接受连接

> TODO：查漏补缺：计网基础

- 了解 I/O 多路复用吗？
    - 比如说说 epoll
        - 说了 epoll 的优势，不需要全量遍历，只需要遍历有事件的文件描述符（这一块说的是这个意思但是有点含糊）

> TODO：查漏补缺：I/O 多路复用的原理，epoll的优势，epoll的工作原理，epoll的使用场景。

- 多进程和多线程的区别？
    - 首先说了进程和线程的区别，进程独占资源，线程共享资源
    - 然后具体在多进程通信和多线程通信上有什么区别？

        - 说了进程通信需要用共享内存，消息队列，信号量等，线程通信直接用全局变量就行
        - 线程间通信更多的考虑的是数据竞争和同步问题
        - 进程切换开销更大
        - 语言和系统的会提供不同的并发模型，并发模型不同，会影响多线程的实现
- C++
    - 虚函数如何实现的?
        - 虚函数表和虚函数指针
        - 会占用实例的内存大小吗？答了如果派生类对象复用基类的函数的话，只占用一个指针的大小
    - 说一说印象最深的 C++ 的特性
        - 从跟 rust 的比较出发说了 C++20 的泛型类型约束，cue 到了 rust 的 trait 和 C++ 的 concept
    - 之前都在写 rust，为什么要转到 C++?
        - 说了用 rust 的原因，项目需要以及未来投资
        - rust 就业市场不佳
        - 说了 rust 是一群用来了 C++ 几十年的人写的，所以 rust 有很多 C++ 的影子
        - C++ 当前生态更成熟更好
- Linux
    - 了解过一些 Linux 的指令吗？随便说了一些
- mysql
    - 有没有建立过 mysql 索引？了解理论但是没有做过大型的业务项目所以没有。

笔试 25min

- 相交链表 判断两个单向链表是否相交
    - 感觉做过但是不太记得具体的那个数学规律，说了大概的思路是快慢指针和相交点到结尾路程相同，依稀记得到达尾部之后要回到头部，达到一个状态。
- 计算需要最少船数 GetMinBoat
    - 一开始的思路是先是用哈希表记录当前还剩下的人然后尽量去做刚好limit的匹配，如果不能实现就找第二大的
    - 后来想了一下直接排序然后从后往前遍历，依次取末尾和头部的数，如果满足就取，不满足就只取末尾的数，更新船数即可

反问

- 今天的表现，有哪些可以提升的地方
    - 算法问我是不是没有准备过（）我说准备的可能少了点（因为确实上一次面完之后就没有再刷题了，但是25分钟时间感觉确实只能遗憾离场了）这方面可以适当再刷一点题，但是也不用刷很多

    - 计算机基础都还ok的，主要是知识的广度这方面可以适当地再扩充一些，比如分布式事务和很常用的一些中间件
- 问技术栈
    - C++ 语言栈
    - 业界常用的中间件，只不过是内部的C++的版本，可以挑一两个练手的项目了解一下，就知道往哪方面去提升了