党凡(00:00:01): 然后还有一位同学先就在下，在下下楼。还有谁说要跟我说要讲兰兰老师说他已经这样的学。后面可能就是线上讲的。我先结束之后。我共享屏幕了。等一下我调低一下音量有点吓人。

陈泱宇(00:00:33): 也是能看得到?

党凡(00:01:03): 491。这一定是要多少钱？十万二十万就买这套各位吗，不够吧，但是什么专的只是软装。因为我们这套东西还挺贵的，这套东西。你看这套东西是另外一个人全部卷，但是软装20万是不？毕竟这个不算，还有对这一套怎么也得上万，那你上次你要你也要教师捐赠吗？我不捐。

党凡(00:01:48): 在墙上被挂出来教师军的最近科总是自己蹦出来，我不知道为什么，我明明没有开任何城市，让客户给我谈资。我不知道。你还要重新定义起点吗？还是已经定义过了，因为文化多少，我忘记主播怎么说了。什么重疾病一系列因为从来没有准时开始过，几乎没有准时开始。

党凡(00:02:31): 老天不给我挂了吗？介绍一下。感觉东西越多话就越多。你说是这样，腾讯没有办法直接推优秀！完了。你回头把它挂到博客上去。我知道我的域名解错了，我没有挂的，我家的加宽挂的布拉利。那我待会可能得现场展示一个给阿里的风险，是因为因为我阿里云多今天给我留了余额报警不能给他退到一半。要个钱。

党凡(00:04:36): 完全胜利，连接状态极佳开始。一开始了吗？一开始。要不然先介绍一下，那你先介绍一下，那我先静音对，或者你来我这介绍也行。

刘晓义(00:04:56): 没事，我在我这里我在。

党凡(00:04:56): 没事，我在。

党凡(00:05:11): 我是不是直接这样说话的好，我在耳朵里还能听到我自己说话非常好，但是大家是不是看不到你没有关系，我们把摄像方面你要把摄像头打开。哈，正好我们要开始了。

党凡(00:05:26): 你不开教室?

陈泱宇(00:05:29): 我试下我的声音，没事，你们继续。

党凡(00:05:31): 不开教室。我这不能调音量，得用座位调，可能是音量控制电脑音频无线手指。Ok. 靠这个。我这个主席给我的脸太大了，我要把脸换一，能不能切换到我只能挡iPhone camera 你没有插没有？那就不用了，就这样吧，这个欢迎各位线上线下的同学今天来这个 OS P 的项目介绍会，然后今年就是每年青奥会参加这个由中科院主办的 OPP，然后基本上。形式上是和 google code 是一样的，我们会在通达这边有一些大的成员来做导师，然后去提一些自己的项目，然后希望来参加的同学们也不一定是现在是特大的成员，如果是特大之外的同学们，如果对我们这边的项目感兴趣的话，也可以来参加。今年我们是有两个项目。一个是跨平台的外报密钥管理器，是党团和党团老师和陈胜奇老师的，我相当于出纳的自有项目，这个是我们之前相当于一直在开发的一套。硬件密钥和软件 FC 的这套方案的一个比较自然的一个后继之后，我们有另一个项目是来自 CIY同学的GCC 的前端的一个工作，然后那个非常的专业我可能不太懂，一会有请 COY 老师来亲自来讲就可以了。

党凡(00:07:16): 然后 ok 那我们首先来邀请党凡老师来介绍一下我们这边跨平台外包密钥管理器这个项目，大家好，谢谢喵喵的介绍。我一般看我看网上那个外国人，他读他们都不读，没关系 it doesn't matter ok 好，那今天我来跟大家介绍一下我们这个项目要解决什么问题，要做什么?

党凡(00:07:42): 首先要回答一个问题就是什么是要回答问题，就得从这件事开始说，从这个 passport list 开始说，大家都知道你单纯用密码是不安全的。这是一个只有你你知道的东西，虽然只有你知道，但是别人也能知道，所以现实中很多大家用到的这些服务都会引入这个。其他的因子来认证，比如说在清华，你要登你的 info账号，那你还要扫这个是用微信是吧，用微信扫码或者是这个收一个微信验证码或者怎么着？这是一类，然后第二个大家做的事情是 ok 只有一个你知道的东西可能不够，所以我们可能还需要引入一个你拥有的可信设备。那这个年代带有的可信设备其实越来越多了，不管是大家可能用我的 UBK 也好，或者是你的 iphone 也好，或者一些安卓机也好都行。然后你如果你的这个电脑里有 T PM，那如果你用这个。windows mac 可能都会有比较好的原生的支持，你有一个可信的设备，那这个时候你就能证明你的身份，但这两者加起来有的时候还不够。比如说我拿到了你的手机就能干点坏事，那这个时候可能还需要就是你的生物特

性。所以随着大家意识到这个密码不够安全，口令不够安全之后就渐渐的再往这个 password list 就是我们去认证，不需要。

党凡(00:09:19): 密码这件事情去发展，那时至今日，我们能怎么用这点，那这个 w3c 就提了一个规范叫。

党凡(00:09:29): 它是有前身的等一下下一页我会介绍它的前身那 webauth 是在 w3c 里已经被完全标准化的一个东西。它背后的别的技术等一下我们会提常见的设备，这里列了一些，比如说 UBKK 你的手机，再比如说你如果用这个 windows 也好，或者是 mac 也好。这些系统里会内置一些东西来帮你做这个认证，那具体解释 webson 是什么？之前我们先来看一下它的上一代协议是一个叫做 U2F 的协议，这个不知道有多少人用过。线上的看不到现场。同学们都用过吗?

党凡(00:10:17): U2F不会只有两个人用过吧？用过优秀没有写过优秀没关系，没写过没关系，这个还是坏人，当时他是换了那个 U PK 卖了300块钱换回来的 U PK 卖你300块钱，这个特别坏。No comments. 这里右边屏幕上右边大家看到的这个是我贴了一个 youtube 的链接，大家也可以自己去搜一下那个你搜王康。

党凡(00:10:48): U2F 就能扫这个东西就是某一次的时候康哥也来讲了一下他在阿里做的一个工作吧，那时候是在阿里做。怎么用 UTF 来做钓鱼，那 UTF 是个什么东西，它其实就是我们刚才讲的对 password list 的一个第一次实践。当然 UTF 它叫它的英文叫 universal to a second factor 是一个通用的二因。

党凡(00:11:16): 比如说它是在你的这个密码或者口令之外的1个 factor，一个因子，他干了一件什么事，如果想登一个服务，你除了要向这个 server 提供你的用户名和密码之外，然后这个 server 还会额外的找你的浏览器去发起一次 challenge 浏览器，会把你这个 challenge 给 relay 到你的某一个外部的设备。比如说刚才喵喵提到的 UB 上。用户去摸一下 UBK 之后，UBK 就会去拿这个 challenge 算一个 response，然后发给浏览器，再会把这个东西 relay 给你的服务器就完成了一次 challenge response 过程。那完成了整个流程之后，你就知道 ok 服务器不仅验证了你的用户名和口令，还验证了你拥有的一个可信的设备。这是它的上一代标准 UTS 那到了 final two 之后，大家发现我们还想做更多事情，不仅能够去完成简单的 challenge response。还希望能在设备这一侧做更多的事情，比如说像刚才讲的这个 U2F 里边所有的这些 challenge response 里涉及的这些密钥。

党凡(00:12:28): 原则上这些设备里是可以不存的，那这样的其实就蛮依赖服务器的，并且你只有提供了你的用户名和口令之后，服务器才能把这个 challenge 带着一些跟密钥相关的数据一起一起给过来。那你还是得知道用户名还要输一次密码，这个就体验不好。所以到了这个就是 web 的现在的一个实现里边。那大家说 ok 我想在你的这个令牌里，这个 token 里存更多的东西。我不需要存一些私钥，我还要存你是谁？你访问哪个网站？所以有了整整个这一套流程之后，我们再用这个东西，那你的设备不管是你的 UBKK 或者你的安卓 iphone 手机里边都会额外再存一些数据说，这个用户他的用户 ID 是什么？他访问的是哪个网站？它的用户名是什么，以及它相应的这个私钥是什么？所以有了这一套流程之后，当你去一个陌生的环境，把你的这些安全设备连到电脑上那。就可以自动的完成整套这个 password list 的登录，这是 F two 的协议，它也分两个主要的过程，一个是 make credential 就是去注册的一个过程。另外相应的是，这个 get assertion 是一个登录的过程，可以看到跟刚才讲的 UTF 很像都有这个 challenge 和 response 的过程，所以其实也是一个密码学安全的认证手段。

党凡(00:13:59): 那 webauth 和刚才讲的 UTF 有哪些不一样，主要有这么三个新的特点。第一个特点是它支持助流式密钥在发到2.1里2.0里，他们管它叫 resident key 在2.2里把它又改了名字叫 discoverable credential，意思我可以在你的 BBK 里。手机里存一个密钥的东西，我还可以把每个网站存1个独立的完全不一样的密钥，加上你的用户名用户 I D 一起放进去，所以叫做主流式密钥，那这可以带给我们拿来干什么？一个显而易见的用处是可以拿它去登 SSH。

党凡(00:14:45): 这个时候你就不需要在电脑里放任何的这些公钥了，因为所有要的东西你的 key 里都有，所以只要插上去用一个比较新版本的 SSH 都可以直接去用。这是一个典型的应用，另外就是我们现在见到的所有的这个 password list login 都是通过这个 resident key 或者是 discoverable credential 来做的。第二点是在更新的版本里，这个 f2.1里还支持更完善的管理功能比。比如说网站也好，或者是其他应用也好，可以去枚举你的这个 key 里都有哪些密钥，然后也可以去做相应的管理，另外也可以存储一些元数据。比如说你可以给一个密钥附带的存几十个字节或者几百个字节的一些附加的信息，这就是外包的新特性。

党凡(00:15:35): 那讲了这个新特性，大家可能会问到底怎么管，我这里只列了一个，目前可能是最好用的方法就是你用 chrome 来管。虽然所有的平台都支持 web，但是几乎没有平台支持 web 的管理。目前最简单的方法就是你在 mac 或者 linux 上打开 chrome，插入你的 UBKK 或者任何这样的东西就可以在里面去管管理你的密钥了。windows 不太行？windows 我不知道提群之后能不能用，好像也不行，没试过，

但是 mac 和 linux 是一定能的，所以我们这个项目的目标就来了。

党凡(00:16:15): 我们希望能做一个跨平台的管理工具，帮大家管理手里的这些 web boss 对不起这一页标题里少了根是 web awesome 不是 we b。对不起，我们希望能做一个快平台的工具来帮大家管这个外包怎么做，这个叫依赖于我们的已有的一些基础了。是我们项目前期做的一些东西，这里列出来，线上同学可能看的稍微清楚一点。我一共列了四个开源的项目，第一个是 flatter nfc k，这是利用 flatter 来去读写 NFC 的一个库，然后是我们在这个 data 里提供了 nf 格式的支持，然后是在 data 上去跨平台的支持 C CI D。

党凡(00:17:09): 阿里 C CI D 是个 flat flatter 的插件，大才是弗拉特？标题里写的是 C CI D 是一个 flatter 的插件，稍等一下。C CI D 是一个 f lat 的插件，我们可以用 flat 的程序调用我们做的这个插件去跟所有的标准的 C CI D 协议的读卡器去做通信，然后最后就是我们这次这个项目想要去完善的一个库叫做等一下我来去介绍一下这个库目前已经做了什么。然后其中前两个库是我们前几年 OS PP 的历史项目，我们分别有两年时间，一次完成比较好的去把这个 NFC 的功能做了实现，另一次是把这个 NF 的功能给它加全了。

党凡(00:18:07): 那这个库我们都干了什么？这个发发这个库目前做的功能特别有限，我们只能支持密钥的枚举和删除，但是刚才讲的这个，其实他要做的东西还蛮多的。这里比较直观的两个缺失，一个是我们不支持密钥的注册和验证，然后也没有一个跨平台的工具，所以这就有了我们整个这个项目的题目，我们要做一个跨平台的管理工具。

党凡(00:18:38): 那我们这个项目的目标是什么？大家在 OS PP 的网站应该能看到，我们一共列了四个大点，第一点是要基于我们发出的仓库补充注册和验证流程的完整功能，然后具体的要求就是我们要把 API 设计的足够好。第二是这个测试要做到覆盖率足够高。第二个大项是我们要用 flatter 来开发一个跨平台的 web 管理器能支持我们的列出验证，注册和删除这样的功能。另外我们前期在 flash 上其实做了很多事情，我跟 harry 还有其他的一些朋友，我们做了很多事情，我们发现在 flash 里其实缺了一些组件，一些复杂的密码学运算，所以在这个过程中。我们可能还需要用一些其他的方式，把这些密码学运算也给它引进来，然后第三点是跟所有的这个 SPP 项目一样，我们要交文档，当然最后要把这个项目能够 pull request 的贡献回去，这个时候。

党凡(00:19:43): 我们的 SPP 才能顺利的完成结题 ok 我的介绍就这么多，一共讲了大概不到20分钟，看看线上线下的同学有没有什么问题。现场捐有的好感动！来朋友们没有问题。在 windows 上刚刚提到。

5000收RTXA5000(00:20:11): 你好，张老师好！能听到吗。

党凡(00:20:14): 可以听到。

5000收RTXA5000(00:20:16): 我的问题是，所以这个项目的重点是后端，我们要给这个库加功能，然后我们还要做前端?

党凡(00:20:26): 对我们是两部分，一个是把里缺失的功能尽量的去补齐，然后第二是做一个能跨平台，帮大家管各种 key 的工具。是这样两个目标。

5000收RTXA5000(00:20:36): 我理解了，谢谢!

党凡(00:20:43): 谢谢申奥的提问，刚才喵喵问什么来着，不好意思在 windows 上提到了一个 chrome 支持，这个是 windows 的技术限制，wi ndows 会接管你的所有的这个跟相关的通信就是和 bug 的通信。如果你的程序提全了，你是可以直接通信的，我们这个程序在 windows 上用，那他就必须要先提取才能用。所以找一个本质的技术限制，对不本质，只是 windows 对他的权限管理稍微严格一点，所以我不知道 c hrome 是不是因为这个原因没有做，耽误我没有试过。

党凡(00:21:20): 我不知道 chrome 是不支持照理说是不是 mac 不安全就不用提坏人能干的事情比较有限。坏人能干的事情就是列出你都有哪些 credential 但他并不能对他只能看一眼列出来哪些，但是它并不能真的去添加或者是调用，因为调用的话需要用户提供 ping，并且需要和你的物理设备有一个物理的交互才可以有点担心的是。随机发一下。但是你说实话，就算系统接管的也不是对，就是因为他走的是这个 HID 协议，这我还真不知道不同操作系统里是怎么去处理它的。会不会有抢占的问题，对，但是像你刚才提这个问题 windows 下确实得提权才能用，不然 windows 不让你访问这个设备压根就不让你用。

党凡(00:22:40): 好像那就也暂时没有什么问题，好，那谢谢大家，我结束共享，然后交给陈老师，另一个这个上面就是它可以它不提选是因为加了。

陈泱宇(00:22:49): 好的 ok ok。你们能不能问问题先回答吧！我可以再等一下。

党凡(00:23:04): 好好的对我们可以把那个给删掉，然后他就，但是 mac 上并没有这样的限制。就比如说你通过 C CI D 接口去访问你通过 P C S 是可以随便访问的 linux 也一样，对他只管就是 U 只管设备的权限，他并不管理边具体的功能，windows 是对功能有审计。windows 如果你写一个 PC SC 的程序，你访问别的 AI D 没有问题，但是访问这个发的 AI D 会被 windows 直接拦下来，所以 windows 在 PC SC 这一层还加了额外的审计。ok 好陈老师你可以继续了。

陈泱宇(00:23:43): 那我们开始吧，然后我想了一下我在今天发了一个赛，就是 ccc fortune 的一个 function 的一个功能实现。然后因为我家人在那个法国开了欧洲峰会，然后所以我只能线上给大家介绍一下。都没关系，大家也可以踊跃提问，然后如果说有什么问题的话，也欢迎大家可以直接的打断，然后进行一个详细的问题的提问和解答。

陈泱宇(00:24:11): 然后我们开始吧，就我先想一下，就我是一个人吧，就是我其实是一个三年级的博士生，然后准备做一些软件协同相关的一些设计，然后 CCC 的同时。对我也是一个，然后可能有大概有2000多行的代码，然后有20多个 commit，大概是这样的一个情况，然后还有很多的那种，正在等待那个上游的 review 的在里面。

陈泱宇(00:24:42): ok 然后我们来回到正题就是讲一下这个 function 是什么，就是我们都知道 CPU 有很多的 IC 扩展，比如说我们大家熟悉的可能，比如叉六上面可能。有什么 SSE 有 f2c512等等这些拓展我们的软件就是很难的去直接把这个扩展利用起来就是之前我。也提到过，然后当时讲过一个 I 的这些原理，但是就是我们这里考虑的是一个性能上的问题，比如说我们的程序有比如说像 for 循环这样的东西，然后我们编辑器开了一个 f2或者 s512这样扩展。然后它这个时候就相应的编辑会做1个向量化的一个操作，就是用我们的这种比较高级的性能指令去完成我们的这个 C 一和 C 加加这些代码或者这些代码。

陈泱宇(00:25:37): 我们如果说直接使用编译器的，比如说杠 M CPU 杠 mac 这样的一个参数去告诉编译器，就说我要。让这个产生什么样的一个代码的话，那这个时候会失去跨平台的兼容性的，我们默认参数可能是一个最 base 的一个 IS，比如叉86上面就是一个。

陈泱宇(00:25:58): 就是一个最差不多64可能唯一的一个这样的 IS，然后就是没有什么 x2512这样的东西。包括 arm 上面和上面也是同样的情况，然后这个问题会在上会非常严重，因为这个实在太弱了，然后叉86上面。也会有类似的问题，然后 arm 上面其实也会有类似问题，但 arm 上面我只跑出来的，就是说它的新扩展会有性能下降的情况，然后我们等会说，所以说我们如果是一个 CCR 程序的话，可以通过向左边这样的代码就是我定义一个函数，然后有一个 attribute。然后现在大概它 close 这边我们可以写，就是给这个父母这个函数去生成一个不同的 IC，比如说我现在有 default 就是我 BC 指定 mac 和 MCD 指定的这个。最默认的这个 IC 的一个东西，然后肯定一般来说，我们是用最通用的方式去生成的，然后我们可以把它加上一个 S s12的实线和 x2的实线 x512的直线。然后这样子我们加上了这个参数了以后，我们的右边就会多出来这么几个函数，比如说这个for点 default sx1点 fx two 负点 fx512s，然后还有一个负点点这样的函数。然后这个函数的作用他会在运行时的时候，我 CPU 的支撑来扩展。

陈泱宇(00:27:33): 然后报报完之后，我去把这个相对应的最适合的函数选出来，然后选一个表里面，这样子我们之后去调用这些函数的时候，就是从停止，然后停止相对应的就是这些对应的。不同的这个版本的这个地址，比如说我一个for的函数运行在一个有 x512的 CPU 上面，然后这个函数就会把这个复点 x50f 写进这个表里面，这个负的函数地址。这样我们去扣这个函数的时候，就会去调到这个 s512f 的这个版本上面去，然后从从而就是我们在支持 s512的这个 CPU 上面就有一个比较好的性能。你就是在做一个这样的事情就是我们给一个函数，根据 CPU 所支持的一个学习扩展，然后去生成一个。函数的不同版本。

陈泱宇(00:28:30): 然后讲一下发，发现你的好处，然后刚好今天出了个新闻，就是可以看到右边就是写出要求，然后就是第二的玩家可以剩下45090这个事情。然后这个事情当时我看到的时候觉得非常难，因为这个指令在上面好像是大概是到3D Bridge。左右才开始有的，之前的老师其实是没有这个决定级的，然后我们如果二进制分发的时候，分发一个这样的指令，那么可能就会导致这个。这个程序会退出，就是不支持这个指令。然后如果说我们在分发程序的这么分发的话，很容易会失去作为对老平台的支持。我们如果比如说我们编译的时候，我们就编一个最 base 的，然后我们只对于那些热点，然后同时又对 asa 扩展比较敏，性能比较敏感的这些函数去生成这个 M 的实现，就是说它既能保证的内容性，然后通信能保证性能。

陈泱宇(00:29:45): 然后性能我再提一点就是说，CPU 扩展全开不一定会生成最快的代码，这个事情就你们如果做过一些算子优化或者是辨析优化的同学应该会非常熟悉，就是说因为我们 CPU 上面可能不同的一个。一个指令去扩展，比如说你你 arm 上面你跑一个标量指令和跑一个什么 SM D 指令，可能它的后端会发给不同的流流，不同的这个流水线。然后对线之间，比如说你要转发数据的时候，或者是比如

说你要转发数据的时候，它的这个延迟可能是不同的。你这个 SM D 的后后面那个线没有实现好的话，比如说我可能就是为了去支持这个扩展，继续去做的这个处理器。然后我保证说软件它能跑，但是我没有考虑过这个扩展，它实现的需要性能，它也会导致 CPU 架构其实对于这些扩展时间是并不好的，然后说我编译器如果是生成这些扩展。也会导致一些性能下降的情况，并且这个 arm 上面其实我最最高测到过10%几的一个性能下降，在 arm 上面的一个 ACL 中的核心。然后如果说开了那个不是你是 N 和 V two 上面。就是有一个测试用例开了一个 sve to 的扩展，然后可以损失10%几的性能。所以说我认为说我们去生成一个多函数的版本，然后说甚至细致到说我对 CPU 架构支持什么，再去选择它，这是有必要的。当然这是一个进阶的工作，就是我们现在只是想说把 for 这个功能也给实现了。

陈泱宇(00:31:29): 我们为什么我们要实现 fortune，因为我们的很多科学计算的这种软件或者这里的代码是用风扇写的。fortune 方式一直没有被 CCL LV M 支持。然后性能上还有另外一点，就是说我们一些比较高级的 IS，比如它有的时候引入更高的初始化代价，比如说像和 arm 的 SVE。这些它都有一些初始化的一些函数，初始化一些指令，然后导致说它的在用了这个扩展之后，它的整个指令。就会变得非常复杂，其实本就是说，导致说你本来这个动态指定数就会增加，然后如果说你的这个程序本来可能只讨论一个情况，两个循环的这种程序，你去增加了这些东西的话，反而会引入一个更高的性能开销。所以说我们去预测这个事情是很有必要的，我想我放一个性能数据，这个东西是我最近。就是刚跑出来的结果就是我自己在做一个这方面的科研，然后自己去提了一套，这一就是一个自动化的一个生成的东西。然后跑一个 SpurEngine 2006，然后在一个的板子上面，然后可以看到我们如果只做一部分的函数去。去生成这个方式的话，它的性能是最好的，可以看到就是这个蓝色的这一条是就是我开，就是说我自动化的一生成就是说选择性的去生成多版本。

陈泱宇(00:33:13): 然后讲一下就是这个最底下这个深蓝色的这一条就是一个最 base 的。对来说，然后这个橙色这条就是一个 base asm，然后再加两个扩展。然后绿色的就是一个比较现代的快速扩展的情况。然后他说我们这里统计的 cycle 数，它这个参数反而是最多的是为什么，因为有一些测试用例，比如说这个436的测试用例，它就是开扩展之后，它会生成一些比较差的代码。还有483这也是，所以说我们要知道就是把扩展开前播那个系统是最好的，然后可以看到，像我们如果自动化是上场这个还是多版本的话？它其实就可以，就不要取其精华气糟粕的这种效果，比如436的这个我可以选用最快的版本就是跟最快版本是差不多，甚至还更快一点点。然后这个48483，这也是就是能显出最快版本来，然后可以避免就是把扩展开全性能下降的情况。然后你们看这个456和482这两个测试用例。也可以看出来，把扩展开显性的性能是最好的，然后我们用多端本的方式，你能得到很好的就是 coverage 就是可以看到这里。它的性能跟其实你不相上下，就是把扩展开前面的情况，然后482这个也是。

陈泱宇(00:34:37): 所以说我们就可以看出来，我们做这个不仅可以让我们的 binary 可以有很好的一个兼容性，然后同时也可以。也可以提高我们的程序的性能，对于那些它虽然说函数本身可能会很敏感，但它不是程序的热点的时候，其实我们是没有必要去加上这个东西的。

陈泱宇(00:35:02): 所以说我们可以做一个手动控制，然后讲1下 C 一和 c300我们是怎么用这个功能就是 CCR 里面有个 Attributes 可以控制这个 function march motive version 的行为，1个是 target close，一个是 target。然后是 target version 它 close 就是我们刚才看到的东西，我们可以传入一个列表就是 xx to x512。

陈泱宇(00:35:33): 我刚才好像网断了一下，你们现在现场还好吗？

党凡(00:35:39): 我开了，还好。就转不过来。还好。

陈泱宇(00:35:43): Ok. Ok. Ok. ok 对我看我网断了一下 ok 就反正 take close 大家也看到过了，就刚才我们讲过的这个东西。然后还有这个 target 和 target version，Target. 就是这个20可能是对叉86的，然后如果是 a64和 type 的话，它用的是 target version。它传入的是一个单一的字符串，然后它用是用于指定该函数的目标是用于说我们，比如说一个函数。我们可能对于不同的一个扩展，可能有不同的实现或者具体的一个函数版本的实现，我们可以去用这样的一个东西，然后我们的函数要在。代码里写很多遍，然后 C 语言里面的话，这个功能是有点小问题，就是说它这个 PPT 我可能没有写好。就是在 C 语言里面的话，这个功能它不会生成函数多版本的支持，它说去给我的这个 function 加上一个 mark 表示说我这个 function 在编译的时候要用哪个扩展去生成 function 的代。它是不能处理这个函数重名和一个 dispatch 的 resolve 的功能的，但他是 C 和 CI 都可以使用的。

陈泱宇(00:37:02): 我们想一下就是 fortune 这边我们缺了什么，就首先我自己在做，就是在测这个东西的性能的时候，这里其实有几个测试问题就是 for 串写的。然后我自己注入了一个方程到它的后端，然后我就发现没有人测过这个 fortune 的支持，就是说这是一个里面的一个加上这个东西，它就有 bug。决于一个编译 pass 的顺序的问题。然后这里我不展开讲的是，如果有兴趣的话可以自己去看我的这个链接

里面的我干了什么事情，然后有这个修复，其实可能还是需要做一些的事情的。然后还有我们在整个里面完全缺少了，甚至比如说像 attribute 它给了他的像原代码级的这个描述。这个事情就是我们最关心的问题，就是我们需要把这个 ar 的描述加到我们这个 fortran 的一个前端里面，然后使得我们的这个 fortran 可以去使用。一个我们这样的描述，然后就是加上这个东西到后端，然后后端代码其实都不用改，因为这是后端的那些他就是这个方向的支持已经非常好了。我们就只要在发展前端去干这个事情就可以。

陈泱宇(00:38:24): 所以说想要下这个任务解析，就是说我们要给加上一个发 motion motion，你拿出来表示知识，然后这里我们会遇到一个问题，就是说 fortune 里面我们除了有 function，还有一个东西叫 subroutine。然后就是 sub routine 的话也是相应的去要去增加这个东西，然后其实这里讲一下就是我们已有的工作就是项目导师自己也是我已经完成了一个外部助的 demo，然后已经帮大家踩了大坑，其实可能大家做起来会好很多。然后要提交批到我们的这个仓库里，然后会他单独不让水平给大家提交，然后 page 也要提交给上游，但是上游的合并是我们不能去控制的事情，所以说直接提的话，我们不要求合并，就只要 PR 到这里，然后我认为审核通过就可以了。

陈泱宇(00:39:15): 然后我想一下，一个扩展任务，就是说我们这个任务本身的话，可能就是相对来说比较简单，然后可能会可能就是会踩点线的坑，需要去修。

陈泱宇(00:39:27): 在扩展任务这边的话，我是希望说能参与进来的同学，我们可以探探索更多的一个方向的一个可能性，就是我们刚才讲到这个东西它用的好的话，既能让我 binary 能有更好的兼容。然后同时也能在支持 CQ 上能找到很好的性能，在不支持它的 CPU 上面也可以保持原有的性能甚至。我在支持这些扩展 CPU 上面有时候可能还能做的更好，因为它的扩展之后，代码可能是会变差的，所以说我们是希望一起去发个这样的 paper 然后其实我自己已经有了想法了，然后也。可能快做完了，我可能还需要很多工程上的实现，我想做一些算法上的事情，关于 B 的一个指定分析的一个事情。但是还需要很多一些，比如说编器里面怎么集成这个东西，然后编器里面怎么去我们 CPU 的一个不同架构，然后进行不同的函数，这个事情都是需要很。我们需要一些帮手来一起做，然后如果说大家加入一些的话，不仅可以贡献开源，然后还可以发 paper 然后。然后这里的话我再讲一下，如果是你发 paper 的话，其实跟开源我觉得是不冲突的，就是我自己的计划是就是。

陈泱宇(00:40:52): paper 的东西重点是讲我们怎么分析这个 binary 本身怎么去生成这个还是多版本的这个一个 profile。然后开源这部分就是讲的是我们怎么在编辑器里面提供我们这样的接口，然后使得边就是说这个我们这个 research 的这个东西，然后能跟边性它能接起来，然后使得说边性能直接使用我们输出的这个结果去生成一个方向的这些代码。所以说我然后是扩展任务，这里的话，我们是做这一块的内容，其实不会影响到这个 paper 的问题，然后甚至也可以让这些成果更早的能被这个开源项目所采用，然后真正的更早的利用到这些开源软件中去。

陈泱宇(00:41:36): 然后想一下我们这个预期实验的效果是什么样的，比如说这是一个破产的代码，然后我们希望就是说我们可以通过一个这样的就是一个的。形式去给这个函数加上一个声明，然后从而说让编辑知道我们这个复函数要生成一个 default access to 这两两个 target 的一个代码。然后这是一个简单的函数，然后这你如果有计算的话，就是能看到一个汇编的一个不同的，然后讲一下我们这个项目有什么技能要求？首先我们基础的话可能会包括这个 shell 和 G 我想应该大家都没什么问题。第二部分就是最关键就是 CPC 代码的开发与调试，然后这块就是 CC 内部代码都是用 C 和 CR 写的。

陈泱宇(00:42:33): C 和 C 加加也不用很 modern 的写法，所以说一般来说，我觉得一般就是正常本科上过那 OP 的课程的同学应该来说这些应该都没什么问题。但是你要注意下内存安全问题，就是因为因为这里面很多时候是4月是要换用的，所以说你可能不太好用一些什么智能指针这样的东西，然后像 GDP 和 RR 这两条工具可能大家也是比较需要的，因为我认为就是调兵器的时候，你可能需要用 GDB 去看调用站，你才能知道。就这个函数是什么时候被调用的，它干了什么事情。对于我们快速理解一个很大的项目是很有帮助的，然后这个工具可以做一个代码的 reply。然后比如说我们的时候，我们之前的一个函数的，比如说一些动态的地址，像我们可能是 mail 出来的地址，或者是这种 new 出来的地址，它都可以保持一个跟原本是一样的。这个时候我们把 R 配合 PPT 可以实现一些那个，然后可以很好去调试一些编辑器，什么时候改了一个什么东西这样的事情。当然，我觉得很可能有一部分同学没有用，但是学习可能也只要很快的时间。

陈泱宇(00:43:58): 不要求你虽然会，但是就是你用我们在做项目的时候可以去了解一下，然后其实内部函数其实是一个最头疼的一个问题，但是我认为这个问题。并不大，就是说项目导师就是我，然后 AI 然后第八个都可以帮你，然后经常你可能去就是你被写上注释，然后 C。这样的 AI 就可以帮你自己补出来，你要调什么样的函数，这样子就非常好。然后还要求你们要对这个 fortune 的语言有个基本的了解，就是说你大概知道语言有哪些关键字，然后这关键字是来干什么，然后你能会写简单的语言就可以了。都是都很快的事情都可能最主要

就是说你的 C 和 CR 基础可能要比较好一些。

陈泱宇(00:44:52): 然后讲了关于扩展任务是什么，就是说扩展任务这边的话，我们是希望这里面都是函数，然后我们希望说我们去找到。这样子的函数，同时它又是我们程序的中国热点，然后同时就是对Alpha扩展很敏感的，然后我们对这些函数去生成我们的。然后最后使得我们二次分发的时候可以做到兼容快速全都要，而且甚至还更好，我们可以去避免这个阿尔法扩展最大覆盖率是一种下降的问题。

陈泱宇(00:45:27): 我要讲的就是这么多，现在是一个提问时间，大家有什么问题？现在聊聊天发文字也可以。大家可以先想问题，我可以先给大家看一些一些，就是这个。我这里加个终端。比如说我可以给大家看一下，比如说我这边有一个一点 C 的这个文件，就是说我有一个负这样的函数，然后它在克隆这里会去。形容这个函数的多个版本，我们比如 C编译。1.41开O3优化？我们可以看看它生成什么样的东西，就我们看到这里会生成就是for的函数的不同版本就是这个负点 default sse to fx to 然后。还有 s512版本，这里我们会编上面，我们还要去 rob 我是有哪些破绽，还有。然后去看，比如说 U512的话，然后我就可以去直接的就是把 f2就是填到这个表里面去，然后诸如此类的这样类推。这里的话也是说我的。我稍微声明说这个东西是一个 IF。这个好像不太一样，反正这最后对前面有对方说，就是说我会。相当于说会汇编里面去生成，去声明这个东西之后，然后再像汇编的时候，会去打这个标记，然后就是调用这个函数的时候，它可以选择第一次调用的时候去跑这个也可以在那个 open 的时候调调这个 resolver。

陈泱宇(00:47:34): 然后 for 传这边的话，反正就是要写成一个，刚才我们展示了一个这样的形式，我们支持这个东西的宾语，当然现在肯定是不行的，就是下边他肯定会报一个错误，然后我们要让他们支持这个东西。

陈泱宇(00:47:50): 这么多。大家问题想好了吗?

党凡(00:47:56): 我提一个问题就是我其实很好奇这个 resolve 所以 resolve 的执行刚刚听起来是 DI 的时候执行，所以是需要 LDSO 就是 inter preter 去帮忙是吗?

党凡(00:48:10): 这个东西相对。我们要和。

陈泱宇(00:48:11): 对是它是用 I 方法实现的。

党凡(00:48:14): 采用 iPhone 的时间也不是严格的是和特定的 link 去绑定的?

陈泱宇(00:48:24): 是的。

党凡(00:48:26): ok 还是比较简单的一个东西。那 ok。

陈泱宇(00:48:28): 但要是你用智力是可以的，比如说你用 M OS L 的话就不支持这个没办法。

党凡(00:48:35): 没有办法，然后 resolver 的现在 GCC 里面是有生成 resolver 这些基础设施的是不是因为我感觉是对每一种架构去搞会比较麻烦?

陈泱宇(00:48:37): 对。是的。对。

党凡(00:48:50): 所以说基本上就需要做的事情是 for 这块可能我们要处理一下 A BI 的问题，然后处理一下这相当于是我们能不能去做去动态的去改这个东西，然后它有没有一点相当于。

党凡(00:49:06): 他在说这块是吧，就是其实看。

陈泱宇(00:49:09): 这些说出来，你不用考虑，因为因为这个是后端的事情，就是说我们 PC 里面去做这个事情的时候，它都会有一个中间的一个 GCC 的一个 IR 去表示它。

党凡(00:49:20): 是因为我有点担心的事情。

陈泱宇(00:49:21): 然后所有的 resolver 都是对于每一个 ISA，这样 ISA 它都会有一个就是去生成这样的函数在里面了。

党凡(00:49:31): ok 是因为我其实有点担心，是不是用的一些 API 要求就是不能做这种事。

陈泱宇(00:49:34): 是没有的。

党凡(00:49:40): ok 乘。那我这边没有什么问题了?

陈泱宇(00:49:42): 如果说这块有的话，其实我们也可以去看我们怎么修复这个问题，我觉得本身就是它本身看起来可能比较简单，就是会编辑器的话，可能几天就能搞定，但是你要去处理这些，比如说 A BI 的这种需求。有什么别的坑，就可能会比较麻烦。

陈泱宇(00:50:10): 然后大家还有什么问题吗?

贺泽邦(00:50:15): 老师好，我之前也是在担心这个 A BI 的问题就是之前说 for 可能会有，比如说像是函数签名这种验证。不知道会不会有问题。然后除此之外，除了这个问题之外，就想问一下，就比如说既然我们这个 multi version 的话，我们如何保证，比如说能够在函数里面去进行这个动态的去切换，或者说怎么去知道这个函数在哪里，还是说现在只能是手动的指。

贺泽邦(00:50:45): 什么样的一种方式?

陈泱宇(00:50:47): 这个问题是我自己 research 的这个东西，目前的话我是有一个 profile 的一个方法，然后还有一些事情，但是我 research 的后面的东西我可能现在不好说。

贺泽邦(00:51:00): ok ok 行。

陈泱宇(00:51:02): 对，然后之前说你关于 A BI 那个问题的话，其实我个人可能没有看太多，就是我们可以在一起交流一下，有问题的话。但我认为是说你，你是只考虑在同一个空里面内部的话，其实我觉得这个是都是可以克服的，都是能解决的。

贺泽邦(00:51:27): ok ok 然后对于比如说 CPU 的这种测性检测，然后我们是大概率还是会就是复用这个 LLVM GCC 里面的，而不是就是新增一块这个。

陈泱宇(00:51:38): 用这些里面就可以了，然后这个东西都不需要我们自己写，就是后端这边的话，其实就只要支持这个就行了，后面后端的话其实都已经写好了。

贺泽邦(00:51:49): ok 行，谢谢！谢谢老师!

陈泱宇(00:51:58): 还有问题吗?

党凡(00:52:14): 那可能都暂时没有什么问题了，那感谢各位老师，要不然今天就先到这里结束，后续如果大家还有什么问题，可以在通过官网上列出的导师邮箱给导师们再做后进一步的检查。然后也欢迎大家报名特大理想的这两个项目，那也谢谢大家今天花时间大大成本地。

陈泱宇(00:52:43): 好耶拜拜。