**下面是课下开源开发实践轨道的大纲，供大家参考。实践部分的内容安排会在预选阶段公布给学生，所以是同学们选课的重要依据。**

**统一题目：基于Rust的KVM轻量级重构(开源项目名称：RustKVM)**

**实践部分：腾讯+下一代高可靠轻量级虚拟化开发实践**

1. 开发任务介绍（背景及意义）

Rust作为内存安全语言已经被业界广泛认可。但Rust语言不只是追求安全，而且在并发和性能方面也有很好的表现，也因此Rust不仅被很多大公司如AWS，Windows，Google等用于应用软件的开发，而且其已经用于构建操作系统，比如Redox OS。Rust也逐渐成为未来底层软件开发的首选语言。目前Rust已经被Linux内核开源社区接受作为开发语言，主要用于Linux内核驱动模块的开发。

Linux是当下最主流的操作系统之一，覆盖了绝大部分的服务器和终端。Linux开创了现代开源的的先河，Linux社区也是当今世界最大的开源社区，聚集了全球顶尖的Hacker和公司，也是最活跃、最具影响力的开源社区。另一方面，Linux内核也蕴藏了巨大的技术能量和宝藏，是技术极客的展示自己的最佳舞台，可以帮助小小程序员实现“改变世界”的大大梦想。

虚拟化技术是云计算的基础，目前大量的公有云以及私有云环境都在使用KVM。KVM作为Type 2型的系统级硬件辅助虚拟化技术（相对于Xen）的典型方案，其有如下的优势：

1. KVM作为一个Linux内核模块方式实现，因此其硬件生态兼容性，可扩展性较好。
2. KVM中没有实现调度等功能，而是依赖Linux 内核的实现（如Linux的调度器），实现充分利用了Linux内核的成熟模块，实现相对精简。

但是随着云原生，serverless等新的场景的出现，KVM现有的实现就显得复杂了。新的场景趋向于以容器为中心，虚拟化技术在新的场景中主要的作用是弥补传统容器技术的隔离性不足的问题。因此对于KVM技术有进一步轻量化的需求。

本课程实践中使用Rust来重构轻量级KVM（Linux内核态），一方面是要充分利用Rust语言的特性提升KVM的健壮性，另一方面是希望轻量化KVM的实现。

1. 任务培养目标
2. 了解虚拟化技术的基础知识。
3. 掌握Rust语言在Linux环境(包括内核)下的编程。
4. 完成Rust KVM的原型构建。
5. 拟培养人数

7-8人

1. 团队导师介绍

蒋彪 腾讯云专家工程师，OpenCloudOS 开源社区技术负责人。操作系统和内核领域10+年老兵，Linux社区资深发烧友和committer。目前从事腾讯云操作系统和OpenCloudOS的研发工作。

联系方式：[benbjiang@tencent.com](mailto:benbjiang@tencent.com)

电话：18780152587

彭浩 腾讯云专家工程师，在Linux内核以及虚拟化领域有10年+的工作经验，主要从事kvm虚拟化的开发，性能优化以及fuchsia 微内核的虚拟化开发工作。在Linux社区以及fuchsia社区都有不少补丁贡献。目前从事腾讯云操作系统和OpenCloudOS的研发工作。

联系方式：[flyingpeng@tencent.com](mailto:flyingpeng@tencent.com)

电话：18123274005

潘睿 腾讯云专家工程师，深耕腾讯云底层技术架构多年，有着丰富的内核trace经验与性能优化实战经验，目前负责腾讯云qemu/kvm技术研发项目。

联系方式：[ruippan@tencent.com](mailto:ruippan@tencent.com)

电话：13684027112

1. 课程计划（包括16周的时间安排、课程具体内容、阶段性培养目标）

|  |  |
| --- | --- |
| 时间 | 内容 |
| 第一周 | 1.根据学习资料初步了解虚拟化以及rust语言。  2. 3课时面授：  OpenCloudOS社区介绍&Linux内核基础  KVM虚拟化基础；  Linux下Rust编程基础；  Kvm重构任务详述; |
| 第二周—第三周 | Linux内核基础编程实践  任务1：完成Linux内核的源码构建  任务2：简单的内核模块编写，并完成模块构建、安装、使用和卸载  Rust系统编程实践  任务1: rust包管理方式构建工程  任务 2: rust unsafe 编程  任务 3: rust mini OS 编程 |
| 第四周--第五周 | Rust在Linux内核下的编程  任务 1:如何在Linux中加入rust模块  任务 2:用rust语言实现一个简单的驱动 |
| 第六周—第十四周 | Rust重构kvm  1.构建kvm 主体框架  2.构建kvm vm接口  3.构建kvm vcpu 接口  4.构建kvm mmu 实现 |
| 第十五周--第十六周 | 联合调试，输出文档 |

1. 评分标准（考核方式，评分细则）
2. 开源代码贡献 （70%）
3. 课程/讨论参与度 （30%）
4. 课程资源（团队可以提供的资源，可选）
5. <https://github.com/OpenCloudOS> 开源社区
6. 腾讯云提供黑石服务器（物理服务器）

实践大纲的补充建议：

* **实践内容，需要安排项目架构解读和源代码阅读（提前公布预习内容）**
* 设置项目实践预习内容，让同学们预先了解项目实践可选择的内容，以便后期选择具体的项目实践任务
* 利用1-2节课时间对实践任务做更加清晰的剖析，对同学们后期实践有帮助
* 可以在实践里加入答题、互动加分环节，调动同学们的积极性
* 可以在上述大纲中加入对同学们选课基础的说明（注意如果太严苛有可能导致团队选择的人数过少）
* 可以在上述相对固定的任务和导师团队的基础上，加入自选题环节，导师可以社区内流动化指导，体现开源优势

实践实施说明和建议：

* 实践部分的授课可以采取线上或者线下两种形式，每周固定授课时间在2小时以上，平时尽可能对同学们的问题及时反馈；
* 在课程预选或者初始阶段尽早进行课下接触可以增加团队凝聚力
* 在实践课的时候录屏，以供同学们反复学习