

秋招日记

一、个人背景

二、学习经历

三、实习经历

四、Offer收获情况

五、投递公司情况

2022年08月10日

2022年08月27日

2022年08月29日

2022年08月30日

2022年09月04日

2022年09月05日

2022年09月06日

2022年09月07日返校

2022年09月08日

2022年09月09日简历改动

2022年09月09日

2022年09月10日

2022年09月13日

2022年09月14日

2022年09月15日

2022年09月16日

2022年09月17日

2022年09月18日

2022年09月19日

六、各大公司笔面试真题

星宸科技

应聘岗位：Linux驱动工程师

2022年08月13日--笔试

2022年08月18日--技术一面

2022年09月06日--技术二面

2022年09月13日--HR面

2022年09月13日--性格测试

凌云光

应聘岗位：图像应用开发工程师

2022年09月10日--笔试

编程题

英文翻译题

简答题

2022年09月21日--一面

2022年09月23日--二面

2022年09月27日--部门HR面

蔚来

应聘岗位：嵌入式软件开发工程师

2022年09月10日--笔试

选择题

编程题

简答题

经纬恒润

应聘岗位：嵌入式软件工程师

2022年09月13日--技术一面

2022年09月18日--笔试

选择题

编程题

2022年09月22日--技术二面

小米

应聘岗位: 嵌入式软件工程师

2022年09月14日--笔试

选择题

编程题

海信科技

应聘岗位: 软件开发工程师(嵌入式)

2022年09月15日--英语口语测试

2022年09月20日--笔试

选择题

编程题

华勤技术

应聘岗位: 嵌入式驱动开发工程师

2022年09月15日--笔试

2022年09月21日--老总一面

2022年09月23日--技术二面

CVTE

应聘岗位: 嵌入式(Linux方向)

2022年09月16日--一面

2022年09月21日--二面

美团

应聘岗位: 硬件开发--嵌入式自动驾驶

2022年09月17日--笔试

编程题

选择题

滴滴

应聘岗位: 自动驾驶创新业务-嵌入式开发工程师

2022年09月17日

选择题

编程题

美的

应聘岗位: 嵌入式软件开发工程师(机器人与自动化事业部)

2022年09月19日--群面 (省会城市的大酒店)

2022年09月20日--测评

2022年09月22日--专业面试

2022年09月24日--收获Offer

中科创达

应聘岗位: 畅行智驾-MCU开发工程师 (校招)

2022年09月20日--笔试

选择题

海康威视

应聘岗位: 软件开发工程师 (嵌入式C/C++)

2022年09月22日--技术一面

紫光同创

应聘岗位: 嵌入式软件开发工程师

2022年09月22日--笔试

选择题

编程题

海能达

应聘岗位: 嵌入式软件开发工程师

2022年09月23日--笔试

选择题

编程题

2022年09月27日--技术一面

秋招日记

一、个人背景

秋招已经告一段落了，写了一些总结，记录了一下秋招期间的每个日日夜夜，总结了每次笔试和面试的经历以及个人的总结，希望能够帮助之后找工作的小伙伴们。

背景：双非一本本科

大三暑假实习：拿到了东信和平(隶属于中电科旗下)的offer，最后去珠海实习了两个月，体验了上市国企的日常工作及生活，对接下来的秋招有了更加明确的方向。

二、学习经历

大一下学期结束后的那个暑假，加入了嵌入式实验室，一待就是三年。从刚开始的玩STM32单片机，学习Linux基础知识；到之后开始接触图像处理相关知识，学习了计算机视觉OpenCV，ResNet、YOLO系列网络模型，学习了Pytorch、tensorflow网络框架；

大二上学期参加了一个喷码机项目小组，通过这个项目学习Python基础语法，学习PyQT的使用，学习了PLC通信的基础知识，在项目期间还学会了如何通过阅读英文手册进行项目的开发(因为我们的项目是要基于使用手册进行二次开发，且使用手册是全英文的)

大二下学期参加并负责中国大学生工程训练大赛智能垃圾分类的视觉代码开发(自己没有参赛报名，但是为这个比赛付出了大半年的心血)

大三上学期学习了ROS机器人操作系统，学习了分布式通信机制，学习了如何绘制PCB板，学习硬件基础知识，自己制作了一台基于ROS的智能小车(基于万向轮、主控为Arduino、上层为Jetson Nano);到后面不断地参加比赛，最印象深刻的是参加了RoboCup智能车比赛并获得了国家一等奖；

大三下学期协助导师开展Linux驱动教学，并制定uboot移植、Linux内核移植、根文件系统搭建、LED驱动开发实验手册，同时学习了FreeRTOS实时操作系统，并且在实训时完成了一个基于FreeRTOS的智能预警项目；在此期间和华南理工合作一个喷头驱动项目，学习了FPGA的使用。

秋招期间总共投递了50多家企业，每天都是早上9点左右起床，到实验室准备笔试面试一直到晚上10点左右回宿舍，持续了一个月左右，很累但是很值得！

三、实习经历

大三下学期，去到了东信和平智能卡股份有限公司实习，实习期间，担任了USBKEY实习小组项目负责人，主要负责了智能卡JavaCard应用的开发，以及协调小组内各个模块的联调。在实习过程中，还是交到了许多朋友，带我的导师也教会了我许多智能卡领域的专业知识，公司的领导对我们实习生还是很关心的，我也从我们副总身上学到了许多管理方面以及待人方面的知识。

总的来说，在我实习的这两个月里，每天都会有项目在Push你向前走，每一天都过的很充实，也十分感谢公司的培养和信任！

四、Offer收获情况

- 08月29日收到超睿科技Offer
- 09月01日收到东信和平Offer
- 09月13日收到星辰科技Offer

- 09月21日收到比亚迪Offer
- 09月24日收到美的Offer
- 09月28日收到紫光同创Offer

五、投递公司情况

2022年08月10日

- 超睿科技--系统软件工程师
08月12日已完成笔试
08月23日已完成面试
08月29日收到Offer
岗位：系统软件工程师；地点：长沙

2022年08月27日

- oppo(简历挂，投的是Linux驱动开发的岗位)
- vivo(简历筛选)
- 紫光同创--嵌入式软件开发工程师
09月02日已完成技术一面
09月22日已完成笔试
09月28日已收到Offer
工作：软件开发；岗位名称：嵌入式软件开发工程师；地点：深圳

2022年08月29日

- TCL--软件驱动工程师(23届提前批)(简历挂)

2022年08月30日

- 比亚迪--软件开发工程师
09月17日已完成面试
09月21日收到Offer
工作：软件开发；岗位名称：软件开发工程师；部门：商用车研究院;地点：长沙

2022年09月04日

- 美团--嵌入式硬件开发工程师
09月17日已完成笔试
- 小米--手机相机驱动开发工程师
已做完测评

已完成09月14日完成笔试

09月28日简历筛选不通过

- 美的--嵌入式软件开发工程师

09月09日已经完成笔试(题型为:单选题+多选题)

09月19日已完成群面

09月22日已完成专业技术面

09月24日已收到Offer

岗位：嵌入式软件开发工程师，部门：机器人及自动化事业部

- 凌云光--图像应用开发工程师

公司简介：凌云光股份（股票号：688400）以**光技术创新**为基础,坚持“**为机器植入眼睛和大脑**”,赋能**电子制造、新型显示、印刷包装、新能源、影视动漫、科学图像、轨道交通**等行业的智能“制造”,基于20余年在光学成像、视觉软件与算法、核心视觉部件等领域的技术积累。

09月10日已完成笔试

09月21日已完成一面

09月23日已完成二面

09月底收到Offer,10月中旬谈薪

小技巧：公司面试的时候可能会问你关于你对公司的了解情况，询问你是否知道公司的一些业务的时候，如果你提前到公司的官网上去查看相关的信息，找到与你岗位相匹配的信息时，在面试的时候会加分哦！

2022年09月05日

海康威视--软件开发工程师（嵌入式）

- 09月07日已完成测评
- 09月08日已完成笔试
- 09月22日已完成技术面

等待HR面

2022年09月06日

- 寒武纪--系统软件开发(简历挂)
- 哈啰出行--嵌软ROS(简历挂，投递简历的时候没有太完善)

- **蔚来NIO--嵌入式软件开发**

09月10日已完成笔试

泡池子ing

- 小鹏汽车--嵌入式软件开发(简历刷)
- CVTE--嵌入式软件工程师(Linux方向)

目前公司的主营业务为**液晶显示主控板卡和交互智能平板**等显控产品的设计、研发与销售，建立了教育数字化工具及服务提供商希沃（seewo）、智慧协同平台 MAXHUB 等多个业内知名品牌。

09月08日已笔试 (CVTE笔试)

09月16日已一面

09月21日已二面

09月24日二面挂

感受: CVTE是面试的时候要手撕编程题的, 所以大家一定要提前做好准备!

- 滴滴出行--自动驾驶创新业务-嵌入式开发工程师

09月17日已笔试

泡池子ing

2022年09月07日返校

2022年09月08日

- ATL新能源(没有参加线下面试, 所以估计是没消息了)

泡池子ing

感受: 有些公司会在各个城市中开展线下面试, 这种情况的话最好要去, 不要只想着通过网申就行。

- 经纬恒润--嵌入式软件工程师

公司简介: 主要从事智能驾驶、智能网联、新能源、动力系统、底盘控制方面, 采用感知-决策-执行一体化技术, 开发了单车智能解决方案, 公司**无人驾驶智能集卡**在唐山港和日照港实现了无安全员的运营服务

09月13日已完成一面

09月18日已完成笔试

09月22日已完成二面

等待HR面

- 浙江大华

09月28日已完成笔试

2022年09月09日简历改动

- 将原来的**基于深度学习的垃圾投放智能小车**项目经历修改为:**基于FreeRTOS+Linux驱动的智能预警系统.**
- 将原来的**基于 STM32 DAC 模块的高精度任意波形实现**修改为**基于 STM32+FPGA DAC 模块的高精度任意波形实现**
- 增加了“抗压能力强”

2022年09月09日

- 全志科技--嵌入式软件设计工程师

泡池子ing

- 海尔(简历挂)
- 富士康

09月09日已完成测评

09月14日完成笔试(考了行测、性格测试、英语四级相关内容)

09月22日给我打了个电话，让我去面试，拒了

- 中兴通讯--软件开发工程师

泡池子ing

- 阿里

泡池子ing

2022年09月10日

- 文远知行（简历挂，投递的是自动驾驶嵌入式开发）
- Momenta（简历挂）

2022年09月13日

- 荣耀--嵌入式软件开发工程师

泡池子ing

- 海能达--嵌入式软件开发工程师

09月23日已完成笔试

09月27日已完成技术一面

- 科沃斯机器人--嵌入式开发工程师

09月14日已完成测评

- 理想汽车--软件工程师(嵌入式)
- 宁德时代--嵌入式软件工程师（简历挂）

2022年09月14日

- 乐鑫科技(简历挂,嵌入式软件开发工程师)
- 360--嵌入式软件开发工程师

2022-10-09 15:00 笔试

- 科大讯飞--嵌入式软件开发工程师

09月15日已完成测评

2022年-10-09 19:00:00 -- 21:00:00笔试

- 商汤科技--(简历挂，投的是嵌入式开发工程师)
- 浙江中控--嵌入式软件工程师

- 京东方--嵌入式开发工程师（整机业务）

09月15日已完成测评

- 海信科技--软件开发工程师(嵌入式)

09月15日已完成测评、英语口语测试

09月20日已完成笔试

泡池子ing

2022年09月15日

- 顺丰科技--无人机应用开发工程师（丰翼科技）

09月15日已完成测评

- 韶音科技--嵌入式软件开发工程师

- 虹软科技--嵌入式驱动开发工程师(之前发过笔试链接我没做，这次被直接挂简历了)

- 华勤技术--驱动开发工程师(BIOS/BSP/BMC)

是全球智能硬件平台型企业。秉承改善人们沟通与生活的使命，为全球科技品牌客户提供产品级、系统级、软硬件研发、运营制造的端到端服务。主要产品有手机、平板、电脑、智能穿戴、AIOT、服务器等，是一家软实力较强的硬件公司。总部位于上海。

09月15日完成测评、笔试

09月21日已完成面试

09月23日已完成技术二面

等待HR面

2022年09月16日

- 字节跳动--C++研发工程师（特效方向）-智能创作

笔试2022-10-09 10:00

2022年09月17日

- 汇川技术--嵌入式软件开发工程师

2022年09月18日

- ThunderSoft中科创达--C/C++开发工程师

09月20日已完成笔试

09月25日简历挂

- 上海云骥智行--系统开发工程师

- 麒麟信安--Linux C/C++开发工程师

2022年09月19日

- 长虹--嵌入式软件开发
- 英威腾--嵌入式工程师(通讯方向)
- 中联重科--嵌入式软件开发

六、各大公司笔面试真题

星宸科技

应聘岗位：Linux驱动工程师

2022年08月13日--笔试

进行了笔试题，选择题主要考的是操作系统的知识、编程题考了求矩形的最大面积、求链表倒数第K个节点。总体来说难度不大，但是要背好八股文以及刷题。

2022年08月18日--技术一面

进行了技术一面，面试的是部门的一个小组长。在面试的过程主要问了很多八股文相关的知识(牵涉最多的就是Linux操作系统相关的知识)、然后就会展开问一下项目相关的知识点。

0.一定要认真复习笔试题

面试官一上来就对自己的笔试题目进行一个询问，问了编程题的具体实现的细节：包括了求矩形的最大面积、求第K个链表的数据

1.进程调度时机？

答：

- 中断处理过程
- 内核线程调用schedule()进行进程切换
- 用户态进程无法实现主动调度，只能通过内核态后某个时机点进行调度，即在中断处理过程中进行调度

2.孤儿进程和僵尸进程？

答：

孤儿进程：

一个父进程退出，而其一个或者多个子进程还在运行，那么子进程将成为孤儿进程。将被init进程(进程号为1)所收养，并且由init进程完成状态收集工作

僵尸进程：

一个进程使用fork创建子进程，若子进程退出，父进程没有调用wait或者waitpid获取子进程状态信息，那么子进程描述符仍然保存在系统中。这种称为僵尸进程

3.请说一下链表跟数组的区别？

答：数组静态分配内存，链表动态分配内存；数组在内存中连续，链表不连续；数组利用下标定位，时间复杂度为 $O(1)$ ，链表定位元素时间复杂度 $O(n)$ ；数组插入或删除元素的时间复杂度 $O(n)$ ，链表的时间复杂度 $O(1)$ 。

4.单链表和双链表的区别

答：单链表只有一个指向下一结点的指针，也就是只能next。双链表除了有一个指向下一结点的指针外，还有一个指向前一结点的指针，可以通过prev()快速找到前一结点，顾名思义，单链表只能单向读取。

双向链表在存储空间上要大于单向链表。

2022年09月06日--技术二面

进行了第二轮的技术面试，面试的是主管。

面试的内容：

1. 问了我一个自己觉得最有成就感的项目，并详细说一下项目中的具体实现的功能
2. 问了我关于Linux驱动中自己最熟悉哪一个驱动框架，做过哪些驱动框架。我面试的时候说我比较熟悉字符设备驱动框架，然后面试官就问我关于字符设备驱动框架自己是否实现了驱动函数，并且询问了我自己是否深究了驱动函数的内存分配，比如问了我ioremap函数的具体实现。

答：

3. 问了我关于虚拟地址转换为物理地址的具体实现，以及多个物理地址转换冲突的问题。

答：

ioremap函数：几乎每一种外设都是通过读写设备上的寄存器来进行的，通常包括控制寄存器、状态寄存器和数据寄存器三大类。外设寄存器通常被连续编址，分为：

- IO映射方式：为外设专门实现了一个单独的地址空，称为IO地址
- 内存映射方式：只实现一个物理地址空间，**外设I/O端口成为内存的一部分**，CPU可以向访问内存单元那样访问外设IO端口，而不需要单独的IO指令

驱动程序并不能直接通过物理地址访问I/O内存资源，而必须将它们映射到核心虚地址空间内（通过页表）， 然后才能根据映射所得到的核心虚地址范围，通过访内指令访问这些I/O内存资源。

ioremap可以将IO内存资源的物理地址映射为核心虚拟地址后，我们可以直接操作RAM那样操作IO口内存资源了，但是我们应该要使用Linux的特定函数来访问内存资源，而不是使用指针来访问，比如writel()、readl()等

多个物理地址冲突问题：

不同进程的虚地址可能对应着同一块物理地址，映射是可以重复存在的，典型的例子就是我说的共享库，操作系统为了节约物理内存使用，对于同一个DLL，在不同的进程中，其虚地址可能是不同的，但其对应的物理内存可能是同一块。**内核能看到所有进程的页表，自然也就能管理所有进程的内存映射，可以保证不冲突。**

4. 问了我关于Linux进程之间如何进行数据的共享的问题，问了我自旋锁、互斥锁的区别，此外还补充了一个读写锁的问题，即读的时候是否可以写入、写入的时候是否可以读取、是否可以同时读取和写入

答：读写锁：给临界区加锁，在进行写操作的时候进行互斥；在进行读的时候可以共享访问。

当前锁状态	读锁请求	写锁请求	读写互斥/写写互斥 即就是，写的时候是独占整个临界区的，其他线程只能等待，而读的时候，是共享的，如果要进行读和写，写的优先级高，因为只有写了，才可以读啊 ----> 写独占，读共享，写的优先级高
无锁	可以	可以	
读锁	可以	阻塞	
写锁	阻塞	阻塞	

1.当没有锁的时候：读锁请求和写锁请求都可以满足
2.当有读锁的时候：读锁可以满足，读操作之间是共享的
读的时候写操作是阻塞，读写之间是互斥的
3.当有写锁的时候：读阻塞，写也会阻塞，因为写操作是之间是互斥的

<https://blog.csdn.net/dangzhongjing92>

5. 问了我关于FreeRTOS实时操作系统和裸机开发的区别，问了我关于Linux操作系统和FreeRTOS实时操作系统的区别

答：FreeRTOS是实时操作系统，主要实现了任务调度、内存管理、IPC、中断管理这几个模块，它要求快速的处理任务。但是GUI、驱动框架、shell这些不是核心功能，而linux则是分时操作系统，是一种软实时系统，能够高效利用计算机的资源

6. 问了我为什么FreeRTOS实时操作系统可以实现多任务的，而裸机的多任务实现又是怎样的

答：裸机：通过中断实现，根据中断优先级实现，可以实现中断的嵌套；而实时操作系统则是通过任务优先级来实现，通过时间片轮转来实现，由任务就绪队列来确定。

7. 问了我关于FreeRTOS实时操作系统做过的项目。问了我Linux驱动做过哪些项目(要体现个人在项目中学到了什么，面试官好像比较看重这个东西，越复杂的话可能越好)

8. 关于FreeRTOS问了我自己实现了一些网络协议栈没，对这种实时性的内核是否进行了一些优化改良。

2022年09月13日--HR面

进行了终面，面试的是HR小姐姐。

面试的内容：

1. 除了投递我们公司，你还投递了其它的公司嘛？现在收到了几份Offer呢？公司的主要业务是啥呢？
2. 在我们公司内你有认识的熟人嘛？他是怎么样和你介绍公司的环境的呢？
3. 聊一聊你的实习公司里面的氛围以及你实习公司里面同事的工作状态？
4. 你觉着上两轮面试官的面试情况怎么样呢？你觉得你发挥得怎么样呢？
5. 你期望的薪资是多少呢？(PS：薪资的话讲一个大概的范围就行)
6. 你还有什么想要问我的嘛？(PS:我主要问了一下公司的交通便利、公司里面的工作氛围、以及吃饭、住宿环境)

2022年09月13日--性格测试

HR面完之后没过多久就发了性格测试的链接，这个月底谈薪资。

凌云光

应聘岗位：图像应用开发工程师

2022年09月10日--笔试

编程题

- ifndef/define/endif作用

答：防止头文件被重复引用。被重复引用是指在同一个cpp文件中被include多次，比如：**a.h文件中使用了#include "c.h",而b.cpp中使用了#include "a.h" 和#include "c.h", 此时就会造成c.h重复引用。**

后果：头文件重复引用会增加编译的工作量，若头文件中重复包含，则会出现编译错误。比如：头文件包含全局变量的定义那么会引起错误。

- sizeof的基本用法， strlen的基本用法

```
char a[] = "abc";
sizeof(a)=4 //字符串的末尾还有一个'\0'
strlen(a)=3 //strlen输出的是字符串的长度，不包括'\0'

//在C语言中，若形参为数组，则传递的是指向该数组首地址的指针，而不是数组本身
void foo(char a1[100])
{
    int c3 = sizeof(a1); //64位系统下，c3 = 8;32位系统下,c3 = 4，因为函数参数a1已不再是数组类型，而是蜕变成指针
}

struct s1
{
    char c;
    int i;
};
sizeof(s1) = 8 //char占1个字节，int占4字节，但是要考虑到字节对齐的情况，即让内存所占字节能够整除最大的变量所占字节数，所以这里必须要为4的倍数。字节对齐作用：减少访存指令周期，提高CPU存储速度。空结构体：sizeof为1

Union U
{
    int i;
    char c;
    s1 s; //8个字节
}
sizeof(U) = 8; //联合体各个成员共享一段内存，sizeof为每个成员sizeof的最大值，同样要考虑字节对齐的问题
```

- memcpy函数的基本用法

- 函数原型： `void *memcpy(void*dest, const void *src, size_t n);`
- 功能：由src指向的地址的连续n个字节的数据赋值到dest指向的起始地址的空间内，可以用来拷贝任意的数据类型的对象

- 头文件: #include <string.h>
- 返回值: 返回一个指向dest的指针
- 说明: dest和src所指的内存区域不要重复, 否则会出现数据的覆盖的问题;与strcpy相比, memcpy并不会遇到'\0'就结束拷贝, 一定会拷贝完n个字节

英文翻译题

摘抄了论文中的一段话, 要求翻译为中文, 大致是关于3D图像重构的内容, 关键字stereo vision(双目视觉)

简答题

- 摄像头物镜的三个重要参数以及作用

答: 焦距、相对孔径、视场角。

焦距: 影响成像的大小, 焦距越长, 所得的像也越大, 放大倍数越大。

相对孔径越大, 通过的光线越多, 在像平面上的像越亮。

物镜的视场角决定了在成像面上良好成像的空间范围当焦距 f 一定时, **视场角越大, 成像也越大**; 同时, 当成像面的尺寸一定时, 焦距越长, 视场角越小。

- 开运算和闭运算的作用

答:

- 开运算: 先进行腐蚀再进行膨胀。作用: 消除小的物体, 平滑形状边界, 并且不改变其面积, 可以很好地去除小颗粒的噪声, 断开物体之间的连接。
- 闭运算: 先进行膨胀再进行腐蚀。作用: 用来填充物体内的空洞, 连接邻近的物体, 连接断开的轮廓线, 平滑边界而不改变其面积。

- 图像分割的作用和三种经典的方法

答:

图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程。图像识别的基础是图像分割, 其作用是把反映物体真实情况的、占据不同区域的、具有不同特性的目标区分开来, 并形成数字特征。图像分割是图像识别和图像理解的基本前提步骤, 图像分割质量的好坏直接影响后续图像处理的效果, 甚至决定其成败, 因此, 图像分割的作用是至关重要的。

图像分割方法主要分以下几类: 基于阈值的分割方法、基于区域的分割方法、基于边缘的分割方法、基于特定理论的分割方法等

- 3D点云和2D灰度图的融合

答:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

可得:

$$Z_c \cdot u = f_x \cdot X_c + c_x \cdot Z_c$$

$$Z_c \cdot v = f_y \cdot Y_c + c_y \cdot Z_c$$

从而可以将图像 (u, v) 位置的点转换到相机坐标系 (X_c, Y_c) :

$$X_c = \frac{(u - c_x) \cdot Z_c}{f_x}$$

$$Y_c = \frac{(v - c_y) \cdot Z_c}{f_y}$$

<https://blog.csdn.net/TimeRiverForever>

```
/**
 * 将彩色图和深度图合并成点云
 * @param matrix 相机内参矩阵3x3
 * @param rgb 彩色图
 * @param depth 深度图
 * @param cloud 输出点云
 */
static void convert(Mat &matrix, Mat &rgb, Mat &depth, PointCloud::Ptr
&cloud) {
    double camera_fx = matrix.at<double>(0, 0);
    double camera_fy = matrix.at<double>(1, 1);
    double camera_cx = matrix.at<double>(0, 2);
    double camera_cy = matrix.at<double>(1, 2);

    cout << "fx: " << camera_fx << endl;
    cout << "fy: " << camera_fy << endl;
    cout << "cx: " << camera_cx << endl;
    cout << "cy: " << camera_cy << endl;

    // 遍历深度图
    for (int v = 0; v < depth.rows; v++)
        for (int u = 0; u < depth.cols; u++) {
            // 获取深度图中(m,n)处的值
            ushort d = depth.ptr<ushort>(v)[u];
            // d 可能没有值, 若如此, 跳过此点
            if (isnan(d) && abs(d) < 0.0001)
                continue;
            // d 存在值, 则向点云增加一个点
            PointT p;

            // 计算这个点的空间坐标
            p.z = double(d) / 1000; //单位是米
            p.x = (u - camera_cx) * p.z / camera_fx;
            p.y = (v - camera_cy) * p.z / camera_fy;

            // 从rgb图像中获取它的颜色
            // rgb是三通道的BGR格式图, 所以按下面的顺序获取颜色
```

```

        Vec3b bgr = rgb.at<Vec3b>(v, u);
        p.b = bgr[0];
        p.g = bgr[1];
        p.r = bgr[2];

        // 把p加入到点云中
        cloud->points.push_back(p);
        //cout << cloud->points.size() << endl;
    }

    // 设置并保存点云
    cloud->height = 1;
    cloud->width = cloud->points.size();
    cout << "point cloud size = " << cloud->points.size() << endl;
    cloud->is_dense = false;
}

int main(){
    cv::Mat cameraMatrix; // 从文件加载相机内参
    cv::Mat rgb;          // 从相机得到RGB彩色图
    cv::Mat depth;        // 从相机得到depth深度图
    PointCloud::Ptr pCloud = PointCloud::Ptr(new PointCloud);
    convert(cameraMatrix, rgb, depth, pCloud);
}

```

- 3D点云拼接

答：3D点云包括了图像的X,Y,Z坐标，需要根据camera_fx、camer计算点的空间坐标，而2D灰度图包括了颜色坐标。

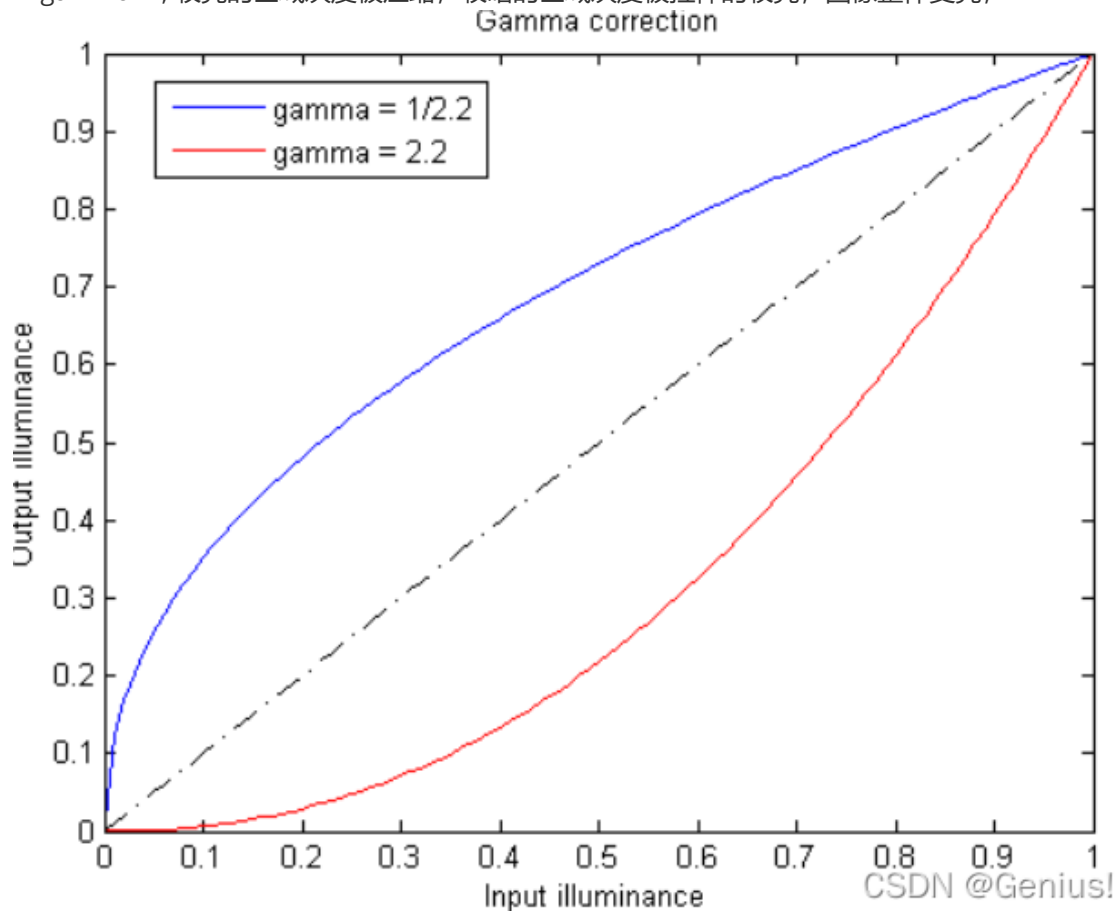
- 通过灰度图直方图来判断灰度图，以及如何改善灰度图的视觉效果

答:通过图像中灰度值的分布区域来划分判断。

我们可以通过灰度变换来使原来图像的某些部分区域特征增强，使图片变清晰。主要算法有：

- 对数变换（对数变换提升低亮区域，压缩高亮区域，使低亮区域的特征更加突出明显）、
- 指数变换（图片低亮度区域将被压缩，高亮度区域将被扩展）、
- gamma变换（提升了暗部细节，简单来说就是通过非线性变换，让图像从曝光强度的线性响应变得更接近人眼感受的响应）

gamma>1, 较亮的区域灰度被拉伸, 较暗的区域灰度被压缩的更暗, 图像整体变暗;
gamma<1, 较亮的区域灰度被压缩, 较暗的区域灰度被拉伸的较亮, 图像整体变亮;



- 直方图均衡化(增加图像的全局对比, 直方图均值化, 将低灰度值归并, 高灰度值拉伸, 当一个图像灰度分布均匀时, 图像的整体细节与质量会提升很多)

步骤: (以灰度深度8为例)

1, 统计每个阶级灰度值的数量

2, 计算每个阶级灰度值出现的概率 $p(r) = \text{sum}(r) / \text{img.size}$

3, 将每个概率的灰度值以前缀和的形式累加 $P(i) = \sum_n^i p(i)$

4, 将图片代入计算, $p[x,y] * \text{img}[x,y]$

2022年09月21日--一面

- 面试官上来先介绍了一下岗位的具体工作: 做手机和电脑的, 我应聘的岗位主要是做视觉方案和应用的开发
- 之后主要进行了一个自我介绍, 然后开始问我自学的OpenCV包括哪些内容, 询问我是怎么样的契机想到自学OpenCV的
- 之后就问我的简历中的项目, 只问了第一个项目: 关于ROS系统的无人驾驶智能小车(因为第一个项目里面包含视觉处理), 询问我项目里面的视觉处理: 主要问了车道线识别的基尔霍夫变换包括哪几种, 然后如何识别的。
- 之后问了我作为项目负责人在项目管理过程中出现项目成员完成不了情况导致项目延期该怎么办?
- 之后问了我三个项目中最有收获的一个项目? 以及收获的点在哪些地方?
- 之后问了几个C++知识点, 主要包括了多态的实现和深拷贝/浅拷贝的区别

答: 多态的实现:

- 静态多态：重载。编译期间确定函数的地址
- 动态多态：重写基类的虚函数实现的多态，在运行时决定，运行时虚函数寻找调用函数的地址

深拷贝/浅拷贝：

- 默认是浅拷贝，系统会调用默认的拷贝构造函数，即浅拷贝，它能够完成成员的一一复制。当数据成员中没有 指针 时，浅拷贝是可行的，但当数据成员中有指针时，如果采用简单的浅拷贝，则两类中的两个指针将指向同一个地址，当对象快结束时，会调用两次析构函数，从而导致指针悬挂现象

```
class widget{
public:
    widget();//default构造函数
    widget(const widget&rhs);//copy构造函数 浅拷贝函数
    widget& operator=(const widget &rhs);//copy assignment操作符
};
```

- 深拷贝会在堆内存中另外申请空间来存储数据，我们需要**重写拷贝构造函数**或者使用运算符重载，**当数据成员中有指针时，必须要用深拷贝**

```
class A
{
public:
    A(int _size):size(_size)
    {
        data = new int[size];//指针变量名 = new type[内存单元个数]
    }
    A();
    ~A()
    {
        delete []data;
    }//析构时释放资源
private:
    int *data;
    int size;
}
```

- 最后问了一下，我对于公司的业务的了解情况
- 反问：
 - 岗位所做的工作：前期：客户给需求，实现硬件方案设计、软件设计、性能调试。后期：项目技术管控、技术管控 大后期：纯管理,只负责调配资源
 - 培养方案：入司(公司集体新人培训)、入行(各部门培训)：包括光、机、电三方面以及平台软件、3D视觉处理、实习期(导师全方位指导)
 - 晋升通道：技术岗(初级、中级、中高级、高级)，管理岗位
 - 后期安排：还有一次终面以HR面

2022年09月23日--二面

- 面试我的是部门的经理，全程没有问我太多的技术问题，主要是和我介绍了一下图像应用开发部门的基本情况，以及询问我的意愿
- 首先是图像应用开发部门主要是做机器视觉这一块的应用开发，9成时间在开发，还有1成时间需要和客户、技术支持等相关人员进行沟通

- 再者是这个部门总共有五六十号人，但是只有5个人是核心成员，为项目的业务负责人，和我说了以下部门的成员发展：技术岗位：前两年提升技术、两年后若优秀则转为管理岗位，承担起部门的一个主要技术发展职责。管理岗位：组长、部门经理等等
- 问了一下我家里人住哪，有没有想过来广东发展(一般公司都会相对注重地域的，因为这牵涉到你是否能够安心在公司里面工作，能在这座城市发展)
- 之后，和我说了以下就是部门里也有本科生在做机器视觉这一块，都是十分优秀的本科生，包括广东工业大学、华南理工大学等等，老总和我说了我的项目的经历比较丰富，发展的前景相对较好(所以项目经历越丰富，你受到的青睐越多)
- 最后和我说了以下，就是他们在做的事情：为工业机器人植入眼睛和大脑，希望通过机器视觉的方式减少人力的劳动，提升生产的效率，老总给我举了iphone14的例子，说iphone14的零部件组装就需要用到机器视觉。
- 反问
 - 公司的培养方案：导师一对一指导
 - 后期安排：HR面谈

2022年09月27日--部门HR面

晚上10点钟左右打电话过来和我洽谈一下工作的事项，主要聊了一下我的工作意愿、岗位的情况。和我说，凌云光主要是招聘硕士为主、本科吸纳优秀的本科生，且本科生一般偏向于前端以及管理路线。

我的工作主要是以python编程语言为主，进行一个定制化的前端任务，后期可以转为研发岗。技术路线的话分为：前期(项目开发)-->后期(研发)，管理路线：2年后可转为管理岗位。

级别：L6~L11。一般两年后可升为L8，然后可以转管理岗或者继续走技术路线。

培养方案：导师带领制度，带领6个月，进行深度学习、3D视觉、2D视觉的开发。

蔚来

应聘岗位：嵌入式软件开发工程师

2022年09月10日--笔试

选择题

1.进程间通信的方式不包括(D)

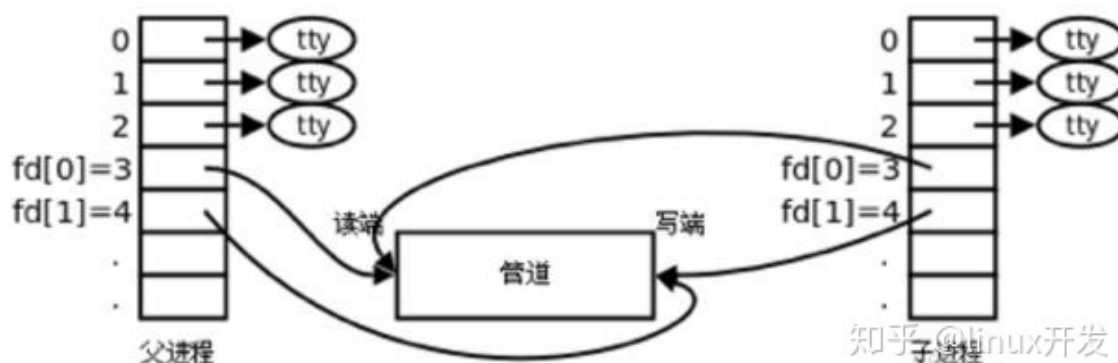
A. 共享内存 B.管道 C.消息队列 D.全局变量

答：进程之间需要数据的共享、消息的发送、进程之间的控制，所以需要进程之间的通信。

方式：

管道(pipe):半双工通信，由pipe函数在内核中开辟一块缓冲区域用于通讯，read(pipefd[0])为读取数据，write(pipefd[1])为写入数据

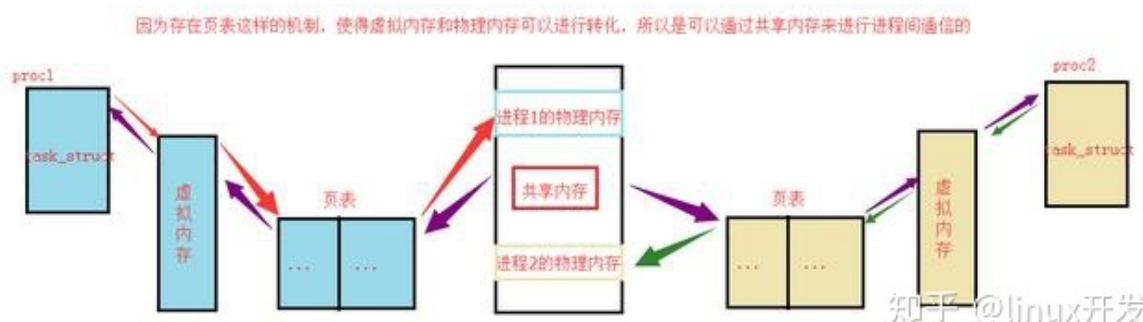
2. 父进程fork出子进程



消息队列：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

信号量：数据操作锁，用来负责数据操作过程中的互斥，同步等功能。**信号量就相当于是一个计数器**。当有进程对它所管理的资源进行请求时，进程先要读取信号量的值：大于0，资源可以请求；等于0，资源不可以用，这时进程会进入睡眠状态直至资源可用。**当一个进程不再使用资源时，信号量+1(对应的操作称为V操作)，反之当有进程使用资源时，信号量-1(对应的操作为P操作)。对信号量的值操作均为原子操作。**

共享内存：共享内存是这五种进程间通信方式中效率最高的。但是因为共享内存没有提供相应的互斥机制，所以一般共享内存都和信号量配合起来使用。共享内存不涉及到程序与内核之间的数据传递，节省了时间。



2. `int *a[10]`和`(int *)a[10]`

答：`int *a[10]` 是数组指针，表示一个数组中有10个`int*`类型的指针,而 `(int *)a[10]` 表示a是一个指针指向了一个包含10个`int`类型元素的数组。

3. `int *fun()`和`int (*fun)()`

答：`int *fun()` 是函数指针，表示函数的返回值是指针类型的，`int (*fun)();`是指针函数,可以通过该指针去调用函数

4. 数组大小动态扩容

答：数组中可以通过`malloc`函数去申请一块大的空间，然后使用`memcpy`将数据拷贝到新的数组中去，实现数组的大小动态扩容

5. ISR可以使用printf嘛

答：ISR为中断服务程序，printf是不可重入的函数，而中断中必须为可重入的函数，所以不能使用printf

注意事项：

- ISR不能有返回值
- ISR不能传递参数
- ISR应该是短而有效率的，在ISR中做浮点运算是不明智的。
- 不能再ISR中使用printf，printf是不可重入的函数，而中断随时都可能发生，而中断内的函数必须是可重入的

拓展：不可重入和可重入

- 不可重入函数指的是该函数在被调用还没有结束以前，再次被调用可能会产生错误。其实现的时候通常使用了全局资源，在多线程的环境下若未处理好数据保护和互斥访问，则会发生错误。
- 常见的不可重入函数有：
printf -----引用全局变量stdout
malloc -----全局内存分配表
free -----全局内存分配表
- 可重入函数：一般可以理解为一个函数在同时多次调用，例如操作系统在进程调度过程中，或者单片机、处理器等的中断的时候会发生重入的现象。

编程题

1. 返回排序后相邻两数的最大差值(桶排序)

题目描述：给定一个整数数组arr，返回排序后相邻两数的最大差值

arr = [9, 3, 1, 10]。如果排序，结果为[1, 3, 9, 10]，9和3的差为最大差值，故返回6。

arr = [5, 5, 5, 5]。返回0。

[要求] 时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$

输入：

4
9 3 1 10

输出：

6

思路：准备 $n+1$ 个桶，最大值放在 $n+1$ 个桶内，其余的放在 $1\sim n$ 个桶内， $1\sim n$ 个桶内元素为 $(val - min) * n / (max - min)$ 号桶。最大差值为当前桶-前一个非空桶的最大值

代码实现：

```
int selectBuket(long num, long min_val, long arr_len, long max_val)
{
    return (int)((num - min_val) * arr_len / (max_val - min_val));
}
```

```
int bucketSort(int arr[], int len)
{
    if(NULL == arr || len < 2)
    {
        return 0;
    }
    int min_val = INT_MAX;
    int max_val = INT_MIN;
    int i;
    for(i = 0; i < len; i++)
    {
        min_val = min(min_val, arr[i]);
    }
```

```

        max_val = max(max_val, arr[i]);
    }
    if(min_val == max_val)
    {
        return 0;
    }
    //标记每个桶的最大值和最小值
    int maxs[len + 1];
    int mins[len + 1];
    bool hasNum[len + 1];
    for(i = 0; i < len + 1; i++)
    {
        maxs[i] = INT_MAX;
        mins[i] = INT_MIN;
        hasNum[i] = false;
    }
    //接收元素分配的桶号，作为maxs和mins的下标
    int number = 0;
    for(i = 0; i < len; i++)
    {
        //获取元素桶号
        number = selectBuket(arr[i], min_val, len, max_val);
        mins[number] = hasNum[number] ? min(mins[number], arr[i]) : arr[i];
        /*如果有元素则比较，没有则加入*/
        maxs[number] = hasNum[number] ? max(maxs[number], arr[i]) : arr[i];
        hasNum[number] = true;
    }
    //找到第一个不为空的桶
    i = 0;
    int last_max = 0;
    int res = 0;
    while(i <= len)
    {
        if(hasNum[i++])
        {
            last_max = maxs[i - 1];
            break;
        }
    }
    /*从第i个桶之后那个值开始寻找*/
    for(; i <= len; i++)
    {
        if(hasNum[i])
        {
            //相邻两个桶之间的最大差值比较，大的赋值给res
            res = max(res, (mins[i] - last_max));
            last_max = maxs[i];
        }
    }
    return res;
}

void testBuketSort()
{
    int num;
    cout << "输入数组长度: ";
    cin>>num;
    int arr[num];

```

```

cout << "输入数组元素: ";
for(int i = 0; i < num; i++)
{
    cin >> arr[i];
}
int len = sizeof(arr)/sizeof(arr[0]);
int value = bucketSort(arr, len);
cout << "排序后相邻两数的最大差值为:" << value << endl;
}

```

2. 返回逆序对个数(快速排序)

题目描述：给定一个序列有n个数，求n个数中逆序对的个数，逆序对的定义： $i < j$ && $a[i] > a[j]$ 。

输入样例：

```

6
2 3 4 5 6 1

```

输出样例：

```

5

```

3. 给定一个字符串，请将字符串里的字符按照出现的频率降序排列，如果频率相同，则按照字符的 [ASCII码](#) 升序排列。

简答题

问：find查找.C结尾的文件，并且将结果显示在标准输出中

答：find ./ -name "*.c" -print

说明：

- -print:将查找到的文件输出到标准输出
- 【-exec command {} \】将查找到的文件执行command操作，{}和\之间有空格
- 【-ok】和-exec相同，只不过在操作前要询问用户

拓展：2>&1

几个基本符号及其含义

- /dev/null 表示空设备文件，可以将不需要的信息丢入到该文件当中去。
- 0 表示stdin标准输入
- 1 表示stdout标准输出
- 2 表示stderr标准错误

2>&1:相当于将标准错误等效于标准输出，&代表的是等效的含义，更加高效。此外，**command 1>a 2>&1**与**command 1>a 2>a**还是有区别的，区别就在于前者只打开一次文件a，后者会打开文件两次，并导致stdout被stderr覆盖。&1的含义就可以理解为用标准输出的引用，引用的就是重定向标准输出产生打开的a。从IO效率上来讲，command 1>a 2>&1比command 1>a 2>a的效率更高。

e.g. find ./ -name "*.c" > /dev/null 2>&1 <==> find ./ -name "*.c" 1> /dev/null 2>&1

经纬恒润

应聘岗位：嵌入式软件工程师

2022年09月13日--技术一面

主要聊了三个项目，附带两个简单的C语言以及C++基础知识

1. 问了个人在项目中所做的事情,所以一定要对项目中的事情进行总结归纳，对里面的相关的技术要熟悉。
2. 问了比赛中的创新点，以及比赛中运用到的知识技术
3. 和我详细聊了自己在项目中所用的技术栈
4. C/C++知识：

- o gcc的编译过程：

- 预处理(Preprocessing): 将宏定义#define的内容进行替换，使用预处理器把源文件test.c经过预处理生成test.i文件

```
gcc -E test.c -o test.i
```

注：文件大小会比源文件大很多

- 编译(Compilation): 将预处理后的文件转换为汇编代码

```
gcc -S test.i -o test.s
```

注：文件大小会比预处理后的小

- 汇编(Assemble): 将汇编代码转换成机器码，这一步产生的文件叫做目标文件，是二进制格式

```
gcc -c test.s -o test.o
```

- 链接(Linking): 使用链接器将该目标文件与其他目标文件、库文件、启动文件等链接起来生成可执行文件

```
gcc test.o -o test
```

一步到位的编译方法：

```
gcc -o test test.c # 输出名为test的可执行程序，然后可以执行./test
```

- o union/struct的区别

struct:各个成员有自己的内存，各自使用互不干扰，遵循内存对齐原则，一个struct的总长度等于所有成员的长度之和

union:各个成员共用一块内存空间，各个变量共用一个内存首地址。一个union变量的长度至少能够容纳最大的成员变量，而且要满足所有成员变量类型大小的整数倍。

大小端模式影响：大端模式 (Big_endian)：字数据的高字节存储在低地址中，而字数据的低字节则存放在高地址中。小端模式 (Little_endian)：字数据的高字节存储在高地址中，而字数据的低字节则存放在低地址中。

5. 反问:

- 我应聘的岗位的主要职责?

答: 公司的主要方向为汽车电子, 嵌入式部门主要负责的是门控制器和座椅加热、车窗控制等车内控制, 主要从事底层驱动的开发、应用层的实现(根据用户的需求来实现)

- 公司的人才培养方案?

答: 会有两个培训: 一个是公司的培训(新员工入职培训), 另外一个事部门的培训(包括专业技能的培训, 会有课程安排的; 此外会分配专门的导师, 可以为你解答疑惑)

- 公司的人才职业发展的通道?

答: 包括技术能力(有五个等级:A、B、C、D、E)以及管理通道的提升, 每年一评

2022年09月18日--笔试

选择题

1. C语言的基本单位:(函数)
2. `int a=b=10`是错误的, 因为只是定义了a, 但是b未定义就赋值。若为`int a,b; a=b=10;`则是正确的
3. 自增

```
#define d 2
d++; //错误的, 宏定义的变量不能使用自增符号
float a = 1.1f;
a++; //float类型可以使用自增
double b = 2.2;
char c = 'a';
int d = 20;
b++;c++;d++; //都是正确的, int char double float都可以自增
```

4. break使用访问: 只能用于case语句或者循环语句中
5. unsigned char占1个字符, 其取值范围为:0~255

编程题

1. 计算斐波那契数列
2. 数组排序(降序排序)

2022年09月22日--技术二面

因为经纬的二面是需要做PPT的, 所以上来我就先通过PPT介绍了一下我的基本情况、掌握的技能、项目经验(最拿手的一个)、实习经验、职业规划。

介绍完后, 面试官就开始对我的PPT上的描述内容进行提问了。首先问了我关于:

- 团队管理过程中出现的问题? 以及如何解决的?
- 讲一讲PID调节的基本原理, 以及三个参数的代表的含义。PID调节这一块为啥要使用20次取平均、PID调节是否有想过用其它的方法进行调节
- 问了我关于平均基尔霍夫变换那一块, 说说创新点在哪以及说说基尔霍夫变换的基本原理
- 最后问了一下我的实习经历, 以及为啥不想再实习的公司继续待下去的原因。

小米

应聘岗位：嵌入式软件工程师

2022年09月14日--笔试

个人心得：不知道为啥，应聘的是嵌入式岗位，但是考的知识点全是Java的

选择题

1. TCP通讯必要的五个信息

答：连接使用的协议、本地主机的IP地址、本地的协议端口、远程主机的IP地址、远程主机的协议端口。

2. 二叉树的三种遍历方式：前、中、后序

答：前序：根左右、中序：左根右、后序：左右根

3. 编译程序的组成部分以及功能

答：一个典型的编译程序有：词法分析器、语法分析器、语义分析器、中间代码生成程序、中间代码优化程序、目标代码生成程序、**表格管理和错误处理程序**

- 词法分析器：输入源程序，拼接、检查、分析单词，输出单词的机内表达形式
- 语法分析器：检查源程序中的存在的语法错误，输出错误处理信息
- 语义分析器：进行语义检查和分析语义信息，并把分析的结果保存到各类语义信息表中
- 中间代码生成程序：按照语义规则，将语法单位转换为一定形式的中间语言代码，如三元式、四元式
- 中间代码优化程序：对中间代码进行优化，等价变换处理
- 目标代码生成程序：将优化后的中间代码转换为目标代码(可执行文件)
- 表格管理程序：负责建立、填写和查找一系列表格工作。表格的作用是记录源程序的各类信息和编译阶段进展情况，编译的信息以及中间结果存储在表格中。
- 错误处理程序：处理和校正源程序中的存在的词法、语法、语义错误，负责报告出错的位置和错误性质等，对发现的错误进行校正，使得编译程序能够继续分析和处理

4. Linux操作系统中的系统资源管理：计算资源(CPU、内存)、存储资源(硬盘)、网络资源(IP、子网掩码、网关、DNS)，**没有中断这种资源**

5. 跳表：增加了前向指针的链表，**是可以进行二分查找的一个有序链表**，可以通过索引来进行快速查找。

6. 堆排序和希尔排序：希尔排序是一种**插入排序**，非稳定排序；堆排序是通过二叉树进行排序，是一种**非插入排序**

7. 操作系统中的缺页中断

概念：要访问的页不在主存中，需要将其调入主存中然后再进行访问。

顺序：

- 硬件陷入内核，在堆栈中保存程序计数器，将当前指令的各种状态信息保存在特殊的 CPU 寄存器中；
- 保存通用寄存器和其他易失的信息，以免被操作系统破坏；
- 当操作系统发现一个缺页中断时，尝试发现需要哪个虚拟页面。通常一个硬件寄存器包含了这一信息，如果没有的话，操作系统必须检索程序计数器，取出这条指令，用软件分析这条指令，看看它在缺页中断时正在做什么；

- 一旦知道了发生缺页中断的虚拟地址，操作系统检查这个地址是否有效，并检查存取与保护是否一致。如果不一致，向进程发出一个信号或杀掉该进程。如果地址有效且没有保护错误发生，系统则检查是否有空闲页框。如果没有空闲页框，执行页面置换算法寻找一个页面来淘汰；
 - 如果选择的页框“脏”了，安排该页写回磁盘，并发生一次上下文切换，挂起产生缺页中断的进程，让其他进程运行直至磁盘传输结束。无论如何，该页框被标记为忙，以免因为其他原因而被其他进程占用；
 - 一旦页框“干净”后，操作系统查找所需页面在磁盘上的地址，通过磁盘操作将其装入。该页面被装入后，产生缺页中断的进程仍然被挂起，并且如果有其他可运行的用户进程，则选择另一个用户进程运行；
 - 当磁盘中断发生时，表明该页已经被装入，页表已经更新可以反映它的位置，页框也被标记为正常状态；
 - 恢复发生缺页中断指令以前的状态，程序计数器重新指向这条指令；
 - 调度引发缺页中断的进程，操作系统返回调用它的汇编语言程序；
 - 该程序恢复寄存器和其他状态信息，返回到用户空间继续执行。
8. UDP首部字段：八个字节，四个字段：源端口号(2个字节)+目的端口号(2个字节)+长度(2个字节)+校验和(2个字节)

编程题

1. 链表中指定区间反转
2. 二叉搜索树转为双向链表

海信科技

应聘岗位：软件开发工程师(嵌入式)

2022年09月15日--英语口语测试

1. 跟着读一段英语文字。比较简单，就是一些常用的英语词汇
2. 听问题并给出你的回答。问题：WHAT DO YOU DO IN YOUR FREE TIME?

答：

Read a book

Whether you have 5 minutes or an hour, picking up a good book and spending some time reading it is a great way to spend your free time in a productive way. If your whole family has a bit of spare time, rather than sitting down to watch TV challenge everyone to pick a book and spend a certain amount of time reading it.

Not convinced that reading is productive?

Reading reduces stress, stimulates the brain, provides you with knowledge, improves your memory, helps improve focus and concentration, creates better writing skills, and of course offers free entertainment for however long you choose.

Exercise

You knew this one was going to be on the list... you were just really hoping it wouldn't be. / *was hoping it wouldn't, anyway.*

Exercising, even if it's just a quick 5-minute walk down the road, is a fantastic way to use your free time. After you exercise you will feel so much more energized, healthy, and ready to face the rest of the day, rather than when you sit on the couch in front of the TV and end up feeling sluggish and lazy for the rest of the day.

3. 看图说话。英文分享如何面对工作的压力

答:

How to reduce stress is an inevitable part of our daily life. Many things, such as natural disasters, war and death, will bring too much pressure to our lives. But according to a psychologist, based on today's situation, stress may come from some small things: taking exams, queuing up, and having too many things to do in a limited time. We all need pressure to increase the taste of life, challenges and opportunities, but too much pressure will seriously affect our physical and mental health, reduce body immunity and damage the brain.

If the pressure lasts too long, it may lead to physical diseases, insomnia, headache, low back pain, ulcer, hypertension. Even if it is heart disease, it is very important to recognize and deal with stress when it really happens. We can try some ways, a lot of sports can reduce the pressure.

It can be very helpful to support our stress together with others. Making time for fun, such as listening to music, may be a good medicine, and sometimes crying is an effective way to relieve stress.

2022年09月20日--笔试

选择题

1. 单片机的时钟由外部晶振决定的(X)

答: 时钟是单片机的脉搏, 在统一的时脉冲控制下进行工作, **由外部时钟(控制外设)和内部时钟(控制内部芯片)共同决定。**

2. 64KB的空间表示范围:0xFFFF,64KB= 64*1024B = 64 * 1024个字节, **故其表示的地址范围最大为FFFF**

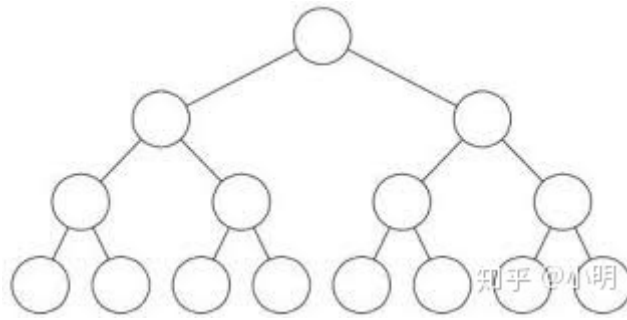
3. 逻辑地址空间

- 程序编译后, 每个目标模块都是从0号单元开始编址, 称为该目标模块的相对地址(或者逻辑地址)
- 当链接程序将各个模块链接在一起成为一个可执行目标程序是, 链接程序依次按照各个模块的相对地址统一从0号单元开始编址的逻辑地址空间
- **用户程序和程序员只需要知道逻辑地址, 而内存管理机制是完全透明的, 只有系统编程人员才会涉及**
- **不同进程可以有相同的逻辑地址, 这些相同的逻辑地址可以映射到主存的不同位置**
- 进程的执行指令和访问数据都要将物理地址从主存中存取, 当装入程序将可执行代码装入内存中, 必须通过地址转换把逻辑地址转换为物理地址, 称为地址重定位

4. 右斜树、左斜树、完全二叉树、满二叉树

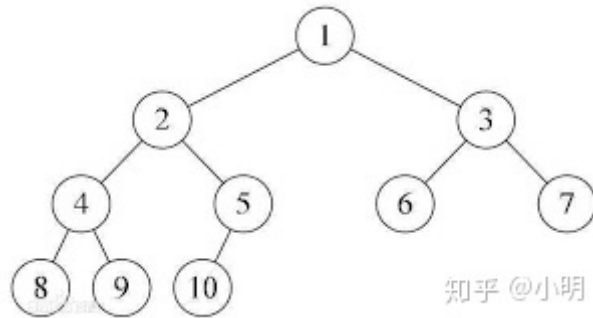
答:

- 左斜树: 所有的结点都只有左子树的二叉树叫左斜树。
- 右斜树: 所有结点都是只有右子树的二叉树叫右斜树。
- 满二叉树



◦ 完全二叉树

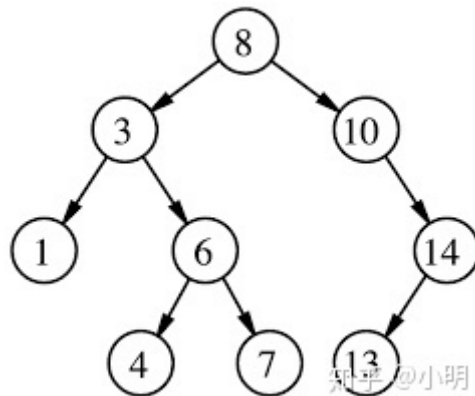
如果二叉树中除去最后一层节点为满二叉树，且最后一层的结点依次从左到右分布，则此二叉树被称为完全二叉树。



◦ 二叉搜索树

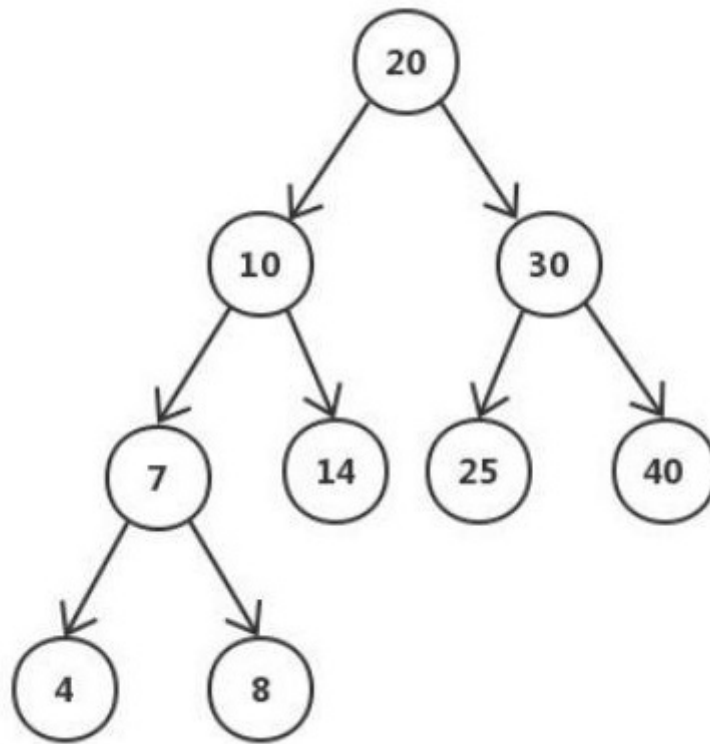
若左子树不空，则左子树上所有结点的值均小于它的根结点的值；

若右子树不空，则右子树上所有结点的值均大于或等于它的根结点的值；



◦ 平衡二叉树

平衡二叉树要么是一棵空树，要么保证左右子树的高度之差不大于 1，并且子树也必须是一棵平衡二叉树



平衡二叉树

知乎 @小明

5. 中断清零

答：中断可以使用硬件清零、软件清零、自动清零方式进行清理，**不只是有硬件清零**

6. 贪心算法活动安排最优规则:相容活动最大的时间

7. linux线程绑定CPU

- 查看进程pid绑核情况: `taskset -p pid`
- 启动时绑定: `taskset -c 1 ./dgram_servr&` //CPU从0开始, 启动时绑定到第二个CPU
- 启动后绑定: `taskset -cp 1,2,5,11 9865` //将进程9865绑定到#1、#2、#2、#11上

8. linux查找/usr/bin/htpasswd文件属于哪个包

答: `rpm -qf /usr/bin/htpasswd`

9. Linux统计物理CPU核数

答: `cat /proc/cpuinfo | grep "physical id" | sort | uniq | wc -l`

10. 可重入函数可以调用不可重入函数

答:

- 可重入函数: 函数可以多线程同时调用, 一般只有局部变量的函数内
- 不可重入函数: 不可多线程同时调用, 一般由全局变量、静态变量, 需要加入互斥锁/信号量/中断
- **可重入和不可重入函数可以互相调用, 但是不可重入函数需要添加互斥**

编程题

求出数组中坏数"0"的个数?

注: 坏数: 即该元素为0, 其左右两个元素的值也为0

华勤技术

应聘岗位：嵌入式驱动开发工程师

2022年09月15日--笔试

1. 各种排序算法的时间复杂度
2. $x*=y+1$, 等价于 $x=x*(y+1)$
3. 空指针不是指针常量
4. 宏定义中未定义变量会在编译时报错
5. $(-5) \% 2 = -1$

2022年09月21日--老总一面

感受：我以为一面都是技术面，结果一上来是老总面(BIOS十年开发经验的老总)，全程没有问我技术上的问题，问的都是生活方面的，问我有没有女朋友，家里住哪，有没有考虑来上海、南昌发展，然后就问我学习怎么样、导师是谁、项目中有啥难忘的事情嘛以及项目中遇到过一些委屈嘛？

反问：

- 岗位职责：BIOS做的是品牌笔记本软硬件开发(从无到有)，竞标，竞选联想、戴尔等大厂的笔记本电脑的生成(从设计到研发到生产)，软硬件测试、项目管理
- 培养方案：专业培训、导师上课(有习题练习)
- 晋升通道：认真做事(每年有两次)

2022年09月23日--技术二面

技术面没有问知识点相关的内容，只是问了一下项目相关的内容以及实习经历

主要是问了第一个项目的相关细节，就是关于ROS无人驾驶智能小车的，说一说你的项目的经历、整体的项目的逻辑以及项目的困难点？

之后就问了一下你的一个实习收获？为什么不考虑留在实习的公司里？

最后问了一下，能不能接受南昌和上海地点

反问：

- 岗位分配：BIOS、BSP驱动、BMS三大岗位

CVTE

应聘岗位：嵌入式(Linux方向)

2022年09月16日--一面

1. 手撕冒泡排序，给了选择题(主要考的是linux基础知识)、填空题(主要靠的是字符串拷贝以及C语言的&&和||操作)
2. 让我开个共享屏幕给他看一下项目的代码
3. 问了项目(问的不深)，就问了你项目中你的代码量是多少，主要的职责是做啥的

感受：面试官最后说自己需要加强一下字符串那块知识的学习，需要多刷刷题目，然后的话呢，就是可能第一次遇到这种手撕题目的面试，还需要多积累积累经验

2022年09月21日--二面

- 手撕编程题(40min)：十六进制字符串转为十六进制数字(CVTE的真实场景题目)
- 问了我一些项目进度延期该怎么办？项目团队成员之间冲突怎么办？对哪方面感兴趣？
- 反问
 - 岗位职责：研发嵌入式，TV板卡(电视智能板卡)，分为三个方向：驱动程序测试、驱动开发、系统开发(全栈)。
 - 培训方案：
 - 新人一个月的成长学院学习(学习西装礼服、西餐、时间管理、古典音乐鉴赏、个人健康、心理咨询)等
 - 下部门：1~2个月学习工具类，第三个月搞项目，进行答辩评估定岗，转为正式期。若定不了岗则可延长到6个月，若还不行则需转岗。
 - 晋升通道：定级(每年3、7、11月晋升)，晋升标准(11.1、11.2、11.3、12等等依次类推)会有晋升答辩(8个维度的答辩)，新人入职从11级开始做起，13级后HR会进行谈话，询问是转技术岗还是技术管理岗。

美团

应聘岗位：硬件开发--嵌入式自动驾驶

2022年09月17日--笔试

编程题

- 考了四道编程题，美团自主命题的

感受：编程题这块，大厂还是相对来说比较重视的，日常中还得多加强这方面的训练

选择题

- 必须要擦除才能使用的存储介质(CD)
A. SRAM B. DRAM C. EEPROM D. Flash

解析：flash芯片不支持覆盖写，只能在擦除过的区域进行写操作，而且擦除一般是按照扇区或者是块进行擦除。根本原因其实是flash芯片的写操作只能将1变为0，而不能将0变为1，擦除之后，flash中是全1的状态，若想将0变为1则只能通过擦除操作。

EEPROM的全称是“电可擦除可编程只读存储器”。flash属于广义的EEPROM，因为它也是电擦除的rom，掉电后数据不丢失，可以保存100年，可以擦写100万次。具有较高的可靠性。

滴滴

应聘岗位：自动驾驶创新业务-嵌入式开发工程师

2022年09月17日

选择题

1. 连续子序列最大和

2. 贪心算法缺点

答：(1) 不能满足求得最后解是最佳的 (2) 不能用来求最大或者最小解的问题 (3) 只能求满足某些约束条件的可行解的范围。

3. 探测主机开放的端口： `netstat -a -n`

4. 字节流数组的使用： `ByteArrayInputStream is = new
ByteArrayInputStream("hello,world")`

5. apache配置多个网站： (1) 基于IP的虚拟主机，每个虚拟主机必须有唯一的IP (2) 基于名字的。只需要依靠客户端发送的HTTP头信息的HOST字段来判断，服务器到底要服务哪个虚拟主机。

6. 代码分析：

```
File file = new File("D:\\test\\1.txt");  
// 如果D:/test 目录下没有 1.txt文件，则创建该文件；如果没有test目录，直接抛出异常，如果  
1.txt已经存在，那么文件创建失败。  
boolean res = file.createNewFile();  
if(!res)  
{  
    System.out.println("创建失败！");  
}
```

7. C++虚函数：使用父类的指针指向的是子类的实例，调用的时候通过父类的指针可以调用实际子类的成员函数

8. 二叉查找树的平均长度： \sum (本层高度元素个数) / 节点个数

9. 内核级线程(KLT)：线程所有的管理工作都由操作系统内核完成

10. mysql锁特性：(1)表级锁：开销小，加锁快，不会出现死锁，并发度最低。(2)行级锁：开销大，加锁慢，会出现死锁，并发度最高(3)页面锁：开销介于表级锁和行级锁之间，会出现死锁，并发度一般

11. mutex为1,表示没有进程进入临界区;mutex为0,表示有一个进程进入临界区，mutex为-1,表示一个进程已经进入，另外一个进程等待进入。

12. 页式存储管理：页式存储系统的逻辑地址是由页号和页内地址两部分组成。

编程题

都是滴滴自命题的编程题，难度适中，需要较好的逻辑思维能力

1. 求墨水污染的数字

2. 刷油漆问题

美的

应聘岗位：嵌入式软件开发工程师(机器人与自动化事业部)

2022年09月19日--群面（省会城市的大酒店）

知识点补充：群面就是一堆人围绕着题目进行讨论，然后总结出规律，最后选出一名代表进行陈述，全程面试官是不参与话题讨论的，只有面试者要进行观点的阐述和总结。

我们总共有7个人进行群面，然后7个人围绕着一个方桌坐下，每个有都会有对应的座位牌，这次群面的题目是：

1. 在数据量极大的情况下，如何进行数据的存储？请从架构方面以及具体技术方面阐述。
2. 你认为是智能家居系统由哪些部分组成？技术上如何实现以及采用哪种手段？

流程：

- 先进行1分钟的自我介绍以及对这两个题目的作答
- 20分钟的组内讨论，大家讨论一下各个细节方面的事项以及交换一下各自的意见。
- 5分钟的总结。选出一名代表进行总结

要点总结：

- 首先一定要针对问题进行剖析，然后一定要鲜明地表达自己的观点，要让面试官眼前一亮。
- 组内讨论的时候一定要参与进来，要积极讨论，积极发言，表达清楚自己的观点
- 总结的时候，若感觉自己总结的比较好的话就积极推荐自己，千万不要怯场，不要因为害怕而不敢去说

我的回答：

个人理解环节：对于第一个问题来说，我觉得可以构建一个分布式的存储系统，采用多节点的方式进行存储，然后使用一个主节点对各个节点进行管理；对于第二个问题来说，我会考虑的更加贴近生活一点，比如从我在外面、进入家门、躺着沙发休息、厨房卫生、打扫卫生、洗浴、睡觉等各个方面都要进行一个智能化普及，比如说在外面通过APP来控制家内家具的状态，进门时通过人脸识别或者指纹识别来实现开门，在室内通过智能语音组手来控制家内的家具，比如沙发按摩、厨房水温调节、扫地机器人打扫卫生、浴室稳定水温、睡眠灯光控制、窗帘控制等。

讨论环节：在家居中添加传感器，实现传感器的数据融合；在家居中添加智能预警系统，当电量过低的时候报警以及实时检测电量情况；相比较于刚刚大家讨论的算法应用，我其实更加关注于底层硬件的搭建，如何实现成本的降低以及如何实现能够高效搭载软件算法的硬件。

总结环节：1. 室外使用手机APP 2. 室内使用语音控制 3. 添加家居的传感器实时检测 4. 硬件的搭建

2022年09月20日--测评

测评主要包括了：性格测试、场景应变能力测试、逻辑思维测试两个部分

2022年09月22日--专业面试

面试主要先问了一个较为温和的问题，总共有两个面试官，两个都是机器人与自动化事业部的技术管，问了一下我在大学期间做的最有收获的一件事，之后就开始问技术了。

第一个面试官主要对我的第一个项目感兴趣，就和我一直讨论第一个项目，而第二个面试官主要对我的第二个项目感兴趣，就和我一直讨论第二个项目。**所以主要要对自己做的项目里面的细节要了然于胸**

反问：

- 岗位职责：做的是库卡机器人，是一种工业机器人，做的是本体的控制器，**所以需要做的就是机器人的控制这一部分。**
- 培养方案：应届生培养计划、部门培养
- 后续结果安排：HR面通过之后发Offer

2022年09月24日--收获Offer

今天中午11点在长沙北辰国萃酒店进行了谈薪，总体听下来福利还不错，本科生20W+，研究生30w+，sp Offer 40w+，而且公司的福利很多，工作时长是955，不提倡加班，这一点的话，总体来说还是相对工资较高的。

签完约后就和部门的同事和HR见面，并开启了破冰之旅，玩了一些小游戏(PS: 酒店的甜点和美食是真的蛮不错的，还有各种饮料^_^)

中科创达

应聘岗位：畅行智驾-MCU开发工程师（校招）

2022年09月20日--笔试

选择题

1. $a=c+2=b$ 合法吗(x)

2. `#include ""`和`#include <>`的区别？

答：`#include <>`直接从系统的类库中查找头文件，若没有则报错；`#include ""`先从工作目录中查找，若没有则从系统类库中查找，若未找到则报错。`#include ""`查找的范围要比`#include <>`要广

3. 作业和进程的对应关系？

答：进程和程序是一一对应的关系，一个作业可以由多个进程组成，且必须至少由一个进程组成，当一个进程不能构建多个作业

4. `realloc`函数用法？

答：将申请的内存空间进行扩大或者减少。

◦ 若为减少，则仅仅改变索引即可

◦ 若为增加，则

■ 若当前内存段后面有需要的内存空间，则直接扩展这段内存空间，`realloc`放回原指针

■ 若当前内存段后面没有需要的内存空间，则在堆中重新申请一块满足要求的内存块，将原来的数据拷贝过去，并释放原来的数据块

■ 若申请失败则返回NULL，此时，原来的指针仍然有效

5. UART波特率是固定的吗？

答：波特率是要依据其时钟频率来确定的，而时钟频率是可调的，所以波特率是可调的

6. `enum{1,2,3}`是正确的吗？

答：是错误的，因为1,2,3是枚举的标号，若这样写的话无法区分是我们定义的变量还是枚举的标号

7. `undef`用法？

答：用来将前面定义的宏标识符取消定义,可以防止宏定义冲突、增强代码可读性、增加调试接口

```
#define MAX 200
printf("MAX = %d\n", MAX);
#undef MAX //使用完后就取消定义
int MAX = 10;
printf("MAX = %d\n", MAX);
```

海康威视

应聘岗位：软件开发工程师（嵌入式C/C++）

2022年09月22日--技术一面

面试官全程问了几个技术问题以及了解一下我的性格方面的具体情况

性格问题：

- 讲一讲你是如何乐于助人的？
- 讲一讲你的学习能力强体现在哪一方面呢？
- 讲一讲你抗压能力强体现在哪一方面呢？
- 讲一讲你的团队协作能力和沟通能力强体现在哪一方面呢？

技术问题：

- 讲一讲冒泡排序的基本思路？

答：

把第一个元素与第二个元素比较，如果第一个比第二个大，则交换他们的位置。接着继续比较第二个与第三个元素，如果第二个比第三个大，则交换他们的位置。

我们对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对，这样一趟比较交换下来之后，排在最右的元素就会是最大的数。除去最右的元素，我们对剩余的元素做同样的工作，如此重复下去，直到排序完成。

- Linux内存申请有哪些方式？

答：

- malloc:在内存中找一片指定大小的空间，然后将这个空间的首地址范围给一个指针变量，这里的指针变量可以是一个单独的指针，也可以是一个数组的首地址,其在逻辑上是连续的

- **kmalloc()**:void *kmalloc(size_t size, gfp_t flags);

kmalloc() 申请的内存位于物理内存映射区域，而且在物理上也是连续的，它们与真实的物理地址只有一个固定的偏移，因为存在较简单的转换关系，所以对申请的内存大小有限制，不能超过128KB。

- **kzalloc()**

kzalloc()和kmalloc一样，实际上只是额外附加了 **_GFP_ZERO** 标志。所以它除了申请内核内存外，还会对申请到的内存内容清零。

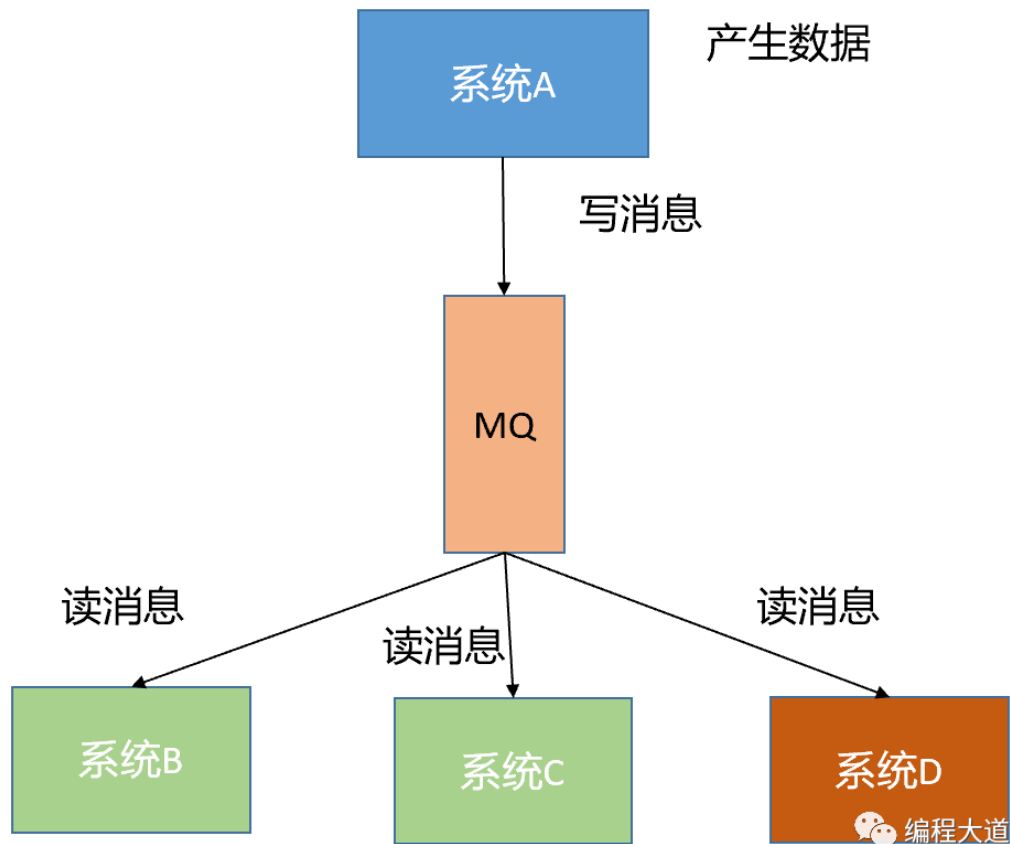
- 数据库用过哪些?如何应对大的数据库的巨大的查询量？

答：Mysql、sqlite。子查询/连接+索引

- 如何实现并发？如何保证上亿数据的并发？

答：

- 系统拆分。将一个系统拆解成多个子系统，使用dubbo来弄，将每个系统连接一个数据库，相当于一个系统多个库
- 缓存。在数据库和缓存中都写一份，读的时候大量读走缓存就行
- MQ实现异步写(消息队列)，实现数据的读和写解耦的操作，从而达到异步的效果



- 分库分表。将一个数据库拆分为多个库，多个库来抗更高的并发；然后将一个表拆分为多个表，每个表的数据量保持少一点，提高sql跑的性能。
- 读写分离。弄个主从架构，主库写入，从库读取。
- Elasticsearch是分布式的，可随便扩容

反问：

- 培养方案：6个月试用期，导师一对一带领入门，分阶段检查，有答辩考核
- 晋升通道：技术(每年一考核)、管理：两年后可以转岗(转为管理)，负责团队管理
- 后期安排：根据需要看是否需要技术加面，后就是HR面，最后入职

紫光同创

应聘岗位：嵌入式软件开发工程师

2022年09月22日--笔试

选择题

1. 派生类析构函数调用顺序?

答：析构函数调用顺序：首先调用派生类的析构函数；其次再调用成员类对象的析构函数；最后调用基类的析构函数；而构造函数恰好相反

2. C++中引用和指针区别?

答：

- 引用必须初始化，但是不分配存储空间，指针不声明时初始化，需要分配空间
- 引用初始化后不能被改变，指针可以改变所指向的对象
- **不存在指向空值的引用，但是存在指向空值的指针**
- **引用没有const，指针有const，const的指针不可变**
- **"sizeof*引用"**得到的是所指向的变量(对象)的大小，而**"sizeof"指针"**得到的是指针本身的大小

3. 虚函数是析构函数吗?

答：构造函数不可以是虚函数，析构函数常常是虚函数。

当类中声明虚函数时，编译器会在类中生成一个虚函数表，虚函数表是一个存储成员函数指针的数据结构。虚函数表是由编译器自动生成与维护的，virtual成员函数会被编译器放入虚函数表中，当存在虚函数时，每个对象都有一个指向虚函数的指针（vptr指针）。在实现多态的过程中，父类和派生类都有vptr指针。

与构造函数不同，vptr已经完成初始化，析构函数可以声明为虚函数，且类有继承时，析构函数常常必须为虚函数。

4. 成员函数被重载的特征?

答：

- 具有相同的作用域
- 函数名字相同
- 参数类型，顺序或者数目不同
- virtual关键字可有可无

编程题

1. 使用新字符替换字符串中的特定字符
2. 链表反转

海能达

应聘岗位：嵌入式软件开发工程师

2022年09月23日--笔试

海能达的笔试总体上来说还是很基础的，只要细心去做就没太大问题。

选择题

C语言和C++的基础知识，考了虚函数相关的内容，父类和子类虚函数调用的顺序

看一个虚函数的例子：

虚函数：指向基类指针在操作它的多态类对象的时候，可以根据指向的不同类对象调用其相应的函数

作用：在基类定义了虚函数后，可以在派生类中对虚函数进行重定义，并且可以通过基类的指针或者引用进行调用，在程序运行阶段动态地选择调用基类或者不同派生类的同名函数，若派生类没有对虚函数的重新定义则继承基类的虚函数并调用

```
#include "stdafx.h"
#include<iostream>
using namespace std;
class Base
{
public:
    virtual void Print()//父类虚函数
    {
        printf("This is Class Base!\n");
    }
};
class Derived1 :public Base
{
public:
    void Print()//子类1虚函数
    {
        printf("This is Class Derived1!\n");
    }
};
class Derived2 :public Base
{
public:
    void Print()//子类2虚函数
    {
        printf("This is Class Derived2!\n");
    }
};
int main()
{
    Base Cbase;
    Derived1 Cderived1;
    Derived2 Cderived2;
    Cbase.Print();
    Cderived1.Print();
    Cderived2.Print();
    cout << "-----" << endl;
    Base *p1 = &Cbase;
    Base *p2 = &Cderived1;
    Base *p3 = &Cderived2;
    p1->Print();
```

```
p2->Print();
p3->Print();
}
/*
输出结果:
This is Class Base!
This is Class Derived1!
This is Class Derived2!
-----
This is Class Base!
This is Class Derived1!
This is Class Derived2!
*/
```

编程题

比较基础，基本一个for循环就可以搞定，主要要注意边界问题

2022年09月27日--技术一面

主要问了我第一个项目的人员个数、你在项目中扮演的角色、项目成员之间是如何沟通的、项目中主要负责的是哪部分的任务、项目进行过程中遇到了哪些困难以及如何解决的、项目中的创新点。

岗位职责：嵌入式软件开发分为：平台部(驱动、上层应用开发)和DSP部(通信、物理层、音频处理等)。

有个半年的适应时间，6个月后转正，2~3年后可以走技术专家路线或者团队管理路线。

培训：部门级和公司级(20个课程的开展)

后期：HR后进行相应的安排