

集团官网 : <https://www.asml.com/asml/en/s427>

HMI官网 : <http://www.hermes-microvision.com/>

面试官 : <https://www.linkedin.com/in/yunxiang-chai-45030b79/> (yunxiang chai)

<https://www.linkedin.com/in/junliume/> (Jun Liu)

<https://www.linkedin.com/in/y8hsieh/> (Yung-Huan Hsieh)

注意事项 :

1. 回答问题一定要精炼, 有支撑点
2. 如果被考到技术问题注意要回答优化算法
3. 不要紧张, 注意作答时间
4. 请穿着衬衫, 保持一个积极向上的外表
5. 多复习在您面试之前, 分配好复习时间, 二叉树问题很重要

第一轮 : 简单自我介绍, coding 问题 : leet code 54题目 . Spiral Matrix

第二轮 : 主要聊个人经历和项目, 技术问题问的较少

问题 : stack 和 heap 内存方面的区别

Stack is used for static memory allocation and heap is for dynamic memory allocation both in RAM.

Stack stores all the local variables and some other bookkeeping data. LIFO order, the most recently reserved block is always the next block to be freed.

Heap is dynamic and no enforced pattern to allocation and deallocation of blocks from the heap. No dependencies between variables inside the heap and they can be accessed randomly, making it harder to keep track of which parts of the heap are allocated or free.

The heap is allocated for each program (process) by the runtime and the stack is allocated when a thread exists. Size of stack is set when a thread is created and size of heap is set when an application (process) starts, but can grow depends on requirement.

C++ 中 vector 和 Linked List 的比较

Vector: dynamic arrays. Good for random access and insertion and deletion in the back (take $O(1)$ time). But bad for insertion and deletion from the front or any other position. Usually laid out contiguously in memroy, so traversing it is efficient because CPU cache.

Linked List: good for delete and insertion at the beginning and the end, once found a particular node you can delete it or insert a new item after it in constant time.

第三轮 : 简单聊聊简历, coding 问题 :

Construct Tree from given preorder and Preorder traversals

就是 给了前序遍历和后序遍历, 要求复原二叉树

一轮电面问题 :

1. C++在private里用Virtual Function在subclass能不能set到?
能

2. Vector和array哪一个overhead少？

Array少， Vector是可变量

3. 动态declare一个variable， 这个variable存在哪里？

存在Heap里， 不在stack

4. 用锁， 影响performance吗？

没有锁

主要问题都集中在OS

二轮Skype面试：

就相当于白板， 在google doc开一个文档或者去开一个code pad敲代码。 三轮三个人。第一个全是coding问题， 问了排序的方法， 随便说出说几个进行code和比较， 问了队列和vector的区别。

Sorting methods:

bubble sort

insertion sort

merge sort

heap sort

quick sort

redix sort

Array contains a specific number of elements and a particular type so the compiler can reserve the required space when the program is compiled. You must deallocate them if dynamically allocate them. They cannot be returned from a function and cannot be copied or assigned directly.

Vector is dynamically sized sequence of objects, push_back and pop_back can operate on the end of the array and they are the basics of vectors. The vectors offers bounds checking with the at function if you need to. Can be returned from a function and can be copied or assigned.

第二个主要聊聊简历， 然后出了一道题， number of island。 用dfs或者bfs解， It原题

第三个没有coding， 聊聊图像处理以前的经验， 遇到的问题， 解决方案， 比较technical， 注重细节， 感觉面试官应该是搞图像的比较懂。还问了deep learning 网络的激活函数有哪些， 进行比较和为什么使用

Binary step function, linear function, sigmoid function, tanh function, rectified linear function, softmax function. Sigmoid functions and their combinations (softmax functions) are good for classifiers. Sigmoid and tanh functions are good at vanishing gradient problem.

第二个面试官和第三个比较友好随和面露微笑互动很好也不让人紧张。总体感觉公司搞的东西有前景， 工作环境还不了解毕竟没有在现场看过

偏重c++编程，但我问了你面试的时候他们说程序语言没那么重要，要看对知识算法的理解和思路，工作中可能要换成c++

Onsite :

2017年Onsite interview

第一轮：30分钟左右

和Director谈，聊简历和背景

第二轮：Li Chuan

深度的背景知识，调度和锁的问题，

Behavior问题：如何解决工作中的难题。

第三轮：Hu Guodong

Coding

1. Array的问题 在里面寻找target（先增后减的思路）

2. Leetcode原题，墙的问题，查砖怎么切可以看下来的最少。

第四轮：Liu Jun

难题：2个array，一个里面都是各种不同形状的螺丝钉，一个都是各种不同形状的螺母，无需排列，如何让他们一一对应。nlogn的问题

问了平行计算，分布式系统，OS system，CNN

HMI职位方向：

OS/distributed system

Parallel computing

Image

加面题目：

1. 无像删除edge，output二叉树（给定一个无向图，是一群pair，链接node，生成二叉树，边的数量有限制）

2. 写KMP

2018年新面经：

onsite HPC/Parallel Computing方向

1. Liu Jun Parallel Computing, OpenMP, SIMD。coding: bolts and nuts

2. Yunxiang Chai, coding: IP string to integer, System design: distributed compression system

3. Guodong Hu, coding: Trapping Rain Water I, II, Number of Islands

4. Wei Fang, resume and project

更新面经：

面试官按顺序：nan zhao, wentian zhou, lingling pu, wei fang

第一个人问的build a binary search tree from an array。第二个人问的delete duplicate element from an array 和 rotate an 2d array 90degree in place

后两个人主要聊的背景和project和图像处理相关的知识

由于时间原因当时后两个面试官只聊了20分钟左右吧要开会

Process

Each process provides the resources needed to execute a program. A process has a virtual address space, executable code, open handles to system objects, a security context, a unique process identifier, environment variables, a priority class, minimum and maximum working set sizes, and at least one thread of execution. Each process is started with a single thread, often called the primary thread, but can create additional threads from any of its threads.

Thread

A thread is an entity within a process that can be scheduled for execution. All threads of a process share its virtual address space and system resources. In addition, each thread maintains exception handlers, a scheduling priority, thread local storage, a unique thread identifier, and a set of structures the system will use to save the thread context until it is scheduled. The thread context includes the thread's set of machine registers, the kernel stack, a thread environment block, and a user stack in the address space of the thread's process. Threads can also have their own security context, which can be used for impersonating clients. Threads have their own different stack pointer and instruction pointer within their process.

电面：

各种C++的基础知识：

vector

popular methods: .begin(), .end(), .size(), .empty(), .front(), .back(), .at(), .push_back(), .pop_back(), .insert(), .clear(),

list

popular method: .begin(), .end(), .front(), .back(), .push_front(), .pop_front(), .push_back(), .push_front(), .insert(), .clear(), .reverse()

smart pointer

smart pointer can make pointers to work in way that we do not need to explicitly call delete. Smart pointer is a wrapper class over a pointer with operator. The objects of smart pointer calss look like pointer, but can do many things that a normal pointer cannot like automatic destrucion, reference counting and more.

process 和 thread
class之间的relationship

class can have a “friend” function to allow an external function to have access to the private and protected members of a class we have to declare the prototype of the external function that will gain access preceded by the keyword friend within the class declaration that shares its members.

Just as we have the possibility to define a friend function, we can also define a class as friend of another class, allowing the second one access to the protected and private members of the first one. Friend relationship is one directional if not defined by both classes in declaration.

Inheritance between classes, derived class have access to public/protected members of the base class, keyword in inheritance defines the minimum level of protection that the inherited members of the base class must acquire in the new class.

Although the constructor and destructor of the base class are not inherited, the default constructor and the destructor of the base class are always called when a new object of a derived class is created or destroyed.

struct和class

The members and base classes of a struct are public by default, while in class, they default to private. Struct and class are otherwise functionally equivalent. Member of a class defined with the keyword class are private by default. Members of a class defined with the keywords struct or union are public by default.

how to speed up query in database

adding indices, explain the query by keyword “explain”.....

onsite :

1. 第一题是multiply两个matrix。第二题给一堆pair，前面那个是child，后面那个是parent，判断是不是一个valid tree。
2. C++的memory leakage，设计一个class有static function，C++的多线程（不会。。。），socket，同步异步什么的，什么都问了，一堆不会的。。。
3. equals和hashCode()，同时有很多人访问网站的话要怎么handle,怎么优化。

Distributed compression system:

1. master-slave setup, master is organizer, slave is where stores the file.
2. files are always large, so choose large size of chunk, say 64MB,\
3. chunk (slave) does store part of metadata, master stores which part of the file is stored on which chunkserver, chunkserver stores the offset on the server.

4. When write the file, cut the input file into the chunk size by client and transfer between client and master. (Read) Master allocates the space for client, replica is needed for reliability when checksum of the first reader chunk server is not giving correct checksum.
5. Actual read and write are conducted between the client and chunkserver, if through master, there will be traffic bottleneck. Single master is OK, but adding one more for replica is not a bad idea.
6. Chunk server strategy to choose, LRU, least disk usage. How to know a chunk server is down? By heartbeat.
7. When write, first write to one captain and the captain will replicate to the others. How to choose captain? Nearest and least traffic.
8. Some optimizer, cache: using 80-20 principles: if 20% of data are visited frequently (80% of the traffic) then caching them inside the servers RAM. sharding: vertical and horizontal sharing, not seems very necessary here. Load balancer: between clients and web server; between web server and file system.
9. As for compression system, think of adding an application server for compressing the files (not familiar with the algorithm), between the web server and master of file system, then need to add a load balancer between app server and FS.
10. Consider the time to find something on the disk, find something on disk takes binary search, \log_2 based (number of entries) every time it read the disk once, take around 10 ms. Use this data to calculate the latency and overall output through then match the QPS requirement. Calculate how many storage you need and how much server you need to handle QPS (based on number of cores and QPS, hardware can do parallel read/write by OpenMP library).

我自己的面经：

电面：

1. fillcups_asml :
https://github.com/xwang322/Coding-Interview/blob/master/asml/FillCups_ASML.py
2. Merge two sorted arrays, Leetcode 原题

onsite:

1. 给一个postorder preorder list 如何构建tree, 答案不唯一 写出一个即可
2. happy number, leetcode 原题

还一些简单的数学, os基本概念题目