

小姐姐味道 2020

故事专辑



摸鱼好时光

《摸鱼好时光》小姐姐味道2020故事合集

鱼，我所欲也。技术，亦我所欲也。

2020年，互联网，并不太平。大到北上广，小到二线、三线城市，大多数公司进入了相爱相杀、相互对抗的境地。

程序员多面对的是虚拟的代码，在遇到超乎逻辑的人和事，更容易情绪失控。你可能受够了无意义的作秀式加班；或者极度厌烦了那些虚伪的做事风格；亦或能力强被用到死但没有对等的报酬。

这都不要紧，因为它们是这个时代，是生命不可分割的一部分。

闲暇时，沏上一杯苦咖啡，品xjjdog故事集。亦苦亦甜，忆苦忆甜，也不妨是一种活着的乐趣。

或许读完这些故事，能够带给你欢乐，或者在笑声中，会感受到一丝凄凉吧。

本合集包含35篇文章，全部虚构。更多，请关注《小姐姐味道》公众号，获取最新技术干货、荒诞故事。为什么我爱你们爱的深沉，因为我也是一个程序员啊。

扫码关注公众号。



01. 程序员画像，十年沉浮

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

十年，转瞬即逝，人生进入下半场。众生皆苦，万相本无。且看风云变幻，慢品苦辣酸甜。小姐姐味道微信公众号首发。

小胡，拼搏者

小胡很久没有笑过了。他在房价的次高点买了套超出自己承受能力的房子，紧接着老婆失业了，失业以后脾气变得特别的不好。

他每天都下班很晚，最近终于鼓足勇气换了份离家近的工作，来省下每天的车费。

他的技术很好，别人也是这么评价，但这次跳槽他并没有涨多少薪。心里乱乱的老觉得堵，家里环境太压抑，老母亲最近又和老婆掐了起来。

他依然每天很晚下班，心烦了就工作。老板看在眼里，给他颁发了一个 拼搏者 奖杯。

今天团建后他开车回家，天很晚，他把头顶在方向盘上呆了很久。推开车门，他静静伫立一刹，把一个包装袋随手扔进垃圾桶。

里面是他刚得的奖杯。

小苗，全栈培训讲师

小苗很久之前就看透了it行业。别人要淘金，他偏要卖淘金的工具。在工作2年之后，通过被培训，他果断进入培训行业，一直到现在，还出了不少书。他可以算是一个全栈培训讲师，懂的东西多如牛毛，就是不深。

“深了也没有用，简单才有市场。有能力自学的谁来培训呀”。小吴咧嘴一笑，他刚给自己镶上了金牙，灯光照耀下显得异常夺目。

小包，BAT梦

小包的学校不好，所以他很自卑，但还是有两把刷子的，所以业绩很好。他的大部分工作都是小公司经历，所以他觉得缺少点什么。随着年龄的增加，他觉得到了去大公司镀金的最后期限。

但天不遂人愿，总有这样或者那样的事牵牵绊绊。准备了很多次面试，都不对口。他觉得在过去的这些公司，技术应该算是上等了，实在不明白这些BAT公司为什么死抠一些冷门而且明显用不着的技术。

这些公司也贱，见到他这种小公司摸打滚爬的人，就像猫见了老鼠一样，想要尽情蹂躏一番，感受一下高高在上的感觉。

“看来我只适合在小公司混啊”，小包回头按了下车钥匙，几米远的保时捷车灯亮了几下。

小殷，佛学院的梦想

对于一个不善言语的俗人来说，龙泉寺一直是小殷的一个梦想归宿，但这样的组织比 Bat 门槛还要高，要能力还要学历。小殷修了硕士，参读经书，在一个低很多档次的寺庙一呆就是五年，希望未来的某刻能结佛缘。

“世人笑我太疯癫，我笑他人看不穿”。他昨天刚结婚，才休息2天，今天就要上班了。

小刘，公司终结者

“我干倒了这么多家公司”，每次小刘说到这里，都惨笑着伸出手，打着手势。这其中的背景意义，是小刘每次工作理想的破灭。

哦，这当然是骗人的，因为每次都有赔偿金。

他选的很准，每次想要弄点期权实现财务自由，可惜结果都不得愿。他的技术并不差，只是运气衰。面试填表的时候，在离职原因那一栏，他都不好意思下手，只能随意编几个说法。

“我要认真准备”，小刘不怀好意的笑笑，“进阿里看看”。

小姐姐味道微信公众号首发。

小李，理想的路上

“无论是干啥，哪怕是卖海鲜，都不会干这一行了”。小李每天都这么想。

可工作太忙了，登上公司的电梯，行尸走肉般送到自己的座位，小李就开始了一天忙碌的生活。

空闲时候，他喜欢逛知乎，看怎么快速赚钱的话题。回了家累的又什么都不想干，只好刷下抖音什么的。

时间匆匆流过，每天他都想着单干，但第二天依然灰溜溜的穿衣上班。

小李觉得，干这一行，就要了解自己的优势。他的优势就是技术，所以还得花时间学习一些烂七八糟的技术，那些叫做 视野 的东西。

“我还没准备妥当”，他每次都这么安慰自己，“万无一失的时候我就自己干”。

小万，互联网爆发户

“风口、风口，都是风口”，小万笑呵呵的说，“我就是那头猪”。

刚毕业，小万找工作就遇到不少的障碍。他只会两种语言的hello world。一个是c，一个是java。

随着每次碰壁，小万的态度变的无所谓起来，他进了一个刚成立的小作坊，工作内容就是整理excel，只要饱自己就行。

公司也实行过一些非人类的制度，但想到出去后还要找工作，就坚持了下来。没想到，公司在短短5年时间里就上市了，作为头部员工，他获得了不菲的回报。

他的编程水平也有进步，现在会四门语言的hello world了。

小江，坚持是无谓的等待

小江终于明白了一个道理。如果你看好一家公司，就一直待下去，和它一直成长。苦也好，酸也罢，不亢不卑，不离不弃。

得出这个结论，是因为小江前几家公司都上市了，前同事都分了不少。每次都在他离职后不久，而且是在经济好像要出现问题的时候。

他的这家公司，据说也是独角兽。虽然从前年就有离职的打算，但还是咬牙坚持了下来。他要好好的待下去，套现，走人。

但也会焦虑，和他一块来的小王，已经被劝退了，那点赔偿都不够塞牙缝的。

“坚持就是胜利，我和它死磕到底”，小江用梳子打磨着自己脱发的前额，看着镜中的自己，说。

小阮，收租快乐

小阮找这份工作，完全是因为无聊。自从村里农改拆迁划入市区之后，他就辞职出去旅游了。

最后见识到更加有钱的人的生活，认识到自己的差距后，也实在是玩腻了，就找了份工作打打牙祭。

他花了很长时间，写了非常精美的微信小程序，开了个公众号，用来收租。房客想要租房，首先得关注他的公众号。同村的七大姑八大婶有了钱以后，基本上也不着家

了，他就都揽过来，替他们打点、收租，仅收取部分管理费。

“这群狗日的租客，老想着沾便宜”。他用力拍了下键盘，夹着的香烟烟灰四散，烫的小阮尖声叫骂。

小姐姐味道微信公众号首发。

小甄，奶爸

小甄真是幸福，朋友们都这么说。小甄的老婆特别牛，但孩子没人带，所以他辞职在家带娃。这一带就带了三年。

他其实挺喜欢写代码的，但程序员工资还是低了点。比起天天在家带娃做饭，他还是比较怀念写代码。

自从辞职后，他就只用python了，主要是做一些爬虫性的脚本，分析一些问题。他庆幸自己懂互联网，所以抖音的号也做起来了。

作为一个男人，他最怕老婆有外遇，那样他就一无所有了，所以变着花样讨好老婆。但他还是有些担心，所以总是偷偷学python。

“过几年我就去做人工智能”，小甄嘿嘿的笑起来。

小冯，兔子的艺术

企业大了，就要讲价值观。就像养狗一样，要有规矩。这时候，一些符合价值观，但没有能力的人，就上台表演了。

小冯深谙职场潜规则，老板表演他拍照，老板群里发言他点赞，老板提要求他表决心，就是不干活。

工作不就是crud么，他看的是编程之外的东西。

兔子本来就是一种悠闲的生物，他只是想要自己的一颗胡萝卜而已。

“我是马云眼中的兔子，是他们逼我这样的”。小冯对自己的定位直言不讳。

小艾，自由职业者

小艾实在是不喜欢天天挤着公交去上班的生活，所以他辞职，做了一个自由职业者。

小公司做事，大公司做人，自由职业者做什么呢？

他什么都做，哪里有钱，哪里就是他的靶子。

股市火的时候，他炒股。

比特币火的时候，他炒币。

更多的是接一些不痛不痒的活儿，勉强度日。即使这样，也是基本收个首付款，尾款根本就回不来。

最近他很焦躁。到处都在割韭菜，收入不增反减，他已经考虑再次打工了。

“我是没认识到自己的能力和资源”，小艾不无感慨的说，“其他人做这一行，应该比我做的好”。

小吴，工作+旅行

有没有想过另外一种生活？每年换一个城市去工作，游览完毕后轻轻的走，不带走一片云彩。

小吴就是这样，当他没钱花了，就去工作。大部分时间，他都在旅行。平均每一年，他就到一个美丽的城市落脚，去品味小城的故事。

在这过程中，他选了几个比较好的城市，都购置了房产。去年为了还贷款，还变卖了一套。

“爸爸”，有一天一个女人带着孩子费尽周折找到他。他头疼的很，而且感觉自己老了。接下来何去何从，他又一次产生了迷茫。

哦，那是梦的感觉。

小裴，职业经理人

一步一步往上爬，最终成为职业经理人，是很多人的梦想。小裴很早就有这样的管理意识，他认识到自己的专业能力很容易遇到瓶颈，也认识到仅靠一己之力，并不能完成高大上的目标，所以他做的准备很早。

他的努力的目标可能和大多数程序员的不同，他培养的是影响力。这些还不够，他还缺少经理人生涯中最重要的一环：有点说得过去的成绩。

可他现在还没有，这也是他一直在这家中型公司一直待到现在的缘故。努力不如借势，等公司名头响了，他就会更加值钱。

“没有成绩，我比你们还要焦虑”。他有时掏心窝子和几个亲信说，但看着他们躲闪的眼神，他不认为有人能懂他。

小宋，自己的事业

公司管理，要向传销看齐。小宋悟道之后，就经常和人分享自己的高见。皇天不负有心人，经过多年的摸打滚爬，小宋终于找到了自己的事业。有了自己的商业模式之

后，他就很少写代码了，但经常画图进行公司的架构设计。他现在非常痴迷一种画饼的技术，他的家里供着关公。

这些二流子程序员完全不知道架构的最高境界就是架构组织和商业模式。小宋曲高和寡，经常有种高处不胜寒的感觉。

“有些事情，你知道就好，千万不要讲出来”。他的好朋友劝他。朋友的身价已经很高了，这样的劝告很有价值，所以他最近收敛了很多。

小丁，精彩的生活

早晨6点钟，小丁一天的生活开始了。他喜欢跑步、登山、吉他、美食，就是不喜欢读书。

小丁的每一天都安排的满满的，他不是科班出身，大学是学兽医的，鬼使神差的做了写代码的工作，主要还是因为工资高点。

没办法，人聪明，编程需要的能力起点又低，小丁感觉这样的工作是为他量身定做的。

“工作是配角，生活才是主角”。每次看到同事加班，小丁就忍不住劝阻。

哦对了，老板马上就结婚了，他马上就可以正式的喊一声姐夫了。

小汪，孤独的行者

小汪的收入并不低，但他觉得很不幸福。高不成低不就的想法，养成了目前单身还有点自卑的他。他本来就不喜欢捣鼓代码这份工作，最近项目老出乱子，经常被批，心理也越来越封闭。他下班时间不喜欢有人打扰，女孩子都不行。

他养了三只猫，非常温顺。养猫不同于养狗，不用出去溜，他喜欢和它们呆在一起点个外卖醉生梦死的感觉。

“这个问题我不知道”，小汪放下工作手机。猫咪脊背上毛发传来的柔软触感，让他舒服的眯起了眼睛。

小祝，三线城市的希望

小祝对爱情看的很重，所以他现在呆在这个三线城市中。鸟不拉屎说不上，起码还是有些禽类的。修电脑的，就是亲戚对他的认知。

虽然在小地方，他可是有理想的，但一身本事只用了3成，就触碰到天花板了。过去几年，干倒了好几家公司，大多数是老板跑了。他现在正在琢磨动用关系，找点外包弄

点外快什么的。

“互联网红利该下沉一下到三四线城市了，机会是给有准备的人”，盯着大街墙壁上油漆未干的红标语，小祝依然充满希望。

小郑，二线城市焦虑

“别飘了。你爸拖关系给你找了个工作”。距离母亲5年前给他打电话，小郑在现在的单位已经呆了4年了。活少不累，每天就是喝茶谈心。代码什么的他已经忘光了，但偶尔，他还会怀念起在帝都通宵调试代码的岁月，那激情澎湃的时光。

“在哪混啊，都是为他人做嫁衣裳”。有时候，他凝视着夜空，不经意间还会深深的叹口气。

主要还是混的不太好。

小吴，国外技术专家

幸亏当时走出国门。小吴每次想到这里，都会对自己翘起大拇指。他是典型的技术男，也不想走什么管理路线。随着自己的技术越来越精湛，研究的内容越来越深入，资历也越来越深。相比较起来，国内的同僚就落魄的多了。

“现在，我可以干我喜欢的事情。或许等我老了，会回到我的家乡生活的”，小吴说，他在几年前就拿了绿卡。

小雷，东南亚彩博

小雷应该是走出国门里面最惨的那一批。当年，为了赚点快钱，他兴高采烈的接受了东南亚某公司的offer。结果，一封闭就是3年。在那里，他承受了超过国内三倍的工作强度，并且没有人身自由，心灵也受到了摧残。

经过一系列努力，他终于在去年逃了出来，一分钱都没拿到。

“现在，我真的能够坦然面对各种不平。世界上有很多灰暗地带，我只关心怎么去躲避”。他直视着远方，他的眼睛非常有神。

小关，赚钱的逻辑

小关的父母都在新加坡，他每年都要飞去看望几次。在国内，他也算得上是留过学的高等人才了，当然走的是不同的路线。刚读了某名校的EMBA，然后找枪手出了几本专业的书籍。这么优秀，他还是非常的忙碌，每天只睡6小时。随着在几个大企业镀金完毕，他也顺利的注册了自己的公司。

“越脏越臭的地方，机会越大”。父母想要他出国发展，他一句话就能说服。

小孟，就喜欢外包

比起其他程序员，小孟一直做外包，他喜欢这份工作。对于职场，他有自己的生存法则。

他专找能够出差，而且补助高的工作，顺便游山玩水。他还兼职开了个网店，所以不能上网的封闭式环境他是不去的。车补、饭补、住宿，这些会搞的都是有油水的。

他最看重的是每年能够连续休息一个月以上，想怎么happy就怎么happy。

“赚那么多钱干什么啊，反正房子又买不起”，小孟甚至连房子都不租，他一直在出差。哦，他还喜欢去相亲，目的不是结婚的那种。

小邬，心中的天平

小邬算是这座城市的成功人士了。通过帮忙打理亲戚的一家公司，他的人生已经开挂，薪资是远高于其他同龄人的。

自己的亲戚天天飞来飞去，一年也来不了几次。最近他学精了，招了个职业经理人前来帮忙，自己偷偷躲在办公室抱着电脑，不知道在捣鼓些啥。

“我哪点不如他啊，不就是靠运气么”。某天醉酒后，他嘴里小声呢喃着。

小顾，我爱996

小顾的生活压力是比较大的，几乎是月光。他和妻子都是独子，双方父母多病，都没有短期死亡的迹象，不是贡献者没有退休金。更要命的是没和老婆控制好，生了3个娃。

他在这家996的公司已经很长时间了，以至于他非常希望996合法化。他现在已经很少参与开发了，天天是一堆琐事，对外面的世界充满了恐惧。

“我虽然已经财务自由了，但我还是不服输，我天天下班这么晚，就是想要为理想拼搏”。小顾每次开会，都要给下属打鸡血。

服输俩字，出卖了他。

小谢， 羊毛党

薅羊毛不犯法。在进局子之前，小谢也这么想，但没想到这次的动静这么大。有奖励的地方，就有利可图。小谢入这一行时间较早，从养号到变现，整个流程都颇为熟悉。这些年来，他都比较小心，虽然也有些被反薅的经历。但每个月，还是会为他带来不菲的收益。这次主要是太贪心，金额有点大，竟然也用了自己的支付宝账号，被别人顺藤摸瓜找到了。

"我没觉得做错了什么，都是为了生活，我有什么错？“。小谢被拘留后很不服气。

他正想着怎么做更隐秘的黑产。

小孟， 转产品

技术转产品是非常容易的，还有不少优势。正因为这个理由，同时基于对代码的厌恶，小孟在三年前就转了产品。

可他有硬伤。他不太喜欢和人交流，只会默默的在背后画原型图；被怼的时候，他会特别生气，以至于一整天的心情都不好；他觉得每天都被那么多繁琐的破事给牵绊，开会、交流、项目管理，真的很烦。

“我的性格就不适合工作”，他把玩着手中的签字笔，“可我没这种命”。

最近他在看 javascript，他觉得前端应该会更简单一些。

小高， 一直走下坡路

小高的职业路最近一直不顺，但十年前也曾达到过巅峰。想当年，他是所有同学里最风光无限的。

回忆起他的第一次离职，他称为 激流勇退。

结果离职后，除了公司光环，啥都没有。

他接着带着暴富的梦想去创业，接连黄了好几家。重回大公司的他，不得已只能做些普通研发工作。最近更加凄惨，领导嫌他的年龄大，性价比不高，正在慢慢孤立他。

他研究孙子兵法，研究毛泽东理论，觉得是因为自己创业的几年把履历给弄坏了。

“人生是场持久战，鹿死谁手未可知”。他咬了口手中的煎饼果子，咽下时能看到喉结的耸动。

小崔，会跳的猿

终于在这家公司干慢两年了。小崔舒了口气，他已经成功的挑战了自己。

过去这十年，大多数时候，每隔半年就会跳槽一次，几乎没超过1年的，公司多的自己都不能够一口气说清。刚开始的时候，他傻不拉几全部都写上，结果简历厚厚的长长的吓人。后来，他才学会了合并这些经历，最近的这些工作找的就比较顺利。他从中得到不少好处，最明显的是薪资，翻了几翻。人又聪明，知识技能并没有拉下。

“我就奇了怪了，他们看重的到底是能力还是经历”。小崔有时候会很纳闷。

小关，专职割韭菜

小关的技术是非常棒的，但他的薪资一般。因缘巧合，认识一个做区块链的老板，一拍即合。

除了结婚，他从来没穿过西服。老板亲自带人来接，去照相馆照了张非常帅的照片，放在了公司官网上。

他的眼睛炯炯有神，自信而有魅力，他是公司的CTO，是精英。

但韭菜涨的有点慢，他就开直播，以技术的角度来解析区块链的好处。老板则以投资的角度讲解ICO的魅力。

他好想吃一张喷香喷香的韭菜饼啊。

“不赌一把，你都不知道自己的底线在哪里”。小关若有所思。

小贝，上车容易下车难

贝总，是大家对小贝的称呼。他们不知道，这称呼后所带的无奈。

小贝是四年前到的这家公司。鉴于看好公司的发展模式，他技术入股，老板承诺了股份，但并没有签署任何协议。

2年前，公司因财务紧张，在领导的忽悠下，他先后投入了200w资金，这部分老板倒是给他折合成书面股份了，但占比很小。

每次和老板谈他的技术入股股份，老板都闪烁其词，他也不好追问。但看到网上越来越多的毁约，他的心头也有不少焦虑。他的履历并不算好，上市前老板透露有美化一下管理层的需求，他就更加担心。

“我要离职，他起码得把200w资金的股份给我吧。况且，我们的交情...”

小贝现在开始研究法律书籍了。

小钟，猝死

小钟兢兢业业的工作，他是最努力的那一个。公司有绩效机制，他想多赚点钱。楼下是地铁，公司楼下也是，直通的那种。他已经好几个月没见过太阳了。

有一天晚上公司出了生产事故，他的手机叮叮叮响个不停，但他再也没有起床。
死在了家里，据说是猝死。

“公司压榨员工，必须给出合理的赔偿”！，小钟的爱人双眼欲泪，旁边小推车里躺着他们一岁的儿子，好奇的看着这个世界。

前几天，他还刚给儿子买了《少儿学编程》系列，这是他最后的期望。

END

人生的目的从来不是享受，也不是承受，而是体验。你的每一次选择，都有自己的理由。你的每一个理由，都会让你热泪盈眶。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

02. 企业内耗成瘾者

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

设计模式里有一种模式叫做反模式。今天我们以第一人称的口吻，欣赏一个反面的，有些熟悉的面孔。如有雷同，纯属巧合，权当消遣。

最近小宋和他的主管说，他快干不下去了。诸事不顺，做什么事都有人百般阻挠。知道这个消息的我，非常的高兴。小宋有着良好的职业素养，看问题通常一针见血。

他要是走了，就等于我砍掉了王总的一条胳膊。

再看一个例子

我可以再透露一下被我折磨疯的小胡。小胡是张总的人，但和我的工作也有交集。

有一天，我发现小胡正在会议室里和其他人讨论技术方案。其实这个方案，早就有结论了，为了这个结论，他已经争论了几个小时，这次就是宣告一下而已。

我微微一笑，把方案相关的小强找来，让他再review一下。小胡没办法，只好又讲了一遍。讲完之后小强不明白了，问为什么要采用这样的方案，小胡只好把讨论过程又重复了一遍。

就这样，1个小时过去了。

就在这个时候，被我偷偷通知的小强领导，出现在会议室门口：你们在讨论这个啊，结论如何？为什么要这么做？

就这样，又一个小时过去了。

这个时候我已经吃完午饭回来了。会议还在继续，我很满意。最终的结论我还是要看一下的，于是，我推开了门。

小胡只好又重复了一遍。我说不对呀，你为什么不“这么”做而是“这么”做？

小胡虽然有些不满，但还是要回答的。于是开启了另一波讨论。

讨论到一半，我的手机铃声响了。暗叫一声太棒了，我出去接了个电话，大约花了半

个小时。

再次回到会议室，我看大家萎靡不振，估计是饿昏了。我说，大家都辛苦了，咱们把结论同步一下，赶紧去吃饭吧。

话虽这么说，该讨论也是要讨论的，直到下午三点半，才结束战斗。

小胡说，不想吃饭了，没胃口。

我摇了摇头。你看这孩子，工作再忙，也要注意身体呀。

我的变化

从前几年放下键盘专职管理以后，我就发现有很多东西使我迷恋。

我喜欢上了说教，我要把我的价值观传递下去，虽然我自己不怎么遵守。就拿日报来说，我就要求他们一日三报：早报、午报、晚报，但我自己可不能写。我那么忙，时间要用在刀刃上。

我对能力强的人，充满了警惕，尤其是公司打包招聘的团队。他们严重威胁了我现在存在。他们竟然明目张胆的在公司拉小圈子，败坏风气，这我可受不了。我得划一个更大的圈。

我还喜欢间歇性发怒，那感觉很棒。处在职场总有一肚子气要受，有一群下属能骂一下，真是一种享受。我不是无事生非的人，骂人我那是有理由的。压力要层层传递才行，不骂人我镇不住这群小崽子们。话说回来，年轻人，没错就不能让我说几句么？

制约公司发展的一些问题，我其实都知道。CEO都不在乎，我在乎个屁呢？保持现状就好，照样有人投资，照样有工资拿，我只要对得起我的这份工作就可以了。怎么才算对得起这份工作？当然是加班了。企业招聘就是筛选一批志同道合的人。认同我的007，你才有机会。

其实，吃完夜宵回来，看到激情满满的办公室，我还是有一些成就感的。

找搜大船跟着起航，真好。公司的业务线，一直在增长。我深知自己的能力，守好我的一亩三分地就行了，但我发现，内心它不允许啊。

内耗成瘾者手记

对于我这种，通过内耗才能生存的人来说，手段有很多。我不关心什么结果，走出去

依然是xx公司的高管，依然有一群死党和我一起去创造未来。我无所顾忌。

首先来说加班。加班原因有千万，什么都比不上攀比性加班有意思。老板要看态度，我们就做给他看。其他部门也加班啊，我们不加，没法和boss交代。其实，对于我们普通人，时间是不值钱的，18块钱的加班餐，可以让一部分人留到7点；50块钱的打车费，可以让更多人留到10点。

要是你管理的好，那就拿一份拿得出手的报表，到老板面前邀功。如果数据不好看，那就在那段时间低调点，乖乖的闭上嘴。

总有人不喜欢加班，那就不是我的朋友。你就做工位上什么都不干就得了，挑挑拣拣，怎么和我一块创造辉煌？

有时候，部门离职率低，我会特别焦躁。没有流动的队伍不好带，新人才会特别的有冲劲。别的部门都那么高的离职率，我搞这么低有什么意义？如果项目有延期，我怎么拿这个来说事？

呵呵，其实我的人，从来就不够用。这就是我给其他人的感觉。

这些都是小case，要想耗的名正言顺，肯定要祭出我的杀手锏的。那就是流程和会议。这个可真是无敌的法宝。

一定要在会议上偏移主题，才能够开的又臭又长。我通常开会都带着水杯的。抿一口香喷喷的茶水，用尖锐的话题撩拨对方，就可以靠在舒服的椅背上，听对方解释。

由于我的技术水平有限，经常被打脸。刚开始我还不太习惯，后来我发现绝招了。千万不要谈细节，抠住几个牛x的名词发问就可以了。比如，高并发、高可靠、异地容灾、备份等。要是实在圆不下去，就直接说：

“这个问题非常重要，xxx你一定把这个问题给记着，回去出个方案。我也不是很懂技术，你们一定要肩负起这个责任”。

你瞧，谦虚点就行了。

另外一个好手段，就是把目标模糊化。企业发展一般都会定一个里程碑，里程碑就是目标。如果这个目标的意义和前景不是很明朗，那就该我这种人上场表演了。反反复复最有意思了，往往在讨论之中，时间点就过去了。

如果像是产品和研发部门这种积怨已久的部门之间的内耗，方式可多了去了。产品部门可以在这一周输出到研发一个需求，然后在下一周把它给改回去。只要没人追责，大可以放心大胆的这么玩。

显然公司是不希望变成这样的，临时的补救办法，就是流程规范。这是大公司才有的东西。小公司要靠它来运转，就已经病入膏肓了。可怜了执行层，他们最苦了。但给别人打工，谁不苦呢。

你会发现很多有意思的东西。大部分时间，大家都在强调规范、制定规范。但是到了遵守的时候，就很多人掉队。

那啥啥啥，自己觉得不好，但老板喜欢呀。老板给你画饼，同时你也在给老板画饼，这没什么区别。

最严重的，就是公司信用透支了。对内对外都臭的要命，就连我们这些混日子的，都无法舒服的待下去了。破败的前景很明朗，没有利益，还耗个什么劲？

End

我这么做很恶心，怎么会没人教训我？也是有的。不过是去年的事情了，我被揍之后就把他弄进了看守所。法律是保护弱者的。

你永远叫不醒一个装睡的人，尤其是喜欢装睡的人。但事情在变化，所以我今天被炒鱿鱼了，灰溜溜的离职，离开了我的舞台。

但是无所谓，多的是平台在等着我。我所拥有的表象，即使是最专业的侦探也无法看透我的内心。每一次职业生涯，我都是一张被抹去痕迹的白纸。因为，造成内耗的，加之让我上瘾的，正是飘忽不定，模棱两可，玩弄权术，利益输送的更高层。

橘生淮南则为橘，生于淮北则为枳。曾经招聘了那么多优秀的人，到最后他们都抹去了锋芒，参与到这场游戏之中，内耗不过是一种自我保护机制。其身不正，虽令不从。

要是遇到真正摊饼而不是画饼的知音，应该毫不犹豫的追随，哪怕飞蛾扑火。

这，才是解救内耗成瘾者的一剂良药。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

03. 《程序人生》杀机！

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

本故事纯属虚构，如同雷同，纯属巧合。

我是一个完美无缺的人，除了有一点爱吹牛的小毛病。我的这个鲜明的特点，成就了今日的我。因为有些东西，了解的人并不多，我只要装作胸有成竹的样子，那些专家就会认怂。

我用我无懈可击的方式，攻城略地，达到了一定的高度。如今的我，已经很多年没有脸红了。

“你确定么”？无论你多牛X，都经受不住我这句话连续问三次。第一次你还可能有点底气，第二次声音就会变小，第三次就会怀疑自己。我要是再多问一次，可能你就会暴走、恼羞成怒。

“我想你不是很明白这个东西！”这种斩钉截铁的论断，通常会让对方败下阵来。我不需要思考一些什么，我只需要装出思考的样子，然后说这句话就可以。但我通常不会这么明显的去说，这很伤人，伤害的这些人日后可能会报复我，所以“你确定？”就成了我当之无愧的口头禅。



我觉得不行

这个世界，有两种情况，对方是无法交流的。第一，对方是个傻X，而我拥有经验和智商的碾压；第二，我是个傻X，对方碾压了我。显然，一个傻X和傻X是相互有认同感的，而我只需要装作是一个对立方就可以了。

需要牺牲的是，有时候可有可无的一些脸面。这样多次之后，周围的人就都摸不清我的虚实：有时候看起来是个天才，有时候又是一个十足的智障。而其中道理，只有我自己会懂。

直到最近来了个家伙，我败下阵来。

我清晰的记着那次集体会议，我垂头丧气的抱着笔记本从会议室逃了出来。假如我有尾巴的话，恐怕我是夹着逃出来的。

一位小组长在会上，表述了他的高并发实施方案，并对其中的细节进行一一介绍。在他讲到一半的时候，我在瞌睡中抖了一个激灵，眼中立马放出了光芒。

“你确定这种方案不会造成什么影响么？”我问。

是你的R！



我希望用这种方式，达到两个效果。

第一，这个家伙平时老是针对我，我要用追问让他产生自我怀疑，降低他的方案权威性。

第二，我想要这个家伙恼羞成怒的反问：“那你说该怎么做？”我当然不知道怎么做，但我却熟知每一套系统中的要点，他的一句反问，我会用更多的反问回击。

“有没有考虑过回滚失败的情况？数据补偿是怎么做的？有没有考虑客服人员的压力？其他相同的产品，有没有看过？我觉得你应该借鉴一下...”

一般人，哦，不，是所有人，都会在我这种高频率的反问下手忙脚乱，最终黯然神伤。然后背地里，给我加上牛X或者傻X的称号。

我从不以为意。

正在我沾沾自喜的时候，从角落里射出一道目光。这道目光一出现，我就觉察到了它的异样。

我惊讶的转了下头，发现了角落里坐着的一个小伙。他并没有什么不同，就像是普通的互联网人一样，格子衫黑休闲裤，有点杂乱的头发和因为熬夜而稍显凹陷的眼眶。他有着圆润的下巴，消瘦的脸颊。懒洋洋的倚靠在旋转座椅上，鼓起了微微发福的肚子。

让我难受的是他的目光，那双死鱼一样的眼睛。那毫无感情的死鱼眼像是往上翻着，又像是随时会进入梦乡。我吃了一吓，顿觉索然无味。



我的这些小伎俩，他可能都懂。在这种目光里，深含着对我的蔑视和嘲弄。他什么都懂，可他什么都不说出来，就静悄悄的躲在角落里看我的笑话，这让我难以忍受。

这是一种不可言传的感觉，我讨厌这双死鱼眼。从这次之后，我就收敛了很多。我一直活在那双鄙夷而嘲弄的死鱼眼中，甚至会在噩梦中惊醒。

但是，由于在一个公司共事，打交道的机会也越来越多。每当我说话时，他都静悄悄的站在旁边，双手若无其事的抱在胸前，似笑非笑的看着我。他不说话，但比世界上任何恶毒的话威力都强大。我拙劣的表演不攻自破，有时候甚至会脑子短路，表达也出现了结巴的现象。

这种现象越来越多。在我兴高彩烈唾液纷飞的进行培训时，他坐在会议室中不停的冷

笑；当我和别人在工位上讨论问题的时候，他转过他的头，毫无表情的聆听。

我受不了了。这个人太厉害了，我完全招架不住。

我必须尽早行动。必须了解这个人的来龙去脉。

很容易从人事的手里搞到了他的简历。我默默的记忆，小心的求证，想看到一些让我心安理得的东西。

从这份简历上，我能够看到他的技术实力是很不错的，只有一定的技术底子，才能发出那么强烈的鄙视攻击。但从他简历的其他信息，看不出有什么不同。经历很普通，不算差，也不是非常出色，这个结果更加让我迷惑。

我开始了一系列的行动。从同事，到人事，到他的直属领导，进行了无微不至的关怀。对他进行了各种障碍设置，甚至关注了他的考勤。

在我的努力之下，他终于被约谈了。如果不出意外，他将消失在我的视线之内。

他离职的那天，我假惺惺的和他谈了一次话。

“接下来有什么打算么？”我问。

“还是做技术吧”，他的话不多，眼神依旧。

“为什么？”我喃喃自语。时至今日，他早应该退下他的伪装，对我再次嘲弄一番。

可是没有，他静悄悄的离开了，就如同他静悄悄的坐在角落。这一刻我恍然大悟，我可能错怪他了，也可能没有错怪。但我没有愧疚感，他不应该拥有这种能力，这双美丽的死鱼眼。

这个发现让我欣喜若狂。

接下来的工作，如往常一样的无聊和普通。激情不再，岁月如常，日子一天天付于流水。

不知多少时日。

同样的会议室，同样的技术评审。在我的小弟完成了设计的宣讲之后，有个新来的家伙晃着大脑袋，满脸傲气的问他：

“你确定这么做没有问题么？”

这是一剂多么无聊透顶的挑战，我简直要笑出声来。躲在角落里的我，精神奕奕，但目光阴冷。这道目光，似有破空之声，又像死在半空的鸟，风驰电掣，飘飘荡荡，死

死粘在发问者身上。

在这道目光之后，是我前不久练就的，一双无精打采，毫无活力的：死鱼眼。

作者简介：**小姐姐味道 (xjjdog)**，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

04. 一切荒诞的傲慢皆来源于认知

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

“一切荒诞的傲慢皆来源于认知”。任何反驳我这句话的人，都会对这句话提供了有力的论据。在说这句充满哲理的话时，我在想着今天的午饭。我特么真没想到会在这里遇到产品。

就是这家面馆，我的最爱，我经常带小伙伴们光顾。老板娘很肥，但也很美，说起话来像是夜莺在歌唱，鸟味十足。

我是一个程序员，带着黑框眼镜，文质彬彬，外表柔弱但内心狂野。今天，我照例带着五六个兄弟，去面馆就餐。由于不怎么吃早饭，所以我们中午吃的都很多，就叫了6碗加大的。

面馆很小，我们就挑中间桌子围坐，挤了个满满当当。

小孙还没有从工作的亢奋中恢复过来，从出门开始，就絮絮叨叨讲个不停。他通常都是这样敬业，但休假了我微信总也找不到他。

- 1、“你昨晚怎么又加班了？”
- 2、“赶工哦，有个客户说这个功能不实现，就要退款。”
- 3、“哦这事我知道，不是前天已经改完了么？”
- 4、“次奥，我还没说完，昨晚又要改回来。说客户觉得还是原来的效果好。”
- 5、“产品经理吃屎的么？把好关啊”
- 6、“这群猪！老子抠代码的时候，他在旁边刷抖音”

我们对面的桌子，同样坐着几个等面的汉子。他们安静的刷着手机，听到小孙的话，有人从鼻子里哼出一股浑浊的空气：“看来码农到哪都一样，以自我为中心，都没点责任感”。

我正好坐在他们对面，这才发现是产品部门刚来的几个小伙。长得人五人六，梳着油

光擦亮的头发。其中一个放下了手机，抬起了头，挑战似得看着这边。说话的正是这位爷。

坐在对面真不好受。此刻很安静，我眼光躲闪，看兄弟们的反应。

小孟前不久刚被这么评价过。他现在紧抿着嘴唇，我能从眼缝里看到他在桌下捻着衣角的大手。他的眉梢拧的很紧，能夹住一片树叶，然而眼睛又瞪的很圆，让我想起了被我家狗误食的弹珠。

小宋大大咧咧，无所谓的撇撇嘴。我知道他家里有矿，来公司就是体验一下而已，大不了拍拍屁股走人。姿态越高，越能容忍蝼蚁的侮辱。

小罗是个急性子，此刻他握紧了拳头，肚子一鼓一鼓的，就要站起来。我停止了观察，心想要坏事，赶紧用脚尖把他踢了一下。

突然，一道犀利的目光射向了我--应该是我面前的拉面。我心想坏了，该来的还是来了。他应该发现了我碗里的煎蛋。果不其然，这人大叫道：“这小子是他们的头，他碗里加蛋了”。我心里更慌了，想要矢口否认蛋又是不争的事实。

我想要争辩几句，但一想到后面的需求，我头皮就发麻。心想还是算了吧，家以和为贵。

“各位大哥，很眼熟啊。咱们应该是一家公司的。要不我请你们吃几个煎蛋吧”。我尽量压低声音，显的谦卑一些。

没想到他笑了起来。

“别整这些没用的，我看你也挺眼熟的。你应该就是xjjdog吧，让我们切磋一下”。说罢，他撸了下袖子，我赶紧后退了一下。

塑料椅子划在地面的声音，格外刺耳。

然后，我听到了几个问题。

1、”如何用程序证明先有鸡还是先有蛋？“

2、“马桶里安摄像头怎么计算水的波动，拍出清晰的图像？”

3、“如何用大数据分析出，王总喜欢什么颜色的内裤？”

其实，除了最后一个问我比较清楚，其他的我都没法回答。我起码也算个高层，知

道公司正在研究一款秘密爆款产品。王总特别喜欢内测，答案我知道，但我不敢说。

所以我沉默着。

我低下了头，眼角穿透了我的脚尖，我从来没有这么窘迫过。一年有12个月，这双拖鞋伴我大半段时光，此刻它已显老旧，像是咧嘴在哭泣。

“你看，这一个小小的功能，仅仅一个小小的功能，你都不能实现”。我的耳边依然有声，“人家A公司怎么可以”。

这个问题，我和他的领导都没法讲明白，所以我不期望短时间能解释清楚。

所以我依然沉默。

面快粘成一个了，我用筷子挑了一下。

这个动作触怒了他。“既然是和你讨论问题，就要拿出态度。你这种态度是不合作的态度，态度太差了”。

“这种不合作的问题我要反映给王总”，我替他脑补了一句。

...

我像被尿布塞住了嘴，说不了话且臭不可闻。

我抬头看了一下钟表，快1点半了，这么耗着不是办法，该回去上班了，下午要打卡。正在我为难的时候，老板娘手里拎着一把锋利的菜刀，悄悄的走到我们身后。我大叫不好，但为时已晚。

“你们刚才一直说的，不是云如厕app么？”。我暗自吃惊，保密这么好的app，不知怎么竟然让她给知道了。看来，我不了解的东西还有太多。

“至于王总的内裤，我知道”，她咯咯笑着，”王总最爱粉红色的“。

所有的人都意味深长的笑着，似乎发现了什么不得了的事情。

对面的小伙目瞪口呆，嘴里能塞进一个鸡蛋。

“至于有鸡有蛋”，老板娘没停下来的意思，“王总说先有鸡，就是鸡先；王总说先有蛋，就是蛋先”。

对面的小伙更是目瞪口呆，嘴里能塞进一个灯泡。

“赶紧回公司打卡，要不扣工资了”。老板娘 拍拍我的肩膀。

此刻的她，笑起来真是好看。像春天的花一样。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

05. 失联的架构师，只留下一段脚本

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

我对Linux非常的精通，尤其是脚本语言比如 `sed`、`awk`、`python` 等，用起来更是炉火纯青。我把它作为自己一个非常特立独行的技能，一个和其他普通程序员区别开来的技能。所以在我写脚本的时候，我都会自豪的抬高我的头，鼻孔朝天冥思精悍的code。

比如，看到别人一遍遍的翻文档安装 `elasticsearch` 这个软件，`xjjdog` 就浑身难受，就写了脚本来加快这个过程。

```
1 mkdir /data
2 useradd es -d /data/es
3 chown -R es:es /data
4 cat > /etc/security/limits.conf <<EOF
5 * soft nofile 65536
6 * hard nofile 65536
7 * soft noproc 65536
8 * hard noproc 65536
9 es soft memlock unlimited
10 es hard memlock unlimited
11 EOF
12 cat > /etc/sysctl.conf <<EOF
13 vm.swappiness = 0
14 vm.max_map_count = 262144
15 EOF
16 sysctl -p
17 chown -R es:es /opt/elasticsearch
```

这种脚本能够让我快速知晓软件安装的要点，不需要再读那些冗长的文档。像这样的事情，我总是在做，久而久之，搞的自己好像很闲一样。来看看我以前分享的命令行吧。

[最常用的一套“Vim”技巧](#)

[最常用的一套“Sed”技巧](#)

[最常用的一套“AWK”技巧](#)

这几天看到小王一直在那里捣鼓 excel，这些数据他已经处理了好几天时间了。客户需要从其他平台迁移到我们的平台，导出了一堆烂七八糟的数据，大概有三四十MB的样子。不知道怎么回事，清洗数据这个活儿，就落在了小王身上。

文件很大，公司的电脑很烂。小王打开之后，电脑的风扇就呼呼直转。他每次都需要使用 `ctrl+f` 找到不太正常的数据，然后把它么拷贝到另外一个文件中。数据多工期紧，昨天晚上，小王就加班干到23点多，直到夜的尽头。

总监对此专门进行了表扬。



我坐在小王的旁边，自然不能对此坐视不理。常年养成的习惯，让我对低效的事情无法忍受，就如同一只常年奔跑的兔子忍受不了缓慢爬行的蜗牛。

只扫了一眼小王的需求，我就判定这个工期三天的任务，使用脚本只需要2个小时就能完成。我并不是乐于助人，实在是我非常的喜欢写这种脚本，还有脚本带来的这种速度差异的快感。

一个小时之后，我把调试好的 python 脚本交给小王。shell里一运行，正确的文件就出来了。好爽的感觉。

小王自然对我拜服，逢人便吹xjjdog如何牛x。

这件事情不知怎么就被总监给知道了，我被叫进了宽大的办公室。看到总监一脸阴沉的脸，我知道事情不妙，但并不知道症结所在。我刚入职这家公司，应该没有在不经意间触碰了不该逾越的底线，我的心中充满了迷茫。

“听说你帮小王解决了个问题”。总监说，“以后少写这样的东西”。

“为什么？”我仿佛不太相信自己的耳朵，“脚本能显著的增加工作效率”。

“就知道你会有这样的疑问。”总监严肃的脸缓和了下来，和我讲了一个架构师的故事。



小宋曾经是这家公司的架构师。有很多三脚猫的架构师并不写代码，所以小宋成为了能码字的稀缺架构师。他的一个绝活就是写脚本，就像我现在干的事情一样。

脚本能增加效率，这是我多年的经验。但 效率 这两个字本身，就根本无法衡量。所以 效率 这两个字，无法被量化。即使你把工期从3天缩减到2个小时，那也不见得你的效率高，因为这只是零散的琐事中的一个小插曲，你省下的时间还是去摸鱼。你的这些效率，打破了正常的研究周期，也断送了想要拼搏的同学的梦想。所以， 增加效率 ，这种有实际功效的做法并不能登上大雅之堂，只能在小圈子里乐呵一下，最后只会变成一个口号。

小宋的脚本第一次是用在一个线上事故的处理上。当时，程序有一个BUG，数据库和缓存中一部分数据错乱，产生了不一致的情况。由于缓存分布在20多台机器上，就不能使用把所有缓存给清掉的方式。

业务经理很着急，经过讨论之后，决定开发定时任务，扫描所有的缓存和数据库中所有的记录，然后修正数据。数据量很大，程序也需要验证，估计修复时间至少需要两天。

小宋说，没那么麻烦。你只需要把问题发生期间，所有的业务日志给我就可以了。

接下来的三个小时，小宋从日志里过滤出了问题发生过程中所有被更新过的key。略一思索，就使用脚本完成了对这一批key的缓存删除操作，非常完美的解决了问题。

这件事之后，小宋就经常被请去写一些脚本来帮助处理疑难问题。他来者不拒，乐此不疲。

一切像是向着良性的发展，直到一次线上的故障。

公司的几百台机器，都是在aws平台上的ec2服务。使用ec2提供的api，可以做很多事情。但ec2的命令实在是太难以理解，所以小宋做了封装。

1 | <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/ec2.html>

使用这个脚本，可以对部分、或者所有的机器，进行批量管理（比如加个分组，开个权限等），就不用登陆到后台做一些管理工作。每当小宋看到黑屏幕上流淌的字符，他就想，这就是效率的魅力。

脚本非常好用，于是得到了分发。有一个运维拿到了这个脚本，鬼使神差的想要在线上验证一把。

他向所有的机器发送了关闭命令。

公司立马就炸了锅，扯皮的事是难免的。但最后的矛头指向了小宋。

脚本是他写的，但他首先是一个架构师。一个架构师，可以不写代码，可以不出成绩，但必须要有风险意识。

可这些命令，并不是我提供的。这是aws平台提供的东西，我对此什么都没有做。去测试这些命令，和用不用这个脚本没有关系。小宋争辩。

但命令确实是通过这个脚本发送出去的，也确实造成了后果。背后那些千回百转的曲折关系，没人会去研究，也没人会关心。那些平常享受小宋脚本的同学，此时也是默不作声，因为这种 责任 问题，如同 效率 问题一样，都是虚张声势，并没有明确的结果。

这种事情，小宋也是体验过的。比如提供了 redis 脚本，有人就喜欢运行 FLUSHALL 命令；提供了docker命令，有人就喜欢调用 docker purge 去做清理。这根本控制不住。

他认为，这是使用者的水平问题，而不是他的脚本问题。与总监吵了几架之后，小宋最后一气之下，愤而离职。

他直接删掉了这家公司所有人的好友，消失了。

“所以，我是对你好，才会和你提这样的要求。比起风险来，这点效率带来的提升，真的是微不足道。讲到这里，总监意味深长的看着我。”我现在就是及时的制止你的行为，免的步小宋的后尘”。

我点点头，这个道理我懂。之前有家公司，就因为觉得Linux的学习成本大，命令危险，就选择了Window系统做服务器。宁可难受、付钱买授权、当2B，也不敢越雷池一步。拿着风险两个字挂在嘴边，这些道理都是一样的。

我突然觉得，这么多年我的信仰就要崩溃了。花了那么大的力气，把命令行学精，到最后落到个学无所用，真是悲哀。

“总监，不好了。, 正在我想要表达些什么的时候，有个小伙子闯了进来，“线上有人使用ansible练手，把根目录的权限全部改成000了!“。



我的脑子里立马出现了这样的命令，这次锅是 linus torvalds 的了。

```
1 | chmod -R 000 /
```

谁知道总监笑了。“你刚来，不了解这里的情况。这个月已经发生好几次这样的事了。是时候换安全可靠的Windows了，我认识采购的人。”

我微笑着点头，不置可否，但虚心受教。

我灰溜溜回到自己的工位，长叹一声，删光了公司所有的人。

今天就提交辞呈吧，毕竟，公司是找不到 linus torvalds 的，但能找到我。毕竟， ansible 这个工具，是我前几天刚给推荐的。

所以，如果架构师是你，你会留下什么脚本呢？

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

06. 架构师写的BUG，非比寻常

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

部门新来了个架构师，BAT背景，住在三环，开宝马上班，有车位。

小伙子话不多，但一旦说话斩钉截铁，带着无法撼动的自信。原因就是，有他着数亿高并发经验，每一秒钟的请求，都是其他企业运行一年也无法企及的。这就让人非常羡慕，毕竟他靠这个比我赚的钱要多。

俗话说，要想在公司不出事故，那就不要写代码。干活多了容易出事，一身轻松无人问津，这就是现实。

但有时候还是要看成果的。新来的研发领导不懂技术，但他懂技术指标，所以就统计大家提交git的数量，如果git活动是一片绿色如A股，那就算过关了。

架构师思来想去，决定领一个**并发量最高的需求**：统计接口的平均响应时间和启动以来的请求数。

为什么说它的并发量高呢？这是因为，它是统计所有接口的，自然比每一个接口的请求数都要大。AOP代码一包，每个接口都得从他这里走一圈。

该我们的架构师上场了。代码如图。

```
public class Monitor {
    public static class MonitorKey {
        public MonitorKey(String key, String desc) {
            this.key = key;
            this.desc = desc;
        }

        String key;
        String desc;
    }

    public static class MonitorValue {
        AtomicInteger count = new AtomicInteger();
        float avgTime;
        AtomicLong totalTime = new AtomicLong();
    }

    private Map<MonitorKey, MonitorValue> monitors = new ConcurrentHashMap<>();

    public void visit(String url, String desc, long timeCost) {
        MonitorKey key = new MonitorKey(url, desc);
        MonitorValue value = monitors.get(key);
        if (null == value) {
            value = new MonitorValue();
            monitors.put(key, value);
        }
        value.count.getAndIncrement();
        value.totalTime.getAndAdd(timeCost);
        value.avgTime = value.totalTime.get() / value.count.get();
    }
}
```

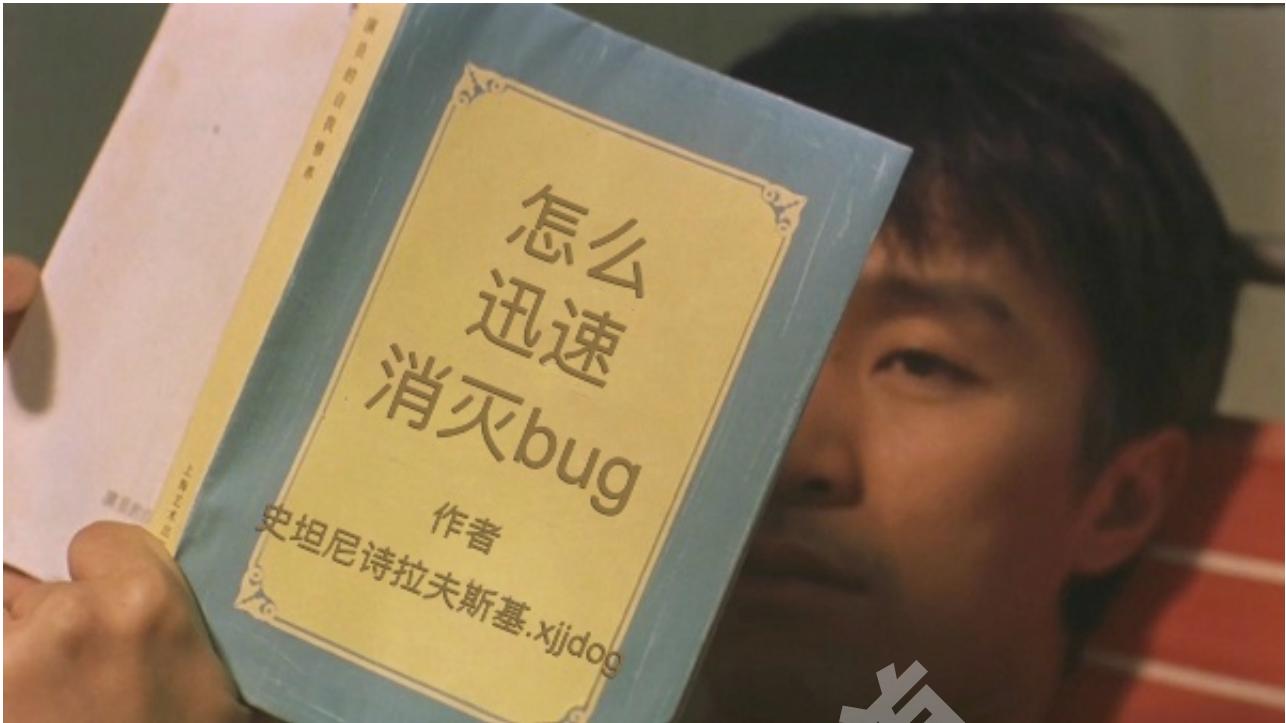
架构师说，我的代码不需要做注释。所谓的注释，都是给垃圾代码用的。我深以为是，他明显是受到了 Netflix 公司的影响。

程序考虑到了高并发场景，使用了线程安全的 ConcurrentHashMap，然后每次通过监控key取出相应的数据，然后在value上递增。这么简单的代码，确实不需要增加什么注释。

作为项目里并发量最高的代码，出于对高级架构师的信任，我们并不需要做什么代码 review，也不需要做什么测试。大家都很忙，代码您呐，到线上遛一遛吧。

我建议你先找一找代码的问题，如果你发现了问题，那就比架构师还厉害；如果你没发现，也不证明你比架构师弱，没有什么好伤心的。

下面插一副图，阻断一下思维。



装B遭雷劈，线上运行一段时间后，内存溢出了。

大家吵吵个没完，毕竟 xjjdog 说过，内存溢出问题的排查周期很长，大约平均需要 40 天左右才能解决问题。在大家开始论证的时候，架构师偷偷的启动了 **Eclipse MAT**。MAT 用来分析内存问题是非常合适的，但前提是你需要把堆栈给捣鼓下来。

架构师会用 `jmap`，最主要的是权限大，于是自己搞了一份拷贝到线下分析。

我能理解到他的心情，毕竟问题定位到自己的代码不是一件什么值得高兴的事情。他发现内存的堆里面，满满的全是 `MonitorKey` 和 `MonitorValue`。

```
1 | Monitor$MonitorKey@15aeb7ab
```

我和架构师关系比较好，于是他问我：咱们的接口是不是特别的多？

我说：不是啊，你别看访问量大，就这么个狗屁业务能有多少接口？几百个撑了天了。

他说：我在堆里发现了几千万个...

说完他就不言语了，因为他发现里面有不少是一样的接口。一定是参数的原因，所以他在代码里加了这个，把 ? 后面的给截断了。

```
1 | key = key.split("\\?")[0];
```

结果发布到线上，过不了多久内存又溢出了。这次终于引起了大牛们的注意，经过大家的分析，发现代码是忘了给 MonitorKey 重写 equals 和 hashCode 方法了。

我不禁脸红起来。作为好朋友，我不应该让他出这个丑。但我又是隐隐快乐的，因为他工资比我高。

所以这就是一个很大的问题。很多同学对HashMap的知识点对答如流，甚至还专门记忆了红黑树。但换一个方式去问，却又一脸懵逼。

其中一种问法是这样的：一个普通的对象，能够作为HashMap的key么？

答案显然是可以的，但需要注意重写hashCode和equals方法。如果忘记重写的话，大概率会造成内存泄漏。

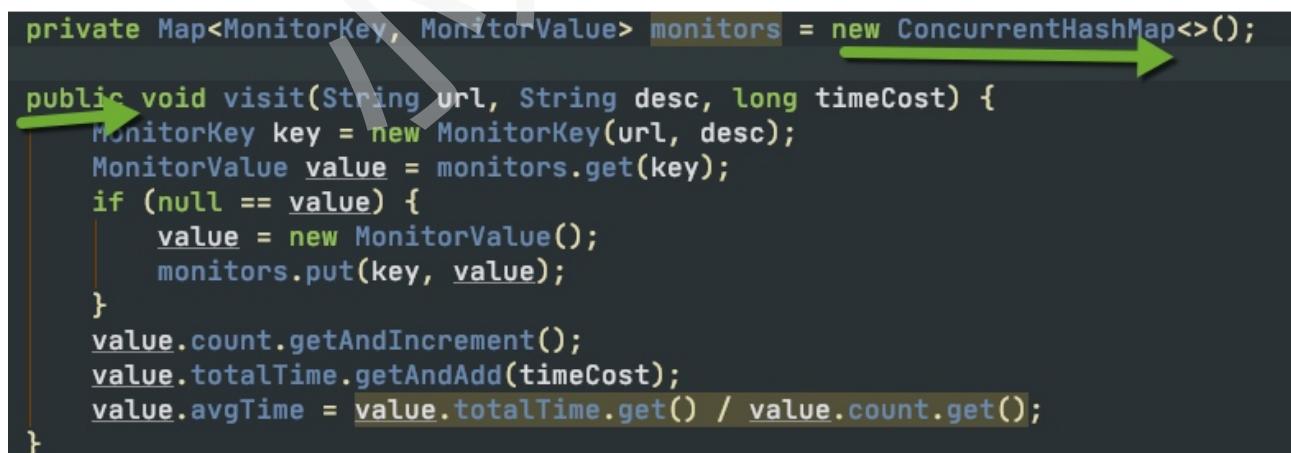
很不幸，现实中忘记的案例很多。大牛架构师也会中招。

代码重写hashCode和equals方法后，线上就再也没发生过内存溢出。

等等，还没完。毕竟是架构师，仅仅这样一个bug还是证明不了水平的。架构师写的bug，肯定非比寻常。

这种事出现的多了，研发领导对技术的权威性就不再是那么感冒。我们决定从并发量最高的代码开始，进行一下代码review。

很不幸，架构师的 visit 代码出现问题了。虽然问题不是很大，但它毕竟是个问题。



The screenshot shows a portion of Java code. It starts with a private field declaration:

```
private Map<MonitorKey, MonitorValue> monitors = new ConcurrentHashMap<>();
```

Below it is a public method implementation:

```
public void visit(String url, String desc, long timeCost) {
```

The code then creates a key and tries to get its value from the map. If the value is null, it creates a new MonitorValue object and puts it into the map. Finally, it increments the count and adds the time cost to the total time.

```
    MonitorKey key = new MonitorKey(url, desc);  
    MonitorValue value = monitors.get(key);  
    if (null == value) {  
        value = new MonitorValue();  
        monitors.put(key, value);  
    }  
    value.count.getAndIncrement();  
    value.totalTime.getAndAdd(timeCost);  
    value.avgTime = value.totalTime.get() / value.count.get();  
}
```

A green arrow points from the first line of the code to the 'monitors' variable, and another green arrow points from the 'monitors' variable to the 'put' call.

在统计数据的时候，代码使用了ConcurrentHashMap，但它并没有什么卵用。

visit方法，首先拿出了key，然后判空，再塞值。这明显不是一个原子操作。

- 1 线程1：获取key为a的值
- 2 线程2：获取key为a的值

```
3 | 线程1: a为null, 生成一个b  
4 | 线程2: a为null, 生成一个c  
5 | 线程1: 保存a=b  
6 | 线程2: 保存a=c
```

此时，B丢了。

业务可以忍受，但严谨的技术大牛们忍受不了，提出了修改的意见。

架构师说，给visit方法加个 `synchronized` 不就成了。

```
1 | public synchronized void visit(String url, String desc, long timeCost)
```

我说不行。有更优雅的写法，效率更高。那就是使用 `putIfAbsent` 方法，代码改动如下：

```
1 | MonitorKey key = new MonitorKey(url, desc);  
2 | MonitorValue value = monitors.putIfAbsent(key, new MonitorValue());  
3 |  
4 | value = monitors.get(key);  
5 | value.count.getAndIncrement();  
6 | value.totalTime.getAndAdd(timeCost);  
7 | value.avgTime = value.totalTime.get() / value.count.get();
```

大家就这两种方式争论了起来。

技术总监托着腮想了半天，看了看争得面红耳赤的同学们，说：这就是我不放心你们的缘故。线上环境要尽量保持稳定性，做最小的变更。既然加个 `synchronized` 就能够很容易简单解决的问题，为啥不直接用呢？下面这种代码改动太大，有风险。

总监接着把头转向我：这个BUG非比寻常，为了让大家引以为戒，你来做整个事故的复盘。把问题的排查和得到的教训分享给大家，让大家向这种至简的架构看齐。我们平常的工作中，也要尽量以结果导向为主，用什么手段无所谓，能漂亮把事情办好就行。

这就是此篇文章的由来，我虚心受教，同时也明白自己的工资是涨不上去了。

你要是点个赞或者友情三连，或许还能安慰我一下下。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

07. 又一批长事务，PO故障谁来背锅？

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

最近几周，发生过多起因为事务问题引起的服务报错。现象为数据库连接池连接占满，数据库连接长时间等待，最终导致请求线程hang住，服务大面积报错。这个时候，服务资源、数据库资源大量空闲，但就是进行不下去，影响是比较恶劣的。

谁来背锅？当然是架构师。因为这次所有的服务都活着，没运维什么事。

面试时，大家可能都会碰到关于**事务**相关的问题，升级版的可能是分布式事务的问题。在互联网行业中，一句马马虎虎的补偿事务就能蒙混过关，毕竟都是些短小精悍的接口。

但在很多企业级应用中，这行不通。我们必须直面惨淡的现实。

为什么要用长事务？

在许多业务非常复杂的后台系统，经常频繁操作DB，为了保证数据的一致性，能够在出错时回滚数据，通常会使用事务。

就拿最简单的单机数据库事务来说。

在事务操作期间，如果持续时间过长，只有等事务结束之后，DB连接才会释放，此类长时间占用DB连接的事务操作，称为长事务。一旦外部有大量请求，并发调用此操作，那么将会有大量的DB连接被持有而没有被释放掉，直到连接池爆满。

这个时候，如果有其他请求到来，那十有八九是以失败告终。

也就是说，连接资源被少数长事务操作占用。在这种情况下，即使是最简单接口查询，都不能够正常进行。

几粒老鼠屎，坏了一锅粥。

一些魔幻的反应

当你去排查这种问题的时候，可能会陷入僵局。jstack显示，多数请求其实是阻塞在tomcat的线程池上，而且是一些访问速度非常快的请求被阻塞。

比如，tomcat的200个线程，有180个阻塞在耗时不到1ms的/**status**接口上。

很多人就一脸懵逼。经验失灵。

jstack此时的输出结果，欺骗了我们。真正造成阻塞的，是那额外的20多个线程。

有哪些改善？

保证事务的短小是一个基本要求，包括但不限于：

应控制慢查询的调用频率，尽量减少慢查询。很多情况下，这条规则是自欺欺人的，需要业务做一些妥协。

事务内不应包含任何RPC调用，减少事务的粒度。通常，一些RPC调用，包括其他非事务资源的调用，耗时非常不可控。如果把它们也纳入事务的范围之内，势必会加剧资源的占用。事务内不应包含其他容易超时或者长时间阻塞的服务，如HTTP调用、IO操作。

次优先级服务如消息队列，不应该放在事务内，避免因为消息队列不可用引起的服务不可用。给类似消息队列的组件，设置一个合理的超时时间的非常有必要的，否则它就会一直等在那里。但即使是这样，也尽量不要把它们纳入到事务操作之内。

跨库、跨类型（如Redis），不应该放在同一事务中，可避免交叉影响。

你可以看到上面的这些描述，有些和我们所追求的数据一致性是相悖的。这不奇怪，依然是CAP原理的权衡。有些业务选择的是宁可卡死不再响应，也不能进入异常数据；有些则首先让业务运行下去，脏数据会通过补偿事务进行修正。

一切看你的选择。

设计总有人背锅，补偿总有人做出牺牲。

解决方式

那么如何来快速解决大事务造成的服务不可用问题呢？

除了扩容，其实是无解。重启大法也不见得好用。因为被阻断的请求，会以更凶猛的

态势再次来袭。

你可能会想到调大连接池的大小。但在实践中得知，也不好用，大事务请求会迅速将连接池占满。

但我们可以提前进行防御。

以Spring为例，事务的使用方式大多数是使用@Transactional注解来控制的，或者是声明式事务方式。我建议以下方式进行预防和发现：

- 1) 重新扫描或者Review业务代码，排查事务中是否有以上提到的各种情况。然后将除DB操作外的其他操作移动到事务之外。
- 2) 每个事务操作都给予足够重视，对于执行复杂度和时间复杂度不确定的事务，添加超时报警，及时发现引起的原因。

同时，还需要加强监控，辅助进行问题排查。

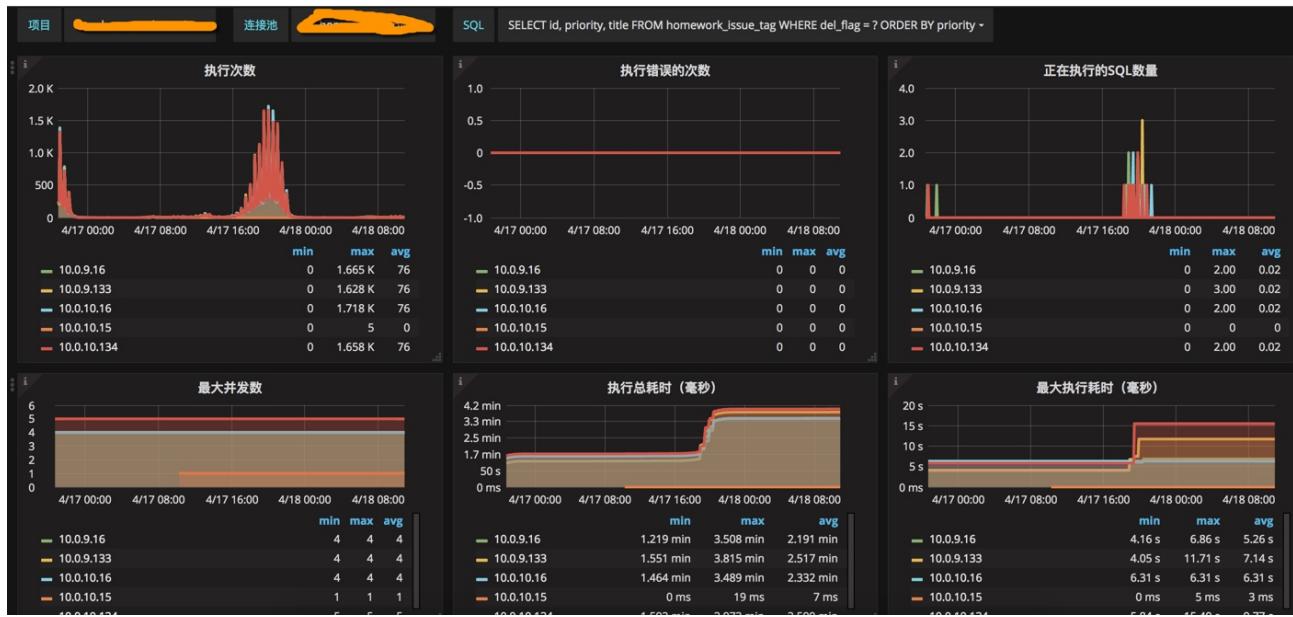
- 1) 业务可以考虑定时将数据库连接池的信息进行打印，通过看日志的方式进行初步排查。
- 2) 使用jstack查询执行栈，找出阻塞的点。
- 3) 排查并联系下游服务，找出主要原因

xjjdog倾向于使用监控快速发现问题。如图，通过连接池监控，可以看到数据库连接池连接数长时间保持在高位不释放，同时等待的线程数急剧增加。发生此种现象多数可以考虑是否是以上原因引起。



发生问题时，应及时（多次）使用jstack定位到线程的阻塞位置，然后排查下游服务是否有问题，或者是否存在慢查询。

最好的情况是服务已经进行了对代码的梳理，那么引起的原因大概率只剩下了慢查询。针对慢查询，druid数据库连接池，提供了sql的聚合，能够查看是每一类查询语句的具体执行情况。如图，短时间内SQL请求飙升，最大执行时长上升，连接池占满：



具体是哪一句SQL所引起的，一目了然。

End

长事务问题的危险级别属于高危型，通常会造成严重的后果，可以通过观察监控，防范于未然。

最优的解决方式，当然是业务模型的改进。但这东西第一涉及到开发成本，第二涉及到跨部门协作。

出钱的老板，无法听懂你这些梦话。

在一些公司内部，这两者都是让人抓狂的事情，还不如痛痛快快背个锅，来得实在。

08. 又一P1故障，锅比脸圆

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

好久没遇到故障了，以至于技术老大在公司会议上，都成了隐形的存在，除了汇报进度一无是处，这是不能容忍的。

只有有问题才能驱动团队的成长。风平浪静的时候，公司领导以为你做的不好，没有暴露风险：**既然已经做的这么完美了，要你何用。**

所以做的太好了，对团队来说未必是一件好事---尤其是对于有受迫害妄想症的公司来说。

1.代码写得好，bug少，看起来就像一个闲人

2.注释多，代码清晰，任何人接手非常方便，看起来谁都可以替代

3.代码写的烂，每天风风火火改bug，各种救火，解决各种线上重大问题，于是顺理成章成为公司亮眼的人才

4.代码乱的只有自己看得懂，才是公司不可替代的人才

5.多写bug，一个好的程序员带动两个以上的兄弟就业，这样整个代码业就会繁荣发展，程序员们也不用像现在这么累

上面这个截图，我已经见过很多次了，展示了一些无奈的现实。对于团队来说，也没啥两样。

及时制造问题，是一个合格的管理者应尽的责任。老板骂你一顿，然后接下来继续倚

仗你；比老板对你客客气气，然后接下来把你进火坑，要妙的多。

打是亲，骂是爱，客客气气要完蛋。

这就是劣币驱逐良币，公司管理要避免出现这种情况。但我们今天的主题不是这，而是一个P1故障。

这不，昨天下午，公司领导的电话就被打爆了：

“商户的账号都登陆不上了，单点登录崩了么？”

“这次影响面很大，用户什么都操作不了了，系统要完玩么？”

“抓紧处理！出了这么大的事，要协调起来！”

“查一下操作记录！是不是有人删库。”

事情很大，有客户直接call了老板。圣旨层层传递，终于到了苦逼的研发部。

躲在角落里的我，一脸蒙圈。赶紧使用测试账号登录后台。

您输入的密码错误，请稍后再试。

这也难怪有人想到了删库。最近互联网戾气严重，频繁出现 玉石俱焚 的事件。在各主管毫无掩饰的压迫和剥削下，这玩意一不小心就中彩了。

所有人都手忙脚乱的忙活起来。技术好的，把终端拖拽到大屏显示器上，噼里啪啦打着命令行；权利大的，叉着胳膊抱在胸前，唾沫四溅指点江山；脾气爆的，找几个管事的负责人劈头盖脸一顿臭骂。

光讨论组就拉了十几个，信息铺天盖地而来。

到了这个时候，保持冷静是非常难能可贵的品质。作为领导，要控制自己瞎指挥的本事，少 call 电话少 @ 人，保持基本的事件追踪即可。排查人会因为回复你这些乱七八糟的指令，错过了问题排查的黄金时段。

不要到最后过足了领导的瘾，领教了老板的狠。

这不，直到10分钟后，才产生第一条有用的信息。

有同学排查发现，所有的用户密码都被重置了。密码被重置的商户，重置时间都出奇的一致。

这也是得益于在数据库表中加了 updateTime 这个字段，证明操作是由业务发起的，

而不是 DBA 人为修改的。

DBA篡改的数据，只能到binlog里去找记录，而不会体现在最后更新时间上。

从这一点也可以说明，要是真有人搞事的话，不是一个DBA，而是一个程序员。

这个程序员不是通过update语句进行的修改，而是直接调用的程序。

大多数人直接想到了阴谋论，因为最近加班太狠了，又不给钱，容易触发到某个人敏感的神经。

能这么想，证明领导们还是有点自知之明的。

接下来，又有另外一个发现：所有的密码被重置之后，密码都是不相同的。

这就更加神奇了。如果是程序员或者DBA写的update语句，那么重置密码应该是一致的才对。而现在，用户密码都是随机的，难道是鬼干的么？

通过员工的挨个调研，发现在故障的这段时间里，并没有代码的发版行为，可以排除因为功能迭代这个影响因素。再说了，这个影响因素也不能乱报，容易让老板对员工的产出产生怀疑：怎么过了这么长时间连个版本都没更新，一TM更新就出问题了。

那大概率是外部搞鬼了。可能是有人在攻击。

很久之前，老板被人安利后，一拍脑袋，决定购买某大厂的安全服务。技术调研之后，发现并没有什么卵用，于是婉拒大厂。

正打算放弃的时候，服务就出问题了。流量和漏洞铺天盖地而来，最终老老实实买了安全检测和安全防护。

怎么就这么巧。

还有一次，有个小伙来应聘，说你们的系统千疮百孔。技术总监认为他在吹牛皮，无视之，水都没喝就让走了。结果，在晚上服务受到了疯狂的攻击，接近瘫痪。

怎么就这么巧。

百因必有果...

但这种得罪人的事，公司没少干，一时间还真想不出是得罪了哪位大神。领导们托着下巴冥思苦想，不得要领。要是捉不住这个攻击者，这锅就得有领导亲自背了。

安全的事情，还是得专业的人来做，该安全团队的人出马上阵了。

黑Term一开，机械键盘一抖，键盘和屏幕就晃晃悠悠滚动起来。一个主机，三个显示器，字符流淌如优美的生活。

这个时候，我相信安全人员的脑子是懵逼的。

因为他发现，事故，是由他自己引起的。

昨天，他刚刚使用了一款安全扫描引擎，然后，他在引擎中增加了一个安全扫描逻辑加弱口令验证检测功能，这个逻辑在执行期间，一直尝试登录，结果达到了重试次数上限，最终触发了bug。搞死了所有用户。

心病还须心药医，解铃还须系铃人。知道了原因，措施就好办。

安全工程师扭扭捏捏把原因告诉了面色逐渐变冷的领导，心想这P1的锅是没得跑了。

领导心想。如果不是他平素里表现良好，尽职尽责，除了给他个锅带，还要给他搞顶帽子，抓个典型。

后续的事情就容易多了，DBA从binlog里找到受影响的记录，然后整理成SQL语句，然后潇洒的一甩手，问题就OK了。

同时，这也证明了系统的脆弱性。假如安全工程师是竞争对手的人，那事情就复杂的多了，所以安全还需要继续建设。背锅的同志，将功补过吧。

由于问题发现及时，并未引起较大恐慌。这极大的体现了组织的协调能力，和应对紧急事件的响应能力。

我相信，领导一定会在公司会议上大放光彩。从接下来几天他红扑扑的脸上，我得到了验证。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

09. 面试官很牛，逼我尿遁

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

庚子年庚辰月，狂风肆虐，飞沙走石。似在风中藏了把刀子，砍翻了我头顶的连衣帽。

今天等来了久违的面试。在我的印象里，面试就是要面对面的。因为疫情的缘故，过去几周我经历了太多的视频和电话面试，总是凉凉。想到一本正经的面试官，下半身可能藏在被子里，或者蹲在马桶上，我就满脸的不自在。

主要还是因为我无法展示自己的风采，这让我很是苦恼。所以今天抓住机会，我要好好的表现一番。

车行飞快，绕城半圈。结账付款，一气呵成。我低下头，用手遮住屏幕看了下地图。心说：到了老弟，接受我的蹂躏吧。

我抄着口袋挤进电梯，不小心蹭到旁边的姑娘。她一脸厌恶的侧过身子，生气的划拉着自己的手机屏幕。为了在面试前保持美丽的心情，我自动将其忽略。

我是xjjdog，别的不会，代码很牛。搞了10年JAVA，组的都是关键零件。多的时候，也有四五十号人，喊我一声 牛总监。

做完自我介绍，坐在我对面的面试官一脸平静，没有任何表情。但我知道，他的内心一定在震颤，正在搜肠刮肚的找问题，来挫挫我的威风。

我向来是不吃这一套的。它强任它强，清风拂山岗。它狂任它狂，明月照大江。

就在我一愣神的时候，他刷刷刷在白板上，写下一段代码，让我分析一下最终的结果是啥。

我瞪大了眼睛一看，只见白板上，有这么个东西。空格还贴心的给我标注出来。

```
1 | int a = 2;
2 | int b = a++ << ++a + ++a;
3 | System.out.println(b);
```

我一看就头大了。最怕这种不加括号的炫技代码，它通常会带我回到痛苦的学生时代。

你应该把主要计算逻辑用括号扩起来，这种代码几乎没法阅读。我建议说。

他看了我一眼，没有言语，刷刷刷，加了几个括号。我定睛一看，惊出一身冷汗。

```
1 | int a = 2;
2 | int b = (a++) << (++a) + (++a);
3 | System.out.println(b);
```

我心想你要我不是？这括号加和没加有什么区别。我尝试着去推导这个过程，越分析越心惊。

面试官看出了我的窘境，他露出了久违的笑容。

我可以直接告诉你结果，是个整数，他的声音仿佛来自远方，结果就是 1024，你可以根据结果分析一下怎么执行的。

亲爱的同学们，你们也不妨反向分析一下。这个问题非常的刁钻，就连我现在平心静气的分析，都要花费很长时间。在当时剑拔弩张的形势下，在面试官死死盯着你的时候，我的脑子一团浆糊。但我的自尊不认输，所以我尝试着从不同角度去倒推。

当时我就想。1024 是 2 的 10 次方，但无论怎么进行分析，都不会得到这样的结果。短暂的思考之后，我认输了。

我们平常不会写这样的代码，一段的沉默之后，我只好拿出杀手锏，在我们代码 review 的时候，会尽力避免这种模糊不清的代码书写方式，这也是我们治理的重点。

面试官笑的更开心了，露出了整齐洁白的牙齿。他对我的回答非常满意。

我期望着他能聊一些比较高级的话题，比如分布式、事务、架构思想方面的问题。结果他一直在和我聊一些具体的实现方式。比如老掉牙的HashMap的底层结构，String的hash算法里，31是什么意思，什么时候会用到红黑树。

我告诉他，31是测试之后概率统计的结果，也是一个比较合适的质数。至于红黑树，它对我们理解和解决问题，并没有什么大的帮助，因为JDK8之前都是用的数组加链表，也没见哪些牛逼人物瞎BB。怎么JDK8一发布，就弄的和圣经一样？

聊了这么多不痛不痒的东西之后，我仿佛虚脱了。

对方好像特别喜欢一些死记硬背的冷门知识，或者是一些Java老版本里是真理，新版本里被推翻的知识点。随着时间的流逝，面试官脸上的笑容也越来越美丽。他仿佛成了一只吃人的老虎，张开了血盆大嘴。

- Java的PECS原则是什么？
- java8的ConcurrentHashMap为何放弃分段锁？又是怎么实现的？
- 能手撸一个准确的二分查找算法么？（纸上写，不给电脑）
- ZGC的染色指针是什么？
- ...

能问到这些东西，证明面试官也是非常的博学，如果我平常不是多看了一些xjjdog的文章，或许就被他给唬住了。

你用过设计模式吧。Java的单例，双重检查，能分析一下么？

听到这里，我再也压制不住心中的怒火。

双重检查，已经是一个反模式，sonar甚至专门有规则去分析。我们平常应该尽量避免去这么写代码，分析这东西有什么用？

你不分析一下，怎么知道是反模式？面试官不依不饶。

我觉得双方应该都需要冷静一下。同时，我也认识到双方之间深深的鸿沟。

这个问题我不太懂。您能给我普及一下么？，我终于认怂了。

他呆呆的看了我几秒，应该是没料想到我会有这样的问题。

我向来忍受不了尴尬的沉默。我捂着肚子，说：肚子有点不舒服，我能先去一趟厕所么？

他几乎是瞬间，点了点头。

我穿过噪杂的办公室往厕所走，看到一个工位上，电梯遇到的小姑娘，正在恭恭敬敬的听着旁边一个秃顶大叔，向她咆哮。

这些年轻人啊，向来没有怜香惜玉的情操。

我是真憋急了。找到厕所，宽衣解带开闸放水一气呵成，顿感无比的舒畅。

电梯就在厕所旁边，违背风水的设计。

刚走下几个匆匆赶来上班的员工，我侧身而入。从电梯缓缓关闭的缝隙中，映射出了我尿遁的事实。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

10. 996的乐趣，你是无法想象的

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

本文 虚构 了一个愤懣猥琐的loser形象，如有雷同，纯属巧合。本文具有魔幻现实主义色彩，优美的艺术来自生活，低于生活。请关爱因工作过度而造成抑郁症和精神衰弱的朋友们。

太阳

最近大楼里，冷清了很多。刚走到公司楼下，一滴水落在了我的脸上。我扶了扶接近1000度的眼镜，抬头望了下虚空。越过30多层的高楼，我看到淡红色的太阳光晕，孤零零的挂在天上。并没有下雨。

我突然想到了一个可能，一种叫做羊驼的生物。抹了把脸放在鼻子下闻了闻，果然是那个味道。我并没有生气，年轻人都很辛苦，这种人畜无害的发泄方式，我觉得挺好。

25层是家理财公司。通常，我需要等上十几分钟，才能挤上装满了大爷大妈的电梯，整个电梯充满了赚钱的正能量，载着平均保质期60多年的肥肉缓缓上升。今天却出奇的顺利，电梯直通31层。我这才想起这家公司已经倒闭了，大爷大妈们早几日已经鸡飞狗跳的大闹过一波了。当时几个大妈哭的非常伤心，我也在角落里笑的非常开心。

这座办公楼，是个大杂烩。有传统的石油公司，也有公务员办公的部门；甚至几个马上要资金断裂的炒币公司，也因为国家对区块链的重视，借尸还魂一样重新焕发了生机。

电梯

白天的工作满满当当，只有到了下班，我才会感觉到些许的无聊和落寞。

大楼从下午四点半开始，就有公司陆续下班了。他们的企业可能效益很好，但像我这样庸俗的人，自由对我来说是成功的约束，这样的生活，我一点都不想要。

我决定步行下楼，同时赠送他们一点礼物。

我从31层往下走。消防楼梯里的灯忽明忽暗，但我繁杂的脚步声，使得所有的灯泡都灼热的燃烧着，为我呐喊助威。对我这样文弱的女生来说，是个体力活。

30层没人，我按了向下的电梯，也按了向上的。
29层没人，我按了向上的电梯，也按了向下的。
28层有人在等电梯，我悄悄收回脑袋，奔向27层。

...

1层终于到了。我看了下秒表，这次记录不错，只花了5分钟。此时电梯停留在20层，非常棒。我超过了它。

结伴做电梯的男男女女，可能会因为这十几分钟，爱上对方，他们应该感谢我；一个老板模样的男人，打着电话焦急的按着电梯。看他的模样，应该是社会还没有教会他什么叫做耐心。

地下还有2层，是有车一族，我要继续...

散步

到了6点半，大楼基本就没人了，只有保安蜷缩在小隔间里，呆呆的抽着烟。

大楼前面是个人行道，但没有红绿灯。6点多之后，我最喜欢在这里散步了。

车很堵。红灯很短。有人很着急，滴滴滴按着喇叭。但是不要紧，我的脚步，踩着违章摄像头的频率前进着。我是个人，首先是个行人，对我这样的物体不进行礼让，交警叔叔会送给他3分。

但也有不吃素的，看见车速快的，我就像猴子一样敏捷的跳开。玩归玩，可不能玩脱了。

现在的汽车很是高级，都安上了行车记录仪。今晚，会有很多人能够记住我。

觅食

7点半就要开饭了。公司统一订饭，外卖准时到达。每当开饭时，我都能想象到我的童年。那个时候，我家养了很多猪，这种抢食的场景，似曾相识。

晚饭并不好吃，但我还是领了我的福利。菜汤很多，分量很大，我只尝了一猫口，就决定让它物尽其用。

我把菜和汤全部装在塑料袋中。我要带它们去旅行。

菜汤滴滴答答从袋中挤出来，奔走在办公室里和走廊上，肆意的散发着它们的味道。

我决定带它们去做我最爱的电梯。自从厕所安装了信号屏蔽器，这几平方的小盒子，

已经成了我的乐园。

电梯完美的到了1层。我欣慰的笑了。因为我，保洁阿姨又该加薪了。

行者

我就这样快乐的到了10点。其实我9点就已经离开大楼了，为了我喜爱的工作，我在附近租了房子。

但我想去远方看看，即使我的肉体不能与我同在。看着被灯光撕裂的斑驳的夜，我打开企业滴滴，打了辆车。

“师傅，夜深了，请你载着我的灵魂，随便到个什么地方去吧。”

快车司机看了我一眼，久久的憋出一句话。“神经病吧”！

我落魄的走开。这个世界，果然没有人能理解我。司机用疑惑的眼神目送我走远，竟默默的消失在夜色之中。

他猜对了。以前我得过精神分裂症，不过已经痊愈了。现在，我们生活的都很好。996的乐趣，只属于我们，他是无法想象的。

祝福

看着高高挂在天空的月牙儿，31层灯光依然，与月争辉。我发出了猪叫一样的笑声。这日复一日的一天，重复着一摸一样的欢乐，但每次都其乐无穷。

此刻的我，竟想作诗一首。5言绝句和7言绝句并不能表达我的感情，那充满了文人骚客的酸臭味。我这种理科生，要使用唯物主义的真假难辨的0和1来表达我的祝福。

浊酒一杯。

一敬苍天。

二敬明月。

三敬岁月静好。

四敬老板双亲身体安康。

五敬公司业务蒸蒸日上。

六敬未来全无过往。

这种祝福，带着我强大无比的精神力，没有时空，没有距离。纵使时光蹉跎百年，祝福对象遥遥万里，瞬间可达。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

11. 领导看了会炸毛的溢出理论

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

大家都知道，程序可以指定启动内存。但如果你玩的太过火，它就受不了了，就会溢出，不和你玩了。这不像是你伴着一席春梦的溢出，溢出的感觉不总是好的。

王小波在《红拂夜奔》里，谈到了领导上的一些趣事。如果你的身心过于健康，没有心脏病，反而会感到无比自责—这是对领导的藐视；同样在现实生活中，你如果太特立独行，没有平庸的潜质，那就是对领导上的侮辱。

怎么说呢？你拥有当领导的潜质，却被领导钳制，这就是根本问题。

不过嘛，领导喜欢某件事，我们就让ta溢出。这点觉悟还是要有。

来看看这2位小伙伴，对领导们的尊敬。

被微信骚扰的小吴

小吴的生活，全被微信给打乱了。

公交车上，他的微信就丁丁响个不停，这预示着他要开始处理问题了。

在睡梦中，甚至在厕所，他也会收到领导的消息。

“领导又在呼唤啦”。为了点开领导的微信，聒噪的女朋友，被他给搞了静音。

小吴苦不堪言。



但无论如何，还是得回家的。但见了女朋友也不怎么聊骚，行尸走肉。

忍无可忍的女朋友，第一次给小吴介绍了溢出理论。



溢出理论：溢出，就是采用正当的手段，极度满足对方的需求，直到对方厌烦并发出终止指令。

卧槽？还有这操作？

小吴就是聪明，稍一思考，定下了几项目标：

1、第一时间，积极回应领导的@。此时，应尽量多的与领导互动，多问为什么。要催促领导回答。如长时间没有答复，则主动拨通领导电话寻求帮助。

2、准备一个长长的任务列表，并追踪好进度。当新的任务到来，则同时回复现有任务，要求领导确认优先级。如果领导对此表示反感，则按照自己的优先级做事，此时问心无愧。

3、早晨起床，第一件事，就是对领导进行求助。可以准备几个棘手的问题，边聊边做公交车。如果领导表示很忙，则过5分钟打给他，表示很急。如果领导说可以上班再说，你就说熬夜到两点，这个问题困扰着睡不着。

4、晚上最后一件事，记得给领导问安。如果喝了杯咖啡睡不着，就可以把对项目的想法，分成几句，每句隔5分钟发过去。重要问题，一定记得打电话。

5、事无巨细，拉领导到所有的讨论组。并@他。

6、联系不上领导，打他家人的电话。

7、扩散此方法，一起表示对领导的尊敬。注意要有个预热的过程，不要突然间有如此的觉悟。

今年三月，领导因为压力太大辞职了。由于他对项目格外上心，现在，他已经成了新的领导。但每个周末，他都小心的自省，做内存整理。

不善交流的小宋

小宋就喜欢默默的写代码，写的不错。

领导表示，小宋不善交流，来了大半年了，没见过他滔滔不绝的样子。应该也没什么朋友，盘他！

绩效C，就给这小子了。

小宋非常沮丧的回到家，愁眉不展。女汉子样的老婆了解到情况，第一次给小宋介绍了溢出理论。



溢出理论：溢出，就是采用正当的手段，极度满足对方的需求，直到对方厌烦并发出终止指令。

卧槽？还有这操作？

小宋对此深以为是，立马组织了实施策略。老婆看后频频点头。

1、每晚练习演讲，训练口才。

2、准备几个话题，踩好时间点和领导聊天。尤其是在他忙的时候。拉住他，打断他的思路，让他厌烦为止。

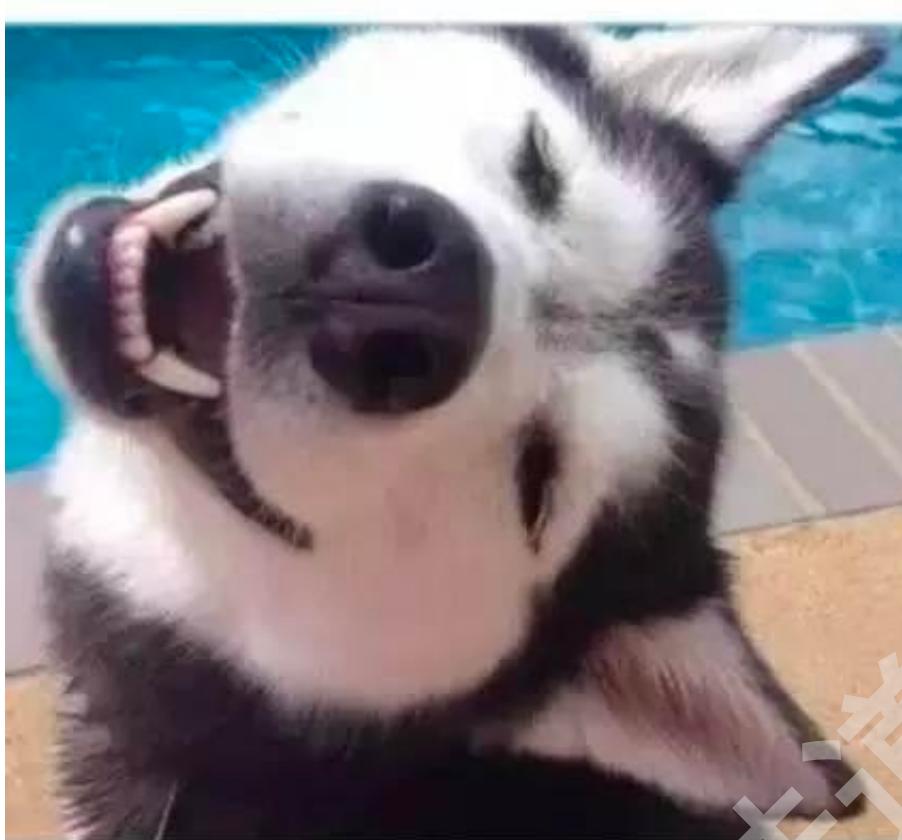
3、主动发送工作总结，并邀请领导进行评估和指导。不要怕被拒绝，因为自己觉得牛b的人，都喜欢教育别人。

4、每天准备几个简单的问题，问领导和周围的人，让他们满足于自己的睿智和你的智障。

5、多输出知识，可以先从写文章开始。念给他们听。

小宋有强大的执行力，所以这些措施，进行的很顺利。他很快就能言善辩，所有人见他张嘴、走近，就寒毛倒竖，因为接下来会预示着至少半小时的丢失。他有口臭，所以这段时间就是煎熬。

小宋很得意，领导表示小宋进步很大。但是经大家反馈，今年还拿了个C。



原因很简单，所有人都溢出了。

End

生活是不是并不总是如意？是不是总有逗比让你想哭又想笑？xjjdog给你介绍一下溢出理论。

溢出理论：溢出，就是采用正当的手段，极度满足对方的需求，直到对方厌烦并发出终止指令。

但也有例外，其中举例如下。

- 1、整个周边环境就是这样，无法独善其身。屠龙者终将为龙。
- 2、对方是类似于虬髯客的精力旺盛的人，你溢出不了他，他却能向下兼容你。
- 3、对方是经过层层压迫的产物，与你并无本质区别。溢出，属于相煎相泣的性质。

正确识别溢出需求，毕竟小年轻的优势就是身体的本钱。但一定要牢记，己所不欲，勿施于人。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

12. 蓝牙如梦

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。温馨提示：非技术。

人生总有尽头，思念没有终点。

在我写下这句话的时候，我在带着我的蓝牙耳机。

蓝牙，是一种高科技的通信方式。虽然它的传输距离很短，但由于它安全且抗干扰，被广泛应用在手持设备上。

我就有一块蓝牙键盘，一对蓝牙耳机，都好用的超乎想象。

我是蒋凡，你最爱的 xjjdog，今天35岁。35岁是一个劫，不管是对 P8 的程序员来说，还是对普通的编码者来说。尤其是庚子年的35岁，更显的与众不同。

牛年的我，首当其冲。

今年的劫难，是情劫。一切都因那对蓝牙耳机而起。

那一天早晨，天刚蒙蒙亮的时候，辛勤的人们都已经起床，慵懒的人们还在梦乡。在油条摊子摆起之前，我就踏上了996路列车。

什么？早餐？对不起，我的通勤时间太长，生活里没有这个概念。

太早了太早了。每次看到公交车司机打着哈欠把车停稳，我都会不由自主的发出感叹。

那天的雾气很大，以至于心情也是蒙上了一层阴霾。薄雾打在窗子上，又四面散开，不留一丝痕迹。挡风玻璃上雾蒙蒙一片，司机面无表情的掌着车，仿佛前方就是寂静岭。

我找了个靠窗的座位坐下，拿出了我心爱的蓝牙耳机。

亲爱的朋友们，不要觉得我啰嗦。从拿出耳机的那一刻起，我几乎每天都在回忆这甜蜜的时刻。它让我的生活不再枯燥，不再是行尸走肉般的无助，它是我新生的起点，是我人生的涅槃。我每天晚上睡觉前，都会用手摩挲着我的蓝牙耳机，那距离是如此的近，那感觉是如此的美妙而真实。

我的耳机名称是：xjjdog，我用的是最便宜的华为手机，只需要800块。

照例打开蓝牙，选择..连接..。这便宜手机，每次都要求我配对，我瞬间按下。

车的路程还长。在漫漫的长路上，我将倚窗前行，闭目倾听我爱的音乐，直到朦胧的到站声将我唤醒。路人上上下下，车辆来去匆匆，没人会注意角落里这具疲惫的身躯，这艘无趣的灵魂。

但是等等？我的耳机里怎么放起了这样的歌曲？谁在我耳边轻唱《一无所有》，是王健林么？

...

是否我真的一无所有

黑暗之中探索你的手

...

这是一首晦气的歌曲，唱它的人都将一无所有。就像35岁魔咒，拿他说事的人，35岁必不太平。

可是，我收藏这样的歌曲了么？我赶紧翻出手机，点了一曲《lost rivers》。

但我根本没听到这美妙的歌曲，反而听到了《忐忑》。

这绝对是灵异事件，我对科技产生了怀疑。难道在我意念的一抖之间，平行宇宙发生了错位？

举目四望，正好碰上了一个眼神。

我顺着眼神看下去，看到一个粉红色的口罩。再往下，是修长的脖子，然后是微微隆起又被紧身毛衣压迫的前胸，最后我的目光定格在她手里拿着的手机。

我愕然的抬起头，又盯上了她的脸。由于带着口罩，我看不到她的表情，但我能看到她慢慢弯起的眼睛。

经过短暂的思考，我明白了。我的手机连上了她的蓝牙耳机，而我的蓝牙耳机连上了她的手机。

时光落寞，擦肩而过之间，可能就是一个宇宙。

我靠在窗上，心扑通扑通乱跳。仿佛是默契，也仿佛是必然，我们都沒有关掉对方的

蓝牙，我的耳边依然响彻着熟悉又陌生的音乐...

啊啊啊咿呀吆...

车就在这悠悠晃晃中到站了。下车时，我回头看了一下妹子，她已经靠在椅背上沉沉睡去，似将生活抛到了九霄云外。

公交车渐渐远去，我听着耳中的音乐从清澈到模糊，最后断断续续直到消失，心中怅然若失。

她是否同我一样，在声音的不可达瞬间，从椅子上惊醒？心中有着一样的，触不可及的温柔？

晚上睡觉时，我将耳机放在枕头，翻来覆去难以入睡。身旁的老婆翻了个身，把腿搭在我的身上。

她睡的很香，还需要补充能量应付明日的繁忙。我也需要尽快入梦，明天还有重要的工作要做。

生活的平庸一直都侵蚀着我的两鬓，像小溪一样流淌，静悄悄发不出任何声响。只有旁边的蓝牙耳机，偶尔闪一下蓝色的幽光，进入我的梦乡，不知是否会进入他人的梦乡...

重复的996路公交车，第二天准时到达。今天天气很是晴朗，司机停下车，伸了个懒腰，我们点头示意。

我开了自己的蓝牙，但没有连接我的手机。

我期待着陌生的音乐到来，但是什么都没有。我向公交车后方望去，依然空空荡荡一如往常。有一束路灯光从车窗里，蛮横的照进来，又瞬间被黑暗吞没。

我摘下耳机，靠窗而坐，耳朵里响起熟悉的音乐。

...

多少冷漠我都尝尽

多少回忆藏在我的眼底

遥远的你是否愿意

为我轻轻点起一丝暖意

痛苦和美丽留给孤独的自己

未知的旋律又响起

...

是否我 真的一无所有

明天的我要到哪里停泊

...

从此，我再也没见过她。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

—13. 狗封喉

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

这是一篇无病呻吟的文章。作者对此厌恶，并深以为是。

初秋的风有点凉。我从后门走出酒店，想要一烟解愁。冷风穿过我的手指，在掌中横冲直撞，手里的烟却怎么也点不着。房顶的电线横七竖八纠缠在一起，将霓虹灯光切割的斑驳。

今晚喝的很尽兴，有些飘了。这生活看似饱和，实则空虚。

这时，我发现墙角里躲着一只小猫，紧张的望着我。使我感到好奇的是，有其他几只颜色相近的小猫，正在不远处嬉戏。看着这团孤单落寞的身影，内心深处的弦被轻轻拨弄。

很久之前，我就是那个喜欢躲在角落的人。

我想起了我的毕业照合影，我单足站在架高的椅子上，勉强入镜。

我想起和朋友的一次最远的旅行。人很多，我在后排努力伸出头，只能从帽沿下的阴影依稀辨认。

还有公司的团建，直到上下车点名，大家才会记起有我这么一个人。

我更加清晰的发现了令我惊恐的事情。

我喜欢坐在公交站的椅子上，呆呆的看着过往的行人。行人匆匆上车，又匆匆下车，匆匆走来又匆匆离去，我却从来没有多收获一个眼神。

外出旅游，总会选一些冷门清净的场所。对于大家趋之若鹜的胜地，还没去过，便升起一股厌恶之意。

唯有在这独处的时光，唯有这一缕冷风扑面的清醒，能让我如垂暮的老人一样，参读记忆。

记忆中满是平庸，如无风的湖水，放眼望去是一望无际的平坦。

最近家里很是热闹，原因是我养了一只活泼的小狗。从这只柯基狗明亮的眼睛里，我看到自己苍白彷徨的中年。青春转眼即逝，还没有开始便已结束，沉沦在无尽的加班岁月里。小柯基还是在耍宝、卖萌，如同一个默默的、尽职尽责的员工，期望获得老板的欢心。我爱惜的摸着它的头，我想要回老家看看，在我还有力气的时候。

说这种话还为时过早，但一锤挥出，十年下落，锤头挥舞的时候，什么都干不了。

家在小山村，离的不远但也不近，驱车大约是4个小时的路程。在全家举迁到新的城市后，我已经整整5年没有回来了。

轻轻推开尘封多年的门，如同垂暮的老人重逢旧地的迷失。尘土四散，阳光也变得恍惚起来。我不由的撇了撇嘴，嘲笑自己的神经衰弱，眼神却是在麻木之后，显现出的那种无比清澈。

这时候，我听到了门外柯基的惨叫。

农村多是恶狗，从你身边跑过却又不凶你，看似气定神闲眼神却充满警惕。现在的柯基，就被一群好奇的恶狗围拢着。从它们营养不良的毛色中就可以看出，这些都是无主的流浪狗。

我弯腰捡了石子，把它们吓跑，把惊慌失措的柯基抱起。

可怜的小柯基。

一只纯黑色的狗，走了不远，又回过头来，向我靠近。我心里发毛，顺手抄过旁边的木棒，慢慢向门里退。我虽文弱，但一只普通的狗，并不能伤我分毫。

等我看过了老家的一切，极度满足了自己无病呻吟。推门一看，却发现那只黑狗还趴在门口。

我愣了一下。在记忆的空间极速搜索，我尝试叫了一下：“小黑？”

“哈哈。小黑？多么俗气的名字！到处都是叫小黑的狗”。我想起七八年前领养了一只田园犬，给它起名字时家人嘲笑的话。

五年前，举家搬迁。我用钳子剪开它的铁项圈，抱了它好久，说了声：“自求多福”，它的眼神满是好奇。五年后，我们都走出了自己的影子，站在前排。

最终，我们从眼神中认出了彼此。听到它的名字，它的眼中闪出了光芒。狗是非常不善于伪装的动物，它们高兴的时候，尾巴就不停的摇晃。

我翻遍了周边的口袋，无以回报这种热情。终于从汽车后备箱里，找出了还没吃完的鸡腿。

小黑连骨头带肉一口吞了下去，甚至没有碰到牙齿。我心里咯噔一下，这种吃法，我无法想象，应该是饿坏了吧。怀里的柯基依然处在惊吓中，对已经吃腻的鸡腿没有任何反应。

接下来满是重逢的喜悦。我走在哪里，小黑就跟在哪里。像是阔别已久的老友，唯有这一刻不停的谈论，才能化解离别的伤痛。

突然，小黑直立站了起来。我记起，这是我教它的第一个动作，现在它的动作僵硬、生疏，应该是多年未曾练习的缘故。我看了下脏兮兮的小黑，犹豫了一下，张开双臂紧紧的抱住它。

good luck! 我默念一句。

一个人两条狗，六只眼睛，三种命运。

缓缓摇起车窗。副驾驶的柯基已沉沉睡去。汽车缓缓起步，在干燥的土路上，划破天际。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

14. 面试官画像（十年沉浮 | 番外篇）

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

有面试，就有面试官。他们的作用，说好听点，就是找到志同道合的人，共同缔造可期待的明天。

但不总是这样。面试官也是人，所以这个过程依然精彩。本篇是《程序员画像，十年沉浮》的番外篇，xjjdog微信公众号首发。

小郑：空手套白狼

小郑目前接手了公司的一个大项目，但苦于没有这方面的经验。项目大，风险也大，组里的几个ppt能人，这个时候不约而同的选择的沉默。HC数量有限，小郑只有靠自己解决了。

他最近花时间最多的地方就是面试，有时候连吃饭都顾不上。他问的问题都特别的细，似乎总能看清事情的本质；有时候又显得像一个十足的笨蛋。功夫不负有心人，通过了海量的面试，小郑终于在脑海里建立出项目的大体实现方案。业内的现状已经摸了个八九不离十，他现在已经有信心对项目进行把控和肢解，付诸实施了。

“你感受过一连几周一直在聊一个项目的感觉么”，小郑笑的和花儿一样，“让人终生难忘”。

小何：谦谦君子

小何热情洋溢，和他打交道的人，都如沐春风。他对人谦虚大度，谈吐得体，理所当然受人欢迎。小何同时也是这方面的技术专家，对很多问题都有独到的见解。

值得称赞的是，他把这个性格也带到了面试中，这给了候选者非常多的期待。通常非常明显得错误，他是不去指正的，候选人看到的永远是他笑呵呵的一面。当候选人茫然不知所措的时候，会受到他鼓励的眼神，能够继续把话题聊下去。普遍反馈的面试感受，是非常棒的。

“以后筛选简历，仔细一点”，走出会议室，他面沉似水，“我的时间很宝贵，这么明显的简历是怎么通过的”。

小罗：难得的休息时间

小罗的工作很多，但他还要抽时间进行面试。他的业务线，不仅需要大量高强度的编码，还需要大量的沟通协调，当然线上支持也少不了。最近的会议又多了起来，一般都是又臭又长抓不住重点下不了结论。每天游走在这些鸡毛蒜皮的事情中，他不由心生倦怠。

面试就是他的放松时间。有时候，他夸张的倚躺在椅子上，很累，感觉什么都听不进去。候选人讲解完第一个项目，他稍微交流几句，就要别人再挑个项目讲讲。他其实现在在想工作中几个棘手的问题。

聊着聊着突然一看手表，惊觉时间消耗已久。不能让面试者无功而返，他稍一观察候选人的谈吐风度，思询良久，在评价表上填上两个字：“优秀”。

“候选者有一股貌似催眠的神奇能力”，小罗不无感慨，“我要让他替我在会议上对付那些难缠的家伙们”。

小崔：你要先预约我才可以

小崔不喜欢有人催他做某件事情，他的面试安排都是提前预约的。这也预示着他面的很多职位，都是比较重要的。

表现的太爱才心切，或者突然把人放了鸽子，是他这种成功人士所不能接受的。

可推到他这里的面试，并不见得能够保证质量。有时候，他需要耐着性子听候选人的絮絮叨叨，表现出一股热情、专注的态度。偷偷看了一下手表，下午的商业预约马上要到时间了。在候选人一脸不解的表情下，他结束了面试。握手言别，一气呵成。

“除非是特别差的，一般我都是过过眼缘。经过了前面的考核，证明专业能力非常棒，可他们都太紧张了”，小崔潇洒的挥起高尔夫球杆，白色的小球在天空中划出一道优美的弧线，又在地上骨碌碌滚了老远。

没有进。

小甄：一张题目，万种变化

小甄向来是讨厌这些耗时的面试的。经理通常让他进行初步的技能筛选，这些细碎的琐事耽误了他不少工作的时间。

为了解决这个难题，他针对不同层次的候选人，准备了一批难度不一的题目。当候选人巴拉巴拉进行自我介绍的时候，他就已经把这些题目圈定了。而候选人的说的啥，他根本就没听清楚。不过这并不影响面试。

技术是一个基本的敲门砖，他一直这样认为。每天都面那么多人，他就是来判定这个人有基本的搬砖能力。他通常只出10道题，候选人答错一道，他就用笔在上面画个叉子。这个过程非常的快，一般10分钟之内就能解决战斗，不过一支烟的时间。

“这么简单的问题都回答不上来，都是吃屎的么”。他的脑子里维护了一个优先级队列，这个题目出现的频率，就让他给调低了一些。

小朱：我就是来客串的

小朱的公司，部门职责特别的模糊。一个普通员工，可能同时有好几个领导。通常一个普通的小问题，都因为信息不对称，噪音的不得了。

最近他友情客串了隔壁部门的面试，这是其中一位领导的拜托。他有着非常丰富的专业知识和面试经验，通常很快就能看到一个人的素质和潜力。

不过挂在他手下的候选人，最近明显多了些。原因无他，隔壁和他的部门，其实是非常严重的竞争关系。是人就总有弱点，他总能找到，并理由充分。

“其实我是在帮他们”，小朱安慰自己，“老虎就应该在森林里，龙的领域就应该是长空。这里的一只小犬，就能把他们撕碎”。

小苏：太优秀的，玩不起

小苏非常看重他目前的这份工作。从底层摸打滚爬上来不容易，他在这家公司工作其实已经很久了。最近公司有些动荡，他不禁回头仔细审视了一下自己，结果吓了一跳。

他是通过不断的招人来扩张自己的地盘。原理是这样的：自己的下属多了，就要建立层级关系，他这个老兵理所当然通过招人确定了自己的地位。

然而，三个臭皮匠，根本顶不了一个诸葛亮，他招的人多，离职的人更多。最后，他成功的聚集起了一群乌合之众。

他想通过招聘，引进一些新鲜的血液，但又怕失去控制，大权旁落。他有过太多对他不是很友好的案例。所以，他干脆把这种苗头扼杀在摇篮之中。只要有人表现的鹤立

鸡群，就被他砍了。

“集中式管理和分权管理都各有利弊”，面对来势汹汹的公司变革，他不无担心。

小唐：要有承担的勇气

小唐算是发现了，公司的坑有点多。这不，前段时间就发生了一次不大不小的故障，公司把责任人挂到他的头上。这种公司环境，真是让人有苦难言。

很多问题其实是历史问题，但目前明显成了他的问题。通过梳理现有的系统，他得出一个令人吃惊的结论：系统风险很大，可以预见的大坑就有很多个。

只有通过招人了，招那些有承担精神的汉子们。删库跑路这样的有职业污点的人当然不能启用，但也要留一些能够在关键时候勇于背锅的人才储备。

太优秀的，他不忍心这么做；太初级的，又没法出活。那这事只能靠缘分了，毕竟某些职位就是为了某些人而设立的。

“人要有承担的勇气”，小唐和候选人说，“这是领导和员工的本质区别”。

小倪：我们在等你

公司引进了一套绩效管理体系，其中就有末位淘汰这一项。可兄弟们工作都很卖力，协作也很好，如何适应公司的硬性规定，这是个问题。

小倪想，我们也搞个人才储备吧，就专招个用来垫背的员工。他有这种想法，就在面试中特别留意。

想法简单，但实行起来不容易。起码，他还是对这种想法有一种愧疚感的。所以进入目标范围的候选人，他还会问一些基本的家庭状况，不至于玩的太过火。另外，候选人一旦入职，工作交流，朝夕相对，毕竟会有感情。到时候，能不能下得去手，是另外一个难题。

“我们等你老久了”，小倪看着你，“我觉得你非常适合这个岗位”。

小郭：要有拼搏的精神

小郭习惯性加班已经很久了。虽然没什么加班费，老板给的承诺也并没有实现，但他就像是滚动的车轮一样，停不下来。偶尔歇下来几天，不是空虚就是生病，生来就是

忙碌的命。

他并没有结婚，因为他没什么时间和姑娘们吃顿饭。但他非常喜欢和女候选人聊天，并和同事们开着略显低俗的玩笑。因为时间问题，他的知识也不成系统，很多技能和观念都有点过时。

如果候选人对加班表现出特别的抵触，他就会显得特别的敷衍。他觉得，一个不知道奋斗的年轻人，肯定不会创造什么价值。所以面试，他一般安排在晚上，或者周末。

“年轻人要让自己忙碌起来”，他带着黑黑的眼圈和隔夜的兴奋，“迎难而上。否则追悔莫及”。

小周：风风火火搞校招

小周是这次校园招聘的总技术负责人。除了要培训一块儿搞校招的面试官们，他还要尽量的宣传他的公司。海报、新闻、现场，都要搞起来，这样才不虚此行。

面对海量的简历和热情的面孔，他每天都工作到很晚，也很累。到现在已经去过了很多学校，这次到的是一个并不太有名的大学。上面的意思是，人就不招了，但宣传还要继续。

“我们的名额已经满了”，面对一个跑到后台追问的男孩，他不好意思的说。

“其实你即使签了第三方，即使你去实习了，公司还有的是办法毁约的”，他想。

小冯：相逢是缘分

第一次接触小冯的候选人都会特别紧张。他会盯着你的脸呆上好久，眼睛连眨都不眨一下。在这种攻势下，候选人的自信心会受到特别大的打击，不自觉的开始怀疑自己。

其实大可不必这样，他们都错怪小冯了。如果你仔细观察他，会发现他的眼神其实是非常迷离的，只是因为他的眼睛小，又带着又黑又大的眼镜，才显得不是那么注目。

其实他现在有可能在看你脸上的一个斑点，他回忆起他的姥姥在脸上的同样部位，也有一个。可惜她没有享受他的孝敬，就去世了。候选人不时迸出的专业名词，就像脉冲一样把他不断拉回现实，让他产生一种似曾相识的感觉。

他心平气和的和候选人聊聊家常，有时候聊的有点过分，会让别人产生一种检查户口本的警觉。还好，他的过滤水平了得，总能找到和自己合得来的未来同事。

“生活都不容易”，他说，“擦肩而过也是缘分，就如同我现在在读xjjdog的文章。一个道理”。

小苗：我是个雏鸟

小苗最近刚接触面试，他很紧张。他觉得自己比当候选人的时候都要紧张，他总怕自己出丑。借着空闲的机会，他把面试的知识重新复习了一下，但还是不放心。他经常听到候选人说一些自己不懂的点和技术，这让他觉得非常沮丧。

他经常收到一些质疑，面对候选者对一些问题的发问，他有些动摇，但用强大的内心压制了下去。他现在，经常作为学习者，旁听组内其他牛人的面试，进步很大。

“对我前段时间面试的一部分候选者，要说一声对不起了”。小苗腼腆的笑了笑，“我还是要继续提高技术水平呀”。

小卫：面试就是学习

小卫非常爱好学习。他坚信，三人行，必有我师。

今天他刚学习了一个知识点，很偏，但他觉得很新鲜。他尝试着在面试中和候选人进行交流，结果发现对方竟然对此十分精通。他们于是在会议室讨论了很久。

小李对新知识有了新的见解，同时对候选人的沟通交流非常满意。虽然有些问题没有考察到，但他觉得，一个有工匠精神的工程师，在其他方面一定不会太差。他希望身边多一些这样的同事，这样他的发展就能够更快一些。

“我还要了他的微信”，小卫很兴奋，“我发现他还是行业的大牛”。

小鲁：一块回家吧

小鲁的家离公司非常的远，他的通勤时间非常长，开车会缓解这一过程，但花费很高。公司倒是有报销，但要熬到10点才行。一旦碰上领导下班时开会，时间上不上不下，那叫一个惨。

偌大一个城市，住在一处的同行还是有的。他在面试中，就特别的留意这一点。遇到住在附近的候选人，就特别的关切。可大部分人虽然家在那，都会在公司附近租个房子。

“一块打个车，对公司来说也省钱啊”，他不好意思的摸了摸脑袋。

End

孙子兵法说，知己知彼，百战不殆。同时，信心比黄金更重要。很多时候，不是你不优秀，是环境太复杂。

虽然现在正规的公司，都会对面试官进行专业的培训，教他们怎么去沟通。但面试的测评，注定客观中带着不可预知的主观。谁也无法确定，坐在你对面的，脑子此时正在何方神游。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

近期热门文章

[《必看！java后端，亮剑诛仙》](#)

后端技术索引，中肯火爆

[《Linux上，最常用的一批命令解析（10年精选）》](#)

CSDN发布首日，1k赞。点赞率1/8。

[《这次要是讲不明白Spring Cloud核心组件，那我就白编这故事了》](#)

用故事讲解核心组件，包你满意

[《Linux生产环境上，最常用的一套“Sed”技巧》](#)

最常用系列Sed篇，简单易懂。Vim篇更加易懂。



15. 一杯苦咖啡 | 公司来了个漂亮女实习程序员

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

我知道，大假期的，很多人都放飞了自我。技术文章太烧脑，在假期容易看过就忘了。所以奉上小故事一枚，感叹一下岁月的蹉跎。温馨提示：本行文来源于现实，但超脱于现实。如有雷同，请勿对号入座。

公司来了个女实习生，叫瑶，说话眸中带笑，漂亮非凡，直男斩！

从她刚来的那一天开始，我就喜欢上了她。程序员本来女生就少，何况是这么鲜艳娇嫩的一朵花。

众所周知，凡是女生出没的地方，周围总是少不了狗的吠叫。一个个腼腆的小伙子，可能就因为电梯里有一位 lady，就能为了一个没有意义的论点，争的面红耳赤，唾沫横飞。

可怜的我们，只在两个地方，毫无保留的勃起自己的羽毛。一个是领导面前，另外一个就是在女孩子面前。



作为小领导，我有特权与她多一些接触。业务上的问题，我尽心尽力解释；技术上的问题，我耐心的教导。短短几个月时间，我说了比过去十几年都要多的话，我觉得自己变成了话唠。

每当这时，她都抿起小嘴，半歪着头似笑非笑，我觉得特别迷人，她让我想起了爱玩 摆头杀 的小奶狗。但问题是，她的领悟能力实在是有点低，一个简单的问题重复了十几遍，还是似懂非懂。

不会是装傻，故意营造和我相处的机会吧。



这样的日子挺好。

问题是，我发现组里的小王，也贴别的殷勤，有事没事和瑶套近乎。老家离了200公里也能称作老乡；还经常下楼给妹子拿午饭，我也闹不清是谁买的。反正是，其心昭然若揭。

我是属于较含蓄的那种人，不善于将自己的感情表达出来。感情越是热烈，我越是喜欢折磨自己，但这热情的火焰，从未冲出过体外。所以我这段时间脸上长满了痘痘。

时间是最好的催化剂，我和小王都能看到彼此眼中的寒光。但是，由于我毕竟是个小leader，是可以给他安排活的，所以每次都给他多分。但这小子活力很大，忙里抽闲，依然对妹子献殷勤。

我觉的这样的日子并不会长久，我需要想个办法，像个男人一样处理这件事情，因为它已经严重影响了我的工作和生活。



刚开完一个无聊的会议，屁股还没做到工位，我就发现办公室炸锅了。

公司服务无数据返回，疑似删库。

我见多识广，对故障见的多了，现在也能够练到面不改色。千奇百怪的线上故障，就是我们存在的价值；管理上的疏漏造成的故障，就是领导们存在的价值。

删库？又能如何？我们有备份，有应急机制，这样的事不用这么猴急吧，看他们一个个都想钻进显示屏里。

但是等等，运维查到的操作账号，来自我们组？

背后的寒毛不禁立马竖了起来。

风轻云淡是一回事，落到自己身上是另一回事，这件事确实应该猴急一下。

我立马行动起来，联系各方资源，尽快的恢复数据，将损失降低到最小。经过一夜的折腾，终于把数据和业务给恢复了。

波澜荡尽，到了追责的时间了。

但事实非常的让我纠结。库，竟然是妹子删的... 也难怪，我给这么一个未上过战场的小战士，分享高权账号，还教她练习危险的命令，明显就是作死行为。

我的冷汗再次冒了出来，嘴里安慰妹子不要太担心，思绪已经游离到九霄云外。

突然间，我有了一个好主意：我可以借这次机会，把小王给干掉！那么，组内故障复盘会不用开了，就有我来直接和领导汇报。

我觉得自己真的像个男人了。说干就干，我推开了领导的办公室…



深刻的反思了这次故障中得到的教训，并对账号管理上的漏洞进行了深刻的检讨。尤其润色在故障恢复中，我起到了关键的协调作用。我含蓄的透露，是小王，将高权账号交给了实习生，没有风险意识，需要得到严肃的处理。

领导笑了，“小王已经和我反应过这个问题了”。

我心里咯噔一下。

“他已经承认了自己的账号透露行为，负这次事故的主要责任。这次事故很严重，我估计公司会给比较严厉的惩罚措施”。领导接着说。

小王这个傻瓜，为了妹子可真是殷勤啊。

“另外，你作为这个组的负责人，抛不开关系。平常是怎么强调风险意识的。要我说，你和小王负一样的责任”。领导的话还没玩。

。。。

三天之后，公司的处理通知下来了。我和小王，由于没有风险意识，对公司造成了非

常大的损失，公司作出辞退处理。保留带薪留职一个月，就可以收拾东西滚蛋了。

我脑子里一片浆糊。哎，认命吧。我坐在座位上发呆，一直到同事们陆陆续续的下班。

我破天荒的加了个班，心情稍微平复了一些。泡了杯咖啡，抄着裤兜，靠在窗子上看城市的夜色。

路灯很亮，减去了月亮的光辉。不同角度的光相互交叉，路上竟然没有影子。借着迷碎的灯光，我看到领导的车停到楼下，瑶从拐角处走了出来，径直上了车，缓缓开走。

我苦笑一下，使劲抿了口咖啡。这苦烈的味道，浓郁悠长。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

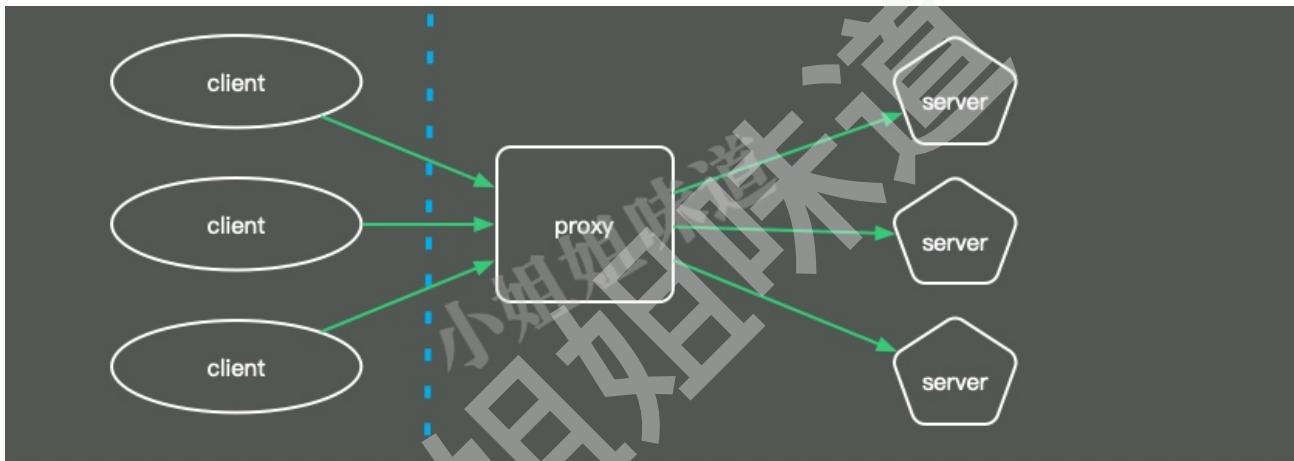
16. nginx工程师，需要上承天命，下召九幽

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

这是一篇典型的 xjjdog 风格文章，扯淡比技术多，记得点赞哦。

nginx很好用，几乎每个公司都在用，它的功能主要有 反向代理、负载均衡、动静分离 等，其中最主要的就是反向代理。

它的主要架构图如下：

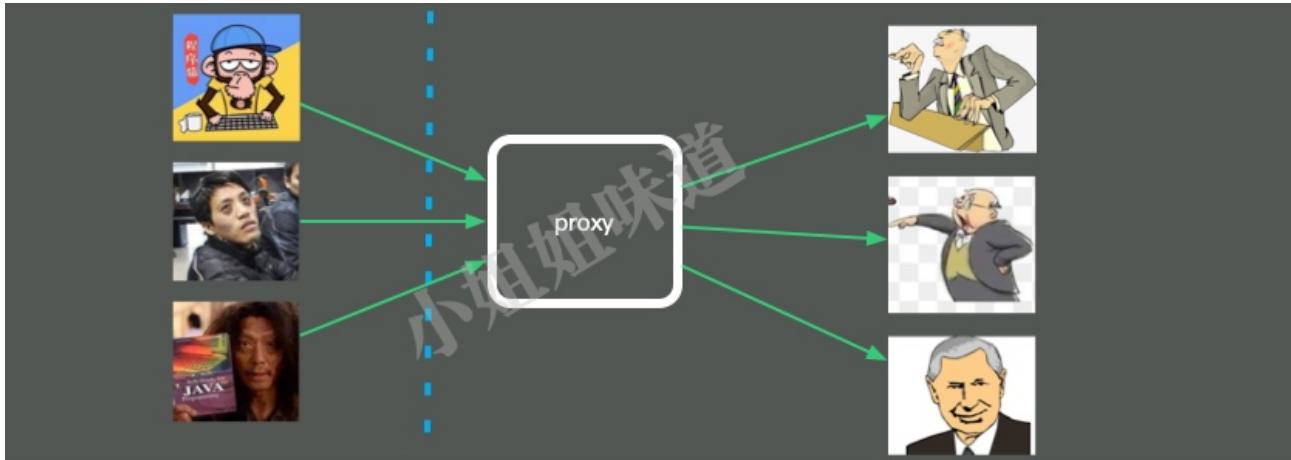


客户端是无法直接访问后端的server的，因为server资源高高在上，client并没有权限。这个时候，就需要有一个proxy，进行代理，将客户端的请求进行转发，然后把结果返回给client。

为什么不能直接让client访问server呢？这就涉及到资源隔离问题。server的一些真实意图，是不想要client知晓的，比如一些受安全保护的敏感资源。也就是说，通过 proxy，server想让client看到啥，client也只能看到啥。

对client来说，功能总是阉割的！甚至是篡改的！

我们将图中的元素替换一下，奇妙的事情发生了。



我们把client换成可怜的 普通码农， 把server换成 牛x的领导， 可以看到6条绿色的箭头， 带着呼啸的山风， 无情的指向了 proxy 。

proxy成为了关键的资源。这个proxy要求可不简单，需要五弊三缺，上承天命，下召九幽。这么牛x的存在，当之无愧称之为nginx工程师。

nginx工程师不需要有自己的想法，但他必须记住领导的每一句话，领会领导的真实意图。传导领导想法的时候，也知道轻重，知道过滤，能够快狠准的将领导的想法广播出去。

不服不行， 这还真不是一般人能做的。

哪些人能够成为nginx工程师呢？

在奴隶社会，无论一个奴隶（slave）工作多么的卖力，能力多么的出众，都没有出头之日。反而是那些拿着鞭子的监工们，更容易获得主人们的赞赏，喝酒吃肉的时候都能够想到他们。

更加奇妙的是，如果把监工们和奴隶们的工作调换一下。奴隶 有可能 会胜任监工的工作，而监工 绝无可能 胜任奴隶的工作。

这是为什么呢？总结下来可以有这么几点：

1. 监工知道更多的内幕资源，更容易投主人所好
2. 监工的工作属性是管理，可操作空间大
3. 监工离权利中心更近，耳边风吹的频率要高一些

你可能会说，这是社会性质使然，现在的社会肯定没这问题。你答对了一半，问题就出在这里。由于每个公司的进化程度不一，程序员在不同的公司，获得的感受和受到的待遇，也是非常大的。你甚至会产生图中爸爸的心态。



监工一般都不是一线，离生产是比较远的，对于client的想法，只能领会一点点。这一点点传导到server那里的，正确性更难以保证。而且，监工由于领会领导意图和汇总excel，耗费了大量的精力，花在真正有意义的事上的时间，就少的可怜。那怎么办呢？只好拿client的成果当作自己的成果。

所以，闷头写代码的程序员，可能会发现自己做了很多工作，经过proxy的一加工一转达，到了领导那里就是个屁。

proxy做的没错，996的client深夜也需要调代码，proxy只需要在一旁刷TikTok就可以

了。工作强度不一样，工时自然就没有衡量对比的价值。

我们可以再看proxy的3种传导场景。

场景1：

```
1 | server: 要加强员工文化建设。  
2 | proxy: 周六、周末去团建, AA  
3 | client: WCNM
```

场景2：

```
1 | server: 公司要勤俭节约。  
2 | proxy: 从明天开始不提供厕纸和开水  
3 | client: WCNM
```

场景3：

```
1 | server: 王xx同学拼搏奋斗, 是公司的楷模。  
2 | proxy: 开启狼性文化, 明起996  
3 | client: WCNM
```

可以看到，这一层的proxy素质尤其重要。如果你不巧碰见了一个水货proxy，你的悲惨程度可想而知。但如果你遇到一个nice的proxy，那就珍惜他，因为所有的压力，都需要由他传导。

那一般的nginx工程师是怎么做的呢？

为了让驴拉磨，人们想出四招：蒙眼睛、捂嘴巴、堵耳朵、抽鞭子。

同样，要想proxy顺利推行，proxy就可以不让你看到某些事情；你想要发表意见的时候，使用暴力手段堵住你的嘴；当你想要聆听领导真实意图，堵住你的耳朵；当你有所懈怠的时候，使用鞭子督促你完成工作。

很多公司由于人员众多，结构复杂，就存在着多层proxy的拓扑。这种公司非常的精彩，一个proxy，可以作为另一个proxy的client，很多时候，竟然会发生server的数量比client的数量多的情况。

我很可怜这些proxy们，他们活的太累了。但由于proxy的工作特性，只需要进行完整的转发即可完成工作，不需要思考--自然有一头乌黑油亮的秀发。加上nginx工程师可以很容易打破35岁魔咒，所以这个职业依然让人趋之若鹜。

你也想做一枚nginx工程师么？

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

17. 9月火气大，能认真写代码么？

不羡鸳鸯不羡仙，一行代码调半天。原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

我在风中藏把刀，斩尽世间秋色。这句注定要流传千古的名诗，是 xjjdog 于2020年9月6日书写，体现了作者想要毁掉秋色，又对其毫无办法的矛盾心理。

秋天很让人讨厌，除了是万物萧条的开始，它还容易让人上火，容易让鼻炎复发。万物都会受到季节的影响，失去它的本性。当然，程序员也不例外。

今天是周末，本应该是不上班的。但中国在 1995 年之前，是没有周末的，周末是无数先烈们用鲜血换来的权益。

周末 并不代表 星期天，而是 周六。星期天其实叫做“礼拜日”，由于 上帝 在第七天休息，所以是西方人在这天去教堂的日子。所以，程序员一般选择周六加班，周日并不加班。

但很明显，这是舶来品！我好像找到加班的理由了！

总之，今天我们来加班了！空调开的很足，是个分享知识的好日子。

今天有同学分享的话题是：为什么日志规范里，要推荐使用 占位符 来书写日志。

```
1 | log.debug("Processing trade with id: {} and symbol : {}", id, symbol);
```

小A说：很明显了，这是因为 {} 这对符号，非常的优雅。通过占位符方式，参与占位的那些参数，就可以被复用。这在设计上是非常优雅的。

小B说：我有点认同小A的观点。通过 {} 这种替换方式，每次都需要耗费CPU去进行查找替换。效率上肯定不是最优的，作者既然提供了这种方式，那肯定是出于代码的简洁性这种trade off进行设计的。

小C说：我觉得作者纯粹是脱了裤子放屁，直接使用 + 进行拼接也是没什么问题的。我用 + 拼接用了好多年了，也没什么问题。这就是作者顺手提供的API而已，并没有什么意义。

小A问小C：你知道 + 在Java中怎么处理的么？。

小C说：不要拿这种老掉牙的问题来问我。+ 最终在字节码上，都是使用 StringBuilder 进行拼接的。我还知道StringBuilder和StringBuffer，其实在效率上并没有什么区别。

小A不服：怎么就没有区别了？一个是同步方法，一个是正常的方法，肯定有区别的。

小C白了小A一眼，没有说话，似乎是不屑于解答这个问题。

别扯远了啊。推荐使用占位符方式，其实还是有非常大的道理的。看一下下面这段代码你就明白了，分享的同学及时的掩盖了这个尴尬。

```
1 | if (log.isDebugEnabled()) {  
2 |     log.debug("Processing trade with id: " + id + " and symbol:  
3 |     + symbol);  
4 }
```

大家热火朝天的讨论起来。由于讨论又臭又长，xjjdog 在此总结一下。

线上的日志级别，一般是使用 INFO 或者 WARN 的，DEBUG 日志一般都不会打印出来，因为它的内容会非常的多非常的啰嗦。上面这段代码，在打印 DEBUG 日志之前，先进行了一步是否能打印 DEBUG 日志的判断。

这个判断，log.debug 方法已经在方法内做了一次了，我们何必要多此一举呢？

注意 debug 中的内容，我们是使用 + 进行判断的，() 里参数的优先级，是高于 debug 方法本身的。如果我们不加上外层的判断，那么 debug 里面的参数，无论是否要打印，都要进行一次拼接。

假如我们现在是 ERROR 级别，里面的这些参数依然要进行拼接。由于 String 对象是不可变的，这些拼接的字符串，就要占用非常多的堆空间，造成浪费。

这样解释 {} 占位符就比较好理解了。因为它生成的字符串数量是固定的，只有在真正需要打印的时候，字符串才会被拼接。

按这个逻辑，如果是 log.error 方法，你怎么玩都可以。

小A小B小C小F，全部都点头称是。

你们这是在这里讨论一些什么狗屁东西？工作进度都正常么？。正在大家讨论的热火

朝天的时候，一个不协调的声音传来。

大家回头一看，原来是项目经理来了。他抱着一台老掉牙的灰色笔记本，屏幕上打开着他最喜欢的 Excel，花花绿绿的统计表格甚是好看。

小A说：我们的进度都正常啊。主要是今天来加班，等的就是需求确认下来。

项目经理说：需求不早就给你了么？王总对这个需求非常的重视，你们得紧张起来。这个星期我都没找到关于这个需求的任务项。

小B说：这个需求根本就不明确，需要先把基本的功能点给订下来。我们加班也不能白加，在这讨论一下技术问题，增加一下技术氛围，有什么问题？

项目经理说：技术氛围有什么用？加班就是用来做需求的，进度不正常写的代码再牛X有什么用？这个功能王总要求下个周就要上，你们还有闲情逸致在这里搞讨论。讨论点需求不行么？

我在一旁静悄悄的看着，项目经理明显是受到秋天的影响了，目前有点火大。这明显违背了项目经理需要情绪稳定这个基本素质，他炸毛了。

反观程序员们，脸红脖子粗，争辩是他们的常态，骨子里的那点骄傲溢于言表。

x你m的，老子忍你很久了，小C跳将起来，一巴掌拍在项目经理的电脑上。花花绿绿的电子表格上面瞬间出现了一道道电子干扰波。



老子也忍你很久了！！项目经理也怒了，指着小C的鼻子吼叫：就TM数你干活慢，整天整些没用的。

一时间口水乱飞，恶语飞贯，振的天花板上灰尘都掉了下来。

你可能会奇怪，他们怎么不动手呢？因为这有前车之鉴。前不久在公司，两个干仗的

都被开除了，而且不给赔偿，所以冒不得这个风险。

过了一会儿，大家都吵累了，怒气冲冲地盯着对方。项目经理并没有因为是一个人就败下阵来，因为其他的程序员都是怂货，只有小C在和他对骂。

这个需求，我和客户聊了一下，觉得不太应该做。今天大家可以早点下班了。正在大家发愣尴尬的时候，王总的声音竟然传了过来。他刚和客户通完电话，没有看到这激烈的场面。

我偷偷跑到王总身边，伸手遮住嘴，悄悄和王总耳语了几句。我得把发生的问题告诉他。



王总的脸沉了下来：一个项目组，就是一个大家庭！不求相爱，也不能相杀。你们这种行为破坏了团结，非常的不可取。项目组的全体成员，这个月绩效全部为C！

一时间，大家幸灾乐祸的看着我。我尴尬的笑了笑，他们应该不是恨我，这根本就是秋天惹的祸。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

18. 人生意义处，无限循环时

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

程序员加班都很严重，但到底你的加班有没有意义？对不同的人来说有不同的答案。小王是我的前同事，是个典型，和我是两个极端。

xjjdog对此故事的评价：如果是真的，我希望他是假的；如果是假的，我希望他是真的。

每当夜深人静的时候，我都会去一趟小公园。公园里有个木头椅子，躺上去很舒服。

背靠木椅，仰望星空，更容易安抚躁动的灵魂。可惜城市的光污染太严重，能看到的星星并没有几颗。不过这也没关系，如墨的漆黑更容易让人抓住躲闪的虚空。

活着是什么意义？我问从草丛里钻出来的虫子，它们静默不语，刚被跳广场舞的大妈惊吓过；我问聒噪的知了，它们对我不理不睬，沉浸在自己单调的声音中。

或许，生活就是沉浸在某件事中不能自拔，并自我催眠，只有自己感觉有意义而已。哪怕是单调的重复。

我突然间就想起小王。对他来说，他最像知了，工作就是生活。正好明天周六，我想和他见个面。

第二天，斑驳的阳光刚撕了了黑暗，我就出发了。

小王是我上家公司的同事，顶着一头营养不良的黄毛，以至于很长时间我我以为他是专门把头发染成这样的。

他是一个喜欢996的社畜。哦不，这是只一个比较理性的评价，他很多时候是熬夜到2~3点的。但你不要把他和那些装模作样的加班狗相比，他呆在公司是真干活的，不像有些人躲在公司蹭空调打游戏。

这种认真仔细的态度，加上他内敛的性格，造成了领导很喜欢把活分给他。但活多了就容易出错，没人会盯着他干了多少活，只会盯着他出了多少错，所以领导并不是很喜他。

这时候小王就摸着他的一头黄毛卷发，很沮丧的向我诉苦。由于我是知心大姐姐

xjjdog，所以能够倾听、并对他的痛苦感同身受。一来两去，我们两个成了好朋友。

这个组合其实很不协调的，我通常不加班，因为晚上我还得躺木椅想事情，加班久了我的脑子容易短路，容易抑郁。小王和我正好相反，满脑子的工作，连喝水的功夫都没有，精神亢奋。他经常成为被效仿的楷模，我对于他就相当于荧光于月光。

对于一个进入了状态的人来说，任何劝阻都是无济于事的。作为他的好朋友，我肯定是为他着想的。我对他说：你每天晚上工作到那么晚，为什么不在工作群里发张加班图片什么的呢？

小王说：这手段太老套太恶心了，我做不来。领导又不是傻子。况且，我的生活就是工作。

我鄙夷的笑了笑，老板都看价值导向，你的价值并没有我这不加班的人价值大。价值和意义是同义词，并没有办法去衡量，我想，只是在我心中这么想而已。小王同样可以觉得他的价值比我大，当他作为劳模的时候。

为什么我总是对小王印象这么深刻？并不是我爱上了他，而是他总能让我想起生命的意义。是活在别人的目光里？还是活在自己的特立独行？是忙碌到无暇思考？还是清闲到失去理智？并没有答案。

和小王产生冲突，是在公司的一次会议后。领导强调，有些人下班走的很早。小王立马跳了出来，提议公司追赶潮流，推行996。他还举了个例子，他自己是10107，996并没有失去什么。

我脑子里瞬间有了几个想法：

- 公司要变相裁员了么？
- 公司要有大单了么？

如果是后者，还真可以搏一搏，毕竟拼上个三年两载换取个一世清闲还是可以接受的。要是前者...哼哼...那也没办法，毕竟我不是劳动局的。

结果都不是，是小王的一厢情愿。他觉得自己的这套理论，可以帮助大家找到奋斗的目标，找到人生的意义。这么政治正确的提议，当然正中领导下怀。

我对小王的看法产生了质的改变，我认为他很坏。

人生是一杆天平，生活在左，工作在右，不能厚此薄彼。小王是个恶心的秤砣，使我的人生被贱卖，失去了活的意义。我能毫不犹豫的从上家公司辞职，小王功不可没。

我活着的意义，就是能够有自己思考的时间，能够没意义的躺在木椅上，感受天地的静谧。如今他也在做着我做的事情，但我想他永远都不会明白这个道理。

其实，我很可怜你，我对小王说。他没有回话。

到现在，你还觉得自己的那一套没错么？他依然保持沉默。

你希望碰见的是马云、是任正非，养狼的时候先喂饱狼。但可惜你碰到的是变种的狼，不过是狈而已。

说到这里，我把鲜花放在他的坟前。希望这一捧灿烂，能够滋润他骤停的心脏。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

19. 气哭老板的顶级密钥存放方案，又做了一件蠢事

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。任何不保留此声明的转载都是抄袭。

老板有一份非常重要的数据，大小不到 1kb，但值得我全力去守护。我问老板：您需要把数据做到哪个安全级别呢？

老板不耐烦的挥了挥手：我像是差钱的人么？按最高的安全等级去做！这事交给你我只要结果！不要问我级别这么细节的问题！

典型的结果导向，我喜欢！看了一眼老板的车，一个车轱辘就顶我一年的工资，我对老板的财力还是非常有信心的。

但我没这个能力做这么牛X的事，只好咨询 xjjdog 做安全的朋友们。

他们告诉我，就按照CA证书私钥存放的等级去做，老板保准满意。

are you kidding?对于这玩意的复杂程度，我还是略知一二的。我想象着老板想要用到这份数据时候的场景，竟然笑了。

密钥有公母，CA是何方神圣？

关于证书，我们每天都在使用。在浏览器上，点击左边锁的小按钮，就能看到使用了 https 的证书的详细信息。

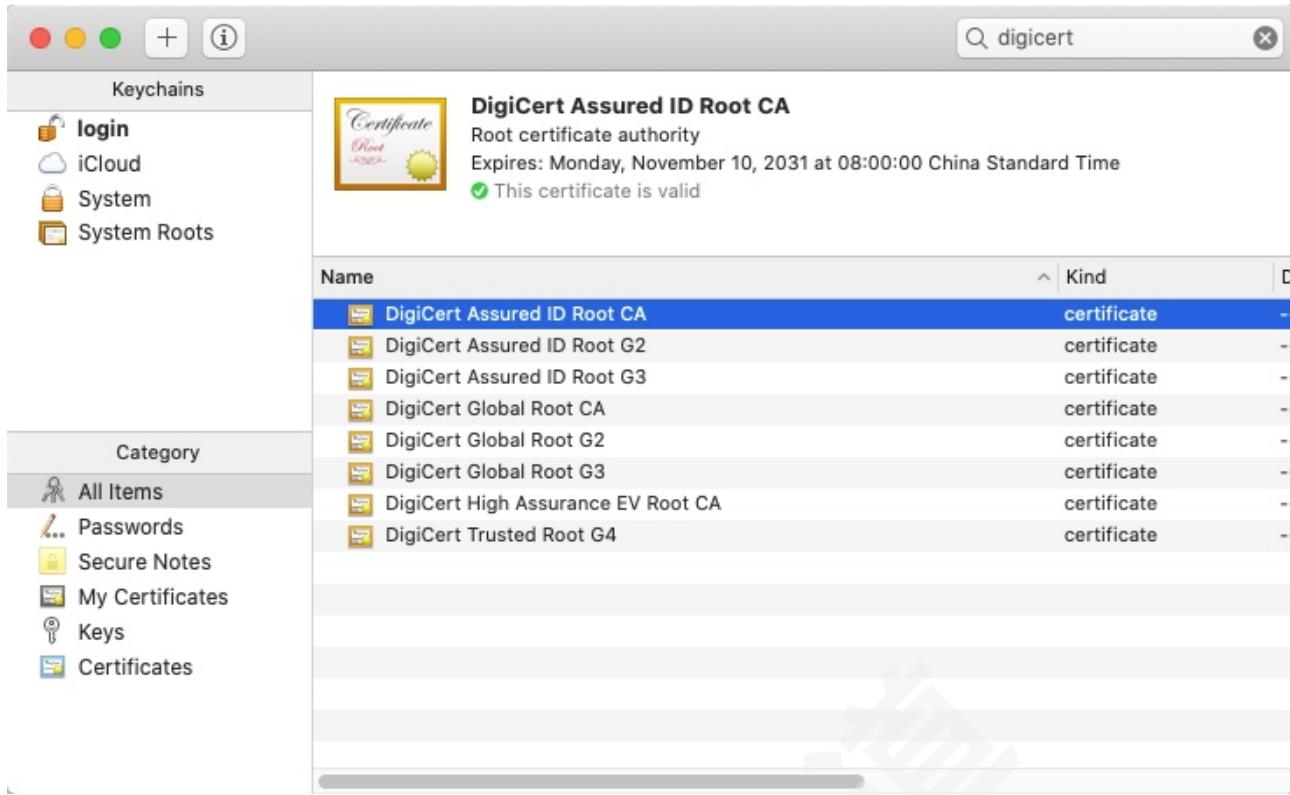


从图中，我们能够看到 Root CA，也就是根签发机构，是 DigiCert。

我们先来看看DigiCert的介绍：DigiCert 是美国老牌的 CA 认证中心，提供 SSL 证书 和 SSL管理工具十多年。

短短一句话，全是关键字，让 xjjdog 无从下手。

像 DigiCert 这么知名的根证书，默认就装在了操作系统上。下图是 Mac OS 上证书管理器中的内容。



不过，我们能够看到的证书，全部是公钥。有公就有母，那母钥是什么呢？

这就需要介绍 RSA 算法了，和 GoF 设计模式、AWK 脚本语言一样，它也是按照发明者的名字拼凑成的，并没有什么实在的意义。

RSA 是一种非对称加密算法，它包含两个密钥：公钥和私钥。这里是公私，而不是公母，可以用任意一个加密，然后用另一个解密。

如果我们想要一个 `Https` 所使用的证书，就需要一家靠谱的机构来给我们签发一张证书和签名。

什么叫 靠谱？就好比你要领导的签名，要有权威性。官越大字越管用。这个靠谱的机构就是 CA。

什么叫 签发？就好比领导挥舞着毛笔写下他特有的轨迹。数字签名就是用私钥（Private Key）来签名一段信息。

使用私钥（Private Key）签名的信息，能够被公钥（Public Key）解密，所以对于私钥的保护成了重中之重。

要多安全有多安全

无论你的过程如何复杂，到最后还是要保存那一串数字！太不安全了。那私钥应该放

在什么地方呢？放在保险柜里么？放在某某云上么（它们都没有资格）？

放在保险柜里同样有一个问题：那就是保险柜的密码。无论你把保密措施做的多强，到最后还是要收口，保存一个特定的值。

密码的作用不仅仅是加密使用，它还需要方便解密才行。

这就麻烦的多，简直和核弹的启动密码保护有的一拼。古巴导弹危机时，美国的核弹密码是多少，八个零你敢信？“核按钮”手提箱共有3只，分别由总统、国防部长和参谋总长掌管----这是我们能够看到的资料，实际上肯定会特别复杂。

人是最不可信的，在加密的步骤人参与的环节越多，密码越容易泄露。根据木桶原理，整个密码体系的安全性，取决于最薄弱的一环。加密环节越多，越容易发起并行攻击。

既然人不可信，这活就可以交给机器去做。

所有的敏感数据，包括密钥的生成，以及私钥的存放，都被锁定在一台特制的机器里。这样就可以保证，除了冰冷冷的硬件，谁也不知道私钥到底是个啥。



下面的链接，是美联邦为这个硬件制定的一份技术标准，长达69页！

1 | <https://doi.org/10.6028/NIST.FIPS.140-2>

其实，我对此还是有一点担忧的。我一点都不怀疑这个东西的安全性，但我非常害怕它是否有容灾的能力。比如突发地震、火灾、或者硬件损坏...这份独一无二的东西被损毁了该如何是好。

CA的私钥没了，会是互联网的灾难。

另外还有一个问题，那就是这台机器被偷走了，怎么办？

这个时候，我们的保险柜终于可以上场了，权力分散的模式也可以使用了。

看过碟中谍的都应该对里面的保险柜印象深刻。它需要突破重重安保，刷各种证件，才能进入。CA私钥的存放与之类似，这台机器就被放在保险柜里。

想要使用的时候，需要多名工作人员同时认证才能打开。想要启动的时候，还需要使

用动态密码。你的每一个动作，每一次操作，都需要录像、审计，相互监督。即使是一只苍蝇不小心跑了进去，也要记录备案，应为这只 `fly` 可能是侦察器。真是繁琐到了极点。

假如我要给 `xjjdog.com` 申请一个证书，就要劳烦这么多牛x专业的人员为我服务，那可真是难以想象。这么多网站申请证书，按照私钥的这种使用方式，我的请求要被排到100年之后...那时候量子计算机都成熟了，要证书何用。

CA遇到了瓶颈，就要发展下线，这个和传销是没有什么区别的。因为这就是个敛财的工具而已，想当年12306使用自签的证书，我还不是照样用的很好（不过现在12306也被DigiCert给签了-.-~）。

CA的下线，按照能量进行分级，从证书体系中，都可以分一杯羹。比如我们最上面那一张图，中间还有一个叫做 RapidSSL 的机构，它们就是中间层的CA。这是使用负载均衡的思想，来完成高并发的证书请求的思路。根CA只需要受理这些中间CA的请求，几年也用不了几次，那么繁琐一点，是可以接受的。

老板的需求

老板要按照最高级别的做法去做，我也只好借鉴CA，把老板的重要信息封装在机械盒子里。算了一下初始成本，起码要1个亿才能完成，还要养一堆安保人员，以及持续的审计工作。考虑到 `gayhub` 将部分代码搞到了北极，我也想追追风，把设备安排在北极。

我把 70页 技术标准和实施方案交到了老板手里，老板给气笑了。

你这是拿我开涮么！你把数据拷贝到U盘里，我自己放到银行的保险箱就可以了！

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

20. 人工智能，是真正的人工智能！！！

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

小王的产品终于研究成功了。他扯了扯领带，尽量让自己放松一些，他将要验证一个伟大的时刻。

面前是一个1/3人高的箱子，箱子侧面有个口，可以轻松的把脚伸进去。

小王穿了一双沾满尘土的皮鞋，把脚放进去后，只见箱顶的灯亮了几下，接着里面一阵轻响。

不一会儿，小王抽出脚，脚上的皮鞋一尘不染，鞋油擦的锃亮发光，直播间里一片惊呼。

小王满意的点点头：这箱子就是一个黑盒。对一个黑盒测试来说，最主要的是测试思想和想问题的周到性，技术是次要的。黑盒里具体发生了什么，测试人员其实也是不关心的。

小王能够突然想到 IT 行业的一些事情，主要原因是他是程序员，这个神奇的箱子，就是他的作品。箱子里装了一个 arduino 控制器，还加了一块树莓派（Raspberry Pi）作为中控，里面集成了使用深度算法训练的人脚识别模型。再加上箱子里面无数的传感器和不断闪烁的灯光，这个箱子里充满了高科技的味道。

无论丑的脚美的脚，不管什么size的皮鞋，保准出来闪闪发亮，让人满意而归。

鞋油的味道很刺鼻，这种重复性的劳动，就应该交给机器去做。

在diy作品大赛上，小王靠他的这些能力，获得了一些成就和奖励。他开源了部分的设计图纸和源代码，但鲜有人能根据他的文章搭建起这样的箱子。

“他一定是隐藏了核心的逻辑”，gayhub上的同行们都这么说。

小王暗自嗤鼻：这种箱子，20年前就有了。

于是思绪纷飞，时间回到了20多年前。那时候，小王还是个孩子。

小王家里很穷，他长得非常的瘦小。如果你看过魔戒三部曲，一定会对 咕噜 有所印象，他就瘦成那样。

当然那时候并没有计算机，甚至电视才开始普及，小王非常喜欢看里面的动画片，可是他的家里很穷买不起。

因为小王的爸爸是擦皮鞋的，他受过伤，只有一只胳膊是能够活动的。干不了重的活，他也只能擦皮鞋。

每天爸爸都会提着沉重的鞋箱，到人流最多的街口，支起一把遮阳伞，开始擦皮鞋。每天晚上，他提着沉甸甸的鞋箱，数着包里赚的钱回家。

小王心想，这个鞋箱，要是能够自动擦鞋该多好。突然他灵机一动，有了主意。

花了一个晚上说服了爸爸，第二天，小王和爸爸打造了一个稍大点的鞋箱。爸爸上班的时候，小王就跟着一起去，去了就钻进鞋箱里。有客人来了，他就伸出沾满了鞋油的小手，帮客人擦干净。

这样过了很长时间，竟然相安无事。

这是一件神奇的事情，大家都把这个鞋箱叫做“神奇魔法箱”，擦鞋的人也多了起来，都想要尝尝鲜。

虽然累，但晚上看到爸爸数钱的时候幸福的模样，小王就觉得很值。

当时的箱子，大小就和现在的差不多，同样是个黑盒。黑盒的好处，就是不需要知晓里面的实现。你虽然觉得神奇，但别人不告诉你你是怎么实现的，你就永远不知道。

谁能想到，20多年后，科技发展到如此模样。

想到这里，小王又无奈的摇头笑了笑。

昨天晚上熬了大半夜，给公司的产品造数据。虽然虚伪，但是老板和股东很需要。

都是自找的，小王叹了口气。

公司的产品推广记录很是寒酸，投资人看一眼就没兴趣带着玩的那种。小王突然间就想到了黑盒理论，突然就灵光一闪。他给老板建议：数据少？我们就造一批吧。

这个建议显然是雪中送炭的。客户和投资人，系统对他们来说就是个黑盒。无论平台怎么造作，他们其实都不知道。在这个虚假的年代，没必要坚持实事求是，只要吹的有理有据，成功就会找上门。

小王看了下表，直播时间差不多了，围观的人群都已经散去。

他隐去了脸上的笑容，突然变得严肃起来。提着笨重的箱子，他走到一个隐蔽的角落。擦了擦脸上的汗水，快速打开了箱子盖。

“出来吧，女儿”。小王的声音有些沙哑。箱子上的灯一闪一闪，在四只眼睛里幻化成星星。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

21. 我做人工智能，我的量化策略爆仓了

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

10月16日，三大股指创新低，科技股重挫。

今天天气很好，没有霾。

夕阳如血的残红，照在脸上火辣辣的。

很痛。

环顾这拥挤的十几平，内心五味杂陈。

角落的桌上，堆着我的书籍。

《机器学习》、《Tensorflow》、《缠论》、《量化交易之路》...

数十本烧脑的书籍代替我的伴侣，曾陪我度过墨色的夜。

桌子一角的台灯还亮着，似与夕阳争辉。

我只想安静的坐着，直到黑夜吞噬我的躯体。

我是个程序员，女的，金牛座。

未婚。年龄有点大。

很宅，消费少。

努力加上运气，过去十来年赚了点钱。

收入不算少，但没有安全感。

IT女认真仔细，我也是。

所以我做AI。（注：人工智能，别想歪了）

这是我的资本。

金牛座吝啬爱财，我也是。

所以我炒股。

这是我的噩梦。

秋天的风很凉。

窗内的我，瑟瑟发抖。

纸上财富，终不过黄粱一梦。

水中之月，终不过一场虚幻。

我爆仓了。

我加杠杆了。
源自我的自我膨胀。
源自我的自信。
一想到这。
顿觉生活索然无味。
我已经不再年轻，可我输了。
就像被蛮横的男人按在墙角，带着无处宣泄的耻辱。

女人投机，终究是不行啊。
我输光了所有，包括我的青春和爱情。
包括我最爱的财富。

论技术，我已经没法再进步了，
所以赚不了更多的钱。
论忽悠，我才疏学浅，
所以无法将技术变现。

我想写一些人工智能的文章分享。
分享我是如何从西瓜书、CS231n学起，
成长为一个专业的AI调参师。
然而当我真正落笔，才发现不过是换一种表达方式来重复别人的知识。
别人认为高深的，我却谓之浅薄。
有什么技术含量，一切不过是手熟而已。
矛盾么？
所以我什么都没写出来。

我深知芯片技术发展到1nm就进入了量子世界，何况现在5nm都不到。
我也了解现在的深度学习依然是在比拼计算资源，谈不上智能。
一切都不过是吹！牛！逼！
可我为什么就信了人工智能的神话，最终栽在人工智能概念上。
我想是“只缘身在此山中”吧。

然后我开始接触量化交易。
我的工作是做AI，自然有优势。
我训练的系统跑赢最牛的投资机构。
我既然这么牛，
为什么不干票大的？

这个时候，招行给我来了电话。
要送我现金，但必须按月还。

欺负我不懂金融么，这不是贷款是什么。

他告诉我是信用额度提升器。

我尝鲜了一下，花了几个月功夫提升到了很高的额度。

正好在16年派上了用场。

我的资金在16年出现了爆炸式的增长。

没错，就是爆炸式。

我小心的控制仓位，从不满仓。

我贪婪的扩大资金，一直满融。

但17年，是我的噩梦。

原因有很多，但我不想总结。

我只买芯片和人工智能，我觉得自己很懂。

最后证明我错了。

我懂得的都错了那我分析不懂的分享给不懂的人岂不是一场犯罪？

我很善良对不对？

我将永远的退出。

那总结还有什么意义？

接下来我必须还清上百万的贷款。

两手空空的去还。

努力工作，为人工智能工作。

每分每秒都撩拨我的伤痛。

然后呢，我已经不再年轻。

也不能像今天刷屏的明星一样，整个容嫁富豪。

或许，我会漂泊到一个陌生的地方，苟且活着。

或许，我会找一个看的过眼的老男人，得过且过。

没有了逆袭的机会。

失去了奋斗的勇气。

如今的我，

行尸走肉。

不就是一条普通的咸鱼么？

你能帮小姐姐一把么？

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

近期热门文章

[《必看！java后端，亮剑诛仙》](#)

后端技术索引，中肯火爆

[《Linux上，最常用的一批命令解析（10年精选）》](#)

CSDN发布首日，1k赞。点赞率1/8。

[《这次要是讲不明白Spring Cloud核心组件，那我就白编这故事了》](#)

用故事讲解核心组件，包你满意

[《Linux生产环境上，最常用的一套“Sed”技巧》](#)

最常用系列Sed篇，简单易懂。Vim篇更加易懂。



22. 猪，是怎么看互联网的！

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

故事虚构，雷同巧合。发表这个声明本非我愿，但不能不声明一下，否则容易对号入座，造成不必要的伤害。

01

7月的天，亮的有些早。我是貂蝉我会飞，还没从王者荣耀的亢奋中醒过神来，闹钟就响了。努力的挑开眼皮，突然间想起，今天早晨9点钟，还有一个重要的会议。

这种突发事件，证明我的时间管理确实有问题。还好是夏天，短袖一套裤子一提，就可以出门了。

不舍得花钱买车位，我把车停在了马路牙子上，经常和交警叔叔捉迷藏。今天早晨出门着急，等跑到车子旁边，才发现车子旁边聚拢着很多人。

我伸长了脖子往里一瞧，发现几个大爷在那里玩豆子，我也好奇的观摩了一下。是这么个玩法。

有个坐庄的大爷捧着两个茶杯，然后顺手抓起几个豆子，用慢的和蜗牛一样的速度放在一个杯子下面。他让别人猜押注豆子，押对了双倍返。就他那手速，不可能猜不到。结果有几个大爷压了100块钱，都翻倍了。

人越聚越多，有人出10块，有人出20，很是热闹。旁边的人装作不小心，碰押注者一下脑袋，豆子就调包了。钱输了下一位来。就这么玩了四五轮。

有人小声说，公安来了…，坐庄的赶紧说：这玩意有风险，别被看到，一会再来。人就都跑掉了。

我看了暗暗点头，大爷将互联网精神学的非常透彻。主要是设立赌博一样的游戏机制，不断刺激参与者的多巴胺，然后找准时机溜之大吉，赚个快钱。

02

想到这里，我就知道自己也必须要努力了。小心的把车锁解开，就骑着上路了。

这个时候，我才发现自己没吃早饭。幸好前面不远，有一家甜点店。到了甜点店门前，看着长长的队伍，我惊的掉了下巴。

赶紧拉过旁边的人，问一下此店缘何如此火爆。这位鄙夷的看了我一眼：“你还不知道吧。这家店每周五搞促销，凭会员卡可优惠6元”。

我脑子里短路的一根筋突然就连上了，我记得自己也办了这里的会员卡，好像还冲了不少钱。那就接在队伍的尾巴上排队吧。

好不容易轮到自己，我掏出自己的会员卡，想要占这个便宜。结果发现平常卖5块钱的面包，今天需要10块钱。脑子里飞速的算了一下，自己是不亏的，瞬间感觉这队排的超值。

“您的会员卡是去年办的”，服务员说，“使用新办的会员卡，今天充值100元的话，可以优惠10元，相当于白吃一顿饭”。

有便宜不占，那不是傻子么。最后，在一顿操作之下，我新办了会员卡，充了钱，买了三个面包两杯咖啡。等我提着快餐袋走出来的时候，我突然间想起来，自己虽然是猪但吃不了这么多。

我暗暗点头，这家店，深得互联网的精髓。使用人们爱贪小便宜的心理，先抬价再优惠，并强行安利一堆不需要的东西，最后都进了垃圾桶。

03

来到会议室，我发现时间还早。我不习惯让别人等，这是我的一贯特色。

我发现，小王比我的来的更早，今天我们两个一块来参加这个会议，目的就是拿下这个项目。由于是第一次来谈这种事情，青涩的我们都有一些紧张。

我看到小王穿了一双形状怪异的皮鞋。一双男士皮鞋，竟然在鞋尖有个毛绒绒的小球，甚是骚气。

看到我的目光，小王得意的给我安利起来。

“你说这双鞋子啊，我从*APP买的，这双鞋子叫做 王子鞋，只有最潮最浪的成功人士才配拥有。穿它出去，相亲绝对成功。”

我赶紧连连点头，低声附和。我心想，这个品牌真是深得互联网精髓，利用少年们爱装逼的需求，体现它们的优越感和虚荣心。把廉价的物品价格炒上天。

我还想多思考一下，但一瞅手表，离会议时间越来越近了。和小王放肆的闲聊了这一小会，心情就放松了下来。

04

对标公司的人也陆续到了，竟然来了两个靓眼的美女。一看到她们低低的胸口，我差点变成了三条腿。

不多时，主办方的人扶着小眼镜进来了。看到美女们，竟不自觉的挑了挑眉毛。我知道他们的心里一定和我一样，瘙痒瘙痒的。

我心想要坏事。本来我们的方案就不是很出彩，再加上和美女们PK，心猿意马，胜率就持续被拉低了。

果不其然，对方准备的很充分，功能和性能碾压我们的方案。本来不怕生的我，说起话来也开始结结巴巴，好不容易熬到结束，额头上都是汗水。

我不禁佩服。

投标公司深得互联网的精髓。合理的利用美色，打擦边球。将所有当事人的头脑不断加热。加上本身就占优势，会把对方碾压的死死的。

得出这个结论，我和小王灰溜溜的走了。

05

这一天真是精彩，互联网技术真是无孔不入。虽然隐隐有一股自豪感，但我不得不面对今天将要竞标失败的问题。

走在大街上，我颓废的拉了拉领带，它已经湿透了。老板对这个项目，是势在必得的，我们的团队熬夜好几天搞了这些资料，没想到今天被我和小王搞砸了。

想到团队的辛苦败在我手里，我的心就开始隐隐作痛。除了心有不甘，还有一股隐隐的羞愧感。

老板会怎么惩罚我呢？今年的晋升或许没指望了。我叹了口气，新款的Xbox是买不到了。

06

第二天，我卡点来到公司，依然是垂头丧气。小王兴高采烈的过来告诉我，中标了！

纳尼？我惊讶的瞪大了眼睛。有点冰火两重天的感觉。美女攻势都没用么？

“不知道吧”，小王说，“老板是有关系的，我们只是去走个形式。无论对方有什么花招，这事本来就是内定的”。

我恍然大悟，互联网的精髓从来没有在此刻让我拜服过。想当年，有多少创业公司有着相同的起点，只有那些有政府和大佬扶持的成长了起来。盘子越滚越大，直到所有局内人无法脱身。

这个消息加固了我在这家公司待下去的决心。因为风口碰上互联网思维，死猪也能飞上天。

何况我是头活的。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

23. Serverless，会将工程师带入“不归路”！

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

别被标题吓尿了。

技术的发展，从来不以个人的意志为主转移，程序员的某些分工也必将随着技术的演进而消失。

在开始之前，我们首先来看一下Serverless是个什么概念。

Serverless直译为“无服务器”，并不是说代码运行就不需要在服务器上跑了。它的意思是，未来的开发，无需关注 服务器 这种比较底层的设施，代码将会直接跑起来。这些资源将变得不可见。

危险了！基础运维人员们。

什么是Serverless

想象一个Java应用的开发过程。你需要经过开发、测试、部署，才能正常上线。如果是一个高并发的应用，就需要预估一个峰值，比如，需要100台服务器，才能支撑用户的请求。有产品经理脑袋一热，加了秒杀的功能，那么你的峰值，就需要按照秒杀的峰值进行设计，即使这个峰值过程持续了不到10秒，一哆嗦之后所有的资源都会被闲置。也就是说，服务器放在那里即使不用，你也要为此付费。

老板捂着心口说痛，猝死了。

事情会更加的复杂，服务器的数量一多，你需要考虑怎么快速部署；每台机器的配置，都需要进行精细的计算，包括 JVM 要分配多大的内存，网络的安全策略要如何配置，高可用，异地多活...，诸多细节需要大量的准备工作和基础设施。

这种现状，我们现在的很多程序员，乐在其中。

就拿基础组件和一些中间件来说，什么分库分表、缓存、数据同步、监控、虚拟化、CI/CD....。内容都相差无几，但每一个上规模的公司，基本上都会自己搞一套。这养活了大量的从业人员，但从整个社会的投入和产出来说，这不合理。

为什么面试要造火箭，因为现阶段这玩意有时候还真用得着。

事情在变化

一个公司在发展到一定阶段的时候，会有大量的内部系统。比如运维系统，财务系统，员工管理系统等。除了少部分的开发人员持续迭代这些系统，后续的使用人员其实充当了变相的“运维”角色。

你需要申请一台机器。ok，填个工单吧。工单审批完毕之后，“运维”人员，在后台点巴点巴，一台虚拟机就创建好了。

你需要拿一下系统的用户列表。ok，填个工单吧。审批完毕之后，“运维”人员为你开放一个令牌，拿着令牌到我们的微服务里去拉信息去吧。

这些抽象的服务器，用户接口，不需要你去准备硬件、准备网络，只需要填个工单，唾手可得，已经有了一点Serverless的样子了。不过是企业内部的，规模也有限。

在这里，我们就需要了解一下几个缩写的名词。

IaaS Infrastructure-as-a-Service(基础设施即服务)。此部分属于基础设施，比如服务器的购买、搭建，与虚拟化、容器等技术息息相关。

PaaS Platform-as-a-Service(平台即服务)。比如操作系统、虚拟机，以及在其之上，提供的与业务弱相关的基础组件。通常，一些耳熟能详的名词，如持续集成、中间件、公共组件、微服务等，属于此列。

SaaS Software-as-a-Service(软件即服务)。生活中，几乎我们每一天都在接触SaaS云服务，但一般是指集中式部署的服务提供者。在这种模式下，商业模式变成了租赁的形态，销售变成了运营，项目变成了产品。

这4个名词的第一个字母，既预示着某项从业环境的改变。比如，IaaS 的完备，使得专攻 基础设施服务 的工程师，逐渐没了发展空间；PaaS 的完备，使得广大 平台开发工

程师 就业路子越来越窄。除了少部分能够享受平台的红利，大部分只能向更高层的抽象去过渡。

这三者大家耳熟能详，但还有两个缩写，这才是Serverless的关键。

BaaS Backend as a Service（后端即服务）。是指公司为移动应用开发者提供整合云后端的边界服务。

这个词非常可怕。按照我们上面的逻辑，等它普及的时候，大部分后端开发工程师将消亡。稳定了这么多年的后端技术栈，终于要自己搞死自己了。

FaaS Functions as a Service（功能即服务）。可以广义的理解为功能服务化，也可以解释为函数服务化。使用FaaS只需要关注业务代码逻辑，无需关注服务器资源，所以FaaS也跟开发者无需关注服务器Serverless密切相关。可以说FaaS提供了一个更加细分和抽象的服务化能力。

这就可以想象到，如果各项设施完备之后，只需要水平一般的前端集成人员，就可以完成企业级的开发。

信息革命的本质就是自我升级然后自我替换，程序员能选择的路子将越来越窄。计算机技术将回归工具的本质，人人得而用之，不再是一部分人的专利。



你可能觉得，这玩意，和现在的云环境有什么区别？区别还是有的，否则不会有这么多大厂趋之若鹜。现阶段，Serverless是云厂商的事，和普通开发者关系不大。但最后，这些新生事物，都会慢慢侵蚀我们传统的领地。

弹性！成本！

我们上面就已经举例了，现阶段的服务端软件开发，需要根据峰值进行设计。买了服务器之后，哪怕是云主机，大部分时间也是空跑。空跑也是要付费的，这也是为什么

企业的IT费用居高不下，它要为很多压根用不着的东西一直花钱。

Serverless是按需收费的，用多少收多少。比如你的服务QPS在10w的时候，给你分配10台机器，降到2W的时候，给你分配2台机器，那就可以足足省下4/5的money。

这种弹性看似神奇，不过也是建立在一些目前的技术上的。比如 kube , docker 等。但这是云厂商的事，对企业来说，那就是服务拥有了巨大的弹性，节省了大量的成本。等老板们尝到这种功能的甜头，还养一堆眼里只有技术的人干什么呢？

时间！敏捷！

Serverless的形态，肯定是大的生态，也就是有非常多的完善的功能积木，开发人员进行组装即可。

云厂商会保证功能函数的正常运行，以及升级迭代，向后兼容等特性，企业压根不会再投入精力到一些既有的功能上来。

比如，建设一个直播的基础设施，是非常昂贵且漫长的。如果Serverless提供了这样的功能模块，企业就可以直接拿来用。

这种租用的方式，不仅便宜，而且快捷。以往需要开发一年半载的产品创意，到现在只需要几天就可以上线。这就是复用的魔力。

一点展望和思考

一些变化

可以看到，在这种模式下，很多职业都将产生变化。

运维工程师？不再需要了，只需要操作配置后台，就能够获取稳定、安全、便宜的主机。

中间件工程师？需求变小了。企业不会再养这部分又贵又难找的人来做这些基础设施。只需要操作配置后台，使用云厂商暴露的功能函数，就可以漂亮的完成工作。

后台开发工程师？需求变小了。不再需要非常复杂的逻辑，做什么前后端分离，做什么复杂的分布式锁和数据同步等。只需要关注业务逻辑就可以了。

前端工程师？这个时候前端还叫做前端么？应该叫全栈。只要会玩乐高积木，就能根据图纸拼接零件，在功能模块足够丰富的前提下，这些都不是魔幻的。

程序员的API，不再是什么Kafka，Redis等，反而会是云厂商提供的一大批自定义的函数。

这对安于现状的工程师来说，真的是挑战。其实从最近几年的云主机普及就能够看出来，有些职业，真的是慢慢的变成了后台管理员。本质上，这与淘宝小二操作的后台，没有什么不同。

本质都是：了解平台的规则（函数），作出相应的推广（集成）。

面向Serverless编程，实在是无趣的多。

在时代的浪潮中，个体总是容易被抛弃的，我们毫无疑问已经进入了终身学习的年代--苦逼的程序员尤其如此。

工具抽象程度高了，并不意味着工程师的素养要求就低了。由于Serverless并没有什么标准，它仅是一种理念，那么各个厂商的实现方式肯定不相同。

作为企业的老板，肯定会有这样的担忧：我不能把鸡蛋放在一个篮子里，受一个厂商的绑架。比如现在有的老板用了阿里云，还会考虑腾讯云、信仰云等等。

使用了A平台提供的功能函数开发的应用，是很难迁移到B平台的。厂商们都希望垄断，往往也不会这么干，那就会造成工程师的学习成本加大。

另外，Serverless厂商的能力毕竟有限，无法覆盖所有的基础业务功能场景。肯定会有厂商，采用开发者平台的方式，吸纳外部的功能组件。开发这部分工功能组件的开发者，也有较高的要求。

我猜测未来会有插件模式的云功能市场，用来丰富这个生态。

各种开源软件的版权也应该有所变化。就像现在的云主机，拿着开源的软件大赚特赚，白嫖了这么多年，而软件开发者却无法从中受益，这是严重不合理的。

那未来的程序员是什么样子呢？

平台开发工程师。搭建云Serverless平台，保证基础设施的完善。这是大厂才玩的事情。

插件功能工程师。按照平台的规则，开发相应的功能，享受售卖的提成；或者受雇于特定的公司，开发此类功能。这是相对头部的工程师。

业务开发工程师。从海量的功能组件中，寻找积木，搭建产品。这和现在在github上寻找功能，集成到项目中是一样的道理。这也就是占比最大的人员。

Serverless一旦普及，就会把大部分工程师带入不归路。还好，由于各大厂商并不是铁板一块，它们相互竞争和攻击，会让建设过程持续非常长的时间。

当然，如果一家独大，也是不必担心的。有三个原因：

第一，天下大势，分久必合，合久必分。大厂的霸道会造成众叛亲离。造成一定程度上的技能回归。

第二，怀旧会让人如痴如醉，很多东西不会消亡。比如我现在还喜欢超级玛丽。

第三，等能活到那一天再说吧。可能和传说中的人工智能一块到来。

不多说了，我要研究 AWS Lambda 去了。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

24. 传统企业的人才们，先别忙着跳“互联网”了

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

最近有创业的朋友咨询，说手里有钱，但不知道要干些啥。想到的点子，放眼四顾，早已经被实现了。我推荐他去当网红，但他又比较内向，抹不开面子，还是想在轻车熟路的互联网行业捞一把块钱。

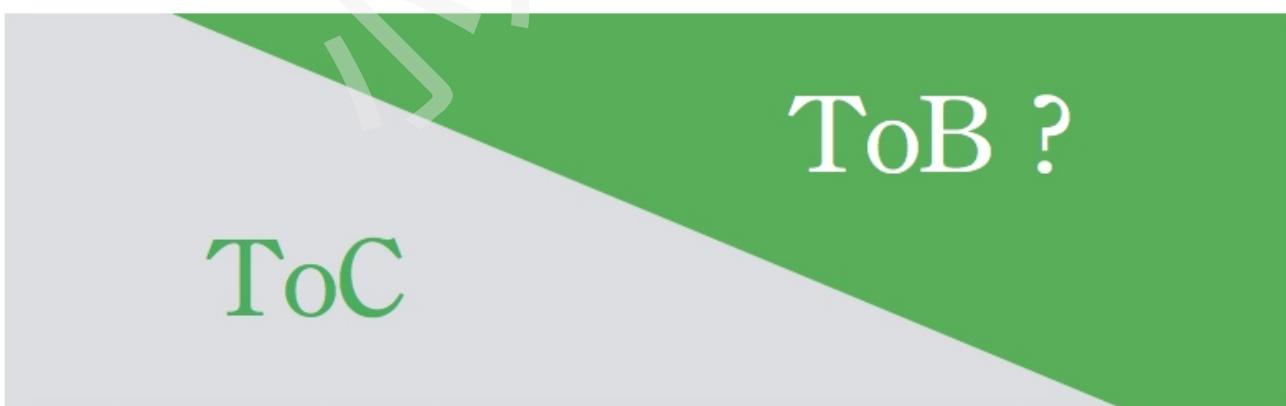
但现在，很难了。

经验并不总是有效的，今日不比往常。一个不小心，就得把过去赚的钱，全部给吐出来。

谈到互联网，从业者首先会想到 2B (toB) 和 2C 两个概念。B和C两个字母隔得很近，但却有着天壤之别。十年河东十年河西，在 2C 逐渐没落的今天，很多多金的大佬，已经把目光瞄准了 企业服务 领域。比如，王兴在2018年就把目光瞄准了B端市场。

只有制造混乱，才能获取利益，在互联网中也是如此。2B行业的乱象，吸引着无数人的眼光。

小姐姐味道



小姐姐味道

2B行业一直不温不火，主要是由于它的业绩太稳定了，利润率有限，不像2C互联网一样大起大落。巨大的波动 吸引着投机者，以及敢于冒险的投资人。他们形成合力，将故事情节一波波推向高潮。

这种情况，在2018年的一哆嗦中，戛然而止。无数亏损的互联网企业，不求继续画饼

了，纷纷忙着上市，希望在资本市场最后收割一场。很难解释这许多企业的统一行为。春江水暖鸭先知，2C什么时候达到高潮，资本最明白。

很多不赚钱的2C公司，错过了2018，后面会活的很痛苦。随后的两年验证了前面的结论，然后进入了寒冬无衣瞎蹦哒的现状。

1、BC的鸿沟

要想深入了解2B和2C的一些特点，必须先了解一下他们的不同之处。两个行业虽同为互联网，但面对的却是两个不同的群体。

下面从四个方面可以下这些差别。

小姐姐味道



小姐姐味道

1.1、服务目标相囧

2C和2B有非常多的区别。首先，是服务目标的不同。一个是 用户，一个是 客户，对待起来自然有所差别。

2C的服务对象，是个人用户，小单大流量，以满足个体的七情六欲为主，所以多会从人性上去考虑产品。做事情先不求回报，以曝光率为主。

很多产品都是凭空想象的，或大众，或小众，只要有目标人群，就有相应的 流量。在这种情况下，产品就是上帝。多数产品在调整之后，虽然会丢失一部分老用户，但会获得更多的用户。

这有点像宗教的集体朝圣，或是追星行为，有些幼稚且无理性。

人类生来就是无聊的，在迷醉的狂欢中追求生的意义。这是2C存在的基础。

小姐姐味道



小姐姐味道

到了2B这里，上帝摇身一变，成为了老板，或者客户。2B是为了解决具体的领域问题，或者复杂的流程而存在的，不会那么天马行空，很多东西决策的流程冗长，有具体的存在意义，才能发展下去。

这是2B存在的基础。

这个领域很难暴富，但一旦成型，利润是稳定的，日积月累下来，并不见得赚的少。2B很不适合心浮气躁的人，因为在这种领域下混个三五年，是没什么用的。

1.2、产品特点各异

产品特点的不同。一个是体验式，一个是租用式。

2C的产品，一般都是集中式的。所有的产品逻辑上统一部署，不同的用户，看到的内容基本上是一致的，差异体现在增值服务上。

2C的产品迭代快，上线后的功能，如果发现用户爸爸不喜欢，可能下周就被撤掉了，开发人员会发现自己做了很多无用功。**大家伙都在为产品的创意买单。**



小姐姐味道

2B这里，平台性质的是少数。更多公司的服务，以项目的方式进行售卖，产品形态上变化多端。加上客户爸爸的自定义需求，同样会让开发人员疲于奔命。

2B产品的需求，比2C的更加不稳定。这个结论，可能对于很多同学来说，难以理解。

但稳定的功能模块，会有非常长的生命周期，大多数功能迭代会趋向于个性化。客户相对于用户，由于是直接金主，话语权更强。**在这种情况下，大家伙都在为项目经理的沟通管理买单。**

1.3、商业模式不同

对2C来说，是最主要的营收通道就是广告。当然也有直播抽成、手续费、订阅这样的变种。目的就是汇总小钱，变成大钱。游戏行业会是付费用户养着免费用户，赚钱还是考头部付费用户。但由于服务的对象，还是个体用户，所以形式上虽然不一，但本质上是一样的。

你可以想象一下，你每天都在免费用着微信，免费玩着王者荣耀，但腾讯每年营收的3000多亿是哪里来的？

所以2C互联网非常注重网络运营，有了流量就有了可能性，总会吸引一定概率的人群进行付费。



小姐姐味道

但2B就可怜的多。传播口径，只在特定范围内存在，售卖主要靠销售、或者靠关系网。主要靠大客户的钱生存。

2B有大B和小B之分，也有混合型的。和2C一样，规模小的服务对象，获得的利益也少；规模大的服务对象，产品话语权会旁落甲方，造成整个团队赚不了多少钱，而且苦闷。

对待不同规模的企业对象，服务配套肯定是不同的。销售拿到的通常都是大单，但钱通常都花的飞快。利润一开始就控制好的，后面会按照步骤一步步走，节外生枝会造成利润不可预料的偏移。因为**2B企业除了服务差价所产生的利润，一无所有**。

花了100块钱，仅收回50块。或者拿到90块，却花了100块，有病不是？

1.4、研发侧重点不一样

对于2C来说，最大的考验，就是量。尤其是一群营销天才高出秒杀这个玩意以后，类似的概念，无时无刻不考验着系统承受高并发的能力。

流量的不确定性，使得研发基础设施的建设，充满了诸多挑战。各种概念，如微服务、中台等，将研发设施逐一进行抽离，并层级的、网状的划分。



小姐姐味道

对2B来说，主要的挑战，就是业务的复杂度，以及可复用组件的建设。前者能够确保项目能正常完工，后者能控制最重要的成本因素。

2、「熵（shāng）」

本文算是对2B领域的一点反思。作为一个常年混迹于2C的互联网人，为什么会有这样的想法？

因为从目前的一些数据观测上来看，投入充满乱象的2B产业，会比2C产生更多的收益。宏观层面，全国信息化建设已达到瓶颈，人口红利不再；政策层面，开始倾向于互联网价值的下沉，以及对实体经济反哺；投资层面，烧钱的模式越来越不值得青睐，陈本和收益成为首要目标。

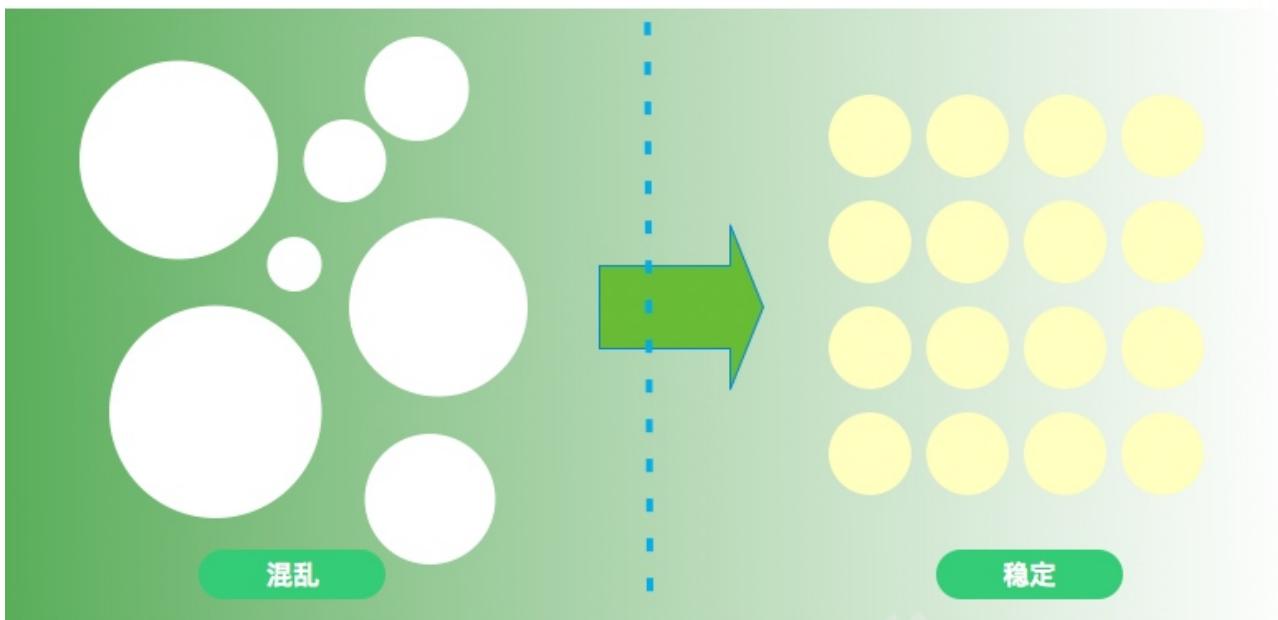
这个过程，我打算使用物理学中的熵进行初步解释。

2.1、混乱变成有序

魔术之所以称之为魔术，因为它违反了自然界的基本定律。在视觉的转换之下，这种明显与经验相悖的东西，总能点燃我们的兴趣，虽然我们知道背后都是些假的把戏。

熵，是热力学中表征物质状态的参量之一。它的物理意义，通俗来说，表现的是热力体系的混乱程度，在控制论、概率论、数论、天体物理、生命科学等领域有重要应用，是十分重要的参量。这也是所有物理体系中，与时间的单向不可逆性相通的唯一的物理概念。

大爆炸理论中，将熵增后的宇宙，描述成毫无生气的一片死寂，任何残存的波动，都会被这种死寂彻底吞没，归于永恒。



小姐姐味道

冰块融化、酒精挥发、蜡烛燃烧、鸡飞蛋打、破镜不能重圆，也是描述的这种单向关系。事物改变之后，就不能再恢复原样，除非有非常强大的外力作用在上面。

脱离了放任即乱，一管就死的宏观调控，熵化是新事物发展的必然趋势。在混乱之中才有机会，在平静中只有死亡。互联网经历了广告业数据化、内容产业数字化、生活服务数字化、行业数字化等阶段，已经慢慢的由混乱变成了死寂，也是事物发展的必然。今天的互联网，急需要新的技术革命的引爆，否则将毫无悬念的变为传统行业。

2.2、热力现状

先看一个简单的互联网现状。

在 2C 领域，目前，中国已经度过了 C2C (Copy to China) 、百花齐放的模式---即中国复制国外的商业模式，进入了存量竞争的阶段。早些年，百度仿造了谷歌、优酷仿造了 YouTube，但外卖和新零售却由进口变成了出口。但 C 端市场的获客难度，加上大厂的技术垄断，几乎断了大部分创业者的梦想。

没有大钱，玩不转了。

2B 领域，大厂凭借资金和成本优势，逐渐分食传统的 关系链 模式。但 2B 的服务并不好做，有些行业壁垒非常深，并无法做成通用的产品，想要通吃非常困难。这个不管叉腿进来的大厂，还是一些传统企业，逐渐轮化为人力外包。

技术领域，随着云端环境的流行，IaaS 、 PaaS 等基础建设已经不再是企业的竞争优势，更多的公司参与到 SaaS 行业的竞争中来。但 SaaS 行业多牛鬼蛇神，客户培养成

本居高不下，多数 SaaS 公司走向没落。

在这互联网逐渐 秩序化 的现状下，能拿得出手的技术仅有： 区块链 、 人工智能 、 5G 、 量子计算机 等。它们有的有具体形态，有的依然在襁褓中。先不要去评价这些技术的价值，它们相当于在平静的湖水中扔进的一块石头，对于大部分错过互联网红利期的朋友来说，是唯一的机会。

混乱中的机会。

3、2B的外在形态

我们在上面，谈到了两种互联网形态，在服务目标、产品特点、商业模式上的不同，但其实在两者之间，也有一条模糊的通道。当服务的目标足够小的时候，就是2C；当服务的目标够大的时候，就是2B。

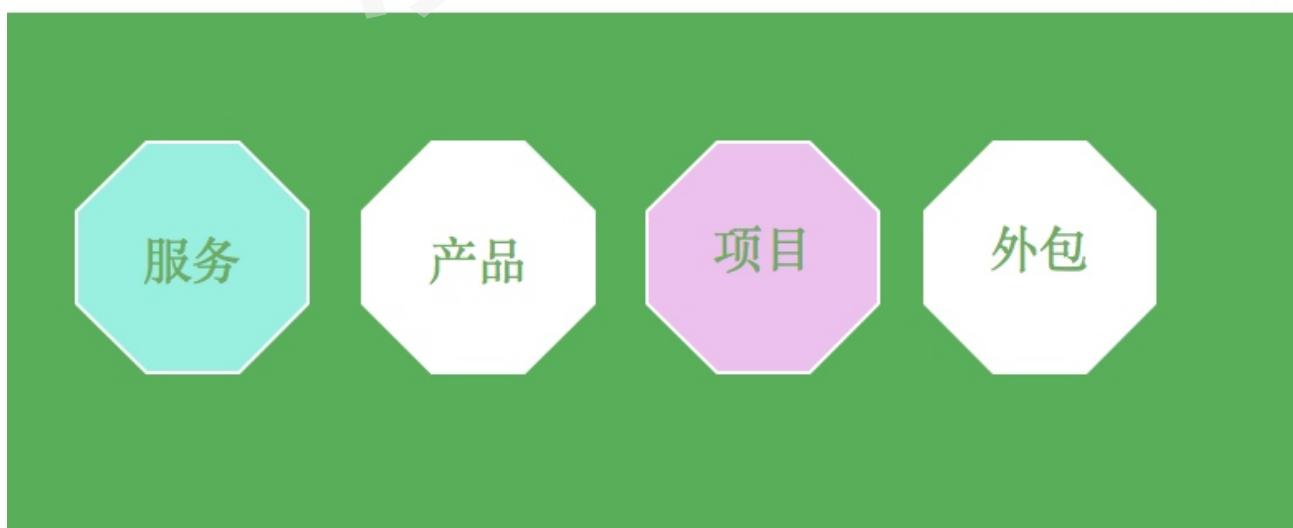
如果你想通吃，不论B、C全部揽到怀里来，往往回消化不良。也就是说，我们一开始，就要找准服务的目标。

如果服务的主要是小型的企业，初创型的企业，那么产品就非常的像SaaS，采用租用的模式获取利润。本质上，还是以量取胜。

如果服务的对象是大企业，非常大的客户，那么产品形态，就是项目。验收完毕，一锤子买卖。

如果服务的对象，领域不一，大小客户都有。那无论公司名叫的再好听，本质上都是外包。

小姐姐味道



小姐姐味道

所以，我们概括性的总结为四个名词：服务、产品、项目、外包。

3.1、服务

SaaS服务是一种理想形态，它采用租用的方式，靠服务订阅而活。由于数据是集中存放的，在大数据的加持下会有更大的想象空间。

SaaS行业比较像2C的互联网，以量取胜，通常不会服务于非常大的客户---那是他们的累赘。

技术属性的服务，比如云服务、直播基础服务；业务属性的服务，比如基于微信的营销平台，都可以做成SaaS服务。

但由于它本质上还是2C互联网的打法，又有利润的诉求，建设起来就有很多阻力。再加上乙方的信任，往往要有关键事件的促进，往往建设时间都比较长。

小姐姐味道



小姐姐味道

由于对客户的培养成本高昂，单位获客成本会更加高昂。除了运营理念的跟进，也会有中间代理商参与到市场的推广中来。但一旦企业采购并推广了服务，短时间内不会再做轻易的迁移，我们可以认为获取了相对稳定的客户。

个人认为，经过短暂的烧钱之后，如果不能实现快速盈利，以及盈利模式的考验，大概率会以失败告终。当然有主动烧钱的金主另当别论。

3.2、产品

另一个比较舒服的模式就是产品。比如SAP，滋润的活了这么多年。但是做一个好的产品是非常困难的，它通常由项目演变而来。

做一些通用的产品，如CRM、供应链、财务等，会陷入到激烈的竞争中来。这些软件的操作体验和价格都相差无几，在产品和业务形态上不会有特别大的创新。

行业应用产品，会有非常大的行业壁垒。通过方法论抽象出来的产品+流程，整体打包后，通常不能直接售卖。多数客户希望获取个性化的自定义功能。

所以，产品的设计要伸缩有度。核心业务要保持绝对的话语权，在一些边缘业务和展现形态上，做到良好的扩展性。就像是，所有的人都会吃饭喝水，但穿的衣服却不尽相同。

小姐姐味道



小姐姐味道

无法做到这一点，产品就会沦为项目。产品属于一直做项目后归纳、抽象的一套业务引擎+数据引擎，属于乙方的叛变。

3.3、项目

服务或者产品演化为项目，我们称作产品的崩坏。是一种与熵增相违背的演进方式。

产品有两种形式的演进。一种是自上而下，先定义产品形态，然后进行市场培养，这需要极高的业内知识沉淀；一种是自下而上，先做项目，甚至外包，然后在此基础上抽象出通用版本。

项目就属于后者。通常是企业没有能力提供通用的产品，或者没有能力在市场上对客户进行培养，而选择的一种折衷方案。

有的企业，会对服务类产品形态心存疑虑，认为数据都存放在别人的平台上，不安全，所以也会采用项目形式。

做项目非常苦逼，碰到阴晴不定的甲方最为痛苦。项目有可能是远程办公，也可能是驻场项目，受到甲方的文化和流程约束。



小姐姐味道

经过销售达成合同，项目即可开始。合同通常不会明确写明系统的功能需求，系统边界不明，造成后期的沟通成本大，功能频繁变动。

做项目和做产品一样，工期会特别长，持续3-5年也是可能的，技术的领先不再是主要的考量点。

3.4、外包

外包是B端市场最主要的形式，占有的市场份额也是最大的。从互联网诞生开始，外包就一直存在，到今天依然坚挺。

企业的数字化，需要用人，但大多数都是临时性的。如果企业不想出这部分长期养人的开支，就会产生外包。

所以，外包几乎无孔不入，任何行业都充斥着它们的身影。这里充斥着数量巨大的伪业务，将开发人员当作工具使用。



小姐姐味道

外包定制模式高（客户上帝），服务半径小（功能有限），对开发人员的要求也是一般，利润也来自于销售和支出之间的差价上。

有些甲方非常过分，会极力的压榨外包的利润，有的甚至会对乙方的管理和人工成本插上一脚。

4、挑战

在2B的企业，面临的挑战不是故事、营销、技术，而变成了产品、管理、成本。在不同层次的B端企业，这三者的重要程度会有不同的侧重点。

但它们有一个统一的外部威胁，就是互联网的集权化。

4.1、集权化

这已经不是老生常谈的问题。拥有流量优势、技术优势、运营优势的大厂，更容易占取利润更大的业务。剩下的又脏又臭的业务，被创业者和外包分而食之。

从最近某大企业频繁的控评就可以看出，垄断对社会、对公民，都是伤害非常大的。

更可憎的是资金优势。通过巨大的资金池，在过去风口受益的，拥有巨大现金流的公司，会通过收购和投资的模式，强行切入到某个行业。

比如阿里，它的触角已经深到了以前互联网不屑入围的政企项目。通过资源拿到项目，然后通过外包完成项目，赚取差价。

今天广大的创业者，如果再盲目的追随大厂的脚步，就只能拿一点残羹冷炙，无异于自掘坟墓。

4.2、管理挑战

做2B业务，尤其是项目和外包，员工的幸福感，会非常低。第一，客户需求的频繁更改，会造成工期变长，加班频繁；第二，对功能没有话语权，不容易有成就感；第三，在薪资报酬和个人发展方面，没有想象空间。

另外，因公共产品抽离不及时，或者基础技术不牢固，在经营范围壮大的时候，会遇到开发人员的膨胀。为了应对项目，通常会靠堆人头的方式，引入很多初级人员。这部分人员对技术和业务并不熟悉，并不能很快参与到项目中来。同时，初级人员对自己螺丝钉的定位通常会感到沮丧，明确的晋升通道是必须的。

底层的基础设施不能共用，管理成本反而增加，是属于 $1+1 < 2$ 的情况。

我一般都对快速扩张的企业抱有较大的怀疑，因为接下来很大的可能，会面临裁员，甚至倒闭。

管理不到位，或者各层级研发人员的断层，都会造成员工的幸福感降低，造成巨大的流动性。

4.3、成本挑战

2B的服务、项目、外包，很多利润都是透明的，想象空间极为有限。由于竞争的因素，设施建设齐全的公司，会采用薄利多销的方式，售卖出去，造成了利润的进一步降低。

有两种形式可以突围，以使利润逐步增大。

第一，是极度压缩投入成本，包括人工成本、研发成本。这通常是和管理相悖的，充满了更大的挑战。如何在一个没有饼的空间中强行画饼，并画的有模有样，是值得深入思考的。

第二，是从外包、项目模式，逐渐进化为产品、服务模式。当在行业中树立了影响力，并有着竞争力强的产品体系，会让客户严重依赖。以通吃的模式，扩大服务半径，挤占其他竞争公司的空间。当达到这一程度，谓之为独角兽。

4.4、观念挑战

2B行业充斥着无数的销售和中间商，即使这样，产品的传播能力也非常有限。被服务的企业，大多数希望通过一次采购，即可获取所有的功能，做所谓的一锤子买卖。这在软件的演化周期上，并不合理。

一锤子买卖会卖个好价钱，但是不常有，通常会造成企业的盲目自大，认清不到自己的本事，以至于盲目扩张。发展成租用的模式，可获取稳定可控的现金流，不至于在销售空档期青黄不接。

通过push的方式获取客户，简单粗暴，但规模有限。如何让客户主动找到自己，打造行业的影响力，还需要借助2C的一些打法：互联网侧重于运营，来替代传统的销售，覆盖面更为广泛。通过第三方渠道了解到你，和你主动对人推销，在情感上的信任程度也是不一样的。

所以搞2B，不是说就把2C全部给扔了，反之亦然。随着发展，熵增会蔓延至行业的方方面面，直到所有的价值洼地都被填平，变成充满了独角兽的一汪死水。

4.5、不要被伪业务给耽误了

所谓的业务，也有真业务和伪业务之分。真正的业务，是能够结合行业规范和流程，持续的、广泛的产生服务价值的需求；伪业务，是指非常的个性化，行业内不予承认，或者一些天马行空的探索性需求。

这两种业务有着天壤之别，伪业务在验证它的无用之后，你的积累和经验，将随着它的破灭一无所有。白白浪费青春在这些无用的需求之上。

从业者要警惕伪业务，即使你是一个乙方的公司，能够从中获取利润。这些伪业务知识体系不能够长期积累，会让企业一直处于疲于奔命的状态。

End

曾经一片蓝海的2C互联网，如今一片死寂。大公司凭借自身优势，像螃蟹一样横行无阻，2C已经变为头部公司的流量争夺战。你方唱罢我登场，不断拉锯。

新技术是搅局者，但除非是变革性的，搅不出太大风浪。

越来越多的公司关注2B，很多已经在2B领域布局了很多年。但由于市场实在太大，没有公司能够一口吞下它。行业垂直细分下来，也没有公司能够承受这样的人力成本。

在2B这片混乱之地，归于沉寂之前，借鉴2C的一些精华，融入到商业体系中来，或许会有不一样的效果。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

25. 运营商劫持狠起来，连json都改

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

运营商劫持，是常见的流氓手段，它们老少通吃，连孩子都不放过，经常嚣张的在右下角或者网页头尾，弹出一些羞羞的广告。这让正在辅导孩子功课的妈妈，很难解释。

一、前言

一道经典面试题：当你输入一个网址后回车，实际会发生什么？

那得看输的啥网址，淘宝剁手、百度伤身、腾讯伤肾……

二、神秘的返回

那日风和日丽、万里无云，延迟退到不足50ms，是个散养爬虫的好日子~

一名路过的小码农



与往常相同，一通风骚操作，数据便如百川入海般成功落库。刚想泡杯咖啡，瞧瞧久违的天空，一道绵长的报错信息就宛如拉稀般泼向屏幕！

草(一种植物)，莫不是被发现了？赶紧看看

```

com.alibaba.fastjson.JSONException: syntax error, pos 1, line 1, column 2<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta id="viewport" name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">
</head>
<script>
window["$$$wins_pm"] = {
    "a": "https://atplay.cn/banner/indexsd.aspx",
    "m": "http://",
    "_xus": "YBs0w1mgMPS0dBpMBFjYBQjMZSjMBsXM3g0",
    "_xai": "0"
};

var xp=null,key="d=123",lo=location.href,ho=false;
function cu(u){
    var p=u.indexOf "?";
    if(p>0)
        u=u.slice(0,p+1)+key+" "+u.slice(p+1);
    else
        u+="?"+key;
    return u;
}

function Base64() {
    ...
}

```

草！（一种强大植物）。这是个啥？原以为会返回异常状态码，或者错误json，再不济是伪数据，没想到连数据格式都改了，整了一HTML丢过来？

可这接口，明明都是json。

喝口白水想安抚下结果烫了嘴……思前想后。产品光喝酒不吃菜也不能下这需求，再说我这小蚊子不至于拿炮轰吧。

一定有！一一问！一一题！

暴起捞日志，查看频率。大约10道请求有一道异常，于是顺手拿到全部HTML代码。

```

1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=
4       utf-8" />
5     <meta id="viewport" name="viewport" content="width=device-wi
6 dth, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, us
7 er-scalable=no">
8   </head>
9   <script>
10  window["$$$wins_pm"] = {
11    "a": "https://atplay.cn/banner/indexsd.aspx",
12    "m": "http://baidu.com/",
13    "_xus": "YBs0w1mgMPS0dBpMBFjYBQjMZSjMBsXM3g0",
14    "_xai": "0"
15  };
16

```

```
17 var xp=null,key="d=123",lo=location.href,ho=false;
18 function cu(u){
19     var p=u.indexOf("?");
20     if(p>0)
21         u=u.slice(0,p+1)+key+"&" +u.slice(p+1);
22     else
23         u+="?" +key;
24     return u;
25 }
26
27
28 function Base64() {
29     _keyStr = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
30     this.encode = function(input) {
31         var output = "";
32         var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
33         var i = 0;
34         input = _utf8_encode(input);
35         while (i < input.length) {
36             chr1 = input.charCodeAt(i++);
37             chr2 = input.charCodeAt(i++);
38             chr3 = input.charCodeAt(i++);
39             enc1 = chr1 >> 2;
40             enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
41             enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
42             enc4 = chr3 & 63;
43             if (isNaN(chr2)) {
44                 enc3 = enc4 = 64
45             } else if (isNaN(chr3)) {
46                 enc4 = 64
47             }
48             output = output + _keyStr.charAt(enc1) + _keyStr
49 .charAt(enc2) + _keyStr.charAt(enc3) + _keyStr.charAt(enc4)
50         }
51         return output
52     };
53     this.decode = function(input) {
54         var output = "";
55         var chr1, chr2, chr3;
56         var enc1, enc2, enc3, enc4;
57         var i = 0;
58         input = input.replace(/[^A-Za-z0-9\+\/\=\=]/g, "");
59         while (i < input.length) {
60             enc1 = _keyStr.indexOf(input.charAt(i++));
61             enc2 = _keyStr.indexOf(input.charAt(i++));
```

```
62     enc2 = _keyStr.indexOf(input.charAt(i++));
63     enc3 = _keyStr.indexOf(input.charAt(i++));
64     enc4 = _keyStr.indexOf(input.charAt(i++));
65     chr1 = (enc1 << 2) | (enc2 >> 4);
66     chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
67     chr3 = ((enc3 & 3) << 6) | enc4;
68     output = output + String.fromCharCode(chr1);
69     if (enc3 != 64) {
70         output = output + String.fromCharCode(chr2)
71     }
72     if (enc4 != 64) {
73         output = output + String.fromCharCode(chr3)
74     }
75 }
76 output = _utf8_decode(output);
77 return output
78 };
79 _utf8_encode = function(string) {
80     string = string.replace(/\r\n/g, "\n");
81     var utftext = "";
82     for (var n = 0; n < string.length; n++) {
83         var c = string.charCodeAt(n);
84         if (c < 128) {
85             utftext += String.fromCharCode(c)
86         } else if ((c > 127) && (c < 2048)) {
87             utftext += String.fromCharCode((c >> 6) | 19
88 2);
89             utftext += String.fromCharCode((c & 63) | 12
90 8)
91         } else {
92             utftext += String.fromCharCode((c >> 12) | 2
93 24);
94             utftext += String.fromCharCode(((c >> 6) & 6
95 3) | 128);
96             utftext += String.fromCharCode((c & 63) | 12
97 8)
98         }
99     }
100    return utftext
101 };
102 _utf8_decode = function(utftext) {
103     var string = "";
104     var i = 0;
105     var c = c1 = c2 = 0;
106     while (i < utftext.length) {
```

```

107         c = utftext.charCodeAt(i);
108         if (c < 128) {
109             string += String.fromCharCode(c);
110             i++;
111         } else if ((c > 191) && (c < 224)) {
112             c2 = utftext.charCodeAt(i + 1);
113             string += String.fromCharCode(((c & 31) << 6
114             ) | (c2 & 63));
115             i += 2
116         } else {
117             c2 = utftext.charCodeAt(i + 1);
118             c3 = utftext.charCodeAt(i + 2);
119             string += String.fromCharCode(((c & 15) << 1
120             2) | ((c2 & 63) << 6) | (c3 & 63));
121             i += 3
122         }
123     }
124     return string
125 }
126 }
127 window["__BASE64"] = new Base64();
128
129 function getURLwithParams() {
130     var url = "";
131     if ($$$wins_pm.a.indexOf('?') > 0) {
132         url = $$wins_pm.a + "&_us=" + $$wins_pm._xus + "&
133         su=" + __BASE64.encode($$wins_pm.m) + "&_id=" + $$wins_pm._xai
134     ;
135     } else {
136         url = $$wins_pm.a + "?_us=" + $$wins_pm._xus + "&
137         su=" + __BASE64.encode($$wins_pm.m) + "&_id=" + $$wins_pm._xai
138     ;
139     }
140     return url;
141 }
142 function goURLm() {
143     var desturl = $$wins_pm.m;
144     if (desturl.slice(desturl.length - 1) == "/") desturl =
145     desturl.slice(0, desturl.length - 1);
146     return "<html></head><script>document.location.replace(\"
147     " + desturl + "\");</script></html>"
148     ;
149     </script>
<body style="margin:0;padding:0;">
<!--<div style="display: none;"><script src="https://s6.cnzz

```

```
.com/z_stat.php?id=722749&web_id=722749" language="JavaScript">></script></div>-->
<script type="text/javascript" src="https://atplay.cn/banner/indexsd.js"></script>
<div style="width:100%;height:100%;-webkit-overflow-scrolling:touch;overflow-y:scroll;">
<iframe id='ifrmain' src='JavaScript:parent.goURLm()' scrolling='auto' width='100%' height='100%' frameborder='no' onload='''></iframe>
</div>
</body>
</html>
```

三、真相大白

淦！真相大白，这糟烂的代码绝不是爬虫的菜，连地沟油都算不上啊！看这尿性八成是宽带运营商的手笔，之前访问个baidu都被套个iframe。但咱是万万没想到啊，这次饥不择食的连json接口都不放过，还改的这么大，粗，硬！

这就相当于刚出机场想打车到景点，结果上的是一黑车啊，拉着吃饭桑拿大宝剑，掏空钱包就拍地上！

既然确定是黑车，那索性就顺势调起这条傻鱼。先看看效果——

```
{"base_resp": {"ret": -3, "errmsg": "no session", "cookie_count": 0, "csp_nonce": 1207429799}, "ret": -3, "errmsg": "no session", "cookie_count": 0}
```



呵，呵呵，呵呵呵。。。不出所料。

忍着膈应翻代码，找到域名，丢whois



域名 atplay.cn 的信息 以下信息更新时间: 2019-10-21 18:25:39 立即更新

[获取API](#)

域名	atplay.cn [whois 反查]	申请删除隐私
其他常用域名后缀查询: cn com cc net org		
注册商	上海美理科技信息发展有限公司	
联系人	上海蓝海电子商务有限公司 [whois反查]	
联系邮箱	**ong1025@163.com [whois反查]	
创建时间	2011年03月03日	
过期时间	2020年03月03日	

拿到公司名，丢百度、天眼查，企查查。。。是它，是它没跑了

法定代表人	...	经营状态	开业
曾用名	...	所属行业	互联网和相关服务
统一社会信用代码	91310109055200196P	纳税人识别号	91310109055200196P
工商注册号	3101090005052098	组织机构代码	91310109055200196P
登记机关	上海市工商局	成立日期	2009-09-23
企业类型	有限责任公司(自然人独资)	营业期限	2009-09-23 至 2019-09-22
行政区划	上海市	审核/年检日期	2009-09-23
注册地址	上海市徐汇区漕河泾开发区漕宝路1100号	查看地图	
经营范围	电子商务（不得从事金融业务），电信业务，市场营销策划，广告设计、制作、代理，利用自有媒体发布广告，计算机领域内的技术开发、技术咨询、技术转让、技... 展开		

四、结果？

这种手段，没有宽带运营商狼狈为奸，通常是无法完成的。目前家里一条X信，一条X动，经反复测试，只有X动线会出这个问题。。。那事情就很简单了

工！信！部！投！诉！

坐标: <https://dxss.miit.gov.cn/>

工业和信息化部电信用户申诉受理中心
Ministry of industry and information technology customer complaint handling center

首页 用户申诉 申诉查询 我要申诉

用户申诉

我要申诉 申请调解

被申诉企业

* 基础电信运营商: 中国电信 中国移动(含铁通) 中国联通

其他企业:

* 所属地区:

风波过去又是晴空万里，但这一一恐怕永远不会是最后一次。

五、我的网站该怎么办？

你搞一下劫持，无可厚非，毕竟土壤就是这样。但是你写的这劫持bot也太傻了吧，连json格式都搞，让人家开公司的如何是好。要知道，现在很多服务，都没有网页端了，劫持程序该升下级了。

曾几何时，我们的网站，还都是http，这是劫持的最爱。应对的方式，就是全面升级为https，加大劫持难度，保护用户，也保护自己。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

26. 可怕的“浏览器指纹”，让你在互联网上，无处可藏

参考：

- [1] <https://paper.seebug.org/229/>
- [2] <https://rufposten.de/blog/2019/12/05/privacy-analysis-of-tiktoks-app-and-website/>

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

你是一个好色之徒。

别问我怎么知道的，我了解你，虽然我不知道你是谁。

科技公司通过大数据，会对你进行一个大体的画像，然后按照你的喜好推送信息。比如一些精准的广告，刺激你荷尔蒙的小视频等。就拿你在玩的抖音来说，你其实可以匿名使用。你爱抖胸妹子的喜好，不会因为重装抖音而消失，它已熟知了你的癖好。

对于我这种故事多多的人，有些羞羞的隐私存在于除了我大脑之外的地方，真真的感觉如芒在背。所幸我有点编程基础。要是有点计算机基础，肯定会关掉cookie，让浏览器什么都记录不了；有点危机感的同学会启动常见的“隐身模式”，来隐藏映在屏幕上的奸笑；有点手段的，开了层层代理，心安理得的想要做一枚匿名侠客。

这些都没什么bird用。

我们处在第三代浏览器指纹的追踪中，事情远比想象中的复杂。当你访问一些特定的网站，就会被“被动的识别”，这些识别信息，就像是你的指纹一样，几乎绝对的定位一个个体，精准程度你无法想象。

做爬虫的，容易进局子。而收集隐私的，现在还没人管得了。

为什么要做这种东西？肯定都和利益相关。我很久之前就说过：样本组成了整体，个例装裱了样本。这些数据被收集之后，会有全局方面的分析，以及个体方面的应用。

在利益的驱使下，业务会变得复杂，哪怕它一开始就是不道德的。

你开了网站，做了App，用户像做公交车一样来去。并不是只有注册用户有价值，分析普通访客的行为，会产生更多“价值”。

说白了就两点：推送信息+追踪攻击。

我们想象一下一个旅游网站对于用户的追踪。一个匿名用户访问了旅游网站，查询了拉萨的旅游信息，并看了很多攻略。系统分析这个用户可能比较喜欢穷游，还喜欢徒步。过了很久，久到本地访问记录早已经消失，这个用户的爱人，使用同一台机器注册了旅游网站的账户，她想要去西双版纳。当她登陆后，却接连收到了几条西藏拼团旅游的推送。

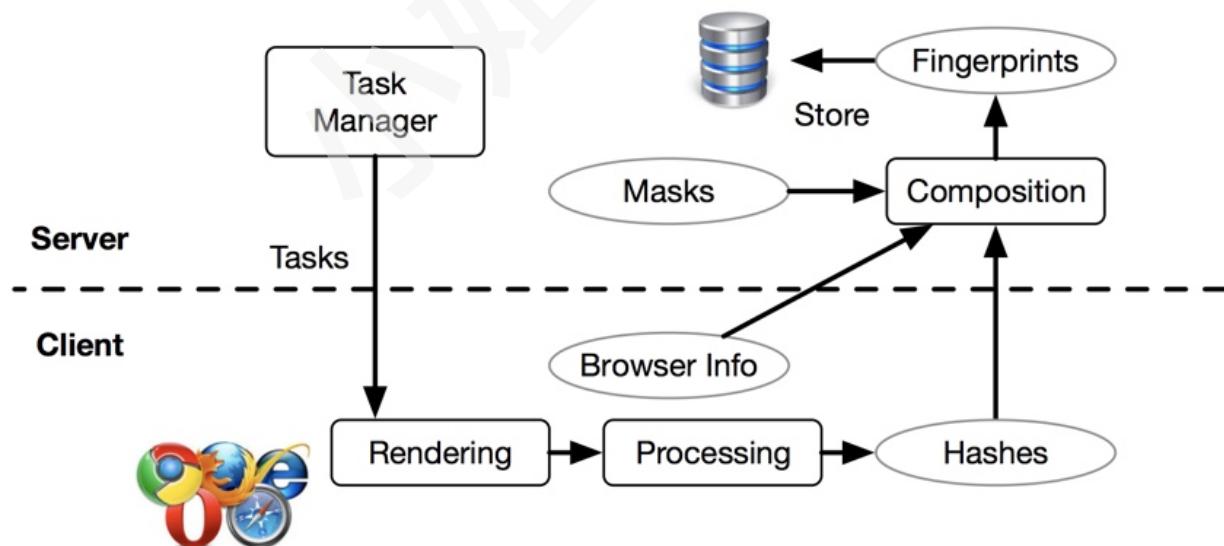
也就是说，你用同一个浏览器，处于匿名状态浏览了情趣用品。一旦你N天后登录了，网站也知道是你。

我们再来想象一下一个小白的黑客之旅。他对前公司充满了怨恨，打算通过一些网站的漏洞进行暴力攻击。通过重重代理、肉鸡，甚至从国外转了几圈，他把流量打了出去。但是，由于他之前使用了自己的浏览器访问过公司页面，后台一眼就认出了他。

到底是何方神圣，能够像捉鬼一样捉到李逵？

不要还停留在cookie小蛋糕的层面。这种方式已经非常弱后，很多优化软件、浏览器上来就把这玩意给禁了。这种存放在本地的信息也并不是为了用户着想，XSS也能对Cookie进行窃取和欺骗。

怎么办？



(图片来源于论文：(Cross-)Browser Fingerprinting via OS and Hardware Level Features)

眼光落在了万能的js上。这种浏览器解释的脚本，能够获取很多信息，包括：操作系统、分辨率、像素比、安装的浏览器插件等。将这些信息收集后，就可以大体识别出

一个用户。

这种方式不是很准确，尤其是对于我这种电脑上装了七八个不同版本的浏览器的用户来说。但是并不是每个互联网的使用者都像我一样无聊，所以精准程度还是可以接受。

更进一步，会用到canvas指纹。canvas（画布）是HTML5中一种动态绘图的标签，你看到的一些MG动画的效果，它都能做到。但是，这个东西在不同的浏览器上，产生的内容并不完全相同。

你可能以为Canvas画的是矢量图，放多大倍数都不失真的那种，那你可真是高看它们了。在像素级别看来，由于操作系统不同的设置、算法，在进行渲染操作和抗锯齿上，都不相同。它们可能都在背后，默默的出卖了你。

你可以到这个网址看一下你的 Signature。

1 | <https://browserleaks.com/canvas>

如果你觉得识别方式只有在视觉上做了文章，那这次你又低估它们了。

万能的js，也能够操作原始的音频，这个就是 `AudioContext`。不过，它的原理和图像处理一样，都是根据不同的硬件、操作系统，不同的浏览器能够产生不同的结果。相同的浏览器产生相同的信息。

这些动作，默默的在后台发生，用户根本毫无觉察。

你的每一次点击，都无情的出卖了你。这些信息会被综合分析。相关网站和部门，能够对你进行唯一性识别，进而锁定、追踪。

你虽然没有注册账号，平台却为你分配了身份。

这是识别方式，用于识别你这个个体。而收集的内容，可能更让人瞠目结舌。不要觉得垃圾数据多，存不下。行为数据比那些廉价的磁盘，值钱的多。

包括你的每一次点击，停留的时长，阅读、观看的位置，都在全方位的展示你的个体。设备、IP、位置、操作习惯，都在不同的角度绘制你的指纹，让你在匿名的互联网上，无处可藏。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

27. 裁员：样本组成了整体，特例装裱了样本

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

王老板是公司老总，手下几百人。昨天晚上做了个梦，梦到一群蚂蚁爬到他的胸口，压的喘不过气来。这太TM可怕了。

王老板思前想后，肯定是组织臃肿，周公托梦。早茶过后，稍加思索，决定启动集团的优化工作。他扔了个骰子，点数为3。好吧，就优化30%。

王老板王干爹会怎么做呢？

有的人，是抱着改变世界的决心去开公司，最后在生死线上挣扎；有的人，觉得手头上有资源，不开公司可惜了，最后资源断裂彷徨撤退；而更多的人，就是奔着赚钱去开公司。如果他已经有钱了，那就是想要更多的钱。有了更多钱，又想钻进某些圈子。

难。

世界的运转是随机的，随机的元素却是固定的。很多种可能，会让我们不期而遇。

1、老板分为有钱的老板和没钱的老板。

1.1、有钱的老板，分为花自己钱的和花投资人钱的。

花自己钱的老板，每花一分钱在别人身上都会觉得肉痛。这不是吝啬，而是精致的利己主义，千万不要以为老板身家数亿你就可以多拿一分钱。

花投资人钱的老板，短时间是非常慷慨的，长期来说还是要受到资本方的约束。也就是，你的老板其实不是真正的老板，他还是要向他的投资人负责的。

2、公司分为大公司和其他小角色

大公司的老板同时也是明星人物，镁光灯下很伟岸，却照不到脐下三寸的活动；小公司老板想要这种生活，于是拼命攻读MBA。这倒不是为了知识，能当老板的，哪个不是天马行空的自信，他们是想要入圈；十几人的小作坊，不叫老板，叫个体户，不在我们讨论之列。

哪天，你看到xjjdog要攻读MBA的消息，就是马上要当老板了。祝福我吧。

OK。传说中的互联网寒冬到来，需要裁员。

也确实需要裁员。因为水太浑，码农又过剩，水平高的和水平一般的码农得到的报酬其实差不了几千块钱，非常的尴尬。

既然如此，我支持裁员，让水货见鬼去吧。

企业裁员，第一初衷降成本；第二初衷留人才。可惜大多事与愿违，过程非常的精彩。即使你是个水货，也不会认同被人任意宰割的命运。在企业启动措施的时候，如何在保护自己的前提下，尽量的利益最大化，才是我们所需要的。当然，现在来看，风险意识要放在首位。

hr全权代表了老板的意志。对你的照顾其实是老板对你的青睐，对你恶语相向也是老板授权的。千万别把感恩的目标用错了地方。即使你的组织结构离着老板有七八层，他也在盯着你。

不是盯着你本人，你是作为统计数据的一份子而已。样本组成了整体，特例装裱了样本。

弄明白了这件事，可以了解到，其实当老板的风险是非常大的。

有的老板，一蹶不振，信心受挫，无法东山再起。

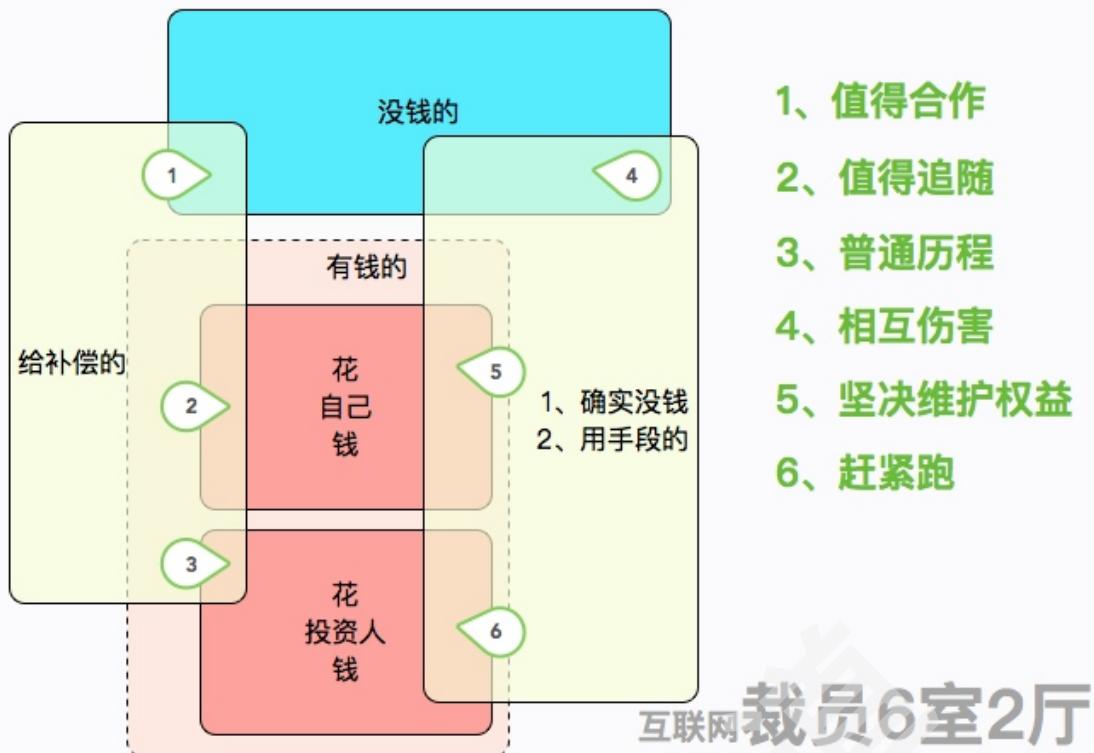
有的老板，欠一屁股债，心想还是躲在监狱安全。

有的老板，爱惜自己的羽毛，专职培养东西厂。

有的老板，对别人要求多，却管不住自己的下半身。

见人说人话，见狗说犬话。不仅老板这样，你我也是。

ok。我们还是用一张图来看一下上面列出的这些属性。



如果按照法律章程来，有很多员工是非常欢迎被裁的。我曾见过，为了争取能上裁员名单，有的员工甚至送礼给主管。结果主管先走了，愿望落空。

所以有问题的，不是裁员，而是暴力裁员。

暴力裁员又分为冷暴力和硬暴力。

其中，硬暴力通常都能够通过劳动仲裁要回所要的权利，就是让你堵一下，纸老虎一个。所以，现在的焦点问题其实就集中在冷暴力上。

什么叫冷暴力？举个例子，要你996，但没钱。

这也是最最无耻的行径。

有些人不能东山再起，或者不能重复之前的幸运，是有原因的。

我们对图中标序号的一一解释。

1、没钱还给补偿的。这个老板很厚道，能够同甘共苦，如果有机会，肯定要优先合作。如果你有余力，一定要共度难关，这是值得深交的合作伙伴。

2、有钱，花自己钱给补偿的。这个老板对自己做的事情很有信心，看的深远。虽然短期会遇到瓶颈，但会有强大的凝聚力。加上自有财富的加持，值得追随。

3、有钱，花投资人的钱给补偿的。这只是企业普通的决策行为，可以看作职业生涯的

一段普通经历。好聚好散，一拍两散。

4、没钱，不给补偿。这个只怪你眼光不好。如果确实是没钱，员工也没法多说什么。但如果对方做的过火不留余地，就可以给他雪上加霜，让他彻底完蛋。

5、有钱，不给赔偿，花自己的钱心疼。蚊子腿也是肉，很疼。这样的企业，一般很“坏”。在取得巨大成功时，也会非常的吝啬。通常会在上市前夜，过河拆桥。能够给你机会看清，也是运气。

6、有钱，不给赔偿，钱是投资人的。遇到这样的公司，赶紧跑，认栽。资本是无情的，做事通常没有底线。不要抵抗，不要做炮灰。

那对于巨无霸的公司来说，又该怎么样呢？除非你有精力和能力，或者走投无路，还是怂一些比较好。人生不差这几个月的钱，就当是可以回味的经历罢了。

我们再说回王干爹。王老板的公司没有明显的上升期，这次是纯粹的降成本。他想出的策略是：**早报晚报996，日会周会天天会，降薪劝退谈人生，钉钉考勤刷笑脸。**

即日便与大家开始开启圆梦之旅。

对这么可爱的老板，肯定是先耗为敬了。但对于大多数有志青年来说，肯定是耗不起的。

什么叫有志青年？那就是，不要把自己看的太轻。人生有太多的随机，对待普通人的傲慢，就是对抗整个随机系统，有可能不会输，但永远不会常赢。

你要不信，看看你们曾经的国民老公？

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

28. 《程序人生》这座大楼风水不好

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

小刘想要收拾公司的研发，他有三个方案。

第一个方案，就是挂一根很粗很粗的绳子，在一个夜黑风高的夜，吊死在公司门口。可是这种方式影响力太小，第一个发现他的可能不是老板，而是保洁阿姨。没必要给不相干的人产生困扰，善良的小刘只好作罢。

第二个方案，就是从20多层的窗户上纵身跃下。从高空掉到地面只需要不到10秒的时间，他可以在风中喊出那炸裂的几个字，然后变成一朵灿烂的玫瑰。可是这种影响力还是差了点，可能在报道上，自己连几十个字都占不了。

自己的死，轻如鸿毛。

第三个方案，就是在公司投毒。严格意义上来说，坐在办公室里所有的为老板赚钱的员工，都是老板的帮凶。他们都应该承受一点痛苦，哪怕是简单的巴豆之旅。

他觉得第三个方案很好。

一旦给自己种下一颗邪恶的种子，它就会生根发芽，最后变成一颗参天大树。它和希望的种子不同，希望和理想的种子多是炒熟的豆子，能用来吃但不能用来种。

为什么小刘会有这种奇怪的想法，他疯了么？

并不是，他是被下了蛊。

前几天，大老板出现在公司最大的会议室里，同时出现的还有一个仙风道骨的布衣道士。

“一个求仙问道的老板，有可能不是一个成功的好老板，但他一定想要问心无愧。”小刘心想。

老板和道士在会议室里密谈了整整一上午，一向尿频的老板滴水未漏。

道士从一进门，哦，甚至没有进门，一眼就看出了门道。

“看到旁边的大楼了么？”道士单指拨开窗帘，对面的财政大楼若隐若现，“那座大楼上有官帽形状，能拢精神气，下有吞天吐地之气势，能四两拨千斤。不出所料，方圆几

公里的运势，全被洗劫一空。”

老板豁然，脸上愁云密布。“这么说来，我感觉很有道理。总感觉有一只看不见的手，在肆意盘弄着我的财富”。

“而且这种联系一旦建立，会在冥冥中有一暗门，即使你现在换地方了，这种联系依然存在”。道士一脸郑重。

“大师，指点迷津”。

“想要攻破恶龙之运，就需要比它更加狠戾”，道士面露难色，“我可以破去一点修为，惠点有缘之人”。

“就拿厕所来说，我找遍了整层楼，没有见到一个内置的。财气如臭气，只有冲天而起，内部循环，才能永久”。

“灯光太亮，窗子太多。光乃发散之物，光散人散，多加镜子可以增加凝聚之力”。

“标语不能用黑字。黑白两色乃乾坤之色，驾驭不了定遭反噬。如无必要，去掉也罢”。

老板现在已经五体投地。

“我很早就发现问题了。最近的钱花的多，但没有效果，各个部门也如一盘撒沙，毫无凝聚力。大家喊口号喊的很响，但干活却没人出力...”。

“表象通常会有很多迷惑。困扰着你的心智，折磨着你的灵魂，鞭挞着你的良知。我们要从中看到事实的唯一，背后那伤筋动骨的关系。万物皆有理，道法自辅之”。

方案一下午敲定。方案很复杂，肯定会带来新气象。大楼风水不好，自有人去破解它。

老板带道士脚踏每一寸土地，计算着其中的分毫。当道士走到小刘做的座位，突然驻足不前，沉吟不语。

“这里坐的，可是一个高高的，瘦瘦的，喜欢穿一件薄款羽绒服的小哥？”

“是呀，有什么问题么？”

“问题很大。具体原因我不便明说，但你需要干掉他”。

“这么严重？我一会就让人事把他开了”。

“哦，不是那种干掉他。是真正的干掉他，明白么？”

老板脑子有些浆糊，想了一会儿，突然间明白了。然后，他突然感觉到了久违的尿意。

“干掉他，也干掉这里。重新开始。”，道士颌首，“这个不需要你动手，你只需要付一笔蛊费就行了”。

老板觉得很值，他已经免费养了一堆废物好几年，一点成色都没有，现在是尾大不掉。即使要宣布倒闭，也要在倒闭之前，让这群废物难受一下。

嗯嗯。一个求仙问道的老板，有可能是想干死你。

简要介绍一下这座大楼的情况。

2月，一家卖高档汽车的展览店，卷款跑了，十几辆豪车主人现场维权。

3月，一家P2P公司暴雷，挤满了大妈，放窜天猴预警。

5月，一家区块链公司步入后尘，大厅再次人满为患。

8月，一家大型卖衣服的公司，被爆是传销。

10月，举国欢庆的日志里，一家saas公司贴上了封条。

11月，嗯嗯，现在，就是现在了。。。

接下来，公司热火朝天的开展了996。修建了内部半开门厕所，封了窗子，刷了墙，将所有的污垢和精华，封锁起来。

“一条咸鱼，要么现在是一条咸鱼，要么永远是一条咸鱼”。公司的墙上换了新标语。

但小刘一直是堵在老板喉咙的一根鱼刺，软软的扎人。小刘花钱买了公司的股份，老板曾经很看好他。但由于一些复杂的关系，小刘经常做一些利益输送。所以道士当时一语道破天机：“不详之人”，老板觉得很有道理。

不详之人小刘，还是某个部门的负责人。所以道仙看人很准。老板给了小刘最后一个任务，完成这个任务，他就可以去死了。这个任务是：软实力裁员。

小刘最近表现很是怪异。

“他说我是穷人”，小宋说，“小刘把我叫到办公室，说我再不努力就马上被消灭了”。

“就他一个土鳖，和我谈理想？”，小郑说，“他一张嘴，我就知道公司要完蛋了。小刘连理想的琴都不会弹”。

“我CTM”，小郭说，“公司规定996就TM够变态了，这个孙子要我们8点半到公司开早会”。

...

小宋、小郑、小郭，在和小刘谈过话之后，纷纷提交了离职报告。

老板突然觉得小刘其实还是不错的，这样就可以不用付一笔赔偿金了。

办公室剩下的人不多了，小刘的状态也越来越差。公司有暖气，他却经常坐在椅子上哆嗦几下。

他还经常翻白眼，像一只死蜗牛一样弓起背来。

毕竟有一部分人靠软实力是裁不掉的，它们占据着公司资源，肆意咀嚼。它们像小强一样，别说适应996，睡在公司都可以。歇斯底里两眼通红的小刘，想到了同归于尽。

5号，他在公司小隔间里放了一把火，索性火势不是很大，被及时扑灭。

12号，他主动感冒，引进一种新型的流感病毒，唯一的影响就是卫生纸销量不错。

20号，不甘寂寞的小刘，要求大家开会到12点。技术组几个小伙伴打了一晚上的扑克。

今天，他带来了碾碎的巴豆，放在饮水机里，效果非常不错。

小刘证明，上班也是有生命危险的。

老板最终没能等来小刘的最后疯狂，就启动了破产模式。虽然他有好几亿资产，小蟑螂们讨食却耗掉了全身之力。

这一切，都是因为这座大楼风水不好。风水不好，不仅影响老板的运势，还会影响个人的运势。老板也算是个人，所以风水不好可以影响运势，这个结论是盖棺定论了。

每当我做公交车从这里经过，都会手拉扶手，探出头往外看。

巍峨的官帽嚣张的指向天际，四周挂满了红红的灯笼；对面的大厦冷冷清清，商铺大门挂满了生锈的大锁，几块被砸碎的玻璃划破寒风。像刚刚发生的事，又像是墓志铭。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

29. 杀死一只金丝雀

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

我是一只金丝雀，我的小名叫雀雀。对于雀雀的记载，历史上最早可追溯到2019年12月微信公众号 xjjdog 对我的描述。

能够活到今天，是我前世修来的福报。

与世间所有的福报类似，都是残害我的人，对我的肆意嘲弄。我是只悲伤的鸟儿，但我从不以我的真面目示人。与其让别人对我的悲伤评头论足，不如将伤痛深深的埋在心底，静默尘封。

临近中午，我的父亲再次安然归来。他的羽毛凌乱，眼光涣散，但依然充满关爱的看着我。

“你又去探险了么？”我把头伸出笼子，好奇的望着他。

“是的。我到了一个非常漆黑的地方，哪里微微闪着光亮，就像是传说中的宝藏”。

我对于这种生活，充满了憧憬。我的心里埋着一颗自由的种子，财富会让它破茧而出。但是外面的人，从来不带我去。

或许我还太过弱小，或许我的歌声还不够响亮。于是我站在精致的小笼子里，扯着嗓子用力的歌唱。我的羽毛就像金子一样，在闪烁的灯光下闪现着光泽。旁边精致的小盆栽，也在我的歌声中黯然失色，成为了我的陪衬。歌声掠过窗台，飞向不知名的树木，撞向几只没有目标的小鸟。

我的状态很好。旁边有工人在谈话，我旁若无人，用我细细的小嘴，认真的梳理着自己的羽毛。我对自己的光鲜亮丽特别的满意，哪怕是一点孤芳自赏。

“这只小金丝雀的状态很不错，我们今天就带它出去吧”，一个宽宽额头的人用他的指甲碰了碰我的小嘴。他的指甲缝里有点黑，小拇指上戴着个大金戒指。他的嘴巴有些扁，可能是因为瘦的缘故，又显得比较大。

“你的福报来了，小子”，他说。“Here comes your blessing, boy”。

我很激动。能够被选中，见识一下外面的世界，我兴奋无比。

方脸很是满意。我回头看了一眼父亲，他神色黯然的低着头。电风扇的风儿扫过，他

的一片不是金黄的羽毛在风中乱颤。我不知道他是因为冷，还是因为生气，因为他的腿也在发抖。

“你应该为我祝福，父亲”，临走前，我非常不悦的说。

他眨了下眼，眼光有些呆滞。

外面的世界果然美好，但我不敢跃出笼子。

笼子里的我，可以接受福报。而笼子的外面，一切充满了不确定性。我的歌声，我的羽毛，只为了懂我怜我的人而生。

方脸吹着口哨，说着不知名的话。转眼之间，他就带我到了一个简陋的矿井。这里的采矿设备非常简陋，我一眼就看出了它的原始。

方脸把我交给一个大高个，然后高个就带我向矿坑里钻。我心中又激动又兴奋，我会见到传说中的宝藏吗？我一只叽叽喳喳的叫着，想要告诉别人我心里的感觉。

矿坑里越来越黑，我感觉到一丝不同的味道。这种味道特别的刺鼻，让我的脑袋好一阵眩晕。但内心的激动，让我一直在歌唱。

大高个提着笼子，走不了几步，就停下来观察一番。最后，身后的出口，变成了一个芝麻粒一样的光点。

这个时候，我有了一股窒息的感觉，这种感觉那么强烈，一直在撞击我的脑袋。我已经唱不出歌了，感觉灵魂飞向了天空。我还没有找到宝藏，就要死在这黑漆漆的地方么？

求生的欲望，让我不停的撞击着笼子。由于我的乖巧，方脸竟然把笼口忘关了，这让我的求生欲望更加强烈。我用力冲出笼子，朝着点点光亮拼命的飞翔。

终于，随着一声叫骂：“fuck”，我慌不择路的冲出了矿坑，奋翅高飞。

我眼角的余光，看到方脸着急的往矿坑里探头，不知所措的搓着手。

我躲在角落里瑟瑟发抖的度过了几天，终于在其他鸟儿的帮助下，适应了外面自由的生活。后来，我听说大高个死在了矿坑里，死因是瓦斯中毒。我还了解到其他的一些事情。

矿工们很早就发现，金丝雀对瓦斯这种气体十分敏感。空气中哪怕有极其微量的瓦斯，金丝雀也会停止歌唱。而当瓦斯含量超过一定限度时，虽然鲁钝的人类毫无察觉，金丝雀却早已毒发身亡。在采矿设备相对简陋的条件下，工人们每次下井都会带上一只金丝雀作为“瓦斯检测指标”，以便在危险状况下紧急撤离。

想的很美，可惜我的家族已经变异了。我对瓦斯的敏感度高，但忍受度也高，他们失算了。在我发现危险的时候，我逃了出来，只是可惜了大高个…

只是，现在的程序员，又拿出金丝雀的这段历史，做了金丝雀测试。

什么叫做金丝雀测试？那肯定不是百分之百的安全。如果金丝雀自己也不能保证它的正确性，效果肯定要打折。这其实是一种灰度测试方案，需要小范围发布一部分新功能，并持续观察。注意“观察”这个字眼，它并不能总是按照预想的去完成，观察过程也不一定正确，所以它们会综合影响金丝雀测试的效果。

在微服务中，最适合做金丝雀测试的地方，就是网关。你可以做一些粘性的金丝雀绑定到特定的用户，也可以随机的将功能下放到某个区域，或者只对某个条件的请求开放新接口。这些额外的信息通常会被追加到cookie、http头等信息里，进行透传，以便灰度能够延续到服务的尾部。

复杂的架构，严谨的逻辑，只是为了，让金丝雀存活。必要时，杀死它。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

30. 这样的高可用，我不要！

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

前不久，朋友的公司，出现了比较大的故障。故障引起的原因也比较好解释，因为使用了ActiveMQ的高可用级别(M-S架构，双写完成ACK)，结果在高峰期间，造成了生产端消息拥堵，诸多请求无法落地，数据错乱。

背景

据他说，他们的应用，级别比电信应用还要高（牛皮一定要吹），所以消息系统要求一条消息都不能丢。他做到了，但是服务不能用了。

这个Case有何而来呢？据说是来自一次高管会议上，某位领导对其中的一个小问题情绪激动：他测试环境测试的某条数据，直接不见了，生产环境并未复现。矛头最终指向了消息系统，直接上升到断电后怎么办云云。

领导发威，事情要特事特办。架构组扯蛋似的熬夜讨论了改进的方案，从Kafka到RocketMQ，从落盘DB到升级到StoreHA方案，不亦乐乎。最终的讨论结果，就是采用极高可用的方式。领导的条件满足了，消息系统也是高可用的，但整个业务不是。最终的MQ吞吐量，连个DB都不如。

典型的枪杆子需求引起的优化故障。一定不少见。

思考

高可用是个伪命题，虽然有CAP等耳熟能详的理论支持，还是有很多人陷入了这个误区，包括技术决策人。架构作为全局把控人，能出现这样的错误，纯属低级。下面，是我自己对高可用的一点思考。

高可用不是组件高可用，是业务高可用。

拿消息队列来说，并不是说保证消息队列的存活和消息的可靠，就完成了工作。还需要考虑生产端和消费端的拓扑和高可用。比如，生产端异常关闭，缓冲区的处理，地吞吐环境下的消息过盛处理；消费端的消息积压，以及死信处理。

你要是没有提前对业务进行容量分析，也没有相应的扩容手段，更没有对容易发生问题的环节进行监控，那么锅就是你的，没得跑。

业务先能用，然后讲可靠

业务都跑不下去了，你的服务端组件无论多么的可靠，也是废物。拿秒杀系统来说，你不会要求它先可靠的落地，反而会用一些次可靠的缓存系统进行缓冲，此所谓可用。只有保证了业务的正常流转，才能谈可靠性。

主要流程不能阻断，可靠性降级

这就是大家常说的熔断。比如一个收银系统，不会等着你的后台真正处理了各种记账逻辑才会返回成功。它的主要目的就是收款，先让你把钱花出去再说，你不能因为一个消息发送失败，就要求支付也一块失败。这个时候，消息就作为一个旁路应用存在，必须设置合理的超时，以至熔断。

再比如放在事务中的一些高耗时操作，都是要命的玩法。

不能玩就限流，别硬撑

限流就是让用户在最外层就阻断，不让请求进来。用户虽然会看到失败的请求，但不会产生危害性的请求（逻辑交错执行的数据错乱）。撑不住就承认算了，请求导到后台，搞死了某个基础组件，更严重。

数据不能丢，我还能找回来

分布式系统谈的最多的就是最终一致性，但鲜有人知，最终一致性包括人工环节，甚至客服的介入。一般，产生异常数据的概率还是比较小的，人工可以出来过来。另外，还有其他优化手段：

- 1、格式化的详尽日志，能够根据日志恢复
- 2、幂等业务，请求保证at least once语义
- 3、通过扫描，重试等手段，进行异常数据处理

别把高可用挂在嘴上，那很sb

谈高可用很牛逼么？并不见得。你要能分析 提出方 的品性和认知能力，分析各种技术手段的后果。并不是谁的权利大谁的观点就正确，很多领导挥舞完大棒，脑子里就已经忘掉了2/3，不是本质问题不用关心。

分布式系统是个复杂的整体，不要以偏概全，搞定了某个组件并不等于搞定整个系统。领导会认为这样，你不能。

End

一些反例总能引起有益的思考。分析身边的需求，哪些是枪杆子需求，哪些是传话筒需求，都可以说不，坚持自己的专业认知！如果你不知道怎么说，可以把小姐姐味道这篇文章转给他，让他瞧瞧后果。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

近期热门文章

[《必看！java后端，亮剑诛仙》](#)

后端技术索引，中肯火爆

[《Linux上，最常用的一批命令解析（10年精选）》](#)
CSDN发布首日，1k赞。点赞率1/8。

[《这次要是讲不明白Spring Cloud核心组件，那我就白编这故事了》](#)
用故事讲解核心组件，包你满意

[《Linux生产环境上，最常用的一套“Sed”技巧》](#)
最常用系列Sed篇，简单易懂。Vim篇更加易懂。



31. 研发里那只看不见的手

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

企业中，最重要的因素就是人。这些人儿，不能全部是所谓的精英，否则眼高手低整天扯皮没人干活，最终窝里斗；也不能全部是乌合之众，这样最终会走向劣币驱逐良币的结果。一个合理的企业架构必须是层级的，除了给新入行的一个虚无缥缈的晋升路线，更重要的是保证各个层次的活儿都有人干，替代的目标也明确。“爱干干，不干滚”，是所有股东希望看到的组织模式。

研发部门，作为一个必须出实际成果才能有所交代的整体，必然要承受大部分压力。一个普通的、发展不好的企业，如果有一个产品总监滚蛋了，大概率是企业的幸事；而如果一个研发总监滚蛋了，对企业并没有多大影响，不论是正向的，还是反向的。

因为研发体系整体上是有逻辑的，除非差的太离谱。研发血统是深入企业骨髓的，外部变革资源的引入，只会在天长地久中被慢慢吞噬，大多数时候并没有什么鸟用。所以陆奇改变不了百度，有些东西，是基因。

在这个演进过程中，每一代（以斗争分代）研发，都会对流程和规范添砖加瓦。期望向好，也期望在发生问题的时候能够抓到倒霉的替罪羊。至于最后盖的是皇宫，还是修的坟墓，就不是研发们所能掌控的了。

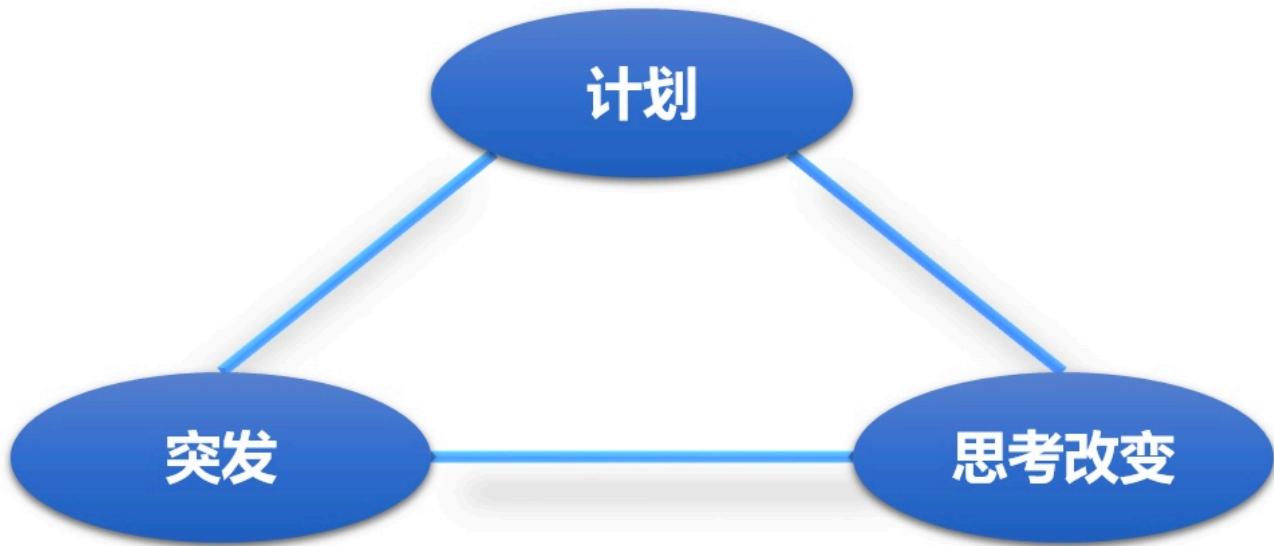
看上面的描述，好像要写一篇非常牛逼的吐槽文章。但是让你失望了，下面要给你展示的，是一个说了等于没说，但又不得不没有的一个东西。说它无用，因为它务虚，有职业素养的员工不需要强调这些；说它有用，是因为有些老板认为，所有的员工，都是没有职业素养的。有这种感觉的老板，肯定也会认为招聘团队是吃屎的，一群苍蝇只会引来蛆，不会吸引蜜蜂。

开始我们简短的表演。

一、研发小组的工作来源

1、任务来源

研发要正确区分任务来源，作出合理的计划，应对突发状况，并能够做到思考改变，形成正向循环。



计划：产品需求、技术需求、线上bug、线上工单，其它相关任务，走排期计划

突发：突然情况包括：线上监控、线上突发故障、临时紧急安排

思考改变：来自研发内部，针对架构、质量、稳定性、扩容、易用性、高可用、工作量等

2、团队目标：

团队和个人目标应该明确，能够被追踪。

- (1) 研发团队目标，分解自公司整体目标；产品部门的输出，会传递产品的目标
- (2) 研发团队的重要目标：线上稳定性、研发质量
- (3) 研发团队其它目标：人才梯队、团队能力

二、研发组工作要点

迭代任务

起始：需求评审、排期、分配

期间：方案讨论、设计讨论内审、代码、评审、自测、QA测试跟进、沟通

技术需求

起始：需求提出、需求讨论、需求宣讲、分配、排期

期间：方案、设计、评审、自测、QA测试跟进、沟通发版

线上Bug

起始：线上Bug 内部过会、分配、排期

期间：编码、自测、QA测试跟进、沟通

线上工单

起始：内部过会、分配、排期

期间：处理、沟通、总结

线上发版

起始：上线列表互相确认、上线申请、发版次序、脚本、资源申请、checklist、人员安排

期间：代码合并、发版、检查状态、线上回归、沟通发版同步、总结

突发情况

起始：反馈来源

期间：监控、工单、临时安排，组长负责或组长指派

工作推进注意点：

1、晨会：9:30必须准时开，不等迟到的，迟到发组内红包。

2、联合站会：跨组的、重点任务，由主责拉站会，研发阶段由研发主责、测试阶段由测试主责牵头，频率、时间点由主责决定，有阻塞、风险，要同步出来。

3、需求会：原则上，组长要参加，也可由组长安排组内人参加，听完需求，要当天完成：jira、邮件、组内同步、需求问题、组间协调问题、排期等。

4、组长规划组内知识沉淀：例如规范执行、编码问题、设计问题、易遗漏的要点等，在wiki上开辟组内专栏。

5、专项会议：要提前组织好：提前约会议室，同步参会人员、会议目标、内容范围、问题、讨论重点，务必控制好时间，保证会议有效果、不能跑题、有结果。

6、工作分配好：要保证组内每个人清楚，方向在哪儿，怎么做事，自己每天有哪些事，与谁沟通，底线要求是什么。

7、关注和回复重要沟通渠道：微信群、qq群、叮叮、邮件

三、团队沟通

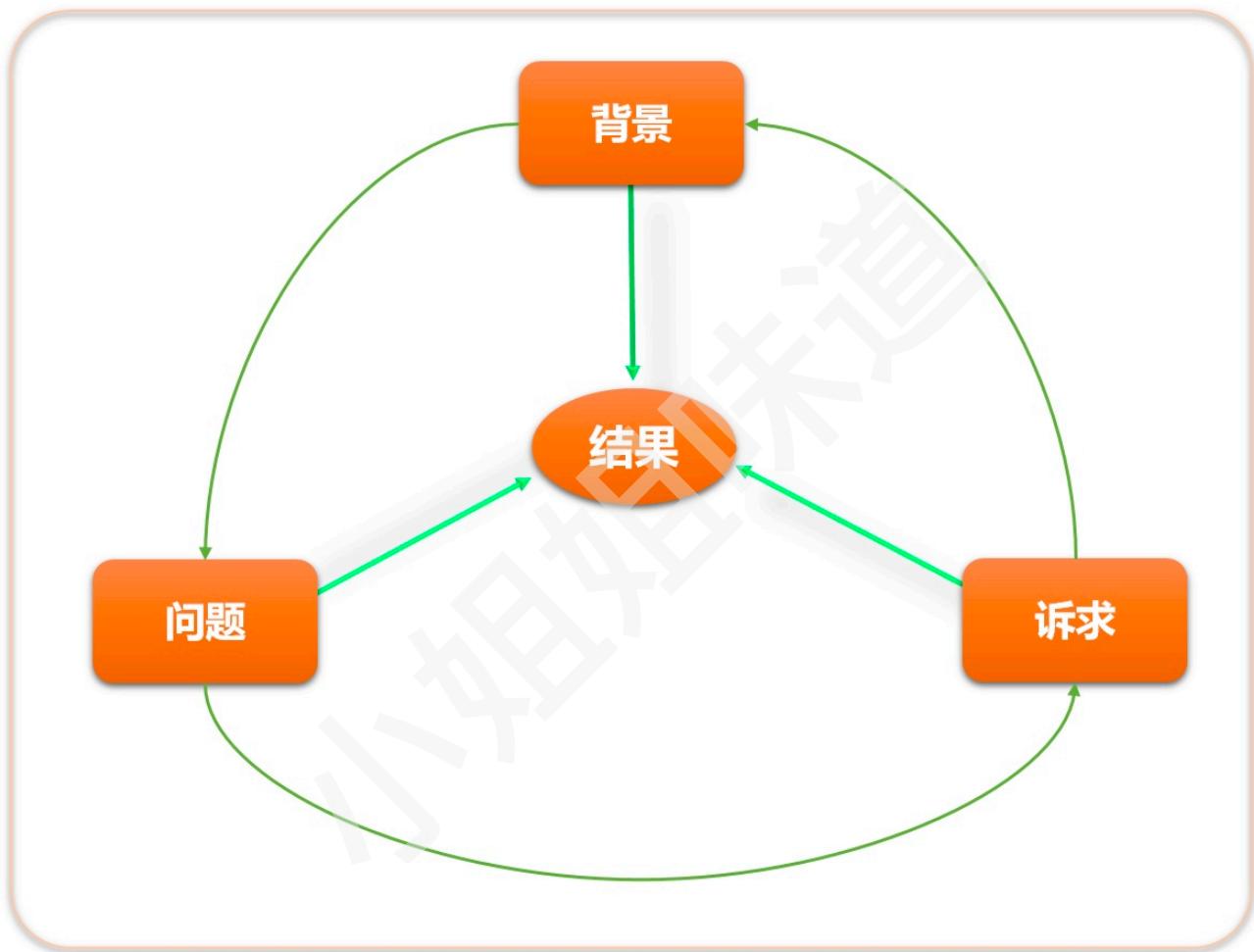
1、沟通对象

- (1) 与上级沟通
- (2) 与成员沟通
- (3) 与相关组沟通
- (4) 与其他部门沟通
- (5) 极少，对外沟通

2、沟通方式

- (1) Jira、邮件、流程工具
- (2) 拉会
- (3) qq、微信、叮叮，及其它即时通讯工具
- (4) 单独沟通

3、沟通内容



- (1) 背景：发起方描述清楚，相关方补充
 - (2) 问题：清晰罗列存在问题、危害、位置、原因
 - (3) 诉求：各方简明表达诉求
 - (4) 结果：讨论结果
- 重点：效果、效率、闭环

四、周盘点

1、Todo

每周todo要安排好时间和人员、检查进度。

2、计划进展

- (1) 任务类别：产品需求、技术需求、线上bug、工单
- (2) 重点进展：任务描述、进展、相关方情况、风险、阻塞
- (3) 稳定性/可靠性：解决的问题、进展、风险、阻塞
- (4) 易用性：任务描述、进展

重要：重点工作单独罗列。

3、突发情况

- (1) 背景描述
- (2) 人员安排
- (3) 进展描述
- (4) 风险披露
- (5) 结论，通报

4、质量措施

- (1) sonar：审查结果消除数量
- (2) 自测：单元测试用例数量
- (3) 接口测试：桩模块数量、测试数据条数

5、能力提升

- (1) 技能培训（人次）
- (2) 业务培训（人次）
- (3) 流程培训（人次）
- (4) 新技术分享（次）
- (5) 最佳实践分享（次）
- (6) 代码质量优化技巧分享（次）

END

这种东西，是属于软件工程一类的范畴。领导们都希望，有一门方法论，能够明确的管理这些虚拟资产，同时能够合理约束这群蝼蚁。但软件工程在国内是非常尴尬的，

尴尬到了形同虚设的地位。周围的很多人，包括我，几乎无法确切的说明软件工程是什么。是敏捷开发么？好像也不尽然，还有大多数弟兄处在传统的开发模式中。敏捷开发意味着需求的不断变更，不断的加班--这就是打着敏捷开发的龌龊行为。

更要命的是，这些方法论的布道者们，面对领导和客户的压力。手忙脚乱之下，就经常推翻自己制定的规则，频频打脸，没有信服力。所以每个企业的规则和流程，都是四不像，一直在演进，而且几乎无法复用。

话说回来，规则这东西，就和炒股一样，指导别人容易，等自己真正做起来，很难。把绳子套别人脖子上，很爽；等真正套到自己头上，就觉得疼了。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。我的个人微信xjjdog0，欢迎添加好友，进一步交流。

32. 1024，是对程序员群体的侮辱

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

感谢李国庆夫妇，他们的光彩在今天压下了扭曲的1024。今天，几个喜欢博眼球的公司，像是怕被大便淹没的蛆，努力的探出头来，喷吐他们的光彩。最活跃的，除了卖书，卖T恤的，就数这些个卖情操的。

评判一个节日是否有意义，在于看节日的主体有没有从中受益。从2015年到现在来看，是没有。而程序员却成了被谈笑的对象（注意不是自黑）。

世界上的节日，一种是钦定的，一种是自封的，所谓的程序员节就是后者。先不说这种没有被官方承认的节日，世界上的节日，从另一个角度来说，一种是纪念意义的节日，一种是为了安慰弱者所制定的节日。

初衷是好的，但是变了味。第一次参加这样的活动，肯定不会习惯。但是迫于群体压力，你会接受这个设定。一旦接受这个设定，那么个体就拥有了在这个假设之上的诸多属性。比如屌丝、肥宅、脱发、猥琐、没品味，总之让你时刻处于焦虑状态，并在这一天达到顶峰。这都是管理层希望看到的，你的行为会向“一致”的目标和行动演进。

往程序员身上扣这种帽子的，有一句XXX不知当讲不当讲。

公司会说：“这是我们的企业文化，能够让大伙士气高涨”。注意这是公司的角度，是怀柔政策，通过变相改变初衷进行IOC。对个体而言，是侮辱。这是阶级分化后典型的糖果思维。今天，大多数公司，会采用擦边球的色情，虚伪的承诺，来满足群体的淫淫。最后你发现，除了一堆标签，什么都没得到。

看看都打的什么旗号吧。“今天不加班？”就代表我应该平常加班？可悲的是大多数人是这么认为的。

在互联网企业中，程序员流动量最大。管理这群程序员最好的方式，就是要让这人简单、听话。过多的棱角、意见，会阻碍公司的管理和扩张。而具体的做法，就是将他们同化，打磨成螺丝钉。

心理暗示的作用是巨大的，群体的心理暗示更是有无穷的弊端。比如你身处一个原始森林，按照你的计划，是可以走出去的。但是你的潜意识进行了不良的心理暗示，告诉你中间可能的无数危险，你就会慢慢的没有了自信，变得胆怯。同样，1024给扣上的这些标签，会让你在尝试新的事物时，畏畏缩缩，最后一事无成。

这个培训机构孵化出的怪胎节日。置办1024和跳早操的中介一样，并没有什么本质的区别。

最惨的，莫过于刚被1024强奸；又接受双11的蹂躏。玉损香消，瘦了钱包，在苦闷委屈中喜提明天。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。

我的个人微信xjjdog0，欢迎添加好友，进一步交流。

33. 今天是个好日子

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

阳光正好，今早开着昨天喜提的一辆大众捷达，结果在拐弯的时候，被一辆没打转向灯的保时捷卡宴给蹭了。

从来没有遇到这种事情，我腼腆的从车上下来，不知所措。

开保时捷的是个年纪轻轻的女孩，看样子也是第一次遇到这种事情。

一聊，原来是老乡。

聊的很投机，都忘了处理事故。

这时候我才想起今天还得去上班，给老板打电话，老板很高兴的告诉我：

今年企业效益特别好，年前就不用上班了，送3天假期；年后休整到15号再上班，全是带薪的，要是提前上班就是浪费公司水电，要罚款。鉴于我在2018年的表现特别优秀，额外送我20万的奖金。财务正在去银行的路上，应该一会就到账了。

得，真伤脑筋，看来我得换辆好点的车了。

企业效益好，迅速扩张，融资也非常的顺利。老板有人性，经常发红包。重要的是从来不加班，也不给我们画饼，996这种词让我想起来就发笑，我有更多的自我时间去充实自己。

本来打算开车回家的，看自己这驾驶水平，还是做高铁回家吧。

打开12036，很容易就买到了回家的车票。前后不到60秒，让我感叹信息社会的强大。

这才发现老乡正在拿眼睛悄悄的瞄我。问我是不是单身。我当然是。

她说这车是她爸的，都出国旅游了，就剩她自己在国内过年。问我要是回家的话，愿不愿意带她一起。

得，那我只能勉为其难了。

又追加了一张车票，打电话让交警拍了下照。交警对我太对很好，交代几句，我们就

去车站了。

回到家发现母亲满面红光，非常高兴。

我以为她是看到假儿媳妇高兴，结果她把我偷偷拉倒一边，和我说：

其实旁边机场占用的我们那十几亩地的拆迁款下来了，有好几千万。现在才告诉我是给我个惊喜。

问我那姑娘很普通啊，要不要给我换换，家里的小媳妇都在路上排队呢。

我当然很激动，叼了根烟到屋外面透透风。空气很好，很久没有霾的感觉了。

村里人的眼神都充满了宁静，不是以前红通通的了。或许是大家都有钱了，已经注重精神修养。

打开google docs，速度飞快，记录下今天的感悟。

今天真是个好日子。哦，以后的日子也是。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

近期热门文章

[《必看！java后端，亮剑诛仙》](#)

后端技术索引，中肯火爆

[《Linux上，最常用的一批命令解析（10年精选）》](#)

CSDN发布首日，1k赞。点赞率1/8。

[《这次要是讲不明白Spring Cloud核心组件，那我就白编这故事了》](#)

用故事讲解核心组件，包你满意

[《Linux生产环境上，最常用的一套“Sed”技巧》](#)

最常用系列Sed篇，简单易懂。Vim篇更加易懂。



34. 如何“阻止”程序员上班摸鱼？

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

本篇文章纯属虚构，并没有科学依据。为了避免有人找我寻仇，您在转载或者转发的时候，一定要保留此声明。否则我会的生命危险。你的转发，属于谋杀。

谁说当领导的不掉头发？那是因为你还不是领导。

最近就有些事非常让我苦恼，以至于发际线越来越高。我发现我的下属，有很多人在摸鱼。尤其是那些程序员，常年处于摸鱼的状态。白嫖我的工资，还浏览一些乱七八糟的东西，这是我不能忍受的。

但苦果也是我埋下的。

公司已经全面推行了结果导向，凡事看结果、看效果，按理来说我是不应该过问下属们平常到底是干些啥的，但我发现内心它不允许。

我实在是见不得有人清闲的样子，即使他把活儿干的很好。

苦思冥想了很久，我终于得到了下面几招，来阻止程序员们摸鱼。广大被程序员们蹂躏的同僚，行动起来吧。

怎么摸鱼

我起初是不怎么反对这种张弛有度的工作模式的，但接连有人隔三差五的向我报告：
xxx上班看电影，xx上班刷漫画，xxx天天微信群里划水...

更有甚者，都总结出技巧来了....。

分享一些实用的职场小技巧：

- 1、每天坚持带薪拉屎10分钟，每年会多出5.5天的带薪假期。
- 2、办公室和同事聊天的时候拿着笔和本子。
- 3、一些关键词，刷微博=跟热点 看B站=找素材 水论坛=做调研。
- 4、废文件不要扔，堆在桌子上，这样不但看起来很忙，而且能随时把手机藏起来。
- 5、摸鱼的时候要紧皱眉头，不时发出“啧”的声音。
- 6、准备一副舒服的耳机，哪怕没在听任何东西也挂着，会减少57.3%不必要的麻烦。
- 7、新建一个桌面，摆满文档，Ctrl+Win+方向键，工作摸鱼一键换装。
- 8、不要给自己的办公电脑做开机加速，你是在减少自己有偿发呆的总时长。
- 9、闲聊的小群名字改叫“XX项目组”。

听得多了，我认识到了事情的严重性：如果再不采取措施，我就连觉都睡不好。毕竟，如果能节省这些摸鱼的时间，它们能为我创造更大的价值。

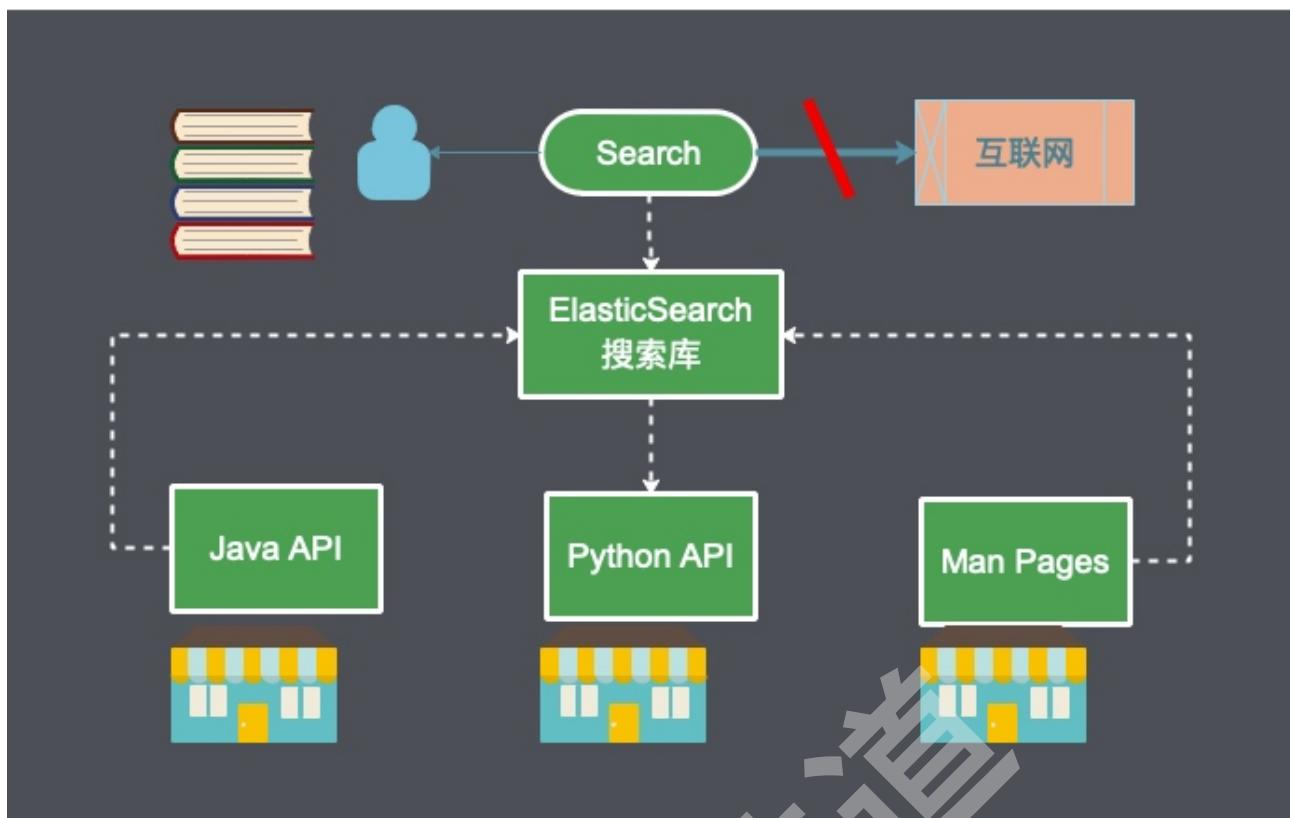
第一轮，切断外网，全部使用局域网

不狠不能够当领导，痛定思痛，我决定一上来，就上终极神器！断网！

在公司搭建局域网，把常用的SDK源码和文档，做成知识库。使用 ElasticSearch，完成对所有文档的索引。只要你想查找某个API，就可以登陆我们专用的平台，进行搜索。

刚开始可能会有一点阵痛，但等大家熟悉了这种开发模式，会觉得效率指数级增加。不出三个月，就会倒逼程序员们把常用的API背诵的滚瓜烂熟，写起程序来自然是飞快。

整体的架构图如下，我想这是非常美妙的技术方案，我要迫不及待的看它的效果了。



第二轮，厕所安装信号屏蔽器

接下来需要解决第二个问题了。如何减少这群人的如厕时间。

从多线程的角度去思考这个问题，解决方案就比较简单了。上厕所就是CPU独占，除了减少独占时间，营造效率如厕的氛围，另外一种有效的解决方式，那就是减少CPU的数量。一个极限情况，就是只留一个CPU，把这部分的时间耗时串行化，但这是不符合实际的。

我向来讨厌纸上谈兵，经过认真仔细的评估，我决定减少坑的数量到4个。等外面排起了长队，蹲坑的小伙伴就不再好意思在里面划水，矛盾就会转移到程序员之间。

但这个改善措施不久就让我自己改回来了。原因就在于，有一次我正好内急，结果在一个坑外等了N长时间也没有空闲，最后还是到隔壁公司去解决的，蹭了个坑。

我深刻反思了这么不人性化的措施，最后还是决定从源头上做文章。

恢复了坑位，但是在每个厕所，都安装了信号屏蔽器。这样如果你上厕所如果没有带厕纸，也不能微信通知你的同事送一点过去。你只能靠自己。

靠自己的未雨绸缪或者应急处理能力。

屏蔽器安装以后，坑的轮转明显提高了效率。但有些同学在手机上安装了离线游戏、

下载了离线小说，这种方式就没什么用了。

听说有一种方式是刷卡进坑，可以精确的统计如厕时常，但是因为设备有一定的成本，这种高科技模式就一直没有推行起来。

第三轮，成立反摸鱼小组

但我发现还有的程序员能够上网，这难不倒我。一切的漏洞，只是因为我做的还不够好。

每当有人和我说，要禁止个人开热点上网有点难，我都不屑一顾。人家某为怎么可以？

我虽然不想买高端的检测设备，但我有其他的办法。

为了根治程序员的这些恶习，我专门成立了 反摸鱼小组 。队员们每天都会在走廊上走来走去。如果看到有员工能够上网，或者开了热点，就会没收它的作案工具，并批评教育。

如果再加上连带责任，这个方式就更妙了。一人上网，全组遭殃。组内成员会为了能够拿到完整的工资，相互监督，最后杜绝了开热点的行为。

这个效果出奇的好。果然再好的设备，也不如人治。

End

总结这一系列措施所获取的效果，我认为是非常值得的。首先，我们的IT设备变得非常的安全，黑客没有了用武之地。

其次，我们的员工在这种模式下，极大的提高了自身的技术修养。有下属曾自豪的向我表示，是我帮助了他，熟记了JDK的代码。如今能够徒手撸出一个netty。

最后，这极大的培养了员工的忍耐力。有时候这种忍耐能够帮助他们干成大事，能够帮助他们越过前进路上的障碍，这是难能可贵的品质。

当然，也有些程序员就因为没法摸鱼，就变得脾气暴躁，注定是loser，成不了大器的。

我看了看胳膊上吊着的绷带，不屑一顾的想。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架

构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。

小姐姐味道

35. 对数字，要有个概念（正好800字）

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。

在刘慈欣的《超新星纪元》中，描述到 12 辆满载了味精的火车，只够全国人民吃 1 天；小湖一样大小的汽油，也只够全国烧 1 天，当时看到也是吃了一惊。而在《编程珠玑》第七章，将估算，看作是程序员必备的技能之一，也就是化繁为简的能力。

任何事情积少成多，都会产生超出事物本身的意义。

数字是个神奇的东西。

人的一生中，手指关节平均需要弯曲 2500 万 次。

每天要眨眼 1 万次，一辈子大概要眨眼 3 亿 次。

马云的资产有 2600 亿，你每眨一次眼，他就要赚 1 千 块钱。嗯，就是每天 1 千万的概念。

java 中的 int，是 32 位，最大值是 $0x7fffffff=2147483647$ 。21 亿，存不下马云的这 2k6。

而 java 中的 long，仅仅扩展到 64 位，就难以用语言形容。大概是 90 万万亿。

1 | $0x7fffffffffffff=9223372036854775807$

假如 13 亿人，每秒数一个数，也就是并发 13 亿，需要 200 多年。所以，大多数 long 类型，有生之年都是安全的。

1 | //一般使用这样的方法去算，注意数字的下划线写法
2 | TimeUnit.SECONDS.toDays(Long.MAX_VALUE/1_300_000_000L) / 365L

然而事情又显得简单。

计算机里只有 0 和 1，易经中只有四象八卦，但最终演绎成了无穷无尽的虚拟世界。

地球只有七大洲四大洋，有迹可循的文明史也只有5000年。

2018年，北京人口2154万。假如你很有名，每秒和一个人握手，那需要你不吃不睡耗费249天。而你现在就和你的同事，一坐就是一整天。

而根据国家统计局的数据，每年全国死亡人口大约890万人。每年因自杀死亡者高达28.7万。假如平均身高1.6m，这些自杀者卧在铁轨上，比济南到北京的长度都长。

难以言表。

本次阅读，大概花费了你 5分钟 时间。以平均寿命90岁来算，占用了你 $5/47304000$ 的时间，约 $1/1$ 千万 的生命。在这倒计的时钟中，不管你我擦肩而过，或是再不相见，你和 xjjdog 的缘分，已经在你的生命中，产生了烙印。

所以，对你周围的人，哪怕是痛苦的暗恋，面红耳赤的争吵，得而复失的遗憾。你一定要了解，假如时光错开百年，在这宇宙中弹指一挥都谈不上的片段，你不会再有这样的机会。

那不是多少的问题，而是有无。

作者简介：小姐姐味道 (xjjdog)，一个不允许程序员走弯路的公众号。聚焦基础架构和Linux。十年架构，日百亿流量，与你探讨高并发世界，给你不一样的味道。
我的个人微信xjjdog0，欢迎添加好友，进一步交流。



如果你喜欢，不要忘了关我我的公众号《小姐姐味道》哦。