

- 迅雷面经
- 前言:
- 比较幸运,一面很简单

跟面试官说明了对 c++ 不熟悉,对 Java 熟悉,但是面试官还是问了很多 c++,遇到不会的自己也尝试取解决

- 一面: 20min 左右
- 介绍一下实习

c++ 的 stl 中 `unordered_map` 和 `map` 的区别,回答了一个是只对 key 进行 hash,一个将 key 进行 hash 之后用红黑树进行存放,所以迭代器进行遍历的时候是 key 有序的

说说你比赛的时候写过的一个题,本来想介绍一下矩阵快速幂,然后问一下面试官懂矩阵快速幂吗?答曰不会,然后开始讲快速幂了, balabala..,讲完之后,然后了解我第二天要回学校,然后好心地帮我询问 hr 是否安排第二面,然后能行

- 二面: 60min 左右

二面面试官感觉很强

跟他了解了一下迅雷的系统架构,感觉很强,可以学到很多

说一说 `unordered_map` 和 `map` 的区别

你刚刚说 `unordered_map` 和 `map` 的查询效率分别是 $O(1)$ 和 $O(\log n)$,一般来说前者比后者查询快,但是会出现前者比后者查询慢的情况吗?

以前没有想过这个问题,然后告知有一个 hash 常数的相关,前者需要 hash,所以有一个常数的概念。后者只需要重载 `<`,所以当数据量小的时候,后者的效率更高些

自己定义的类需要做为 `unordered_map` 的 key 的时候,需要重载 `==`,写一个 hash 函数

刚刚查阅资料:

复制代码

1. 数据量不大的情况下, $O(n)$ 并不见得比 $O(\log n)$ 快多少,有可能还要慢
2. rehash 问题, hash 需要将 key 通过 hash 函数映射成 `size_t`,再存储到一块预先申请好的连续空间上,因此当这些空间不够用了以后就需要重新分配空间并对原有元素进行重新计算 & 分配.这是很耗时间的. 如果能预估出所存元素的最大个数,可以用 3rehash 或构造函数一次申请足够的空间,这样就可以避免 rehash.
3. 冲突,当数据量很大时不可避免地会有冲突,这会一定程度上影响 hash 的性能,但事实上 C++ 已经把这个问题解决好了,我们可以完全透明地使用.

怎么判断一个进程所用的内存

用过 Linux 吧,说说你用过的一些命令,然后问了一下 top 命令的中的与内存相关的几个参数

c++ 与 Java 的区别,没回答上,大致说了一下 Java 语言的特性

Java 的反射说一下

c++ 的多态是怎么实现的,然后回答了一下虚表与虚指针

多态是 c++ 里边的,然后如果让你在 c 语言中实现一个多态,你怎么做呢?构造类似于虚表的数据结构,然后指向函数指针

一个系统,每秒只能处理 10w 的请求,然后这个时候有 50w 的请求过来了,你打算怎么处理,不使得这个系统发生异常

往流量控制这一块说了说,比如用一个阻塞队列。后边被问还有什么方法吗?面试官提醒了一个降级控制,就是对每一个任务设置一个级别,优先处理级别高的,级别低的任务的话可

能价值不高或者没有价值，所以可能不执行或者稍后执行
进程和线程

网络的 tcp 四次握手的 time_wait 状态说一下

讲讲 tcp 和 udp 的区别，和各自的应用场景，应用场景回答了一两种，感觉面试官不是很满意

讲讲 https，回答了加了 ssl 层，然后说了说两端进行通信的过程

· HR

闲聊

你了解公司为什么是技术氛围

说说一下你的大学生活

其他不是很清楚了