



开放原子开源基金会
OPENATOM FOUNDATION

openKYLIN

技术全景案例集

(2024)



OpenAtom openKylin社区

引言



数字经济已成为全球经济发展主引擎，操作系统作为数字基础设施的底座及灵魂，成为推动产业数字化、智能化发展的核心力量。在新一轮产业革命背景下，万物互联、数据驱动、软件定义、平台支撑、智能主导的发展趋势日益凸显，操作系统作为重构IT信息系统的底座及灵魂，将被重新定义。在这场变革中，我们不再是旁观者、见证者，而是这场伟大变革的参与者、实践者。

在软件定义未来世界，开源决定软件未来的当下，OpenAtom openKylin开源操作系统根社区以“共创”为核心，在开源、自愿、平等、协作的基础上，通过开源、开放的方式与企业、高校等伙伴构建合作伙伴生态体系，联合产业优势力量共同推动Linux开源技术及其软硬件生态的繁荣发展。

《OpenAtom openKylin技术全景案例集》（以下简称：“案例集”）是由openKylin社区牵头编撰，众多产业领域优势企业、知名院校、杰出开发者共同参与，汇聚开源操作系统智慧成果的结晶。该案例集每年发布一次，持续收录openKylin社区优秀技术创新项目、行业应用场景、生态适配成果案例、用户使用案例等，致力于为产业生态从业者、Linux开源爱好者、开源操作系统关注者提供参考和借鉴，欢迎更多伙伴一起集智众创！

目录

引言	01
目录	02
1 社区简介	05
1.1 社区介绍	06
1.2 社区发展历程	06
1.3 组织架构	07
1.4 版本管理	08
1.5 上下游贡献	08
2 社区共建	09
2.1 企业会员	10
2.2 SIG组	10
2.3 高校站	10
2.4 用户组	10
2.5 个人爱好者	10
3 根社区基础能力	11
3.1 核心组件选型及维护	12
3.2 “可控开源”体系	14
4 技术创新项目	15
4.1 底层技术创新	16
4.1.1 “分级冻结”机制	16
4.1.2 VirtIO-GPU硬件视频加速框架	16
4.1.3 VirtIO-GPU AV1解码功能	18
4.1.4 不可变系统	20
4.1.5 GB18030-2022支持	21
4.2 桌面场景创新	22
4.2.1 UKUI桌面环境	22
4.2.2 wlcom合成器	25
4.2.3 虚拟键盘	26
4.2.4 麒麟输入法框架及OK输入法	27
4.2.5 麒麟极点五笔输入法	28
4.2.6 多端协同	29
4.2.7 系统多语言支持	29
4.2.8 奥云蒙古文智能输入法	30
4.3 生态技术创新	31
4.3.1 RVtrans二级制翻译技术	31
4.3.2 KARE生态兼容技术	32
4.3.3 开明软件包格式	32

目录



4.3.4 openKylin Wine助手	34
4.4 开发工具创新	35
4.4.1 KylinCode集成开发环境	35
4.4.2 openSDK	37
4.4.3 青霜Web框架	38
4.4.4 UraSDK	39
4.4.5 嵌入式版本构建工具	40
4.4.6 RISC-V统一镜像烧录工具	41
4.5 安全能力创新	43
4.5.1 Genmai	43
4.5.2 火焰卫士	43
4.6 智能融合创新	45
4.6.1 语音助手	45
4.6.2 桌面AI大模型插件	45
4.6.3 AI框架安装助手	46
4.6.4 麒麟AI模型管理工具	47
4.6.5 openKylin AI SDK	48
4.6.6 麒麟AI助手	49
4.6.7 其他实用AI功能	50
4.6.8 启元九格大模型	51
5 行业生态适配案例	52
5.1 整机适配	53
5.1.1 ROMA笔记本适配	53
5.1.2 如意BOOK笔记本适配	53
5.1.3 MuseBOOK笔记本适配	54
5.1.4 联想开天P90z G1t台式机	54
5.2 处理器平台适配	55
5.2.1 兆芯KX-6000、KX-6000G系列处理器适配	55
5.2.2 海光C86-4G系列处理器	55
5.2.3 Intel 最新系列平台适配	56
5.2.4 飞腾D3000处理器适配	56
5.2.5 龙芯3A6000处理器适配	57
5.2.6 此芯科技P1处理器适配	57
5.2.7 中科通量RV64G架构DW1000处理器适配	57
5.3 国产显卡适配	58
5.3.1 景美JM9100显卡适配	58
5.3.2 格兰菲Glenfly Arise 1系列显卡适配	58

目录

5.4 生态软件适配	59
5.4.1 搜狗输入法NG麒麟桌面版适配	59
5.4.2 亿图软件适配	60
5.4.3 红莲花安全浏览器适配	60
5.4.4 讯飞星火大模型适配	60
6 社区爱好者构建	62
6.1 Cutefish桌面环境移植适配	63
6.2 KernelBuilder项目	63
7 基础设施平台建设	64
7.1 基础服务平台	65
7.2 一站式编译构建平台	66
7.3 开源合规管理平台	68
7.4 学习成长平台	69
编后语	71
致谢	72
商标声明	73
免责声明	73

01

社区简介

1.1 社区介绍

1.2 社区发展历程

1.3 组织架构

1.4 版本管理

1.5 上下游贡献

1.1 社区介绍

OpenAtom openKylin（简称“openKylin”）是由开放原子开源基金会孵化及运营的开源项目，由基础软硬件企业、非营利性组织、社团组织、高等院校、科研机构和个人开发者共同创立。社区以“为世界提供与人工智能技术深度融合的开源操作系统”为愿景，旨在于开源、自愿、平等、协作的基础上，共同打造全球领先的智能桌面开源操作系统根社区，推动Linux开源技术及其软硬件生态繁荣发展。

1.2 社区发展历程

2020年

注册 openKylin 商标。

2021年

起草社区章程，确定社区理念；
成立首个 SIG 组、成立咨询委员会。

2022年

成立理事会并召开首届理事会会议；
社区正式成立；
央视报道openKylin，桌面根社区
登录微博热搜；
0.7、0.7.5、0.9版本发布。

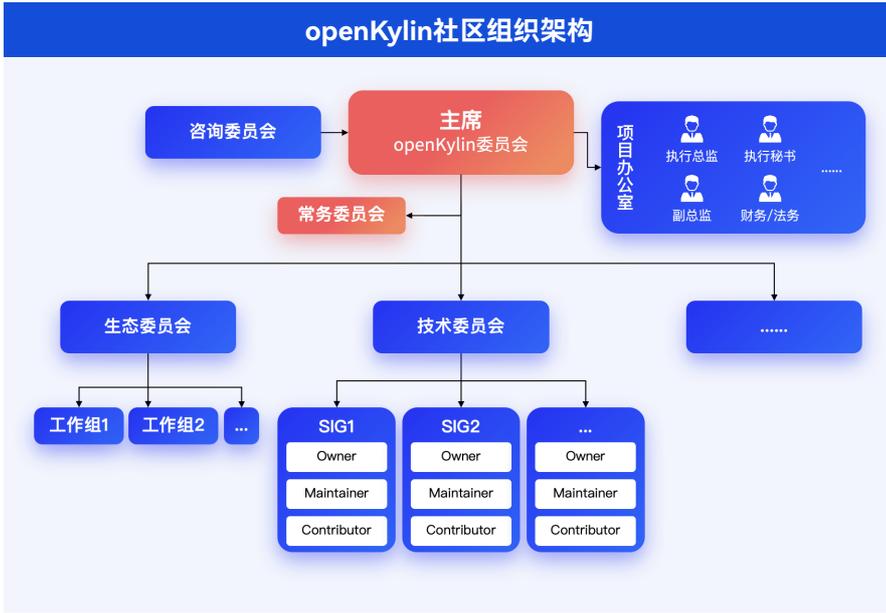
2024年

入选国资委“2023年度央企十大国之重器”；
捐赠给开放原子开源基金会；
再获央视报道；
openKylin 2.0版本发布；
社区会员突破800家。

2023年

召开首届咨询委员会会议；
“openKylin 1.0”正式发布；
第二次获得央视报道；
标志着我国具有系统组件自主选型、
操作系统独立构建的能力。

1.3 组织架构



职责说明

- **openKylin项目工作委员会**（简称“openKylin委员会”）为社区的最高决策机构，负责社区的重大决策；
- **咨询委员会**，负责社区技术发展和战略方向的指导；
- **技术委员会**，负责社区技术决策和技术创新等工作；
- **生态委员会**，负责openKylin品牌宣传和生态拓展等工作；
- **项目办公室**，负责执行openKylin委员会的各项决议，并负责社区例行运作和管理。



1.4 版本管理

openKylin开源操作系统根社区每年都会发布一个正式版本，将社区众多SIG组的技术创新成果持续地合并到openKylin版本中去。通过这种持续集成创新的方式，可以帮助新技术或者新项目在openKylin社区快速孵化、成熟。同时，这种开源、开放的开发模式也吸引了大量爱好者参与进来，提出建议或者问题，促进我们的开源项目更好地发展，更符合用户的需求。

openKylin社区目前最新稳定版为2.0 SP1版本，提供X86、ARM、RISC-V、LoongArch四个架构的版本镜像，支持Intel、AMD、海光、兆芯、飞腾、龙芯等最新主流CPU型号，以及格兰菲Arise 1、景美JM9100、芯动风华2号等多款国产GPU。同时还支持DC ROMA、Muse Book、如意Book、VisionFive2、LicheePi4A、HiFive、SG2042 EVB、Louts2等RISC-V开发板和Raspberry Pi、Cool Pi、Chillie PI、Phytium Pi等ARM开发板。openKylin版本生命周期管理策略如下：

创新版本	LTS版本
每年发布1个版本， 提供1年被动更新支持	每3年发布1个版本， 提供1年主动更新支持 + 1年被动更新支持

1.5 上下游贡献

openKylin致力于打造全球领先的智能桌面开源操作系统根社区，积极参与上游社区贡献，如InputMethod SIG组与Fcitx社区合作，为Fcitx5框架贡献了虚拟键盘支持的代码；Virtualization SIG组为Virtio-GPU方案贡献了硬件视频编码功能，可使虚拟机借助物理机侧的显卡进行硬件视频编码，大幅提高虚拟机的视频编码性能，相关代码已贡献到Virglrenderer、Mesa、qemu等上游开源项目。

同时，openKylin社区在国内外多个主流开源社区扮演重要角色，其中，openKylin社区的明星开源项目UKUI桌面环境已导入Debian、Ubuntu、openEuler等多个主流Linux发行版社区，在国内外拥有众多用户和爱好者。

02

社区共建

2.1 企业会员

2.2 SIG组

2.3 高校站

2.4 用户组

2.5 个人爱好者

2.1 社区会员

openKylin社区以“为世界提供与人工智能技术深度融合的开源操作系统”为愿景，并推动国产操作系统产业生态健康发展。操作系统的生态构建并非一日之功、一家之力，需要通过社区合作、共创共享的开源治理模式，联合产业上下游生态伙伴力量，共创可持续发展的操作系统产业生态。诚邀广大生态合作伙伴加入openKylin社区，携手共建，推动国产操作系统产业生态繁荣发展。

目前，850多家操作系统产业链生态伙伴已加入社区，涵盖了基础硬件、基础软件、应用软件、安全、AI等全产业链领域，并在技术创新、生态共建、社区推广等方面与社区开展了多维度合作。

2.2 SIG组

openKylin社区目前已组建125个SIG组，涵盖内核/硬件/架构/驱动、基础组件、桌面环境/设计、开发框架、系统安全、人工智能等多个技术方向。截至2024年12月，社区各个SIG组累计成员数达到700+，累计创建仓库3800+，累计提交PR14600+，累计提交issue14500+，累计召开SIG组公开会议260多次。

2.3 高校站

openKylin高校站是openKylin社区在高校建立的合作站点，通过推进高校开源技术的产学研融合，构建起学生Linux的基础知识架构，再通过开源活动+项目实践的方式，为学生积累实践经验，并对职业规划等方面进行详细讲解，通过理论+实践的形式，培养卓越创新能力的开源人才。

截至目前，已有北京大学、上海交通大学、南京大学、北京航空航天大学、东南大学、天津大学、南开大学、西安电子科技大学等101所985、双一流、普通本科和头部职业院校加入openKylin社区并建立高校站，与社区开展了多种形式的合作。未来，openKylin欢迎更多高校合作伙伴加入，一起建立产学研融合的开源创新人才培养体系，为实现国内开源事业可持续发展蓄势储能。

2.4 用户组

openKylin用户组，简称OKUG，是为方便区域开源爱好者交流openKylin系统版本及用户体验、Linux技术及生态建设，探讨开源操作系统产业趋势及开源技术贡献而成立的区域型城市组织，定期或不定期举办线上及线下交流活动，提供更加丰富多样的社区参与体验及开源实践机会，凝聚各个地区的openKylin社区开源爱好者，推动本地开源项目的合作等。截止目前，openKylin已建立30个用户组，其中14个为国际用户组、16个为国内用户组。

2.5 个人爱好者

openKylin社区一直以来积极鼓励个人爱好者的参与和贡献，他们在社区的发展中起到了重要的作用。社区目前120多个SIG组中有15%是由个人爱好者成立的，他们是学生、公司职员、自由职业者等等，在社区中参与文档编写审核、协助收集用户的问题反馈、负责桌面环境移植、发掘和修复安全漏洞、孵化创新项目等多个方面的贡献，为openKylin社区版本的改进提供了很多有益的帮助。

此外，个人爱好者还在社区的宣传和推广方面作出了贡献。他们通过自己的渠道和方式，积极传播openKylin系统的优势和特点，吸引更多的人加入社区，扩大了openKylin的用户群体和影响力。

03

根社区基础能力

3.1 核心组件选型及维护

3.2 “可控开源”体系

3.1 核心组件选型及维护

背景概述

众所周知，开源操作系统是由众多开源软件构建而成，而核心组件选型可以简单理解为选择一些关键软件（如 kernel、glibc等）的某个版本来作为根基构建我们的操作系统。因此核心组件自主选型及维护对操作系统根社区来说是一项基础且重要的工作。

选型策略介绍

开源软件引入策略

从合规性、兼容性、重要程度、活跃度、质量、安全性以下六个维度评估判断是否引入该软件到openKylin社区。具体评估依据如下：

评估项/级别	A级	B级	C级	D级
合规性	项目完全符合开源协议	项目剔除极少部分内容后，可符合开源协议	项目存在开源风险	项目不符合开源协议，或者与已有的开源协议不兼容
兼容性	与当前版本完全兼容	与当前版本部分基础库存在不兼容的情况，可通过少量修改进行移植	与当前版本部分基础库存在不兼容的情况，需要大量移植工作	完全不兼容且需要的移植工作量过大
重要程度	属于系统常用功能	属于可扩展增强功能	与已有功能重复且可能存在冲突	与已有功能不兼容或者会破坏已有功能
活跃度	非常活跃，每日代码更新，bug修复及时，不断推出新功能	比较活跃，重要bug修复及时，保持每周都有新的代码，6个月内至少1次小版本升级	维护阶段，重要bug或者反馈2周内无开发者响应，遗留大量一般bug	无人维护，bug和反馈1个月内无人响应，遗留大量bug
质量	开发团队规模大，软件已经存在多个开源社区，功能稳定，质量有保证	软件已经在某个社区存在多年，功能基本稳定，质量有一定保证；全新软件，但代码质量高、文档完善，经过了各项ci检测，功能基本稳定	代码质量不高，缺乏相关文档，有功能缺陷	有较大功能缺陷，无法正常运行
安全性	无已知安全漏洞，且新发现漏洞一周内可修复	少量已知安全漏洞，且新发现漏洞一月内可修复	已知安全漏洞较多，且新发现漏洞三个月内无响应	大量已知安全漏洞，且新发现漏洞半年内无响应

- 持续引入全“A”软件（或版本）；
- 鼓励引入仅含“A”和“B”的软件（或版本）；
- 谨慎引入含“C”的软件（或版本）；
- 拒绝引入/及时删除含“D”或者“C”超过3个的软件（或版本）。

3 根社区基础能力

开源软件版本选定策略

对于新引入的软件项目或者已引入的软件项目，项目维护者会在每个openKylin主版本开发阶段前期预选该软件的引入版本。软件预选版本时遵循以下四点原则：

- 及时跟进上游社区动态，优先选择开源组件当前稳定分支的最新版本；
- 兼顾生态，软件版本自主选型的同时，考虑国内外主流桌面发行版创新版本的选型策略；
- 兼顾稳定，软件版本选型时需考虑和其他软件的兼容性，如需其他组件配合升级的，应及时沟通，统筹规划，避免后期导致严重问题影响交付；具体见第4部分；
- 软件版本一旦选定后，若无特殊情况，在openKylin主版本的生命周期内不再做大版本变动，仅采取特性、补丁回合方式解决质量问题。

开源软件分级及兼容性原则

软件级别的划定主要根据该软件的在openKylin操作系统中的影响域，具体见下表：

级别	范围	兼容性原则
一级（核心） 影响域：整个系统	kernel、glibc、gcc、zlib等	API和ABI在主版本的生命周期范围内保持稳定，并且在接下来的一个主版本中也尽量保持稳定
二级（重要） 影响域：系统的某个重要功能模块，如网络	dbus、openssl、bzip2、network-manager等	API和ABI在单个主版本的生命周期内保持稳定，依赖其的应用程序在单个主版本生命周期内无需重大修改
三级（普通） 影响域：软件本身	其他软件	API和ABI在单个主版本的生命周期内不强制保持稳定，存在依赖关系的应用程序保持联动升级

应用场景

作为openKylin社区基础库选型工作规范，指导社区开发者在满足合规性、兼容性、重要程度、活跃度、质量、安全性的要求下，完成基础库选型及后续维护工作。

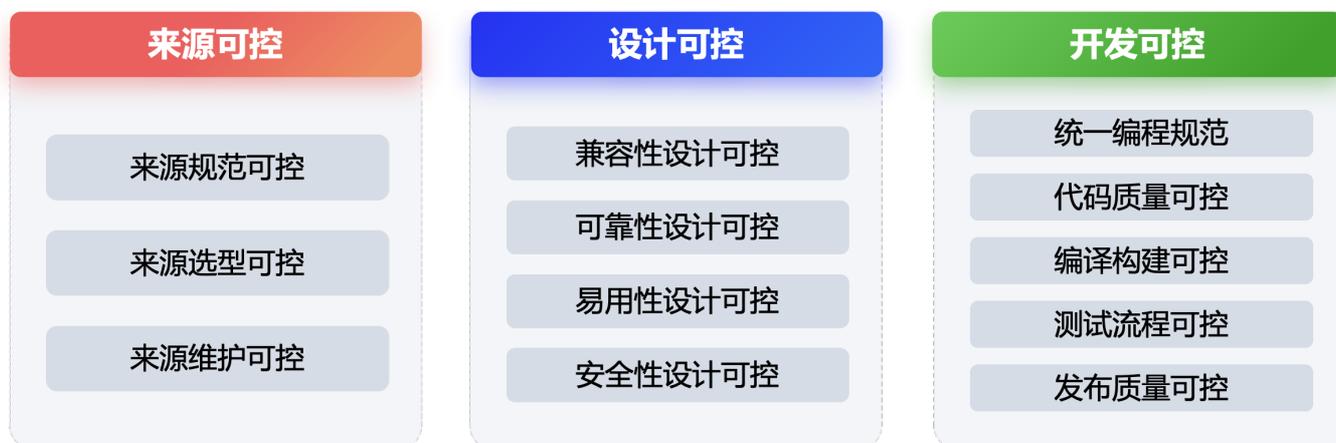
3.2 “可控开源”体系

背景概述

以开源软件为基础构建信息系统成为主流趋势，开源软件存在迭代快、安全开发机制欠缺、维护人员不足等现状，导致全球开源安全事件频发，威胁着使用者信息安全，也带来了隐私信息泄露的风险。因此，开源安全风险已成为全球化挑战，是开源项目首要关注的风险点。而要降低开源风险，保障开源安全，就要求开源社区必须做好开源的安全机制。为此，openKylin社区推出了“可控开源”体系，为开源安全保驾护航。

“可控开源”的特性

“可控开源”体系从代码的来源、设计和开发三个维度，围绕代码流通的全链路进行安全管理。



来源可控

从软件的源头出发，包括了来源选型规范、来源选型可靠性检测、合规性检测、稳定性检测、安全性检测、可维护性检测等方面，确保引入的开源组件来源清晰、透明、合规、安全和可靠。

设计可控

在开源软件设计阶段对兼容性、可靠性、易用性、安全性等方面的设计进行约束和规范，确保开源软件在兼容性、可靠性、易用性、安全性方面达到一定水平并拥有较高的统一性。

开发可控

提供统一的编程规范、开发合规检测、代码质量检测、编译构建检测、测试发布过程检测等机制，减少开源软件在开发过程中的潜在风险，提升开源软件产品质量，保障开源软件的稳定性和安全性。

应用场景

面对日益严峻的开源安全挑战，“可控开源”体系可以让开发者能够更安全、更高效地在openKylin社区平台上开展研发工作。接下来openKylin社区将持续建设完善“可控开源”体系，助推国内开源领域迈向安全创新的新阶段。

04

技术创新项目

4.1 底层技术创新

4.2 桌面场景创新

4.3 生态技术创新

4.4 开发工具创新

4.5 安全能力创新

4.6 智能融合创新

4.1 底层技术创新

4.1.1 “分级冻结”机制

背景概述

在传统Linux操作系统中，应用进程的生命周期主要由应用本身直接控制。然而，系统资源（包括CPU、I/O、存储等）是有限的，当我们运行大量的I/O密集型或CPU密集型应用时，经常会出现系统卡顿甚至“假死机”的现象，这大大影响了用户的操作体验。为此，我们在openKylin操作系统中设计了“分级冻结”机制，用以管控应用生命周期。

功能介绍

“分级冻结”机制目前由openKylin社区UKUI SIG组成员负责维护。“分级冻结”机制会根据应用的不同状态进行分级处理，对不同状态下的应用进行CPU、内存和磁盘I/O等资源限制，以保障系统的稳定性和当前使用应用的流畅性。

- **焦点应用**：表示当前正在操作的应用，一般情况下，不对其做资源限制，充分保证其流畅度；
- **前台应用**：只存在PC模式下的状态，表示当前运行的未最小化但没有获得焦点的应用；
- **后台应用**：PC模式下指窗口最小化状态且持续时间 $\leq 30\text{min}$ 或者前台应用持续时间 $\geq 30\text{min}$ 的应用；平板模式下指非当前打开且持续时间 $\leq 30\text{min}$ 的应用进程；
- **缓存应用**：指处于不可见状态且持续时间 $> 30\text{min}$ 的应用；
- **休眠应用**：被冻结并且放入swap分区的应用。

应用场景

在系统资源紧张，比如运行大量的I/O密集型或CPU密集型应用时，分级冻结能够保证系统组件、重要应用和当前用户正在操作的应用的资源分配，保障其流畅度；当想要提升电池的续航时间时，可以打开冰冻模式，进一步降低系统功耗；当系统可用内存不足时，分级冻结能够主动回收长时间不操作的应用，释放更多的系统可用内存。

项目地址

“分级冻结”项目地址为：<https://gitee.com/openKylin/kylin-process-manager>

4.1.2 VirtIO-GPU硬件视频加速框架

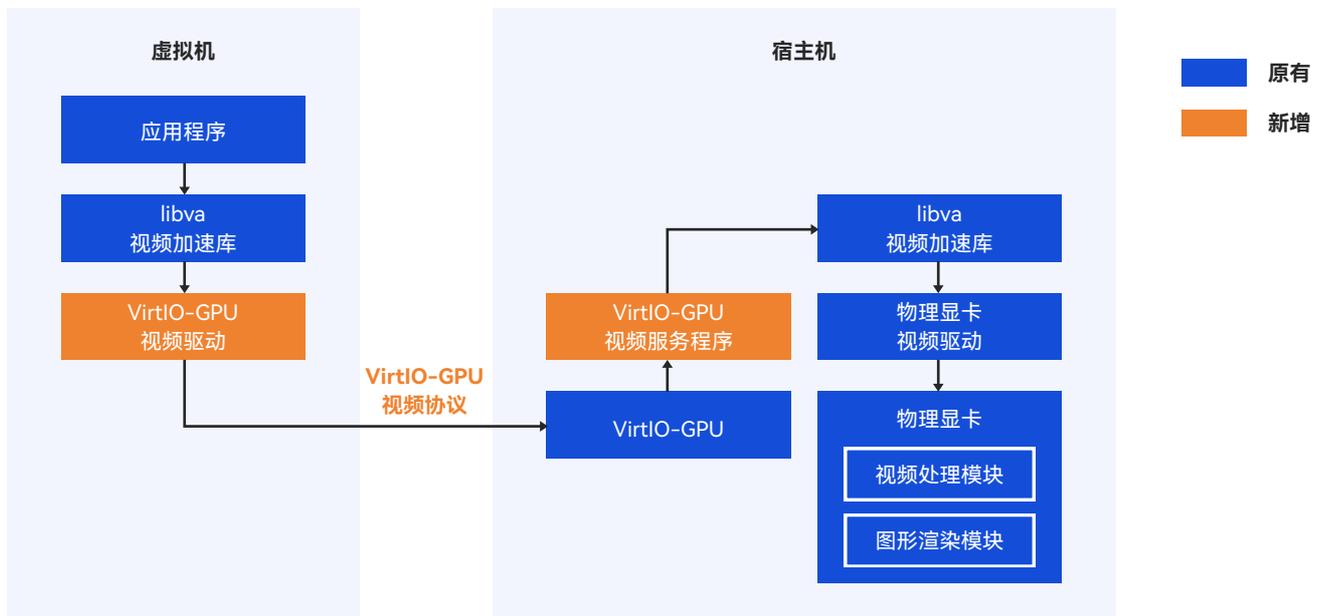
背景概述

传统虚拟GPU优化方案采用了基于API转发的VirtIO-GPU虚拟化技术，并借助virglrenderer组件实现3D硬件加速。这种方式虽然可以大幅地提升虚拟机的图形性能和用户在网页浏览、文件办公和游戏等场景下使用体验，但是虚拟GPU不支持硬件解码的弊端却导致软件解码CPU占用率过高、画面不流畅，甚至丢帧等现象。

为此，openKylin社区Virtualization SIG组为VirtIO-GPU创建了一套采用前后端架构的硬件视频加速机制，为其增加硬件编解码功能。

功能介绍

VirtIO-GPU硬件视频加速框架前端为“VirtIO-GPU视频驱动”，后端为“VirtIO-GPU视频服务程序”。前后端之间采用“VirtIO-GPU视频协议”进行通信，该协议主要定义了编解码相关的一些命令，如创建编解码器、创建视频缓冲区、解码比特流等。



目前，Virtio-GPU硬件视频加速框架已支持H.264和H.265视频规范的解码和编码功能（仅支持YUV420图像格式），暂不支持VP8/9等视频规范：

	H.264	H.265	VP8	VP9	AV1
解码功能	✓	✓	待定	待定	待定
编码功能	✓	✓	待定	待定	待定

对于解码功能，Virglrenderer已经集成在了0.10.1及以后的版本中，而Mesa预计在正式版本22.3.0中集成。对于编码功能，Virglrenderer预计在0.10.4中集成，Mesa预计在23.0.0中集成。具体如下：

	解码功能	编码功能
Virglrenderer	支持版本：≥0.10.1	预计版本：0.10.4
Mesa	支持版本：≥mesa-22.3.0-rc1	预计版本：23.0.0

应用场景

通过为VirtIO-GPU增加硬件编解码功能，可使虚拟机借助物理机侧的显卡进行硬件视频编码，大幅提高虚拟机的视频编码性能，从而提升用户在视频直播、视频聊天和视频剪辑等场景中的使用体验。

项目地址

openKylin社区Virtualization SIG: <https://gitee.com/openkylin/community/tree/master/sig/Virtualization>

Virglrenderer项目补丁: https://gitlab.freedesktop.org/virgl/virglrenderer/-/merge_requests/940

Mesa项目补丁: https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/18831

4.1.3 VirtIO-GPU AV1解码功能

背景概述

AV1是下一代开源的、版权免费视频压缩格式，由开放媒体联盟（AOMedia）于2018年初联合开发并最终定稿。AV1支持高清、4K和8K等视频，在相同的比特率下比VP9、H.265等有更好的视频质量，或在较低的比特率下有类似的视频质量。由于AV1良好的性能以及背后厂家的支持，AV1已经逐渐流行起来。目前，谷歌、网飞等厂商都已经采用，各显卡制造商也陆续增加了对AV1的支持，常见的支持AV1规范的显卡有AMD Radeon RX 6600以及NVIDIA GeForce RTX 3060等。而openKylin社区在Virtualization SIG团队成员的努力研发下，成功为VirtIO-GPU增添了AV1解码功能，并受到了国际媒体Phoronix的追踪报道。

The screenshot shows a Phoronix article page. The main content includes the title, author (Michael Larabel), and a summary of the AV1 video decoding feature. A terminal window screenshot shows the command to run a video file and the resulting output, including the video title and supported profiles. The article also mentions that the feature will be in Q4 as the next major feature series for this open-source GPU driver stack.

phoronix

ARTICLES & REVIEWS NEWS ARCHIVE FORUMS PREMIUM CONTACT CATEGORIES

ENHANCED BY Google Search

AV1 Video Decoding Added To Mesa's VirGL For QEMU/KVM Guests

Written by Michael Larabel in Mesa on 10 August 2023 at 06:24 AM EDT. 6 Comments

 Merged for Mesa 23.3 is support for the VirGL code to handle accelerated AV1 video decoding within guest virtual machines.

Thanks to this code merged into the Virglrenderer and this now-merged Mesa code, there is support for AV1 video decoding that can be enjoyed by QEMU/KVM guests -- assuming the host GPU is capable of AV1 decode.

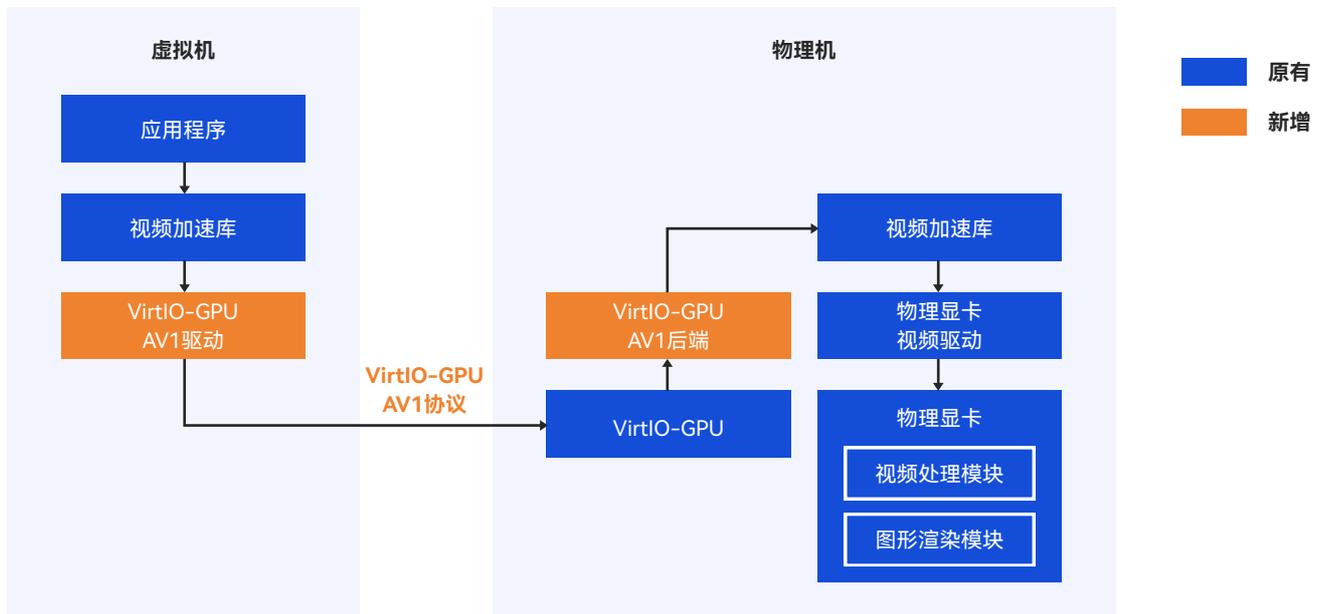
Feng Jiang with China's Kylin OS project worked on this AV1 decode hook-up and shared this screenshot of the Blender film Big Buck Bunny running successfully with AV1 decode inside a QEMU/KVM guest:

```

feng@fedora:~$ mpv --hwdec=vaapi /avi/Big_Buck_Bunny_Sunflower_version.mp4
[+] Video --fd=1 (+) (av1 1920x1080 30.000fps)
File tags:
Artist: Blender Foundation 2188, Janus Bager Kruse,
Commit: Creative Commons Attribution 3.0 - Int'l
Composer: Sacha Bondegeburu
Genre: Animation
Title: Big Buck Bunny, Sunflower version
Using hardware decoding (vaapi):
VO: [cpu] i960x1080 vaapi[mx12]
VP: vdpau / 0000010 (770)
[+]
[feng@fedora ~]$ vaInfo
Trying display: wayland
Trying display: x11
vaInfo info: VA-API version 1.16.0
vaInfo info: Trying to open /usr/lib64/dri/irdriva
vaInfo info: found test function _vaDriverInt_1
vaInfo info: vaOpenDriver() returns 0
vaInfo info: VA-API version: 1.16 (libva 2.18.0)
vaInfo info: Driver version: Mesa Gallium driver 23.2
2.0.0 (merit2), LLVM 16.0.4, DRM 3.49, D3D11
vaInfo info: Supported profile and entropypoints
VAPRProfile24ConstrainedBaseline: VAMrtypp
VAPRProfile24Main: VAMrtypp
VAPRProfile24High: VAMrtypp
VAPRProfile24Low: VAMrtypp
VAPRProfile24LowPower: VAMrtypp
VAPRProfile24LowPower2: VAMrtypp
VAPRProfile24LowPower3: VAMrtypp
VAPRProfile24LowPower4: VAMrtypp
VAPRProfile24LowPower5: VAMrtypp
VAPRProfile24LowPower6: VAMrtypp
VAPRProfile24LowPower7: VAMrtypp
VAPRProfile24LowPower8: VAMrtypp
VAPRProfile24LowPower9: VAMrtypp
VAPRProfile24LowPower10: VAMrtypp
VAPRProfile24LowPower11: VAMrtypp
VAPRProfile24LowPower12: VAMrtypp
VAPRProfile24LowPower13: VAMrtypp
VAPRProfile24LowPower14: VAMrtypp
VAPRProfile24LowPower15: VAMrtypp
VAPRProfile24LowPower16: VAMrtypp
VAPRProfile24LowPower17: VAMrtypp
VAPRProfile24LowPower18: VAMrtypp
VAPRProfile24LowPower19: VAMrtypp
VAPRProfile24LowPower20: VAMrtypp
VAPRProfile24LowPower21: VAMrtypp
VAPRProfile24LowPower22: VAMrtypp
VAPRProfile24LowPower23: VAMrtypp
VAPRProfile24LowPower24: VAMrtypp
VAPRProfile24LowPower25: VAMrtypp
VAPRProfile24LowPower26: VAMrtypp
VAPRProfile24LowPower27: VAMrtypp
VAPRProfile24LowPower28: VAMrtypp
VAPRProfile24LowPower29: VAMrtypp
VAPRProfile24LowPower30: VAMrtypp
VAPRProfile24LowPower31: VAMrtypp
VAPRProfile24LowPower32: VAMrtypp
VAPRProfile24LowPower33: VAMrtypp
VAPRProfile24LowPower34: VAMrtypp
VAPRProfile24LowPower35: VAMrtypp
VAPRProfile24LowPower36: VAMrtypp
VAPRProfile24LowPower37: VAMrtypp
VAPRProfile24LowPower38: VAMrtypp
VAPRProfile24LowPower39: VAMrtypp
VAPRProfile24LowPower40: VAMrtypp
VAPRProfile24LowPower41: VAMrtypp
VAPRProfile24LowPower42: VAMrtypp
VAPRProfile24LowPower43: VAMrtypp
VAPRProfile24LowPower44: VAMrtypp
VAPRProfile24LowPower45: VAMrtypp
VAPRProfile24LowPower46: VAMrtypp
VAPRProfile24LowPower47: VAMrtypp
VAPRProfile24LowPower48: VAMrtypp
VAPRProfile24LowPower49: VAMrtypp
VAPRProfile24LowPower50: VAMrtypp
VAPRProfile24LowPower51: VAMrtypp
VAPRProfile24LowPower52: VAMrtypp
VAPRProfile24LowPower53: VAMrtypp
VAPRProfile24LowPower54: VAMrtypp
VAPRProfile24LowPower55: VAMrtypp
VAPRProfile24LowPower56: VAMrtypp
VAPRProfile24LowPower57: VAMrtypp
VAPRProfile24LowPower58: VAMrtypp
VAPRProfile24LowPower59: VAMrtypp
VAPRProfile24LowPower60: VAMrtypp
VAPRProfile24LowPower61: VAMrtypp
VAPRProfile24LowPower62: VAMrtypp
VAPRProfile24LowPower63: VAMrtypp
VAPRProfile24LowPower64: VAMrtypp
VAPRProfile24LowPower65: VAMrtypp
VAPRProfile24LowPower66: VAMrtypp
VAPRProfile24LowPower67: VAMrtypp
VAPRProfile24LowPower68: VAMrtypp
VAPRProfile24LowPower69: VAMrtypp
VAPRProfile24LowPower70: VAMrtypp
VAPRProfile24LowPower71: VAMrtypp
VAPRProfile24LowPower72: VAMrtypp
VAPRProfile24LowPower73: VAMrtypp
VAPRProfile24LowPower74: VAMrtypp
VAPRProfile24LowPower75: VAMrtypp
VAPRProfile24LowPower76: VAMrtypp
VAPRProfile24LowPower77: VAMrtypp
VAPRProfile24LowPower78: VAMrtypp
VAPRProfile24LowPower79: VAMrtypp
VAPRProfile24LowPower80: VAMrtypp
VAPRProfile24LowPower81: VAMrtypp
VAPRProfile24LowPower82: VAMrtypp
VAPRProfile24LowPower83: VAMrtypp
VAPRProfile24LowPower84: VAMrtypp
VAPRProfile24LowPower85: VAMrtypp
VAPRProfile24LowPower86: VAMrtypp
VAPRProfile24LowPower87: VAMrtypp
VAPRProfile24LowPower88: VAMrtypp
VAPRProfile24LowPower89: VAMrtypp
VAPRProfile24LowPower90: VAMrtypp
VAPRProfile24LowPower91: VAMrtypp
VAPRProfile24LowPower92: VAMrtypp
VAPRProfile24LowPower93: VAMrtypp
VAPRProfile24LowPower94: VAMrtypp
VAPRProfile24LowPower95: VAMrtypp
VAPRProfile24LowPower96: VAMrtypp
VAPRProfile24LowPower97: VAMrtypp
VAPRProfile24LowPower98: VAMrtypp
VAPRProfile24LowPower99: VAMrtypp
VAPRProfile24LowPower100: VAMrtypp
VAPRProfile24LowPower101: VAMrtypp
VAPRProfile24LowPower102: VAMrtypp
VAPRProfile24LowPower103: VAMrtypp
VAPRProfile24LowPower104: VAMrtypp
VAPRProfile24LowPower105: VAMrtypp
VAPRProfile24LowPower106: VAMrtypp
VAPRProfile24LowPower107: VAMrtypp
VAPRProfile24LowPower108: VAMrtypp
VAPRProfile24LowPower109: VAMrtypp
VAPRProfile24LowPower110: VAMrtypp
VAPRProfile24LowPower111: VAMrtypp
VAPRProfile24LowPower112: VAMrtypp
VAPRProfile24LowPower113: VAMrtypp
VAPRProfile24LowPower114: VAMrtypp
VAPRProfile24LowPower115: VAMrtypp
VAPRProfile24LowPower116: VAMrtypp
VAPRProfile24LowPower117: VAMrtypp
VAPRProfile24LowPower118: VAMrtypp
VAPRProfile24LowPower119: VAMrtypp
VAPRProfile24LowPower120: VAMrtypp
VAPRProfile24LowPower121: VAMrtypp
VAPRProfile24LowPower122: VAMrtypp
VAPRProfile24LowPower123: VAMrtypp
VAPRProfile24LowPower124: VAMrtypp
VAPRProfile24LowPower125: VAMrtypp
VAPRProfile24LowPower126: VAMrtypp
VAPRProfile24LowPower127: VAMrtypp
VAPRProfile24LowPower128: VAMrtypp
VAPRProfile24LowPower129: VAMrtypp
VAPRProfile24LowPower130: VAMrtypp
VAPRProfile24LowPower131: VAMrtypp
VAPRProfile24LowPower132: VAMrtypp
VAPRProfile24LowPower133: VAMrtypp
VAPRProfile24LowPower134: VAMrtypp
VAPRProfile24LowPower135: VAMrtypp
VAPRProfile24LowPower136: VAMrtypp
VAPRProfile24LowPower137: VAMrtypp
VAPRProfile24LowPower138: VAMrtypp
VAPRProfile24LowPower139: VAMrtypp
VAPRProfile24LowPower140: VAMrtypp
VAPRProfile24LowPower141: VAMrtypp
VAPRProfile24LowPower142: VAMrtypp
VAPRProfile24LowPower143: VAMrtypp
VAPRProfile24LowPower144: VAMrtypp
VAPRProfile24LowPower145: VAMrtypp
VAPRProfile24LowPower146: VAMrtypp
VAPRProfile24LowPower147: VAMrtypp
VAPRProfile24LowPower148: VAMrtypp
VAPRProfile24LowPower149: VAMrtypp
VAPRProfile24LowPower150: VAMrtypp
VAPRProfile24LowPower151: VAMrtypp
VAPRProfile24LowPower152: VAMrtypp
VAPRProfile24LowPower153: VAMrtypp
VAPRProfile24LowPower154: VAMrtypp
VAPRProfile24LowPower155: VAMrtypp
VAPRProfile24LowPower156: VAMrtypp
VAPRProfile24LowPower157: VAMrtypp
VAPRProfile24LowPower158: VAMrtypp
VAPRProfile24LowPower159: VAMrtypp
VAPRProfile24LowPower160: VAMrtypp
VAPRProfile24LowPower161: VAMrtypp
VAPRProfile24LowPower162: VAMrtypp
VAPRProfile24LowPower163: VAMrtypp
VAPRProfile24LowPower164: VAMrtypp
VAPRProfile24LowPower165: VAMrtypp
VAPRProfile24LowPower166: VAMrtypp
VAPRProfile24LowPower167: VAMrtypp
VAPRProfile24LowPower168: VAMrtypp
VAPRProfile24LowPower169: VAMrtypp
VAPRProfile24LowPower170: VAMrtypp
VAPRProfile24LowPower171: VAMrtypp
VAPRProfile24LowPower172: VAMrtypp
VAPRProfile24LowPower173: VAMrtypp
VAPRProfile24LowPower174: VAMrtypp
VAPRProfile24LowPower175: VAMrtypp
VAPRProfile24LowPower176: VAMrtypp
VAPRProfile24LowPower177: VAMrtypp
VAPRProfile24LowPower178: VAMrtypp
VAPRProfile24LowPower179: VAMrtypp
VAPRProfile24LowPower180: VAMrtypp
VAPRProfile24LowPower181: VAMrtypp
VAPRProfile24LowPower182: VAMrtypp
VAPRProfile24LowPower183: VAMrtypp
VAPRProfile24LowPower184: VAMrtypp
VAPRProfile24LowPower185: VAMrtypp
VAPRProfile24LowPower186: VAMrtypp
VAPRProfile24LowPower187: VAMrtypp
VAPRProfile24LowPower188: VAMrtypp
VAPRProfile24LowPower189: VAMrtypp
VAPRProfile24LowPower190: VAMrtypp
VAPRProfile24LowPower191: VAMrtypp
VAPRProfile24LowPower192: VAMrtypp
VAPRProfile24LowPower193: VAMrtypp
VAPRProfile24LowPower194: VAMrtypp
VAPRProfile24LowPower195: VAMrtypp
VAPRProfile24LowPower196: VAMrtypp
VAPRProfile24LowPower197: VAMrtypp
VAPRProfile24LowPower198: VAMrtypp
VAPRProfile24LowPower199: VAMrtypp
VAPRProfile24LowPower200: VAMrtypp
VAPRProfile24LowPower201: VAMrtypp
VAPRProfile24LowPower202: VAMrtypp
VAPRProfile24LowPower203: VAMrtypp
VAPRProfile24LowPower204: VAMrtypp
VAPRProfile24LowPower205: VAMrtypp
VAPRProfile24LowPower206: VAMrtypp
VAPRProfile24LowPower207: VAMrtypp
VAPRProfile24LowPower208: VAMrtypp
VAPRProfile24LowPower209: VAMrtypp
VAPRProfile24LowPower210: VAMrtypp
VAPRProfile24LowPower211: VAMrtypp
VAPRProfile24LowPower212: VAMrtypp
VAPRProfile24LowPower213: VAMrtypp
VAPRProfile24LowPower214: VAMrtypp
VAPRProfile24LowPower215: VAMrtypp
VAPRProfile24LowPower216: VAMrtypp
VAPRProfile24LowPower217: VAMrtypp
VAPRProfile24LowPower218: VAMrtypp
VAPRProfile24LowPower219: VAMrtypp
VAPRProfile24LowPower220: VAMrtypp
VAPRProfile24LowPower221: VAMrtypp
VAPRProfile24LowPower222: VAMrtypp
VAPRProfile24LowPower223: VAMrtypp
VAPRProfile24LowPower224: VAMrtypp
VAPRProfile24LowPower225: VAMrtypp
VAPRProfile24LowPower226: VAMrtypp
VAPRProfile24LowPower227: VAMrtypp
VAPRProfile24LowPower228: VAMrtypp
VAPRProfile24LowPower229: VAMrtypp
VAPRProfile24LowPower230: VAMrtypp
VAPRProfile24LowPower231: VAMrtypp
VAPRProfile24LowPower232: VAMrtypp
VAPRProfile24LowPower233: VAMrtypp
VAPRProfile24LowPower234: VAMrtypp
VAPRProfile24LowPower235: VAMrtypp
VAPRProfile24LowPower236: VAMrtypp
VAPRProfile24LowPower237: VAMrtypp
VAPRProfile24LowPower238: VAMrtypp
VAPRProfile24LowPower239: VAMrtypp
VAPRProfile24LowPower240: VAMrtypp
VAPRProfile24LowPower241: VAMrtypp
VAPRProfile24LowPower242: VAMrtypp
VAPRProfile24LowPower243: VAMrtypp
VAPRProfile24LowPower244: VAMrtypp
VAPRProfile24LowPower245: VAMrtypp
VAPRProfile24LowPower246: VAMrtypp
VAPRProfile24LowPower247: VAMrtypp
VAPRProfile24LowPower248: VAMrtypp
VAPRProfile24LowPower249: VAMrtypp
VAPRProfile24LowPower250: VAMrtypp
VAPRProfile24LowPower251: VAMrtypp
VAPRProfile24LowPower252: VAMrtypp
VAPRProfile24LowPower253: VAMrtypp
VAPRProfile24LowPower254: VAMrtypp
VAPRProfile24LowPower255: VAMrtypp
VAPRProfile24LowPower256: VAMrtypp
VAPRProfile24LowPower257: VAMrtypp
VAPRProfile24LowPower258: VAMrtypp
VAPRProfile24LowPower259: VAMrtypp
VAPRProfile24LowPower260: VAMrtypp
VAPRProfile24LowPower261: VAMrtypp
VAPRProfile24LowPower262: VAMrtypp
VAPRProfile24LowPower263: VAMrtypp
VAPRProfile24LowPower264: VAMrtypp
VAPRProfile24LowPower265: VAMrtypp
VAPRProfile24LowPower266: VAMrtypp
VAPRProfile24LowPower267: VAMrtypp
VAPRProfile24LowPower268: VAMrtypp
VAPRProfile24LowPower269: VAMrtypp
VAPRProfile24LowPower270: VAMrtypp
VAPRProfile24LowPower271: VAMrtypp
VAPRProfile24LowPower272: VAMrtypp
VAPRProfile24LowPower273: VAMrtypp
VAPRProfile24LowPower274: VAMrtypp
VAPRProfile24LowPower275: VAMrtypp
VAPRProfile24LowPower276: VAMrtypp
VAPRProfile24LowPower277: VAMrtypp
VAPRProfile24LowPower278: VAMrtypp
VAPRProfile24LowPower279: VAMrtypp
VAPRProfile24LowPower280: VAMrtypp
VAPRProfile24LowPower281: VAMrtypp
VAPRProfile24LowPower282: VAMrtypp
VAPRProfile24LowPower283: VAMrtypp
VAPRProfile24LowPower284: VAMrtypp
VAPRProfile24LowPower285: VAMrtypp
VAPRProfile24LowPower286: VAMrtypp
VAPRProfile24LowPower287: VAMrtypp
VAPRProfile24LowPower288: VAMrtypp
VAPRProfile24LowPower289: VAMrtypp
VAPRProfile24LowPower290: VAMrtypp
VAPRProfile24LowPower291: VAMrtypp
VAPRProfile24LowPower292: VAMrtypp
VAPRProfile24LowPower293: VAMrtypp
VAPRProfile24LowPower294: VAMrtypp
VAPRProfile24LowPower295: VAMrtypp
VAPRProfile24LowPower296: VAMrtypp
VAPRProfile24LowPower297: VAMrtypp
VAPRProfile24LowPower298: VAMrtypp
VAPRProfile24LowPower299: VAMrtypp
VAPRProfile24LowPower300: VAMrtypp
VAPRProfile24LowPower301: VAMrtypp
VAPRProfile24LowPower302: VAMrtypp
VAPRProfile24LowPower303: VAMrtypp
VAPRProfile24LowPower304: VAMrtypp
VAPRProfile24LowPower305: VAMrtypp
VAPRProfile24LowPower306: VAMrtypp
VAPRProfile24LowPower307: VAMrtypp
VAPRProfile24LowPower308: VAMrtypp
VAPRProfile24LowPower309: VAMrtypp
VAPRProfile24LowPower310: VAMrtypp
VAPRProfile24LowPower311: VAMrtypp
VAPRProfile24LowPower312: VAMrtypp
VAPRProfile24LowPower313: VAMrtypp
VAPRProfile24LowPower314: VAMrtypp
VAPRProfile24LowPower315: VAMrtypp
VAPRProfile24LowPower316: VAMrtypp
VAPRProfile24LowPower317: VAMrtypp
VAPRProfile24LowPower318: VAMrtypp
VAPRProfile24LowPower319: VAMrtypp
VAPRProfile24LowPower320: VAMrtypp
VAPRProfile24LowPower321: VAMrtypp
VAPRProfile24LowPower322: VAMrtypp
VAPRProfile24LowPower323: VAMrtypp
VAPRProfile24LowPower324: VAMrtypp
VAPRProfile24LowPower325: VAMrtypp
VAPRProfile24LowPower326: VAMrtypp
VAPRProfile24LowPower327: VAMrtypp
VAPRProfile24LowPower328: VAMrtypp
VAPRProfile24LowPower329: VAMrtypp
VAPRProfile24LowPower330: VAMrtypp
VAPRProfile24LowPower331: VAMrtypp
VAPRProfile24LowPower332: VAMrtypp
VAPRProfile24LowPower333: VAMrtypp
VAPRProfile24LowPower334: VAMrtypp
VAPRProfile24LowPower335: VAMrtypp
VAPRProfile24LowPower336: VAMrtypp
VAPRProfile24LowPower337: VAMrtypp
VAPRProfile24LowPower338: VAMrtypp
VAPRProfile24LowPower339: VAMrtypp
VAPRProfile24LowPower340: VAMrtypp
VAPRProfile24LowPower341: VAMrtypp
VAPRProfile24LowPower342: VAMrtypp
VAPRProfile24LowPower343: VAMrtypp
VAPRProfile24LowPower344: VAMrtypp
VAPRProfile24LowPower345: VAMrtypp
VAPRProfile24LowPower346: VAMrtypp
VAPRProfile24LowPower347: VAMrtypp
VAPRProfile24LowPower348: VAMrtypp
VAPRProfile24LowPower349: VAMrtypp
VAPRProfile24LowPower350: VAMrtypp
VAPRProfile24LowPower351: VAMrtypp
VAPRProfile24LowPower352: VAMrtypp
VAPRProfile24LowPower353: VAMrtypp
VAPRProfile24LowPower354: VAMrtypp
VAPRProfile24LowPower355: VAMrtypp
VAPRProfile24LowPower356: VAMrtypp
VAPRProfile24LowPower357: VAMrtypp
VAPRProfile24LowPower358: VAMrtypp
VAPRProfile24LowPower359: VAMrtypp
VAPRProfile24LowPower360: VAMrtypp
VAPRProfile24LowPower361: VAMrtypp
VAPRProfile24LowPower362: VAMrtypp
VAPRProfile24LowPower363: VAMrtypp
VAPRProfile24LowPower364: VAMrtypp
VAPRProfile24LowPower365: VAMrtypp
VAPRProfile24LowPower366: VAMrtypp
VAPRProfile24LowPower367: VAMrtypp
VAPRProfile24LowPower368: VAMrtypp
VAPRProfile24LowPower369: VAMrtypp
VAPRProfile24LowPower370: VAMrtypp
VAPRProfile24LowPower371: VAMrtypp
VAPRProfile24LowPower372: VAMrtypp
VAPRProfile24LowPower373: VAMrtypp
VAPRProfile24LowPower374: VAMrtypp
VAPRProfile24LowPower375: VAMrtypp
VAPRProfile24LowPower376: VAMrtypp
VAPRProfile24LowPower377: VAMrtypp
VAPRProfile24LowPower378: VAMrtypp
VAPRProfile24LowPower379: VAMrtypp
VAPRProfile24LowPower380: VAMrtypp
VAPRProfile24LowPower381: VAMrtypp
VAPRProfile24LowPower382: VAMrtypp
VAPRProfile24LowPower383: VAMrtypp
VAPRProfile24LowPower384: VAMrtypp
VAPRProfile24LowPower385: VAMrtypp
VAPRProfile24LowPower386: VAMrtypp
VAPRProfile24LowPower387: VAMrtypp
VAPRProfile24LowPower388: VAMrtypp
VAPRProfile24LowPower389: VAMrtypp
VAPRProfile24LowPower390: VAMrtypp
VAPRProfile24LowPower391: VAMrtypp
VAPRProfile24LowPower392: VAMrtypp
VAPRProfile24LowPower393: VAMrtypp
VAPRProfile24LowPower394: VAMrtypp
VAPRProfile24LowPower395: VAMrtypp
VAPRProfile24LowPower396: VAMrtypp
VAPRProfile24LowPower397: VAMrtypp
VAPRProfile24LowPower398: VAMrtypp
VAPRProfile24LowPower399: VAMrtypp
VAPRProfile24LowPower400: VAMrtypp
VAPRProfile24LowPower401: VAMrtypp
VAPRProfile24LowPower402: VAMrtypp
VAPRProfile24LowPower403: VAMrtypp
VAPRProfile24LowPower404: VAMrtypp
VAPRProfile24LowPower405: VAMrtypp
VAPRProfile24LowPower406: VAMrtypp
VAPRProfile24LowPower407: VAMrtypp
VAPRProfile24LowPower408: VAMrtypp
VAPRProfile24LowPower409: VAMrtypp
VAPRProfile24LowPower410: VAMrtypp
VAPRProfile24LowPower411: VAMrtypp
VAPRProfile24LowPower412: VAMrtypp
VAPRProfile24LowPower413: VAMrtypp
VAPRProfile24LowPower414: VAMrtypp
VAPRProfile24LowPower415: VAMrtypp
VAPRProfile24LowPower416: VAMrtypp
VAPRProfile24LowPower417: VAMrtypp
VAPRProfile24LowPower418: VAMrtypp
VAPRProfile24LowPower419: VAMrtypp
VAPRProfile24LowPower420: VAMrtypp
VAPRProfile24LowPower421: VAMrtypp
VAPRProfile24LowPower422: VAMrtypp
VAPRProfile24LowPower423: VAMrtypp
VAPRProfile24LowPower424: VAMrtypp
VAPRProfile24LowPower425: VAMrtypp
VAPRProfile24LowPower426: VAMrtypp
VAPRProfile24LowPower427: VAMrtypp
VAPRProfile24LowPower428: VAMrtypp
VAPRProfile24LowPower429: VAMrtypp
VAPRProfile24LowPower430: VAMrtypp
VAPRProfile24LowPower431: VAMrtypp
VAPRProfile24LowPower432: VAMrtypp
VAPRProfile24LowPower433: VAMrtypp
VAPRProfile24LowPower434: VAMrtypp
VAPRProfile24LowPower435: VAMrtypp
VAPRProfile24LowPower436: VAMrtypp
VAPRProfile24LowPower437: VAMrtypp
VAPRProfile24LowPower438: VAMrtypp
VAPRProfile24LowPower439: VAMrtypp
VAPRProfile24LowPower440: VAMrtypp
VAPRProfile24LowPower441: VAMrtypp
VAPRProfile24LowPower442: VAMrtypp
VAPRProfile24LowPower443: VAMrtypp
VAPRProfile24LowPower444: VAMrtypp
VAPRProfile24LowPower445: VAMrtypp
VAPRProfile24LowPower446: VAMrtypp
VAPRProfile24LowPower447: VAMrtypp
VAPRProfile24LowPower448: VAMrtypp
VAPRProfile24LowPower449: VAMrtypp
VAPRProfile24LowPower450: VAMrtypp
VAPRProfile24LowPower451: VAMrtypp
VAPRProfile24LowPower452: VAMrtypp
VAPRProfile24LowPower453: VAMrtypp
VAPRProfile24LowPower454: VAMrtypp
VAPRProfile24LowPower455: VAMrtypp
VAPRProfile24LowPower456: VAMrtypp
VAPRProfile24LowPower457: VAMrtypp
VAPRProfile24LowPower458: VAMrtypp
VAPRProfile24LowPower459: VAMrtypp
VAPRProfile24LowPower460: VAMrtypp
VAPRProfile24LowPower461: VAMrtypp
VAPRProfile24LowPower462: VAMrtypp
VAPRProfile24LowPower463: VAMrtypp
VAPRProfile24LowPower464: VAMrtypp
VAPRProfile24LowPower465: VAMrtypp
VAPRProfile24LowPower466: VAMrtypp
VAPRProfile24LowPower467: VAMrtypp
VAPRProfile24LowPower468: VAMrtypp
VAPRProfile24LowPower469: VAMrtypp
VAPRProfile24LowPower470: VAMrtypp
VAPRProfile24LowPower471: VAMrtypp
VAPRProfile24LowPower472: VAMrtypp
VAPRProfile24LowPower473: VAMrtypp
VAPRProfile24LowPower474: VAMrtypp
VAPRProfile24LowPower475: VAMrtypp
VAPRProfile24LowPower476: VAMrtypp
VAPRProfile24LowPower477: VAMrtypp
VAPRProfile24LowPower478: VAMrtypp
VAPRProfile24LowPower479: VAMrtypp
VAPRProfile24LowPower480: VAMrtypp
VAPRProfile24LowPower481: VAMrtypp
VAPRProfile24LowPower482: VAMrtypp
VAPRProfile24LowPower483: VAMrtypp
VAPRProfile24LowPower484: VAMrtypp
VAPRProfile24LowPower485: VAMrtypp
VAPRProfile24LowPower486: VAMrtypp
VAPRProfile24LowPower487: VAMrtypp
VAPRProfile24LowPower488: VAMrtypp
VAPRProfile24LowPower489: VAMrtypp
VAPRProfile24LowPower490: VAMrtypp
VAPRProfile24LowPower491: VAMrtypp
VAPRProfile24LowPower492: VAMrtypp
VAPRProfile24LowPower493: VAMrtypp
VAPRProfile24LowPower494: VAMrtypp
VAPRProfile24LowPower495: VAMrtypp
VAPRProfile24LowPower496: VAMrtypp
VAPRProfile24LowPower497: VAMrtypp
VAPRProfile24LowPower498: VAMrtypp
VAPRProfile24LowPower499: VAMrtypp
VAPRProfile24LowPower500: VAMrtypp
VAPRProfile24LowPower501: VAMrtypp
VAPRProfile24LowPower502: VAMrtypp
VAPRProfile24LowPower503: VAMrtypp
VAPRProfile24LowPower504: VAMrtypp
VAPRProfile24LowPower505: VAMrtypp
VAPRProfile24LowPower506: VAMrtypp
VAPRProfile24LowPower507: VAMrtypp
VAPRProfile24LowPower508: VAMrtypp
VAPRProfile24LowPower509: VAMrtypp
VAPRProfile24LowPower510: VAMrtypp
VAPRProfile24LowPower511: VAMrtypp
VAPRProfile24LowPower512: VAMrtypp
VAPRProfile24LowPower513: VAMrtypp
VAPRProfile24LowPower514: VAMrtypp
VAPRProfile24LowPower515: VAMrtypp
VAPRProfile24LowPower516: VAMrtypp
VAPRProfile24LowPower517: VAMrtypp
VAPRProfile24LowPower518: VAMrtypp
VAPRProfile24LowPower519: VAMrtypp
VAPRProfile24LowPower520: VAMrtypp
VAPRProfile24LowPower521: VAMrtypp
VAPRProfile24LowPower522: VAMrtypp
VAPRProfile24LowPower523: VAMrtypp
VAPRProfile24LowPower524: VAMrtypp
VAPRProfile24LowPower525: VAMrtypp
VAPRProfile24LowPower526: VAMrtypp
VAPRProfile24LowPower527: VAMrtypp
VAPRProfile24LowPower528: VAMrtypp
VAPRProfile24LowPower529: VAMrtypp
VAPRProfile24LowPower530: VAMrtypp
VAPRProfile24LowPower531: VAMrtypp
VAPRProfile24LowPower532: VAMrtypp
VAPRProfile24LowPower533: VAMrtypp
VAPRProfile24LowPower534: VAMrtypp
VAPRProfile24LowPower535: VAMrtypp
VAPRProfile24LowPower536: VAMrtypp
VAPRProfile24LowPower537: VAMrtypp
VAPRProfile24LowPower538: VAMrtypp
VAPRProfile24LowPower539: VAMrtypp
VAPRProfile24LowPower540: VAMrtypp
VAPRProfile24LowPower541: VAMrtypp
VAPRProfile24LowPower542: VAMrtypp
VAPRProfile24LowPower543: VAMrtypp
VAPRProfile24LowPower544: VAMrtypp
VAPRProfile24LowPower545: VAMrtypp
VAPRProfile24LowPower546: VAMrtypp
VAPRProfile24LowPower547: VAMrtypp
VAPRProfile24LowPower548: VAMrtypp
VAPRProfile24LowPower549: VAMrtypp
VAPRProfile24LowPower550: VAMrtypp
VAPRProfile24LowPower551: VAMrtypp
VAPRProfile24LowPower552: VAMrtypp
VAPRProfile24LowPower553: VAMrtypp
VAPRProfile24LowPower554: VAMrtypp
VAPRProfile24LowPower555: VAMrtypp
VAPRProfile24LowPower556: VAMrtypp
VAPRProfile24LowPower557: VAMrtypp
VAPRProfile24LowPower558: VAMrtypp
VAPRProfile24LowPower559: VAMrtypp
VAPRProfile24LowPower560: VAMrtypp
VAPRProfile24LowPower561: VAMrtypp
VAPRProfile24LowPower562: VAMrtypp
VAPRProfile24LowPower563: VAMrtypp
VAPRProfile24LowPower564: VAMrtypp
VAPRProfile24LowPower565: VAMrtypp
VAPRProfile24LowPower566: VAMrtypp
VAPRProfile24LowPower567: VAMrtypp
VAPRProfile24LowPower568: VAMrtypp
VAPRProfile24LowPower569: VAMrtypp
VAPRProfile24LowPower570: VAMrtypp
VAPRProfile24LowPower571: VAMrtypp
VAPRProfile24LowPower572: VAMrtypp
VAPRProfile24LowPower573: VAMrtypp
VAPRProfile24LowPower574: VAMrtypp
VAPRProfile24LowPower575: VAMrtypp
VAPRProfile24LowPower576: VAMrtypp
VAPRProfile24LowPower577: VAMrtypp
VAPRProfile24LowPower578: VAMrtypp
VAPRProfile24LowPower579: VAMrtypp
VAPRProfile24LowPower580: VAMrtypp
VAPRProfile24LowPower581: VAMrtypp
VAPRProfile24LowPower582: VAMrtypp
VAPRProfile24LowPower583: VAMrtypp
VAPRProfile24LowPower584: VAMrtypp
VAPRProfile24LowPower585: VAMrtypp
VAPRProfile24LowPower586: VAMrtypp
VAPRProfile24LowPower587: VAMrtypp
VAPRProfile24LowPower588: VAMrtypp
VAPRProfile24LowPower589: VAMrtypp
VAPRProfile24LowPower590: VAMrtypp
VAPRProfile24LowPower591: VAMrtypp
VAPRProfile24LowPower592: VAMrtypp
VAPRProfile24LowPower593: VAMrtypp
VAPRProfile24LowPower594: VAMrtypp
VAPRProfile24LowPower595: VAMrtypp
VAPRProfile24LowPower596: VAMrtypp
VAPRProfile24LowPower597: VAMrtypp
VAPRProfile24LowPower598: VAMrtypp
VAPRProfile24LowPower599: VAMrtypp
VAPRProfile24LowPower600: VAMrtypp
VAPRProfile24LowPower601: VAMrtypp
VAPRProfile24LowPower602: VAMrtypp
VAPRProfile24LowPower603: VAMrtypp
VAPRProfile24LowPower604: VAMrtypp
VAPRProfile24LowPower605: VAMrtypp
VAPRProfile24LowPower606: VAMrtypp
VAPRProfile24LowPower607: VAMrtypp
VAPRProfile24LowPower608: VAMrtypp
VAPRProfile24LowPower609: VAMrtypp
VAPRProfile24LowPower610: VAMrtypp
VAPRProfile24LowPower611: VAMrtypp
VAPRProfile24LowPower612: VAM
```

功能介绍

VirtIO-GPU AV1解码功能在“VirtIO-GPU硬件视频加速框架”的基础上开发，实现了AV1前端驱动、AV1后端服务程序等，并扩展了VirtIO-GPU视频协议，整体结构如下：

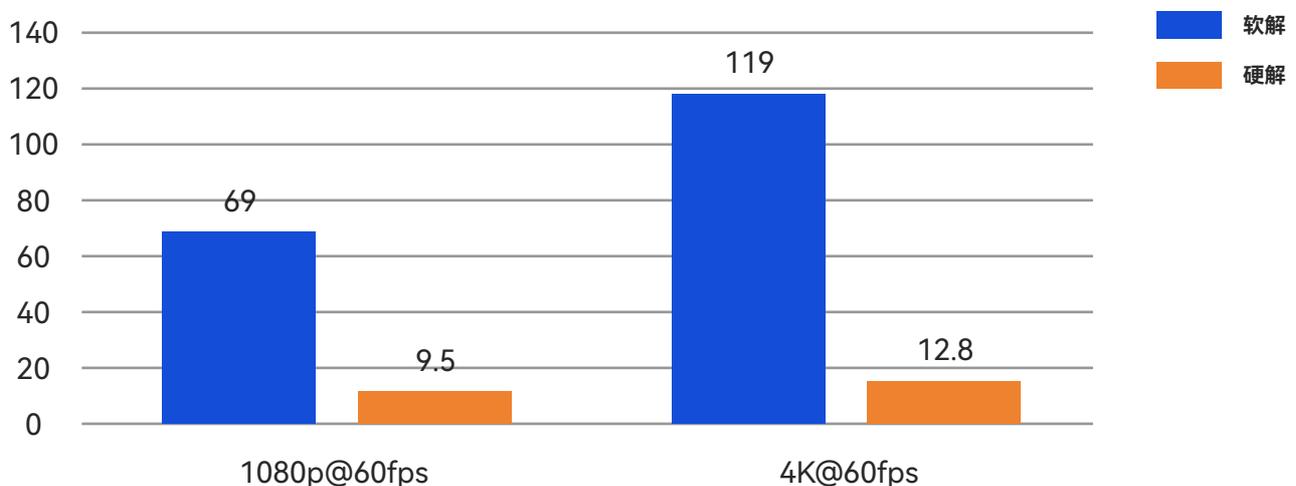


其中，VirtIO-GPU AV1 驱动主要负责接收来自虚拟机应用程序和视频加速库的视频解码请求，并将其封装成VirtIO-GPU AV1协议报文，通过virtio队列发送给VirtIO-GPU设备。VirtIO-GPU设备接收、解析报文后，继续将请求转发给AV1后端。AV1后端则调用物理机侧的视频加速库，借助物理显卡进行硬件解码，并最终将解码后的图像原路传回。

应用场景

在x86环境下，经测试使用VirtIO-GPU AV1硬件解码功能后，可显著降低虚拟机内播放器等视频程序对vCPU的占用，提高系统流畅度和用户体验。

播放器的CPU平均占用率 (4C8G)



项目地址

Virtualization SIG主页: <https://gitee.com/openKylin/community/tree/master/sig/Virtualization>

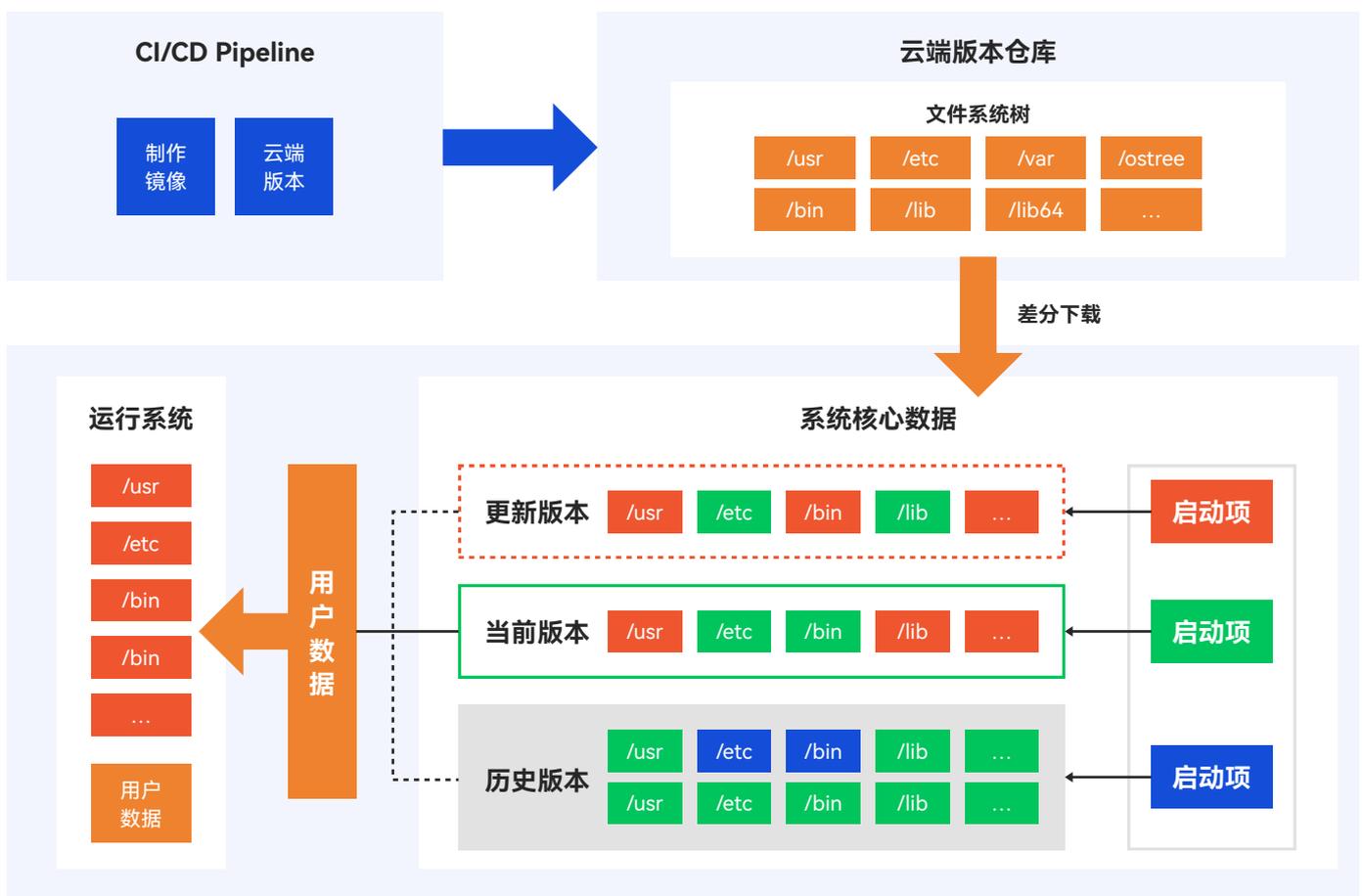
Mesa补丁: https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/23386

Virglrenderer补丁: https://gitlab.freedesktop.org/virgl/virglrenderer/-/merge_requests/1135

4.1.4 不可变系统

背景概述

不可变系统是一种设计为不可更改且只读的操作系统，即用户或应用程序不能直接修改正在运行的系统的核心文件和目录，用户通过原子更新实现系统的更新升级，在更新失败时能够迅速回滚，确保系统不会陷入异常状态。应用程序通过容器技术与核心操作系统相互隔离，确保一个应用程序所做的更改不会影响核心系统。



功能介绍

稳定性

借助OSTree技术，openKylin在2.0阶段实现了对系统架构的不可变性的支持，可以确保每次系统更新都是无风险的、原子级的操作，大大降低了系统运行中出现问题的可能性，为用户提供了更加强大的稳定性保障。

安全可靠

通过不可变系统架构，openKylin有效应对了恶意软件和系统漏洞的威胁，为用户的数据和隐私安全提供了坚实的保障，极大地提升了系统的安全性。

应用隔离

通过容器化技术实现应用程序与核心操作系统隔离。确保应用程序所做的更改不会影响核心系统或其他应用程序。

持续创新

openKylin基于OSTree技术的不可变系统，将持续为用户提升系统升级、应用程序（开明包）安装等功能，为业务和应用的持续创新提供了坚实的技术支持。

应用场景

不可变系统主要应用在对安全性、稳定性、系统可预测性等条件要求较高的场景，如云计算、物联网设备、嵌入式等等。

项目地址

openKylin OSTree项目地址：<https://gitee.com/openkylin/ostree>

4.1.5 GB18030-2022支持

背景概述

GB18030全称为《信息技术中文编码字符集》，是我国自主研发的超大型中文编码字符集国家标准。该标准规定了中文字符及其他常用字符的二进制编码及其对应的字形或图形。2022年7月19日发布的GB18030-2022是其最新版本（简称新国标），并于2023年8月1日正式实施。openKylin社区在GBCharactersEncoding SIG组的主导推动下，实现了openKylin操作系统对GB18030-2022的完整支持，包括文字输入、文字显示等多个方面。

主要工作介绍

为了完整准确支持GB18030-2022标准，需要从底层基础库到应用程序进行全面修改才能实现该目标。其涉及到的模块如下图所示：



在openKylin社区GBCharactersEncoding SIG组的主导推动下，openKylin 1.0版本完成了文本编辑器（pluma）、GTK3框架、harfbuzz以及glibc库的相关修改，实现了对GB18030-2022的完整支持。同时，为了支持输入法新国标字符，GBCharactersEncoding SIG、InputMethod SIG联合开发了GB18030输入法，使用户可以通过国标内码输入任意新国标字符。

来自开源，回归开源，openKylin在增加GB18030-2022支持的过程中，还向pluma、GTK等开源项目贡献了相关补丁，帮助其解决了无法正确识别、设置非ASCII字符字体名的问题。

应用场景

openKylin操作系统在国内率先完成了GB18030-2022标准的支持，可以满足用户对GB18030-2022标准的需求场景。GB18030-2022标准相比旧版GB18030-2005标准，新增了17000多个汉字，共收录87887个汉字，228个汉字部首。新国标大幅扩充了汉字字符数量，尤其是生僻字数量，基本上满足了人名、地名、古籍等场景中生僻字处理的需求。

项目地址

GBCharactersEncoding SIG组主页地址：

<https://gitee.com/openkylin/community/tree/master/sig/GBCharactersEncoding>

4.2 桌面场景创新

4.2.1 UKUI桌面环境

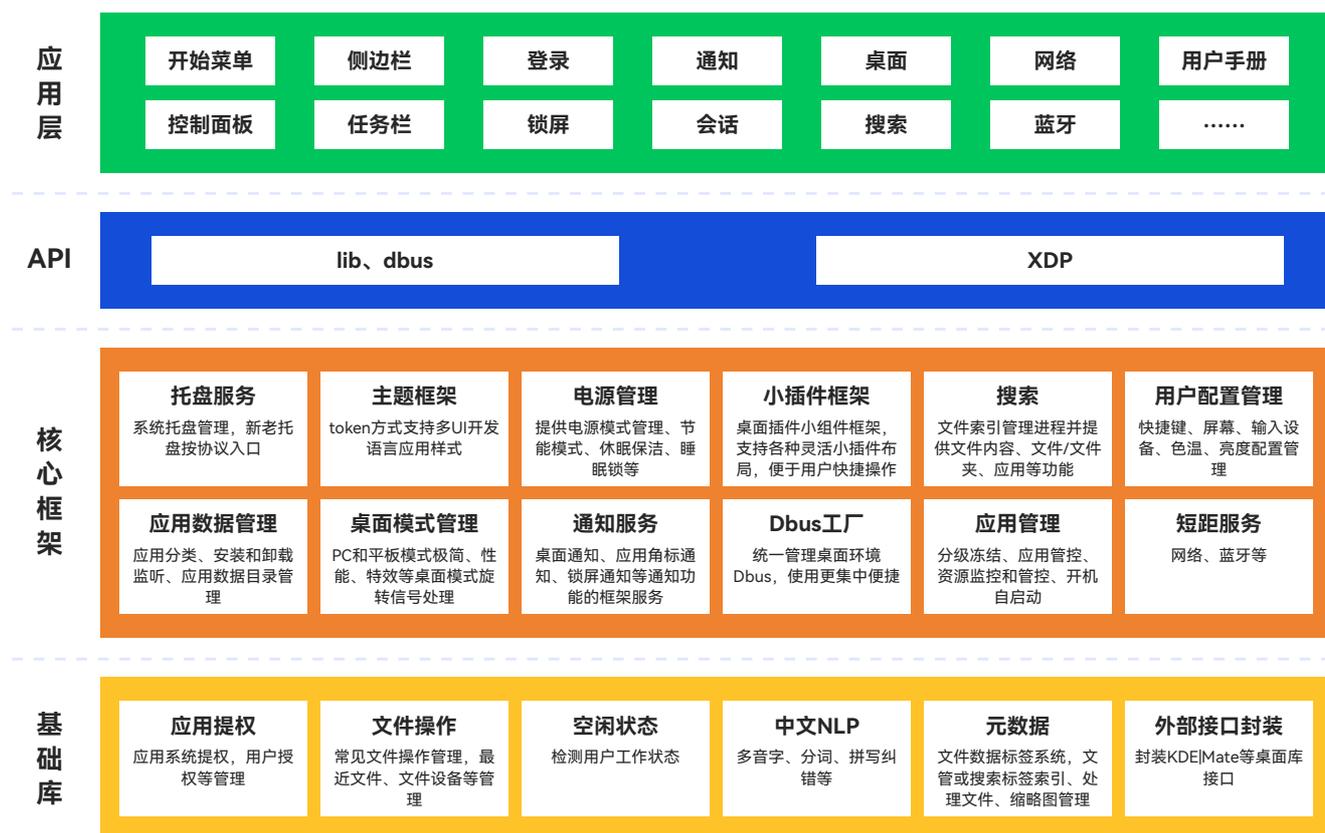
背景概述

桌面环境是桌面操作系统的核心组件，国内外有很多优秀的桌面环境以及搭载了这些桌面环境的众多发行版，但是这些桌面环境在UI设计、交互方式等方面都不太符合国人的使用习惯，缺乏中国特色，使用门槛和学习成本很高，难以在国内推广。因此，UKUI团队开发UKUI桌面环境的初衷就是要做一个真正符合中国人使用习惯的、有中国特色的、简洁易用的轻量桌面环境，让用户可以快速适应新的操作系统。

功能介绍

UKUI (Ultimate Kylin User Interface) 桌面环境目前由openKylin社区UKUI SIG组负责维护，致力于UKUI桌面环境相关软件包的规划、维护和升级工作，开发维护不断满足各种设备和用户需求的桌面环境程序。该项目主要包含程序启动器（开始菜单）、用户配置、文件管理、登录锁屏、桌面、网络工具、快捷配置等，为用户提供基本的图形化操作平台。桌面核心组件开发工具以Qt、C++为主，宗旨是始终如一地提升系统的操作体验，提供集稳定性、美观性、流畅性和便捷性为一体的桌面环境。UKUI桌面环境技术架构如下：

4 技术创新项目



目前UKUI桌面环境已更新到4系列版本, 具有以下特性:

轻量化设计理念

视觉聚焦、交互轻松。UKUI Framework+轻量化UI, 支持使用更“简单的”语言, 打造更有活力的视觉效果, 同时保证快速迭代的稳定性与可靠性。

数据共享, 统一管理

业务层与功能高度解耦, 支持快速迭代开发和定制, 支持业务层插件化开发, 更好地满足各类需求。

支持PC/平板双模式

UKUI支持平板和PC模式一键切换, 扩展更多硬件使用场景。

兼容X11/Wayland双显示协议

UKUI兼容X11和Wayland双显示协议, 可以在每次登录时快速切换, 满足更多显示场景需求。

应用场景

UKUI桌面环境默认搭载在麒麟系列桌面操作系统上, 包括openKylin、优麒麟开源操作系统以及银河麒麟、中标麒麟商业发行版中, 同时也支持Ubuntu、Debian、openEuler等国内外主流Linux发行版, 兼容通用X86、鲲鹏、海光、飞腾等多个架构平台。最新的UKUI4.0技术路线特点是实现了最大化的重用, 可以根据功能进行分类实现, 将业务层

与功能高度解耦，核心框架层封装所有的桌面环境功能实现，保证桌面环境数据的一致性，UI层只需关心业务需求，统一从核心框架获取数据和配置管理。这样的框架开发模式，支持快速迭代开发和定制，支持业务层插件化开发，更好地满足各类需求。

项目地址

UKUI SIG组主页：

<https://gitee.com/openkylin/community/tree/master/sig/UKUI>

UKUI各组件项目地址：

<https://gitee.com/openkylin/ukui-session-manager>

<https://gitee.com/openkylin/ukui-session-managerukui-settings-daemon>

<https://gitee.com/openkylin/ukui-search>

<https://gitee.com/openkylin/ukui-biometric-auth>

<https://gitee.com/openkylin/ukui-biometric-manager>

<https://gitee.com/openkylin/ukui-interface>

<https://gitee.com/openkylin/ukui-control-center>

<https://gitee.com/openkylin/ukui-greeter>

<https://gitee.com/openkylin/ukui-menu>

<https://gitee.com/openkylin/ukui-panel>

<https://gitee.com/openkylin/ukui-power-manager>

<https://gitee.com/openkylin/ukui-screensaver>

<https://gitee.com/openkylin/ukui-sidebar>

<https://gitee.com/openkylin/ukui-window-switch>

<https://gitee.com/openkylin/ukui-kwin>

<https://gitee.com/openkylin/ukui-kwin-effects>

<https://gitee.com/openkylin/ukui-bluetooth>

<https://gitee.com/openkylin/ukui-system-monitor>

<https://gitee.com/openkylin/ukui-notification-daemon>

<https://gitee.com/openkylin/ukui-clock>

<https://gitee.com/openkylin/ukui-globaltheme>

<https://gitee.com/openkylin/ukui-touch-settings-plugin>

<https://gitee.com/openkylin/ukui-input-gather>

<https://gitee.com/openkylin/ukui-app-widget>

<https://gitee.com/openkylin/ukui-system-appwidget>

<https://gitee.com/openkylin/ubuntuKylin-default-settings>

<https://gitee.com/openkylin/kylin-tablet-desktop-general>

<https://gitee.com/openkylin/kylin-app-manager>

<https://gitee.com/openkylin/kylin-app-cgroupd>

<https://gitee.com/openkylin/kylin-status-manager>

<https://gitee.com/openkylin/kylin-nm>

<https://gitee.com/openkylin/kylin-usb-creator>

<https://gitee.com/openkylin/kylin-device-daemon>
<https://gitee.com/openkylin/kylin-user-guide>
<https://gitee.com/openkylin/peony>
<https://gitee.com/openkylin/peony-extensions>
<https://gitee.com/openkylin/youker-assistant>
<https://gitee.com/openkylin/libpwquality>
<https://gitee.com/openkylin/qt5-gesture-extensions>
<https://gitee.com/openkylin/qt5-ukui-platformtheme>
<https://gitee.com/openkylin/biometric-authentication>
<https://gitee.com/openkylin/kwin>
<https://gitee.com/openkylin/dmz-cursor-theme>
<https://gitee.com/openkylin/openkylin-theme>
<https://gitee.com/openkylin/chinese-segmentation>
<https://gitee.com/openkylin/time-shutdown>

4.2.2 wlcom合成器

背景概述

图形显示系统是桌面人机交互的核心模块，目前广泛使用的X-Window（也常称为X或X11）显示系统，最初由麻省理工学院1984年研发，历经近40年的修补与扩展，代码量庞大，维护困难，设计也已略显“陈旧”。针对X中的上述缺陷，开源社区于2008提出了替代方案—wayland显示服务器协议。该协议下，wayland合成器与wayland显示服务合为一个整体组件，通信开销低；客户端之间的输入和输出隔离，提高了客户端的安全性；客户端页面刷新时，合成器只处理数据有更新的区域，提高了显示合成效率。鉴于wayland相对X的巨大优势，目前国际通用linux发行版本已经把wayland合成器作为默认的显示服务器。但目前社区开源wayland合成器在兼容性及稳定性方面存在不足。一方面大多数基于X协议编写的游戏、图形密集型等专业应用程序在wayland环境上兼容性较差，同时wayland各桌面环境定义了大量的私有协议，协议扩展难；另外，虽然wayland项目已经进行了十余年，在使用上仍然不够稳定，存在各种问题。

为解决上述问题，openKylin社区Wayland SIG基于wlroots研发了wayland合成器：wlcom。

功能介绍

wlcom合成器具有以下开发愿景：

- 依赖少，未使用QT或者GTK进行UI编写。只需要一些基础的库，例如pixman、cairo、pango、librsvg等；
- 按需设计应用与合成器之间的协议，可方便快捷增添协议，减少因协议库更新不同步或者版本冲突带来的问题；
- 特效支持，支持最大化最小化等特效，以插件形式加载；
- 完整的中文输入支持，支持input-method v2和text-input v1/v3，支持input popup，支持chromium/electron应用；
- 快捷键和触摸手势支持，支持键盘快捷键，触摸板和触摸屏手势设置；
- 输入设备支持，支持鼠标、键盘、触摸板、触摸屏、数位板；
- 多语言国际化支持。

整体框架如下图所示：



应用场景

wlcom合成器目前已基本完成与UKUI桌面环境核心组件的融合，并适配兼容主流多媒体、社交、办公等类型的X11协议应用。wlcom合成器项目将率先应用到openKylin 2.0版本上。

项目地址

wlcom项目地址：<https://gitee.com/openkylin/kylin-wayland-compositor>

Wayland SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/Wayland>

4.2.3 虚拟键盘

背景概述

随着移动端设备的不断普及，增加平板模式支持已成为桌面操作系统发展的必然趋势，因此openKylin操作系统在发布之初便默认支持一键切换平板模式。不同于传统的PC设备+传统PC操作系统，移动端设备+平板模式下主要的文字输入方式为虚拟键盘而非实体键盘，因此一款好用、易用、稳定的虚拟键盘将大幅提升移动端设备+平板模式下的使用体验，但遗憾的是传统的Linux生态中一直缺少一款这样的产品。

为此，openKylin社区InputMethod SIG和Fcitx社区核心开发人员取得联系，并一起讨论分析了当前情况，最终双方决定由InputMethod SIG组主导，基于最新的Fcitx5输入法框架进行扩展，增加Fcitx5对虚拟键盘的支持，然后在此基础上开发了一款openKylin虚拟键盘。

功能介绍

openKylin虚拟键盘具有以下主要功能：

- 支持自由拖拽
- 支持位置记忆
- 支持屏幕旋转
- 支持悬浮球唤醒
- 支持快捷键功能
- 支持小语种输入

应用场景

openKylin虚拟键盘主要应用在平板模式下文字输入场景，可以满足用户在平板模式下快速、便捷输入文字的需求。

项目地址

openKylin虚拟键盘项目地址：<https://gitee.com/openkylin/kylin-virtual-keyboard>

InputMethod SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/InputMethod>

4.2.4 麒麟输入法框架及OK输入法

背景概述

在传统Linux桌面操作系统，输入法框架只提供了基本的物理键盘输入法能力，而在虚拟键盘、手写、语音等输入方式并没有提供统一的标准，每种输入法只能根据自己的情况用不同的方案去实现，导致和系统整体上并不协调，无法融合在一起并给用户带来更好的输入体验。随着 AI 技术的发展，输入法框架也需要提供接入AI的能力。

当前Linux系统上，主流输入法厂商的输入法并没有完全按照输入法框架规范实现，部分功能是通过各自的私有规范实现，会导致很多奇怪的输入问题以及和其他应用生态适配的问题。

功能介绍

麒麟输入法管理框架开发愿景：

- 兼容fcitx5、fcitx4、ibus等输入法框架；
- 兼容JAVA、Python、Electron、QT、GTK等GUI框架；
- 兼容X和wayland输入法协议；
- 提供国标内码、多语言、少数民族语言等解决方案；
- 提供系统级AI能力，支持AIGC的AI引擎插件、手写识别AI插件、语音识别AI插件；
- 提供统一、规范的输入法开发指南文档。

OK拼音输入法已实现功能：

- 支持全拼输入；
- 支持简繁转换、中英标点和全角半角；
- 全新输入窗口和状态栏；
- 支持状态栏拖拽、位置记忆和多屏幕使用。

应用场景

- 降低输入法厂商的适配成本，只需要一次适配就可以在多平台无缝迁移；
- 使openKylin操作系统拥有自己的输入法，界面风格更统一；
- 灵活组合，便于开发定制，任何开发者都可以定制属于自己风格的输入法。

项目地址

OK输入法项目地址：<https://gitee.com/openkylin/ok-input-method>

InputMethod SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/InputMethod>

4.2.5 麒麟极点五笔输入法

背景概述

极点五笔输入法，全称为“极点中文汉字输入平台”，是由杜志民先生在21世纪之初开发的一款完全免费的，功能强大的，以五笔输入为主，拼音输入为辅的中文输入软件，支持智能造词、字典功能以及命令操作等。自诞生以来，极点五笔输入法凭借其高效、稳定的输入体验，赢得了广大中文用户的喜爱。杜志民先生作为这一优秀软件的创造者，一直致力于为用户提供更加便捷、智能的输入解决方案。目前，杜志民先生已加入openKylin社区InputMethod SIG组，共同推进极点五笔输入法在openKylin社区的开源维护工作，推出为openKylin及麒麟用户定制的“麒麟极点五笔输入法”。

功能介绍

支持三种输入方案

纯五笔、五笔双拼以及双拼。

支持其他多种输入模式

临时英文、自动英文键盘、快速输入、常用汉字输入、非常用汉字输入以及笔画输入等。

支持短语编辑功能

添加短语、删除短语、自定义短语、快速删除短语等。

应用场景

习惯使用五笔输入法和对高效输入有强烈需求的用户。

项目地址

麒麟极点五笔输入法项目地址：<https://gitee.com/openkylin/freewb>

InputMethod SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/InputMethod>

4.2.6 多端协同

背景概述

多端协同应用由openKylin社区Connectivity SIG组负责开发，主要面向AIOT实现连接性的技术和应用场景。为用户提供高效、便捷的设备间互联互通能力，实现不同设备间的协作，进而提升工作效率。目前主要支持PC与PC、PC与Android设备间的连接与协同。

功能介绍

手机与PC协同场景

- 支持手机扫码连接PC
- 支持手机投屏到PC
- 支持手机与PC文件互传
- 支持PC查看，搜索，管理手机文件
- 支持手机使用PC输入法
- 支持通过PC控制手机

PC与PC协同场景

- 支持PC与PC之间的屏幕分享
- 支持PC与PC之间的文件分享
- 支持PC上跨设备搜索文件
- 支持PC上管理其他PC文件

应用场景

- 方便用户手机电脑之间互传文件
- 支持用户的Android设备投屏到openKylin设备，并支持PC反控Android设备

项目地址

多端协同项目地址：<https://gitee.com/openkylin/kylin-connectivity>

Connectivity SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/Connectivity>

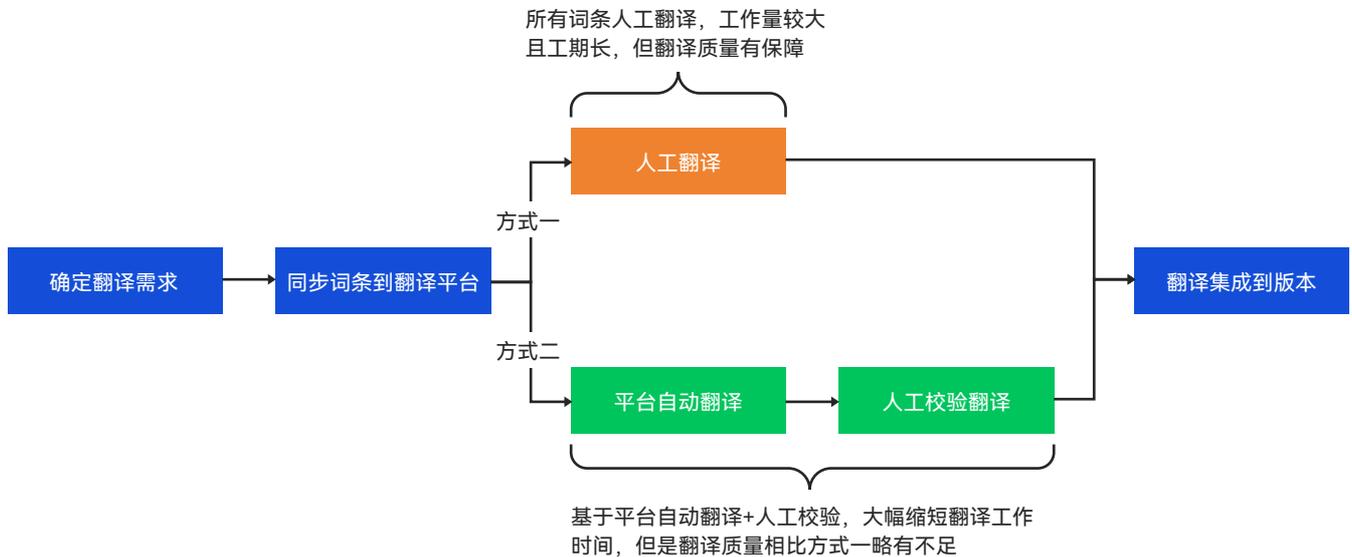
4.2.7 系统多语言支持

背景概述

在全球化日益盛行的今天，多语言支持在各类软件特别是操作系统上已经成为不可或缺的重要功能。我国作为一个拥有56个民族的多民族国家，除了汉族外，还有55个少数民族，其中很多民族拥有自己的语言和文字。因此，在软件开发中增加少数民族多语言支持，不仅是对多元文化尊重的体现，也是满足不同用户需求、提升用户体验的关键。但国外主流的Linux操作系统大部分都不支持符合我国少数民族地区用户使用习惯的系统语言。因此，openKylin社区在I18N SIG组的主导推动下，联合西藏大学、内蒙古大学等相关高校，在openKylin 2.0阶段完成了对藏文、蒙古文、维吾尔文、哈萨克文、柯尔克孜文等少数民族语言的原生支持。

openKylin多语言构建流程

openKylin操作系统多语言构建由社区I18N SIG组主导，通过openKylin翻译平台进行多语言管理、自动化翻译、人工审核、翻译词条同步等工作。具体构建流程如下图所示：



应用场景

满足我国少数民族地区用户对多语言开源操作系统的使用场景需求。

项目地址

I18N SIG组主页地址：<https://gitee.com/openkylin/community/tree/master/sig/i18n>

4.2.8 奥云蒙古文智能输入法

背景概述

我国是多民族、多语言、多文种的国家。随着科技的不断进步和全球化的发展，语言和文字的保护、传承和发展变得尤为重要。这些语言和文字是这些地区文化和身份认同的重要组成部分。然而，由于蒙古语数字化进程的滞后和缺乏相应的工具支持，这些民族语言和文字在数字化时代面临着许多困难和挑战。奥云蒙古文智能输入法旨在为用户提供一种便捷、高效、准确的输入体验，帮助用户轻松输入蒙古文，并满足他们在不同场景下的输入需求。

功能介绍

支持多种输入方式

支持语音输入、模糊输入、联想输入、特殊符号输入、自定义短语快速输入等多种输入能力。

支持用户导入词库

针对用户大量导入自定义常用短语需求，输入法允许用户批量导入词库，实现特定领域术语的快速输入。

支持国标与共享工程等多种编码

输入法参考多个编码标准文件，提供国标编码与共享工程编码等多种编码选项，确保在不同环境下的字形正确显示。

支持多语言虚拟键盘

提供了包括传统蒙古文、托忒文、满文以及西里尔文等多个语种的虚拟键盘。

支持多种自定义字体

提供了8种字体，供用户选择。

蒙古文竖排排版

对所有配置项以及输入文字均采用竖排排版，符合用户的文字使用习惯。

应用场景

满足蒙古文用户在日常使用场景下对蒙古文输入的需求，并提供便捷、高效、准确的输入体验。

项目地址

奥云蒙古文智能输入法项目地址：<https://gitee.com/openkylin/fcitx-aoyun>

4.3 生态技术创新

4.3.1 RVtrans二进制翻译技术

背景概述

RISC-V软件生态处于起步阶段，构建独立且完整的RISC-V生态需要大量且长期的工作。二进制翻译技术能够实现软件的跨平台兼容运行，达到降低开发成本、快速迁移的目的，对于不同硬件平台间的生态兼容具有重要意义。因此，基于二进制翻译技术，实现X86软件生态向RISC-V平台的高效平滑迁移，能够对RISC-V应用生态建设进行有益补充。

功能介绍

RVtrans技术以提高软件兼容性能和效率为核心，研究动静结合的二进制翻译优化技术，实现软件的快速翻译执行；基于相关性原理对动态指令翻译进行优化，研究基于时空局部性的指令Cache管理策略和基于地址关联的跳转指令翻译优化方法；研究动态库本地化映射封装和软件隔离兼容运行技术，以及面向RISC-V的兼容解决方案，实现X86架构软件在RISC-V平台上高效兼容运行。

应用场景

预期基于openKylin开源操作系统实现高效的RISC-V平台兼容运行环境，实现在RISC-V系统上高效运行X86架构软件。

项目地址

RVtrans项目地址：<https://gitee.com/openkylin/rvtrans>

RISC-V SIG组主页地址：<https://gitee.com/openkylin/community/tree/master/sig/RISC-V>

4.3.2 KARE生态兼容技术

背景概述

KARE是一种让用户能够在操作系统上运行旧版本软件的兼容环境，旨在解决兼容性问题，提高软件的可用性和可靠性。KARE出现的主要目的是解决旧软件在新的操作系统上无法正常运行的问题。KARE通过模拟旧的操作系统环境，使得旧的软件可以在新的系统上正常运行，从而避免了因为软件不兼容而导致的应用崩溃和数据丢失等问题。

功能介绍

KARE当前主要通过容器方案来运行跨系统版本应用，容器技术通过利用操作系统提供的虚拟化功能（如Linux上的namespace和cgroups）来实现资源隔离和限制。容器技术将操作系统的资源（如CPU、内存、文件系统等）划分为独立的命名空间，每个容器都拥有自己的独立命名空间，从而实现了应用程序之间的隔离。

在容器技术中，容器镜像包含了应用程序及其依赖库的所有文件和配置信息。容器镜像当前是通过Docker等工具创建的。在运行容器时，Docker等容器引擎会将容器镜像加载到隔离的环境中，并启动容器内的应用程序。

总的来说，KARE通过容器技术实现了应用程序的隔离和资源限制，从而可以在不同版本的操作系统上运行跨系统版本的应用程序。KARE的核心原理是利用操作系统提供的虚拟化功能来实现资源隔离和限制。

应用场景

- 允许用户在系统上运行旧版本的软件或不兼容的软件；
- 避免因为软件不兼容而导致的应用闪退和数据丢失等问题；
- 使用户能够在新系统上继续使用他们喜欢的旧软件。

项目地址

KARE SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/KARE>

项目正在开源筹备中，敬请期待！

4.3.3 开明软件包格式

背景概述

传统deb/rpm软件包之间依赖关系错综复杂，没有明确的系统与应用的界限，会出现安装某个软件后，系统工作不正常问题；各种衍生的发行版非常多，采用的开发库版也不尽相同，开发库及版本碎片化问题严重；传统打包对解决软件的兼容性问题没有帮助，存在系统升级后某些软件无法使用的风险。为了应对传统软件包格式的这些缺陷，openKylin社区开发了一种新的软件包格式：开明软件包格式。



功能介绍

应用隔离更兼容

一次打包，多处发布。将应用软件与系统隔离开，根本上实现向后兼容。

多格式融合

支持 Linux 存量应用、Applmage和Flatpak等多种格式应用的导入，统一管理。

细粒度权限管控

支持应用间隔离和更细粒度的权限管控，提高安全性，保护用户在每个应用中的隐私。

安全治理

逐步减少软件库的碎片化，跟踪修正基础库安全漏洞，让应用软件更健康。

应用场景

- 为应用软件提供高兼容性、高安全性的解决方案；
- 支持大量存量 Linux 应用的导入和管理；
- 有助于治理发行版碎片化问题；
- 方便用户安装升级应用软件。

项目地址

KaiMing SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/KaiMing>

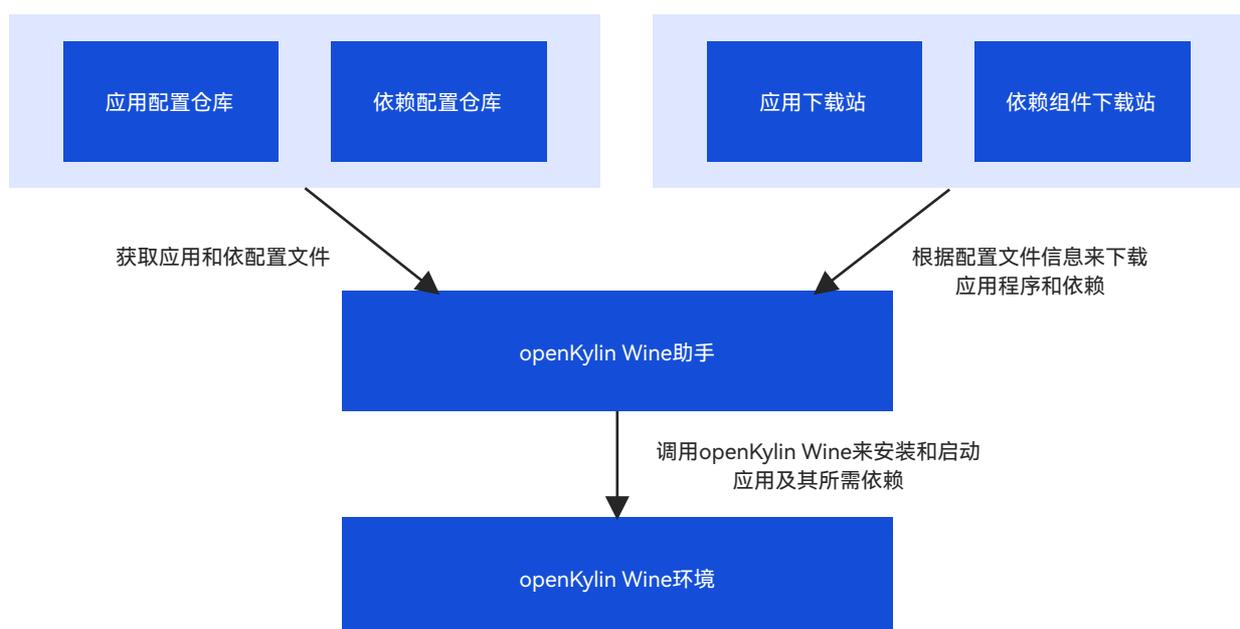
4.3.4 openKylin Wine助手

背景概述

Wine是一个开源软件，可以在Linux系统上兼容运行Windows应用程序。然而，在实际使用中，它存在一些不便之处。首先，安装Windows应用程序后，仍然需要进行一些配置，如图标和注册表设置等。这些配置因应用程序而异，对于某些用户来说，使用Windows软件并不方便。其次，Wine相对于Windows来说，其API是基于Linux系统调用设计的，因此存在一些缺陷，只有部分Windows软件可以兼容运行。因此，需要设计一个桌面软件，用于展示适配良好且可以正常使用的软件。为了提供用户更好的Windows应用程序安装和使用体验，openKylin社区CompatWinApp SIG组设计了一款集软件展示、下载、安装、管理等功能于一体的桌面软件。

功能介绍

openKylin Wine助手基于Wine提供的Windows应用运行环境，为用户提供软件展示、下载、安装、管理等功能，可以让用户在openKylin系统上方便地下载、安装和使用Windows应用程序。openKylin Wine助手框架如下：



openKylin Wine助手首先从应用配置仓库和依赖配置仓库下载配置文件。随后，将应用程序展示在窗口中，当用户点击应用程序后面的安装按钮，openKylin Wine助手就会根据配置信息下载相应应用程序和依赖，并最终调用Wine进行安装。

安装后，应用程序后面的按钮会变成启动按钮，点击启动按钮后，openKylin Wine助手会调用Wine启动该应用。

应用场景

openKylin Wine兼容环境可以完成国产操作系统在Intel、AMD、海光、兆芯平台上运行x86 Windows程序的功能。Box64、Box86和Wine相配合，可以完成国产操作系统在飞腾平台（CPU平台需支持32位ARM指令）运行x86 windows 程序的功能。

项目地址

openKylin Wine助手项目地址：<https://gitee.com/openkylin/wine-assistant>

openKylin Wine程序配置仓库地址：<https://gitee.com/openkylin/win-program>

4.4 开发工具创新

4.4.1 KylinCode集成开发环境

背景概述

KylinCode是由openKylin社区Framework SIG组基于Code-OSS定制开发的一款工具。有小伙伴可能会问，已有VSCode、VSCodium等工具，为什么不直接拿来用？虽然直接使用主流的VSCode、VSCodium等工具可以满足大部分的需求，但在国内还存在以下问题：

- VSCode协议限制较大。从微软网站下载的VSCode二进制发行版软件并非完全开源，它所使用部分插件也不是开源软件，例如常用的C/C++支持插件、remote-ssh插件等；
- 需要支持更多的国产平台，VSCode和VSCodium仅支持X86和ARM平台，LoongArch、申威等国产平台需自行支持，重新编译构建Code-OSS则不可避免；
- 面向国产操作系统，我们有增加、改造、定制功能的需求，仅依靠VSCodium无法满足需求，需要对源码进行改造。例如为常用操作增加工具栏；从OpenVSX开源插件市场中检索到我们针对国产系统适配、改造的插件等。

功能介绍

技术架构

Kylin-Code整体上由4部分组成，如下图所示，包括IDE基础平台、插件管理器、插件、插件仓库。



其中，IDE基础平台提供基本图形界面、插件架构，用户通过插件管理器检索、安装、管理插件，通过插件依赖管理器安装插件依赖。

需要注意的是，仅安装IDE基础平台只能完成很有限的编辑功能，需要安装插件并在操作系统上安装插件依赖才能实现更多的开发功能。

桌面应用模式

与常见IDE一样，IDE基础平台以桌面应用软件形式安装运行

- 1、安装方式一：openKylin系统1.0版本已预装Kylin-Code
- 2、安装方式二：在openKylin系统的软件商店中找到Kylin-Code进行安装
- 3、安装方式三：前往：<https://gitee.com/openKylin/kylin-code/releases>，下载安装包，然后执行：`sudo dpkg -i.deb`

远程开发模式

类似微软remote-ssh功能，IDE基础平台以远程服务形式安装运行，用户在本机通过SSH连接开发远程机器上的代码，体验和本地开发大体一致，远程开发模式需要插件（插件名称：Kylin remote development，插件ID：remote-dev，插件发布者xhafei）和服务端配合实现，插件会自动或引导用户安装服务端。

- 1、服务端下载地址：<https://gitee.com/mcy-kylin/kylin-ide-server/releases>
- 2、通过远程开发模式，安装openKylin系统的机器可以作为本地开发机，远程开发X86、ARM、LoongArch架构、银河麒麟操作系统V10机器上的代码

应用场景

KylinCode作为一款集成开发环境（IDE），兼具了桌面开发模式和远程开发模式，并支持安装插件等常用功能，可以充分满足社区开发者在openKylin操作系统上的开发需求。

项目地址

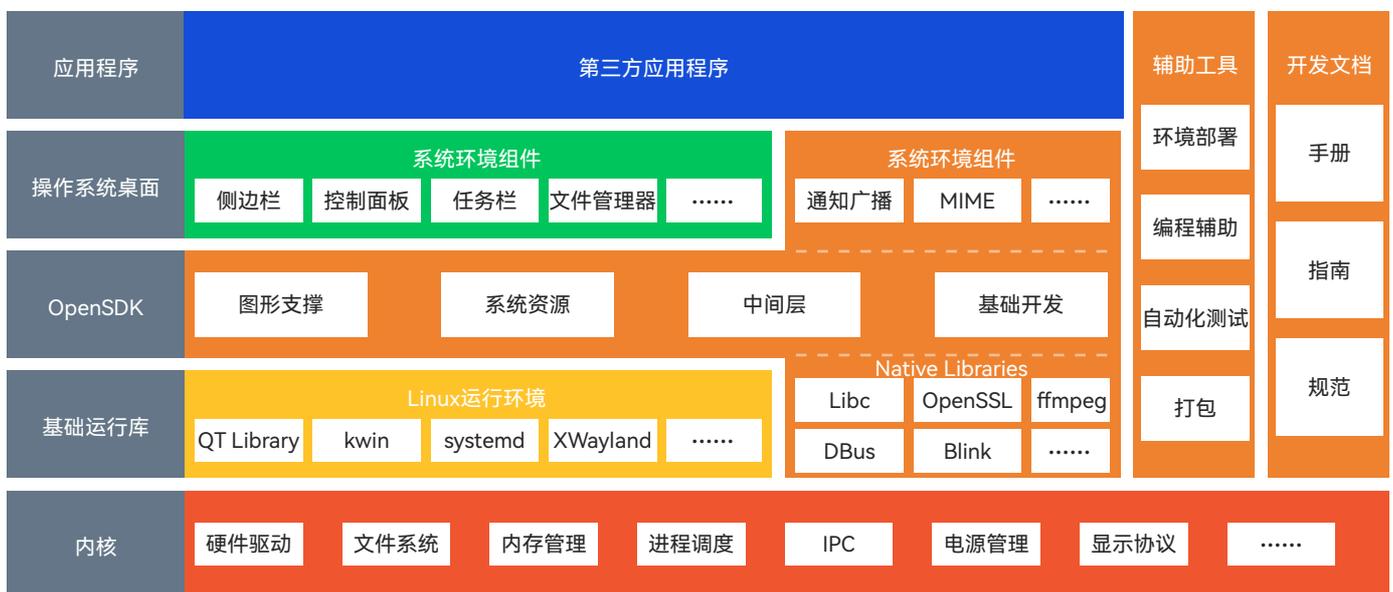
KylinCode项目地址：<https://gitee.com/openkylin/kylin-code>

Framework SIG组主页：<https://gitee.com/openkylin/community/tree/master/sig/framework>

4.4.2 openSDK

背景概述

openKylin自研开发者套件（以下简称openSDK）是在openKylin社区操作系统上，为生态建设与软件开发提供安全、可靠、快捷、稳定的开发者接口。相比于社区中其他的开发者套件或框架，openSDK更加聚焦于解决开放麒麟桌面操作系统的兼容、适配、移植、优化等方面的问题。



功能介绍

openSDK 当前聚焦三大模块，包括应用支撑 SDK、系统能力 SDK、基础开发 SDK，同时充分考虑 openSDK 的兼容性。具体见下表：

模块	描述
应用支撑 SDK	聚焦于应用显示层，为开发者提供麒麟自研控件，向图形化应用提供图形化开发功能，可使用统一的 UI 框架，进行应用窗口的管理以及与系统进行互动等，降低应用开发与应用迁移学习成本；
系统能力 SDK	聚焦于为开发者提供更多系统能力，开发者可快速获取基础的系统、硬件信息、当前的运行时信息等，提升开发效率，助力开发更聚焦实际业务内容；
基础开发 SDK	聚焦于应用开发过程中，为开发者提供日志管理、封装字符串处理等能力，提升开发效率；

应用场景

openSDK主要用于为开发者提供安全、可靠、快捷、稳定的接口，降低开发者在openKylin操作系统上的适配开发成本，并提升安全性和系统兼容性。

项目地址

openSDK项目地址：<https://gitee.com/openkylin/libkysdk-system>

4.4.3 青霜Web框架

背景概述

青霜Web引擎/软件基础架构研发，属于解决“卡脖子”问题的计算机系统底层核心基础模块。2021年，获得中科院软件及工信部信发司两个国家队全力支持及孵化。

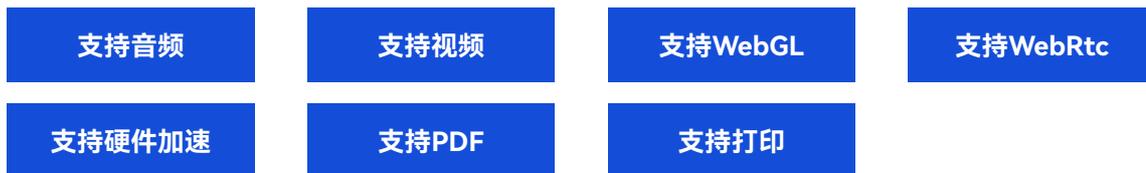
团队核心成员全部来自金山、腾讯等大厂，均拥有15年以上一线大规模C++开发经验。邀请澳门大学首席教授，IEEE终身 Fellow、IARP Fellow及AAIA Fellow，唐远炎教授全职加盟，带领博士后团队负责计算机图形基础算法研究。

目前是国内唯一的Web引擎研发及创业公司，Web引擎为现代OS必备核心组件，儒特科技与国内主要国产OS厂商均已建立深度合作关系。

功能介绍

微内核架构，仅保留最基础的排版能力，其他功能全部以插件化方式提供，同时提供“全性能/低功耗”弹性伸缩机制，未来还将提供远程渲染能力。

1、插件化模块



2、弹性伸缩模块



应用场景

1、作为简易Web页面显示组件

- 对标Windows系统的WebKit2组件或Android的WebView组件；
- 用于软件开发过程中，加载某些Web接口的功能组件，如微信登录等；
- 用于软件开发过程中，显示某些展示性内容，如产品说明书或版权声明等页面。

2、作为软件开发基础框架

- 对标Flutter、React、Electron等软件开发框架；

■ 用于构建“原生+H5”模式的软件，现代软件几乎均基于这种模式开发。

项目地址

<https://gitee.com/beijing-root/qingshuang>

<https://gitee.com/openKylin/community/tree/master/sig/QsFramework>

4.4.4 UraSDK

背景概述

当前，随着终端用户手上各种形态的设备越来越多，设备上的操作系统也多种多样，移动端有Android和iOS，桌面端有Windows、MacOS、Linux。开发者不仅需要面向多形态终端开发应用，还要面对跨操作系统和跨编程语言开发带来的困难。因此，openKylin社区的HTML5 SIG组正在开发一款Hybrid技术解决方案的SDK：UraSDK，它能够提供良好的外部插件扩展，拥有事件机制、JS API注入、Web和Native双工通信等基础能力，并拥有足够丰富且易用易扩展的插件生态。使用UraSDK开发出的应用可在多个平台上有贴近原生应用的UI效果，实现一致的功能使用体验，解决应用多端兼容的问题，让开发者不用再依照不同的前端框架或小程序平台定义的DSL（Domain Specific Language）来编写应用。

功能介绍

脚手架 (ura-cli)

提供从项目生成、插件和组件管理、编译平台管理、到应用编译烧录调试的全套功能，以及其他提升开发效率的插件，并帮助组件模板生成、测试示例应用生成。

功能插件库 (ura-plugins)

提供自研插件库支持，包括原生硬件功能、互联互通功能的自研插件库支持，提供 Capacitor、Cordova官方及社区平台的海量插件支持。

UI组件库 (ura-components)

聚焦前端用户体验和UI交互，提供开箱即用的UI组件支持，目前已完成五十余款Lit版本的组件，基于Vue和React版本的组件也正在开发中，ura-components当前已在openKylin社区代码库开源。

应用模板 (ura-templates)

UraSDK提供包括Lit（基于原生Web组件）、React和Vue的应用框架和应用模板支持，不限制使用哪一种前端UI框架。

UI组件库 ura-components	原子组件	lit原生套件	内置应用 模板框架	原生应用模板	
	复合组件	react框架套件		react应用模板	
	功能组件	vue框架套件		vue应用模板	
功能插件库 ura-plugins	自研插件库		原生接口调用		
	Capacitor官方插件	Capacitor社区插件	互联互通调用		
	Cordova官方插件	Cordova社区插件	跨平台支持		
脚手架 ura-cli	核心脚手架 ura	项目生成	目标平台管理	脚手架 套件	生成插件模板
		插件管理	跨平台应用编译		生成插件示例应用
		组件管理	应用烧录调试		生成UI组件模板

应用场景

- 已有桌面端或移动端的SPA Web应用，期望转为原生应用使用；
- 已有桌面端或移动端的SPA Web应用，期望可跨端跨平台使用；
- 期望可以通过一次应用开发，即可实现各个平台功能体验一致；
- 期望可在短期内开发出视觉交互良好的功能开发验证应用。

项目地址

UraSDK UI组件库项目地址：<https://gitee.com/openkylin/ura-components>

UraSDK项目其他组件正在开发中，敬请期待！

4.4.5 嵌入式版本构建工具

背景概述

为方便开发者基于openKylin开源操作系统二进制软件包构建嵌入式操作系统镜像，降低系统定制门槛，openKylin社区Embedded SIG组开发了一款嵌入式版本构建工具，用于快速构建以openKylin软件源为基座的嵌入式操作系统版本。

功能介绍

易用性强

openKylin嵌入式版本构建工具V1.0使用shell实现，结合实际项目经验设计构建接口。构建人员可快速构建，验证，发布嵌入式操作系统。

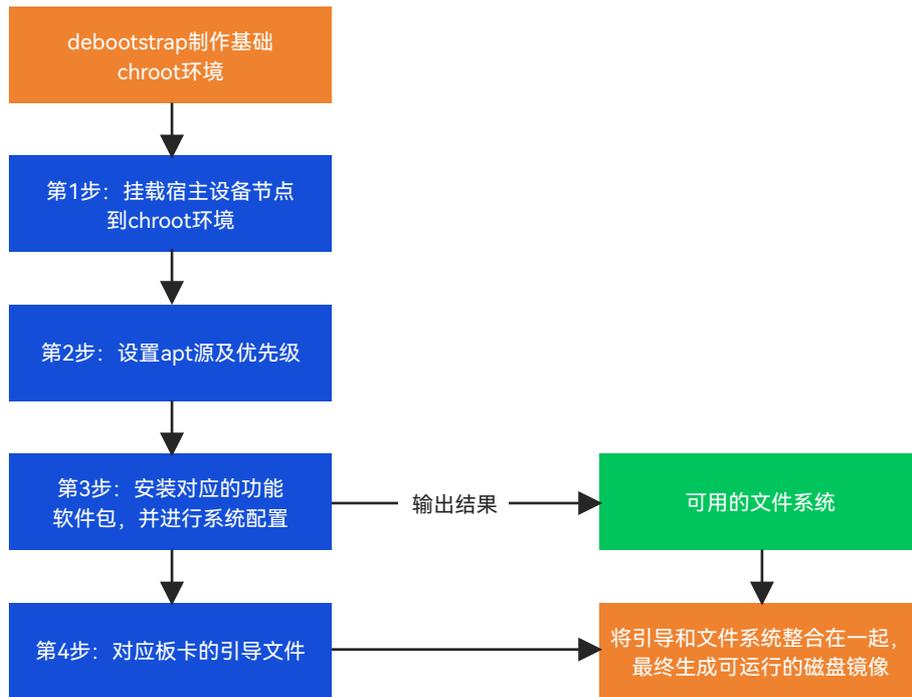
扩展性强

openKylin嵌入式版本构建工具V1.0依赖openKylin软件源构建嵌入式操作系统基座，构建人员可通过叠加额外软件源、定制系统文件等方式扩展嵌入式操作系统功能。

支持多架构

目前已支持ARM64架构（双椒派、树莓派）、RISC-V架构（VisionFive2、Lotus2）开发板的镜像生成。

openKylin嵌入式版本构建工具工作流程图



应用场景

基于openKylin嵌入式版本构建工具，各位开发者可以根据自己的实际需求，灵活构建想要的嵌入式操作系统镜像。

项目地址

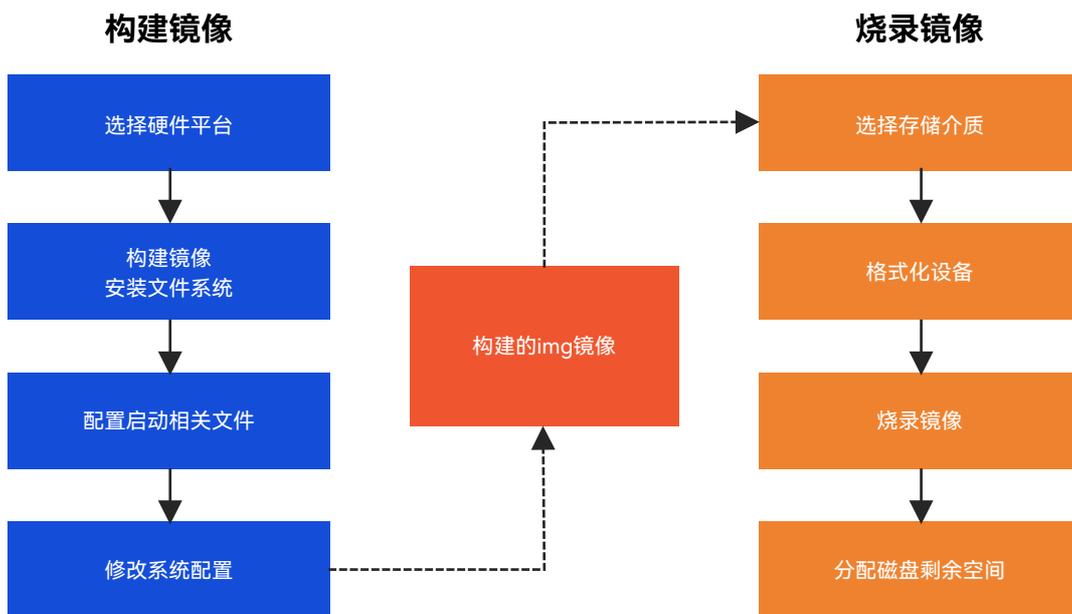
openKylin嵌入式版本构建工具项目地址：<https://gitee.com/openkylin/openkylin-embedded-builder>

Embedded SIG组主页地址：<https://gitee.com/openkylin/community/tree/master/sig/Embedded>

4.4.6 RISC-V统一镜像烧录工具

背景概述

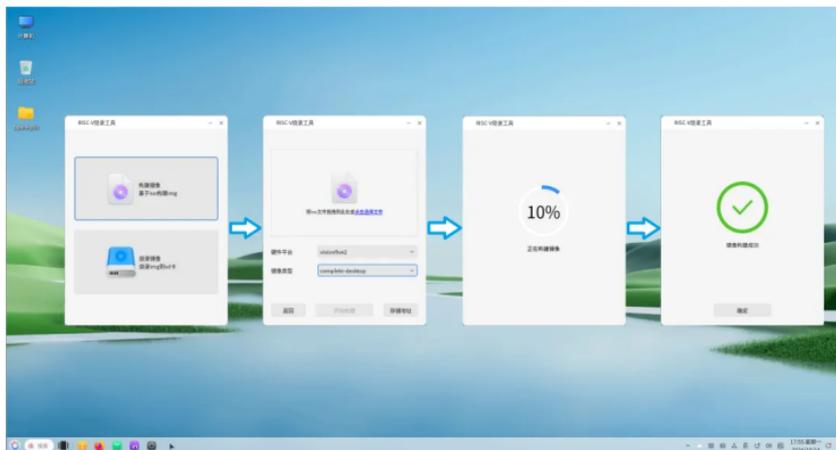
为解决RISC-V镜像碎片化问题，海河实验室主导的RISC-V SIG团队开发了RISC-V统一镜像烧录工具，通过发布RISC-V统一镜像烧录工具以及RISC-V架构统一的iso镜像，用户可以使用烧录工具并基于统一iso镜像来构建和烧录适用于某款RISC-V硬件平台的img镜像。



功能介绍

构建镜像

目前烧录工具支持VisionFive2、LicheePi4a、Milk-v-pioneer和Spacemit K1四款平台的镜像构建功能，之后社区RISC-V SIG团队会逐步对其他RISC-V硬件平台进行支持。下图分别为烧录工具首页以及构建功能界面。



烧录镜像

目前烧录工具支持烧录镜像到sd卡中，并通过sd卡启动RISC-V硬件平台。下图为烧录镜像功能界面。



应用场景

RISC-V统一镜像烧录工具目前支持VisionFive2、LicheePi4a、Milk-v-pioneer和Spacemit K1四款平台，后续还会增加支持更多的平台。

项目地址

RISC-V统一镜像烧录工具项目地址：<https://gitee.com/openkylin/kylin-riscv-burner>

RISC-V SIG组主页地址：<https://gitee.com/openkylin/community/tree/master/sig/RISC-V>

4.5 安全能力创新

4.5.1 Genmai

背景概述

Genmai项目是一个基于go语言开发的自动化安全检测框架，旨在提供一种简单易用的方式来进行系统的安全测试。该框架目前支持多种安全测试，包括内核poc安全检测、系统软件包的poc或版本检测以及安全基线检测等，同时也支持多种linux常用架构。

功能介绍

Genmai自动化安全检测框架主要功能包括：

- 支持多种功能检测，如安全基线检测、系统检测、内核安全检测等
- 支持多模式检测，如版本匹配模式、poc检测模式
- 支持多种UI界面，如CS模式或BS模式都可以进行检测
- 支持多种测试类型，包括单模块检测或全系统的检测
- 提供远程评估：提供远程评估模块，可以同时对不同架构和版本的系统进行检测

应用场景

Genmai自动化测试工具适用于需要进行安全自动化测试的多种Linux系统的多种架构，如x86，arm64，mips64le，riscv64等。它可以对系统进行内核poc检测，系统poc及版本比对检测以及安全基线配置项检测。

项目地址

Genmai项目地址为：<https://gitee.com/openkylin/genmai>

4.5.2 火焰卫士

背景概述

火焰卫士是由openKylin社区Defend SIG组规划开发的一款针对openKylin系统的综合防护软件。旨在保护openKylin操作系统免受病毒、间谍软件、广告软件等恶意软件的攻击和感染。为用户提供简单、便捷的使用方式，给三方软件提供基于系统的安全接口。

功能介绍

火焰卫士的主要功能框架规划包括：

实时监控

通过实时监控，检测和预防恶意软件的入侵。它可以监控文件、进程、网络等，以确保系统的安全。

恶意软件检测

后台服务内置基础的恶意软件检测模块，提供基础的恶意软件检测功能，检测且清除病毒、间谍软件、广告软件等恶意软件。

隔离和删除

当检测到恶意软件时，火焰卫士可以将其隔离或删除，以防止恶意软件对系统造成进一步损害。它还可以恢复已隔离的文件和文件夹。

自定义安全级别

火焰卫士可以帮助用户调整操作系统安全设置，以提高系统的安全性。它可以配置安全设置，如防火墙、文件检测强度、网络过滤强度等。

应用场景

火焰卫士的主要应用场景如下：

1、个人家庭场景

- 系统防护：火焰卫士可以帮助个人用户检测和清除恶意软件，并帮助用户提高系统的安全性。
- 修复损坏的系统：火焰卫士可以帮助用户检测和清除恶意软件，并帮助用户修复损坏的系统或应用程序。
- 个人信息保护：火焰卫士可以在系统层面防止恶意软件对系统敏感信息的读取，从源头避免个人信息泄露。

2、网络应用场景

- 防止网络攻击：火焰卫士可以基于端的角度做网络检测，检测流向当前端的数据，防止对当前端的网络攻击。
- 旁路网络监测：火焰卫士通过集成三方流量监测软件，针对流经当前网络设备的流量做旁路监测。

3、企业用户场景

- 防止未经授权的访问：火焰卫士可以帮助用户提高系统的安全性，防止未经授权的访问和数据泄露。
- 保护企业计算机系统：火焰卫士可以帮助企业保护其计算机系统免受恶意软件的攻击和感染，并确保数据的机密性和完整性。

4、恶意软件扫描

- 恶意软件查杀：火焰卫士通过集成三方病毒查杀软件，针对指定文件做病毒的查杀。

项目地址

火焰卫士项目地址：<https://gitee.com/openkylin/defend-management>

Defend SIG组主页地址：<https://gitee.com/openkylin/community/tree/master/sig/defend>

文档地址：<https://gitee.com/openkylin/defend-management/wikis>

4.6 智能融合创新

4.6.1 语音助手

背景概述

语音助手是由openKylin社区AIFramework SIG组为openKylin操作系统开发的一款系统级语音指令识别工具，目标是为openKylin系统增加智能输入能力，用户可通过语音快速下达指令，触发应用功能。

功能介绍

openKylin语音助手服务端采用开源模型库paddlespeech作为语音识别服务，将客户端上传的流式语音进行识别并且返回给客户端；客户端采用Qt的Multimedia模块进行语音识别，然后进行格式封装上传到服务端，根据识别结果处理对应操作。目前，在openKylin 1.0版本上已实现的主要功能有：

识别用户的语音输入

支持快速打开应用

应用场景

目前主要应用在openKylin操作系统上通过语音唤醒打开应用的场景，有兴趣的开发者可参考适配说明完成您的应用与openKylin语音助手的适配。

项目地址

项目地址：<https://gitee.com/openkylin/kylin-asrassistant>

AIFramework SIG地址：<https://gitee.com/openkylin/community/tree/master/sig/AIFramework>

4.6.2 桌面AI大模型插件

背景概述

openKylin社区开始尝试对AI大模型产品的生态兼容和系统级融合，探索操作系统的AI智能化发展新路径。目前，已在openKylin软件商店上架推出桌面AI大模型插件，安装后将会在开始菜单里增加AI助手模块。

功能介绍

开始菜单AI助手模块支持用户使用自己申请的AI大模型API Key和Secret Key，调用AI大模型接口，实现便捷的聊天机器人功能。

应用场景

用于在openKylin操作系统上体验基于AI大模型的聊天机器人功能。

项目地址

桌面AI大模型插件项目地址：<https://gitee.com/openkylin/ukui-menu-extensions>

4.6.3 AI框架安装助手

背景概述

以大模型为代表的人工智能技术迅猛发展，成为全球技术焦点，越来越多的开发者投入到人工智能技术的学习和应用中。但是，目前在操作系统上安装AI框架并不是件容易的事情，需要安装合适的显卡驱动、加速工具、依赖软件以及配置系统环境等。每一个环节出问题都会导致AI框架安装失败。为了让AI开发者能够更快地投入到算法设计和AI开发当中，openKylin社区AI4OS SIG组开发了AI框架安装助手，助力AI开发者实现一键安装AI框架。

功能介绍

智能推荐

用户只需选择需要安装的AI框架，应用根据系统硬件智能推荐最佳安装方式，同时支持用户选择。

一键自动

软件自动安装最适合的硬件驱动，加速工具，依赖软件，并且配置好系统环境。免去了繁复的安装过程和系统环境配置。

无需值守

由于网络等原因造成的安装失败，能自动恢复继续安装。

过程可见

软件显示安装的各个步骤，以及各个步骤的进度等信息。

节省资源

支持资源断点继续下载，过程断点继续安装。

支持热更新

确保能及时更新AI框架版本以及安装方式，同时能以最快的速度解决软件可能存在的BUG。

界面友好

采用简洁、直观的AI框架选项，并提供良好的安装过程反馈。

应用场景

AI框架安装助手可以帮助开发者快速安装和配置深度学习框架，如TensorFlow、PyTorch等，从而简化开发环境的搭建过程，提高开发效率。

项目地址

openKylin AI框架安装助手项目地址：<https://gitee.com/openkylin/ai-frame-installer>

AI4OS SIG主页：<https://gitee.com/openkylin/community/tree/master/sig/AI4OS>

4.6.4 麒麟AI模型管理工具

背景概述

麒麟 AI 模型管理工具可以自动下载自然语言处理、语音识别和文本检索等本地大模型。性能较高的机器可以使用本地大模型体验 AI 助手！

功能介绍

一键下载

麒麟AI模型管理工具支持一键下载，只需点击一键下载按钮，然后输入管理员密码即可。

高级下载

用户也可以根据自己的定制化需求，自行搭配选择和配置不同的模型类型，并支持单个模型下载安装。

模型检查

用于检查下载模型的完整性。如果所有模型均存在且文件正确，则输出全为成功；若输出有失败，请重新下载对应模型。

项目地址

麒麟AI模型管理工具项目地址：<https://gitee.com/openkylin/kylin-ai-model-manager>

AISubsystem SIG主页：<https://gitee.com/openkylin/community/tree/master/sig/AISubsystem>

4.6.5 openKylin AI SDK

背景概述

openKylin社区积极布局探索AI+OS深度融合技术。在AISubsystem SIG组的主导下，打造了openKylin AI子系统，通过SDK的方式为AI应用提供能力。社区开发者可通过AI SDK提供的C语言接口将AI能力集成到应用中。这种方式有助于开发者便捷的集成和调用，同时也支持大家根据AI SDK开发openKylin AI应用！

功能介绍

提供C语言接口

openKylin AI SDK将AI能力统一抽象封装为C语言接口，屏蔽了各个大模型的接口差异，开发者甚至不需要有AI相关的知识储备即可完成AI能力的集成，降低了使用门槛，目前均支持异步接口。

支持模型切换

openKylin AI SDK默认提供端侧模型和云端模型的能力，云端包括百度和科大讯飞等厂商，开发者可根据需求进行切换，也可以通过AI助手的模型配置界面进行配置。

支持AI引擎订制

openKylin AI SDK中所有的AI能力通过插件的形式进行集成，开发者可根据自身的需求实现具体的插件完成模型对接。

自然语言处理

提供了基础的文本对话能力，内置了多种提示词，开发者不需要编写具体的提示词，通过指定提示词模板id即可，包括中英文翻译、文本扩写、文本润色、内容质检、邮件回复、总结概括、工作总结、代码生成和会议信息提取等，提供了10个主要接口。

图像处理

提供了文生图和OCR等能力，共计12个主要接口。

音频处理

包括语音合成/识别等能力，共计16个主要接口。

模型配置功能

支持设置AI能力使用的模型、设置模型的具体配置信息等，共计8个主要接口。

项目地址

openKylin AI SDK项目地址：

<https://gitee.com/openkylin/kylin-ai-sdk>

<https://gitee.com/openkylin/kylin-ai-runtime>

<https://gitee.com/openkylin/kylin-ai-engine>

AISubsystem SIG主页：

<https://gitee.com/openkylin/community/tree/master/sig/AISubsystem>

4.6.6 麒麟AI助手

背景概述

麒麟AI助手支持多种不同类型的大模型配置，包括语言模型、视觉模型、语音模型等，并支持云端、本地、私有化部署等多种部署方式。同时，openKylin社区与讯飞星火大模型积极合作，为openKylin系统所有用户提供了免费AI大模型试用账号福利，只需要简单几步操作即可免费获取并配置试用账号！

功能介绍

AI文本对话

- 百科知识问答
- 代码编写
- 文本处理

AI助手文生图

- 文字生成图
- 生成图下载
- 生成图预览

AI语音对话

- 语音朗读
- 语音问答
- 语音控制

AI助手图片处理

- 支持拖拽上传
- 多种图片添加方式
- 支持图片黑白扫描件处理
- 支持图片抠图处理

AI会议助手

- 说话人识别
- 语音转文本
- 会议记录
- 会议总结

AI助手文本划词处理

- 文本扩写，文本润色
- 中英文翻译，内容质检
- 总结概括，邮件回复
- 工作总结，代码生成，新建日程

项目地址

麒麟AI助手项目地址：<https://gitee.com/openkylin/kylin-aiassistant>

UKUI SIG主页：<https://gitee.com/openkylin/community/tree/master/sig/UKUI>

4.6.7 其他实用AI功能

背景概述

除麒麟AI助手外，openKylin 2.0系列版本还引入了众多实用的系统级AI功能和应用，包括AI画图、智能剪切板、数据管家、AI看图、智能模糊搜索等等，可以大幅提升用户的操作体验和办公效率！

功能介绍

AI画图

基于KolourPaint画图工具和VISION视觉大模型，开发实现了智能文生图功能，用户可以在输入框输入自己的画图需求后，一键生成图片并插入到画图工作区，方便后续二次创作，大幅提升艺术工作者的画图效率！

智能剪切板

智能剪切板功能可以帮助用户在复制文本、图片等数据到目标应用时，通过使用“Win+v”快捷键，实现数据智能转换，目前支持：图片内文字转文字、文本内容转图片。

数据管家

面向用户数据内容的智能数据管理，提供对用户文档和图片内容进行智能分类管理。当前已实现基本功能，包括创建、编辑数据空间，动态聚合文件等。后续还将结合AI技术，实现语义搜索等更多高级功能，助力用户更加智能、快速、高效地管理个人数据。

AI看图

基于自研看图应用，融入AI特性，支持直接对图片进行处理，提供智能人像抠图、一键切换背景切换及尺寸裁剪等功能。

智能模糊搜索

基于系统全局搜索组件，增加智能模糊搜索功能。当您忘记要搜索的文件的准确信息时，可以通过自然语言描述进行模糊搜索，系统将智能为您进行搜索匹配。

项目地址

AI画图项目地址：<https://gitee.com/openkylin/kolourpaint>

智能剪切板项目地址：<https://gitee.com/openkylin/ukui-clipboard>

数据管家项目地址：<https://gitee.com/openkylin/ukui-intelligent-data-manager>

AI看图项目地址：<https://gitee.com/openkylin/kylin-photo-viewer>

智能模糊搜索项目地址：<https://gitee.com/openkylin/ukui-search>

4.6.8 启元九格大模型

背景概述

启元九格大模型由启元实验室牵头，联合清华大学、哈尔滨工业大学、中国科学院计算技术研究所、北京大学、南开大学等优势单位共同研制。具有高效训练与推理和高效适配与部署的技术特点，具备文本问答、文本分类、机器翻译、文本摘要等自然语言处理能力。目前已开源两个参数级别模型，分别是百亿级通用基础大模型为8B（80亿）和端侧模型2B（20亿参数）。

由国防科技大学OS4AIPC SIG牵头，基于openKylin开源操作系统+启元九格8B开源大模型+whisper开源语音识别套件等，研制了全开源的openKylin AIPC版本，初步具备了离线高效推理、系统深度融合、端侧应用赋能、端云统一调度等能力！

功能介绍

离线推理能力

通过模型轻量化，在本地部署端侧模型，并基于消费级显卡开展参数调优和领域知识增量微调，支撑多场景AI赋能。

系统融合能力

通过开展系统级推理优化、操作系统指令生成等技术实现AI与OS的深度融合。

应用赋能能力

通过开放统一接口，支撑无网络状态下WPS等应用层软件调用大模型的能力。

端云调度能力

引擎同时支持云侧大模型调用，联网状态下可以提供更强大的AI能力。

项目地址

启元九格大模型项目地址：<https://www.osredm.com/jiuyuan/CPM-9G-8B>

OS4AIPC SIG主页：<https://gitee.com/openkylin/community/tree/master/sig/OS4AIPC>

05

行业生态适配案例

5.1 整机适配

5.2 处理器平台适配

5.3 国产显卡适配

5.4 生态软件适配

5.1 整机适配

5.1.1 ROMA笔记本适配

产品介绍

ROMA是由RISC-V基金会牵头，深度数智开发、鉴释科技调试的全球首款原生RISC-V开发笔记本电脑，可让您体验原生RISC-V开发及RISC-V软件生态系统。

- 12nm(专业版)/28nm(标准版)SoM封装;
- 配备GPU、NPU和功能加速器的四核RISC-V笔记本电脑;
- 可升级 SoC 主板。



适配情况

为完成openKylin操作系统与ROMA笔记本的适配，RISC-V SIG组成员联合ROMA项目成员进行了技术攻关，攻克了多项技术难题，包括系统启动问题、显示问题、视频播放卡顿问题等，另外，RISC-V SIG组成员还为ROMA笔记本适配了包括Liberoffice、FireFox和钉钉等多款软件，满足了客户日常办公的需求。

合作伙伴

深度数智科技（珠海）有限公司、上海鉴释科技发展有限公司、先进计算与关键软件海河实验室

5.1.2 如意BOOK笔记本适配

产品介绍

如意BOOK是由中国科学院软件研究所主导研发，搭载基于玄铁C910处理器的曳影1520芯片原型及玄铁VirtualZone软硬全栈安全平台，支持4TOPS@INT8，具备通用AI算力、TEE等可信计算特性；如意BOOK首次打通了从底层芯片到操作系统到商用软件的RISC-V全链路，可主要应用于远程协同、移动办公、教育学习等场景。

- SoC: 玄铁C910;
- 玄铁VirtualZone软硬全栈安全平台;
- 具备4TOPS@INT8本地AI算力;
- 具备TEE等可信计算特性。

适配情况

为完成openKylin操作系统与如意BOOK的适配，RISC-V SIG组成员联合如意BOOK项目成员进行了技术攻关，攻克了多项技术难题，包括系统启动问题、GPU适配问题、视频播放卡顿问题等，另外，RISC-V SIG组成员还在如意BOOK上适配了包括搜狗输入法、福昕阅读器、Chromium等多款软件，可满足客户日常办公的需求。

合作伙伴

阿里巴巴达摩院（上海）科技有限公司、先进计算与关键软件海河实验室

5.1.3 MuseBOOK笔记本适配

产品介绍

MuseBOOK是由进迭时空主导研发，搭载全球首款 8 核 RISC-V 处理器 SpacemiTKey Stone K1。可主要应用于远程协同、移动办公、教育学习等场景。

- SoC: SpacemiT K1;
- 屏幕: 14.1 英寸 1080P 60Hz LCD, 覆盖 72% NTSC 色域, 亮度 250 尼特;
- 网络: 基于 RTL8852BE 的 WiFi 6 模块;
- 重量: 1.36 千克;
- 续航: 38Wh 电池 + 65W 充电器。

适配情况

为完成openKylin操作系统与MuseBOOK的适配，RISC-V SIG组成员联合MuseBOOK项目成员进行了技术攻关，攻克了多项技术难题，包括系统启动问题、GPU适配问题、视频播放卡顿问题等，另外，RISC-V SIG组成员还在MuseBOOK上适配了包括搜狗输入法、福昕阅读器、Chromium等多款软件，可满足客户日常办公的需求。

合作伙伴

进迭时空（杭州）科技有限公司、先进计算与关键软件海河实验室

5.1.4 联想开天P90z G1t台式机

产品介绍

首款国产AI PC联想开天P90z G1t台式机搭载兆芯开先KX-7000处理器，最高工作频率3.6GHz，集成GPU、支持双路4K硬件解码。采用国内显卡，支持10亿级参数大模型运行。支持四屏显示、两路4K视频输出。采用了全蜂窝式风洞设计17升塔式机箱，还配有全固态电容、USB静电防护、网口防雷击，支持接口主板直出、双SSD方案，可选多串口、选配 Wi-Fi。

适配情况

openKylin 2.0系列版本与联想开天P90z G1t台式机完成适配，并以此为基础打造了银河麒麟桌面操作系统AI版，亮相2024年8月8日中国操作系统产业大会，完美运行openKylin AI子系统和端侧模型，支持麒麟AI助手等众多系统级AI功能和应用。

合作伙伴

联想开天科技有限公司

5.2 处理器平台适配

5.2.1 兆芯KX-6000、KX-6000G系列处理器适配

产品介绍

兆芯开先®KX-6000系列处理器核心采用超标量、多发射、乱序执行架构设计，兼容最新的x86指令集，可支持64位系统，以及 CPU 硬件虚拟化技术。同时支持SM3/SM4国密算法，可提供基于硬件的数据加密保护，满足多种市场的应用需求，主要面向高性能桌面、便携终端、嵌入式等市场应用领域。

兆芯新一代开先®KX-6000G系列处理器，集成处理器核心、双通道DDR4内存控制器、高性能3D图形加速引擎、高清流媒体解码器与显示接口以及PCIe 3.0、SATA、USB等高速IO接口，可以支持单芯片解决方案。非常适合一体机、笔记本、便携式终端以及物联网计算平台等兼顾高性能、低功耗需求的应用领域。

适配情况

兆芯KX-6000、KX-6000G系列处理器完成了与openKylin 1.0版本操作系统的适配工作，相关驱动已合入openKylin内核中。相关成果后续也将逐步合并到openKylin后续主线版本中。

合作伙伴

上海兆芯集成电路股份有限公司

5.2.2 海光C86-4G系列处理器

产品介绍

海光C86-4G 3400系列处理器集成4-16核心，具有超线程特性，支持8-32线程，支持32xPCIe5.0，2个DDR5内存通道，单核最高加速频率达到4.0GHz；3400系列处理器满足可信计算、隐私计算等安全领域特性，芯片具有SM3/SM4国密指令集，并提供其他硬件加解密加速功能，3400系列芯片主要应用于入门级服务器、工作站、工业控制等市场，满足党政企业及关基行业的办公业务场景，基于海光终端处理器，海光也为各企业 and 专业人员提供高效解决方案。

适配情况

海光C86-4G 3400系列处理器完成了与openkylin 2.0版本操作系统的适配工作，相关驱动已合入openKylin内核中。相关成果后续也将逐步合并到openKylin后续主线版本中。

合作伙伴

海光信息技术股份有限公司

5.2.3 Intel 最新系列平台适配

产品介绍

Raptor Lake是Intel基于混合架构的第13代和第14代Intel酷睿处理器的代号，采用Raptor Cove性能核心和Gracemont效率核心。与Alder Lake一样，Raptor Lake也是使用Intel的Intel 7工艺制造的。Raptor Lake具有多达24个核心（8个性能核心加16个效率核心）和32个线程，并且与Alder Lake系统（LGA 1700、BGA1744、BGA1964）兼容。

适配情况

Intel SIG共完成openKylin 6.1 & 6.6 Kernel合并Intel最新平台支持补丁600多个，包括最新图形，声音，蓝牙，无线，电源管理，网络，串口，多媒体等硬件的支持。同时linux-firmware合并Intel最新平台支持补丁170多个，包括Intel最新显卡，无线网卡，蓝牙，声音等的硬件固件支持。目前openKylin 2.0系列操作系统已成功适配第13/14代Intel客户端平台，Meteor Lake、Raptor Lake系列，以及ADL-N（12代 refresh）。具体机型如下所示：

- 支持Intel Meteor Lake AIPC平台；
- 支持Intel最新13/14代CPU型号；
- 支持Intel ADL-N, RPL-S,RPL-P, RPL-U, RPL-HX等最新平台；
- 支持相关OEM惠普，联想等基于Intel 13代机的相关机型。

测试期间，openKylin 2.0系列操作系统在以上Intel开发机平台上整体运行稳定，功能、性能及兼容性方面表现良好。

合作伙伴

英特尔亚太研究有限公司

5.2.4 飞腾D3000处理器适配

产品介绍

飞腾D3000是飞腾新一代高效能桌面CPU，集成了8个飞腾自主研发的新一代高性能处理器核FTC862，兼容64位ARMv8指令集并支持ARM64和ARM32两种执行模式，支持单精度、双精度浮点运算指令和ASIMD处理指令，主频达到2.5GHz，支持硬件虚拟化，支持商业和工业分级。

搭载飞腾D3000/2000的整机与openKylin 2.0 SP1版本相互兼容良好，整体运行稳定，各项功能及性能方面均表现优异！

适配情况

飞腾D3000处理器完成了与openKylin 2.0 SP1版本的适配工作，相互兼容良好，整体运行稳定。相关驱动已合入openKylin 6.6内核中。

合作伙伴

飞腾信息技术有限公司

5.2.5 龙芯3A6000处理器适配

产品介绍

龙芯3A6000是龙芯第四代微架构首款处理器，面向高端嵌入式计算机、桌面、服务器等应用。采用自主龙芯指令集（LoongArch™），龙芯3A6000处理器SPEC CPU 2006 Base单线程定/浮点分值分别达到46.1/57.7分，达到国际市场主流水平。

适配情况

龙芯3A6000处理器完成了与openKylin 2.0 SP1版本的适配工作，相互兼容良好，整体运行稳定。相关驱动已合入openKylin 6.6内核中。

合作伙伴

龙芯中科技术股份有限公司

5.2.6 此芯科技P1处理器适配

产品介绍

此芯P1 SoC芯片采用12核的Armv9架构处理器设计，异构集成了CPU、GPU、NPU，AI算力达45TOPS，支持百亿参数以内的端侧大模型，具备强大的多媒体处理及高效的功耗管理能力。

适配情况

搭载此芯P1芯片的AI PC原型机与openKylin 2.0 SP1版本相互兼容良好，整体运行稳定，各项功能及性能方面均表现优异！此次适配过程中，双方首次基于开源的Linux 6.6内核版本合作。基于此芯P1强大的AI异构算力之下，该平台可本地运行Qwen、Llama 3等多种主流大语言模型以及文生图大模型。

合作伙伴

此芯科技集团有限公司

5.2.7 中科通量RV64G架构DW1000处理器适配

产品介绍

北京中科通量科技有限公司基于64位高性能RV64G架构DW1000 处理器成功运行openKylin操作系统。RV64G是RISC-V重要的基础指令架构，采用简化指令译码逻辑，能够更容易预测指令预取模式，进而减少内存访问次数、提高指令执行速度，降低功耗，为广泛的通用计算领域提供了简单且完备的指令集，使开发者可以更好的对高性能计算进行优化。

适配情况

自2023年投身于openKylin RV64G SIG组以来，中科通量持续加大投入力度，积极主动应对系统使用过程中出现的各种问题，不断拓展硬件适配的范围和深度，推出多款系统镜像，并强化软件源生态建设。截至目前，中科通量已与openKylin社区合作完成了近1500个源码包在RV64G架构上的编译，为用户提供了涵盖从基础编译、系统管理、桌面环境到图像处理、视频处理等多样化的服务，其中包括gcc、g++、rustc、systemd、ukui-desktop-environment、ffmpeg、opencv等工具，极大地丰富了RV64G软件生态体系。

合作伙伴

北京中科通量科技有限公司

5.3 国产显卡适配

5.3.1 景美JM9100显卡适配

产品介绍

JM9100是景美推出的第三代GPU，在前两代的基础上进一步优化性能与功耗，支持两路4K@60Hz独立显示输出，支持1GB-4GB LPDDR4/DDR4显存，支持一路4K@60fps硬件解码，支持H.265/264等常用格式，支持OpenGL4.0、Vulkan1.1、OpenGLES3.2等图形编程接口，支持OpenCL3.0计算编程接口。全面支持国产CPU、国产操作系统和国产固件，可广泛应用于PC、工控机、图形工作站等计算机设备。

适配情况

景美JM9100显卡完成了与openKylin 1.0和2.0系列版本的适配工作，相关驱动已合入openKylin内核中。相关成果后续也将逐步合并到openKylin后续主线版本中。

合作伙伴

长沙景美集成电路设计有限公司

5.3.2 格兰菲Glenfly Arise 1系列显卡适配

产品介绍

Arise1系列显卡芯片采用国内领先的基于28纳米工艺，具备自主知识产权，内置自主设计研发的新一代图形处理引擎，适用于桌面、商业显示以及通用计算等中高端应用场景。

- 支持DX11、OpenGL4.5、OpenCL1.2等主流API;
- 拥有64/128/196位的显存位宽，可提供最大57.6 GB/s的显存带宽和最大6GB的DDR4显存容量;
- 支持HEVC/H.264/MPEG2/MPEG4/VC1/MJPEG等解码格式，同时支持HEVC和H.264编码;

- 硬件平台支持涵盖主流CPU架构和知名的CPU生产商，如X86（兆芯、海光、Intel、AMD等），ARM（飞腾等），MIPS & LoongArch（龙芯）等；
- 搭配该系列显卡的整机稳定可靠，各项应用兼容适配，能够满足各种日常显示需求。

适配情况

目前已完成了Glenfly Arise1系列显卡驱动与openKylin 1.0和2.0系列版本的适配兼容工作，驱动已合入openKylin内核、mesa等组件中。相关成果后续也将逐步合并到openKylin 后续主线版本中。

合作伙伴

格兰菲智能科技有限公司

5.4 生态软件适配

5.4.1 搜狗输入法NG麒麟桌面版适配

产品介绍

搜狗输入法NG麒麟桌面版是基于跨平台输入解决方案开发的新一代输入法工具。跨平台输入解决方案提供了包括面板、引擎等输入解决方案所需要的组件支持。其中，跨平台用户界面致力于提供一个平台相关层分离的跨平台用户界面业务框架，主要包括如下模块：

业务逻辑模块

与平台完全无关，实现UI绘制、事件处理的全部业务逻辑；

平台相关层

基于具体平台特性，完成包括渲染、绘制、窗口管理等平台相关实现，并以插件的形式提供给业务逻辑模块调用；

用户界面插件

根据实际业务需要，定制实现的一组UI组件，并在皮肤描述中被引用和配置。

适配情况

上海涛略信息技术有限公司与openKylin社区InputMethod SIG组开展合作，目前已完成搜狗输入法NG麒麟桌面版X86、ARM架构与openKylin系统的适配，RISC-V架构适配正在进行中。目前搜狗输入法NG麒麟桌面版目前已上架到openKylin软件商店，并在openKylin 1.0.1和2.0版本上预装。

合作伙伴

上海涛略信息技术有限公司

5.4.2 亿图软件适配

产品介绍

深圳市亿图软件有限公司成立于2014年，自成立以来一直致力于绘图创意软件的研究和开发，凭借多年在图形可视化领域积累的经验，亿图已经成为国内外专业的绘图创意类软件提供商。其主要产品包括亿图图示EdrawMax、亿图脑图MindMaster、项目管理软件EdrawProj、平面设计软件菲果、组织架构图软件OrgCharting。

适配情况

亿图软件与openKylin社区Compatibility SIG组合作，完成了openKylin开源操作系统V1.0与亿图图示、亿图脑图和亿图项目管理三款软件的适配，并上架到openKylin软件商店。测试结果表明，亿图系列软件在openKylin操作系统上运行稳定，使用流畅。

合作伙伴

深圳市亿图软件有限公司

5.4.3 红莲花安全浏览器适配

产品介绍

海泰方圆红莲花安全浏览器，融入我国国产密码算法和一系列安全功能，打造了更安全的架构，支持自主网络信任体系。全面兼容龙芯（MIPS）、龙芯（LoongArch）、飞腾、兆芯、鲲鹏、海光、申威等国产处理器；全面兼容麒麟操作系统；全面兼容金山WPS、数科、书生、点聚等电子公文、电子签章、电子文档、中间件以及数据库等基础软硬件产品。

适配情况

海泰方圆与openKylin社区Compatibility SIG组合作，完成了openKylin开源操作系统V1.0与红莲花安全浏览器V5.0的适配测试工作，并上架到应用商店。测试结果表明，红莲花浏览器在openKylin操作系统上运行稳定，使用流畅。

合作伙伴

北京海泰方圆科技股份有限公司

5.4.4 讯飞星火大模型适配

产品介绍

科大讯飞成立于1999年，是亚太地区知名的智能语音和人工智能上市企业，公司总部位于安徽合肥。自成立以来，一直从事智能语音、自然语言理解、计算机视觉等核心技术研究并保持了国际前沿技术水平；积极推动人工智能产品和行业应用落地，致力让机器“能听会说，能理解会思考”，用人工智能建设美好世界。

讯飞星火认知大模型是科大讯飞发布的大模型。该模型具有7大核心能力，即文本生成、语言理解、知识问答、逻辑推理、数学能力、代码能力、多模交互。目前最新版本为V4.0。

适配情况

科大讯飞星火认知大模型，通过讯飞开放平台向开发者全面赋能。加入openKylin社区后，星火大模型V4.0最新版本已接入openKylin AI子系统，为众多系统级AI功能赋能。同时，讯飞星火为openKylin社区所有用户提供免费token体验专属额度，方便社区用户快速体验和探索桌面AI功能!

合作伙伴

科大讯飞股份有限公司

06

社区爱好者构建

6.1 Cutefish桌面环境移植适配

6.2 KernelBuilder项目

6.1 Cutefish桌面环境移植适配

项目背景

Cutefish是国内一款简洁、美观、实用的开源桌面环境项目，主要包括Cutefish Core、Cutefish Fishui等核心组件及Cutefish Terminal、Cutefish File Manager、Cutefish Video Player等系统应用。openKylin社区个人爱好者孙圣博在社区申请成立了Cutefish SIG组，负责移植Cutefish桌面环境及其组件到openKylin社区，丰富openKylin操作系统的。

成果介绍

目前，Cutefish SIG已经完成Cutefish桌面移植工作，共计维护23个软件包。并在openKylin社区、赛昉科技、CoolPi提供的技术及设备支持下，完成了对Cutefish桌面环境在X86，ARM，RISC-V三个架构的适配工作。

用户类型

openKylin社区个人爱好者

6.2 KernelBuilder项目

项目背景

日常使用Linux系统的过程中，可能经常因为系统的内核不够新导致一些新的硬件无法被驱动，或者是想尝试一下新版内核中的一些特性、比如部分RISC-V的驱动在新内核中才被添加进去，及其他的一些原因需要使用到比较新的内核。针对这种场景，openKylin社区爱好者熊世洁在openKylin社区成立了KernelBuilder SIG组，致力于研究如何主动从kernel.org获取到最新的内核并打包成可以安装的deb包。

成果介绍

- 已通过Github actions实现全自动内核获取并编译打包（无需人工干预），同时在打包的过程中新增了对ZSTD压缩算法的解压重制打包步骤（由于部分发行版暂未支持ZSTD算法，故将ZSTD转化为LZMA压缩算法）；
- 采用直接从<https://www.kernel.org/>获取mainline、stable、longterm三支tarball的形式进行打包、后续将迁移至直接由git仓库进行打包、由源码切换分支或tag实现，更好的代码复用，同时也能直接编译出deb包提供给ppa、apt源。（需要完成对debian/changelog等信息的自动填写）；
- 已在yangtze、nile版本引入distcc实现分布式编译；
- 引入multistrap、mmdebstrap用以制作rootfs，rootfs后续可为wsl和docker镜像服务。（制作rootfs及docker镜像也已实现自动化）；
- 向proot-distro项目上游，即用于在Termux中管理Linux发行版安装的实用程序，提交关于openKylin的代码，将openKylin发行版引入Termux，同时该项目也能为我们提供rootfs。

用户类型

openKylin社区个人爱好者

07

基础设施平台建设

7.1 基础服务平台

7.2 一站式编译构建平台

7.3 开源合规管理平台

7.3 学习成长平台

7.1 基础服务平台

背景概述

社区的基础设施平台是一组关键工具、资源和服务，旨在支持开源软件项目的发展和协作。这些平台为开发者、贡献者和用户提供了一套协同工作的框架，使他们能够共同推动开源项目的成功。

解决方案

openKylin社区基础设施平台具备以下能力：

统一认证管理

- 整体平台单点登录
- 用户管理和权限控制
- 合规性和数据隐私

一站式编译、测试执行和分析、构建的开发场景流程闭环

- 成熟完整的开发环境
- 自动化构建、测试与集成

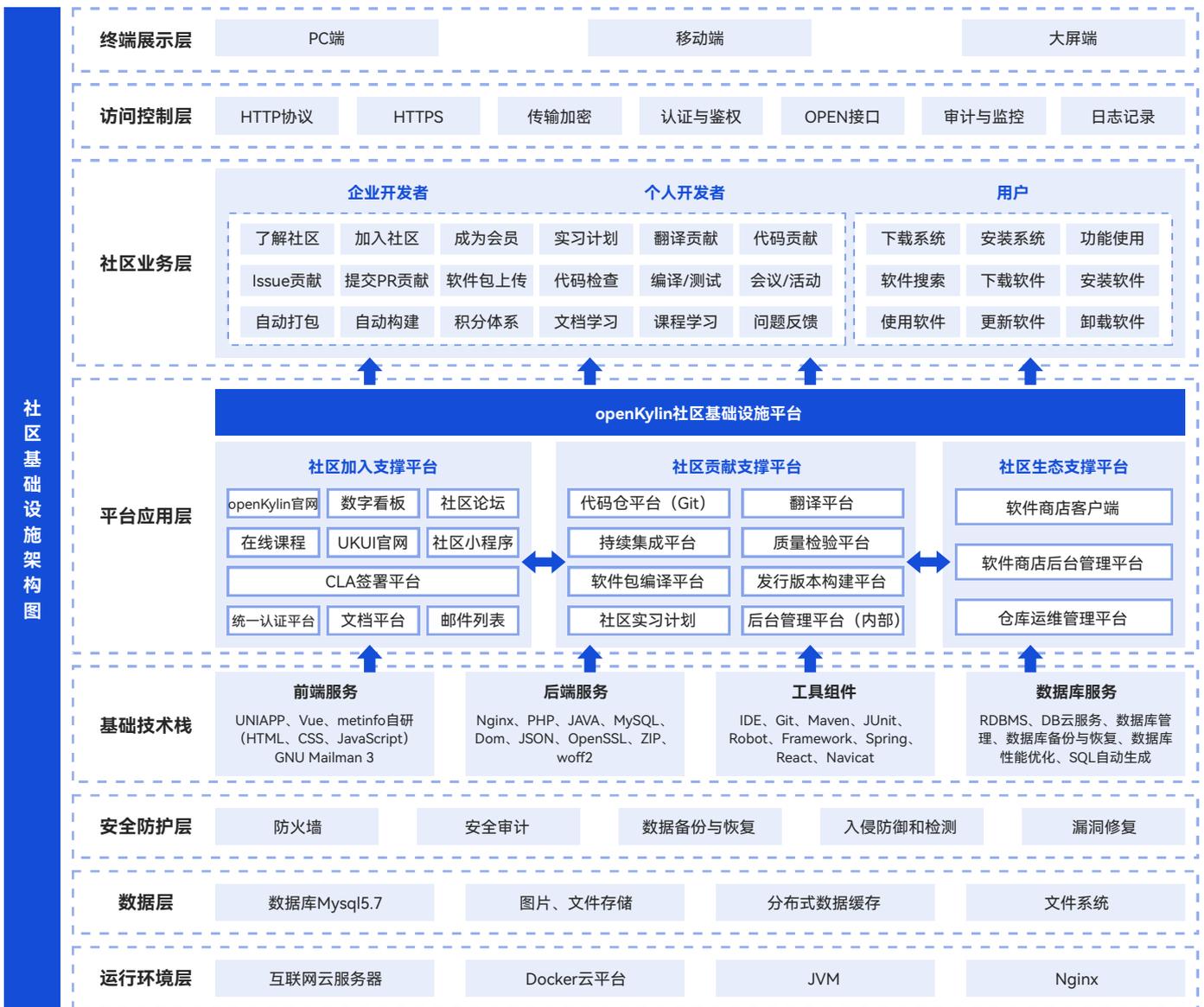
从了解—加入—贡献—收获，无缝对接的平台协作模式

- 吸引新成员，包括那些对社区和项目不太了解的人
- 提供简单、清晰的路径，以便成员逐步参与
- 帮助成员找到适合他们技能和兴趣的任务，从而提高贡献的质量
- 鼓励和奖励贡献者，以增强他们的满意度和忠诚度

强大的资源服务框架

- 利用弹性云资源为社区的开发者和用户提供稳定流畅的服务支持
- 完善的资源监控工具和警报系统，及时发现问题、进行性能分析，并采取必要的行动
- 严格的安全策略，确保资源和数据的机密性和完整性

整体社区基础设施架构图



7.2 一站式编译构建平台

背景概述

开源社区汇集了全球各地的开发者和贡献者，他们协作开发和维护各种项目，从操作系统到应用程序和库。这种开源模式已经改变了软件开发的方式，使得项目更具可访问性和协作性。社区的一站式编译构建集成平台用于协调和自动化软件项目的编译、构建和集成任务的综合工具。平台的主要目的是简化和优化开源项目的开发流程，以确保高质量的软件交付。

功能介绍

编译和构建自动化

支持自动执行编译和构建任务，无需手动干预。包括编译源代码、生成可执行文件或库文件、打包和部署软件。

多语言和多技术栈支持

支持多种编程语言和技术栈，适用于各种不同类型的开源项目。

持续集成 (CI)

支持持续集成，允许开发者频繁提交代码，并确保每次提交都经过自动构建和测试。有助于尽早检测问题并减少错误。

持续交付 (CD)

支持持续交付，可以自动部署到不同的环境，从测试环境到生产环境，实现快速交付。

多环境支持

支持适应多个操作系统和硬件平台，确保软件在不同环境下的稳定性和可用性。

报告和监控

提供详细的报告和监控工具，支持开发者了解编译和构建的进度，以及任何问题的详细信息。

优势和价值

提高开发效率

自动化编译和构建过程减少了手动操作的需要，使开发者能够更专注于编写代码。

保证软件质量

持续集成和自动化测试确保代码的质量，减少了潜在的错误。

快速交付

持续交付功能允许快速将新功能和修复推送到生产环境。

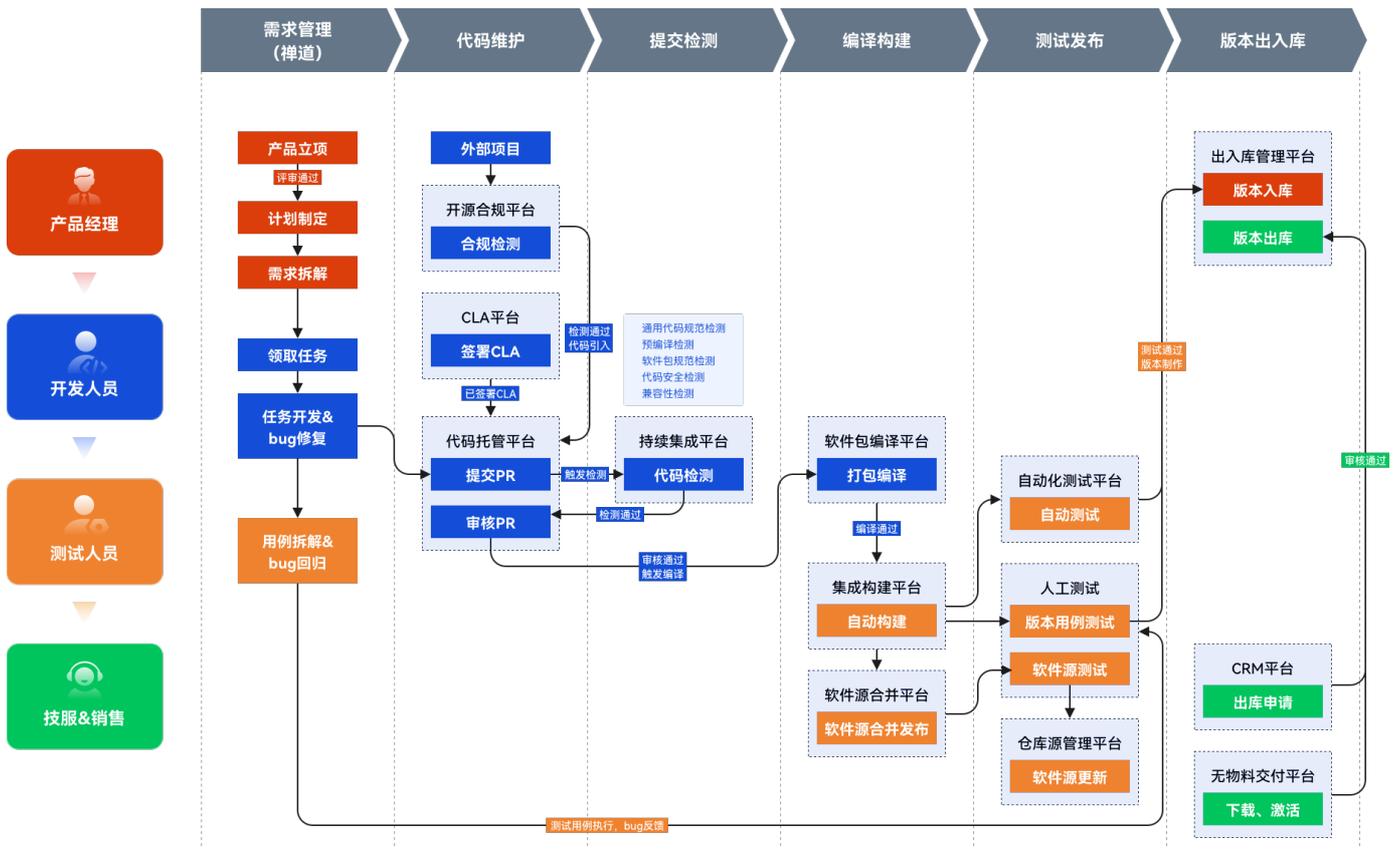
协同合作

提供了一个集成工具，有助于多个开发者协同工作，确保代码和构建的一致性。

跨平台兼容性

支持不同操作系统和硬件架构，使项目适用于更广泛的受众。

执行过程



7.3 开源合规管理平台

背景概述

开源合规管理平台旨在解决参与openKylin社区贡献的企业对已引用开源软件的元数据治理缺失，获取项目SBOM（软件物料清单）需求迫切，开源软件证书合规性、兼容性管理缺失，以及自开源建设以来开源合规法律风险性不断增加、开源软件片段引用尚未有效治理等相关问题，提供一套统一完整的开源合规管理解决方案。

功能介绍

单包扫描

单包扫描提供单个软件包的扫描能力，支持两种模式：

- 第一种模式，用户在前台界面上直接上传需要扫描的软件包文件，系统在接收到文件后，自动执行软件包的解包和扫描任务，并将扫描结果展示给用户；
- 第二种模式，用户在前台界面上输入软件包的下载URL，系统从该URL下载软件包，自动执行软件包的解包和扫描任务，并将扫描结果展示给用户。

Git仓库扫描

用户在前台界面上输入Git仓库的URL地址及分支，并配置相应的访问凭证（如用户名、密码、SSH密钥等），系统会

自动连接到指定的Git仓库，并递归地扫描仓库中的所有文件和提交记录。Git仓库扫描的常规流程如下：

- 初次扫描：对指定的git仓库进行全量扫描，识别所有的合规风险，并将结果存储到数据库中，建立仓库与风险的关联关系；
- 增量扫描：在后续的扫描中，系统会比对当前仓库状态与上一次扫描结果的差异，只对新增和修改的文件进行扫描，并更新数据库中的风险记录。

包列表扫描

用户在前台界面上输入多个软件包的下载URL，填写完成后系统自动从这些URL下载软件包，并对这些软件包执行解包和扫描操作。

API被动扫描

除了上述提及的单包扫描、Git仓库扫描和包列表扫描外，为了方便与外部系统集成，开源合规管理平台还提供了API被动扫描能力。支持第三方平台如Gitlab、Gitee、Gerrit等，通过该能力可以方便地将合规扫描接入到各自平台的流水线中。

优势和价值

通过实施该解决方案，企业可有效降低开源软件带来的合规性风险，提高开源软件的使用效率和质量，促进内部开源文化的发展，并更好地参与和回馈开源社区，实现商业利益与开源生态的双赢。

7.4 学习成长平台

背景概述

社区的学习成长平台为技术爱好者和开发者提供的宝贵资源，帮助他们获取知识、提升技能和参与开源项目的机会。平台不仅为个人提供了知识和技能的获取途径，还鼓励协作、分享和贡献。爱好者和开发者们能够积极参与社区项目，解决问题，分享经验，并一起推动技术的发展。社区的学习成长平台代表了开源精神，即知识应该是开放的，可自由获取和分享，为整个技术社区的进步作出贡献。

功能介绍

社区学习成长平台主要包括以下功能：

文档中心

文档中心是一个知识的宝库，包含了关于各种项目、工具和技术的详尽文档。提供了安装、配置、使用和故障排除的详细指导，以帮助开发者更好地理解和应用这些技术。

在线课程

社区的在线课程提供了深入学习的机会，涵盖了广泛的主题，从了解社区—加入社区—贡献社区各环节的支持，以各种形式呈现，包括视频教程、文字教程、手动实践。

源动实习生计划

“源动力实习生计划”是由openKylin社区发布的线上实习项目，旨在为高校学生及开源爱好者提供一个“辅导+实训+激励”的一站式开源实训平台，让学生在实际的开源项目需求中，验证自己学到的技能，通过实际案例提升项目经验积累，逐渐成长为优秀的开源人才。

成长收益

- 凭实习积分可获得实习工资。
- 资深导师指导参与社区贡献，与资深开发者交流学习，快速获得开发技能提升。
- 实习成果将集成进openKylin主线版本，供全球用户下载使用。优秀者将授予社区资深开发者的荣誉和额外激励，获得社区offer的机会。

编后语



随着数字时代的到来，世界发展的速度、软件的迭代和维护乃至创新都迎来了前所未有的挑战。我国高度重视开源产业发展，2021年，开源首次写入“十四五”战略规划，越来越多的产业力量开始关注开源、参与开源。

目前，openKylin开源操作系统根社区已聚合800多家单位会员，涵盖操作系统、数据库、办公软件、CPU、GPU、整机、人工智能优势企业及高等院校，拥有13000多位开发者、200多万社区用户遍布全球188个国家和地区。广泛的产、学、研、用各领域力量加入社区共建之中，也将为操作系统根技术创新奠定智囊基础。

未来，openKylin社区愿发挥平台支撑作用，聚合产、学、研、用及国内外开源力量，共同推动openKylin开源操作系统根社区建设，为加速构建众研众用众创的开源软件生态而不遗余力，欢迎广大伙伴、开发者与我们一起，携手走向同国际顶尖开源社区比肩的星辰大海。

致谢

最后，衷心感谢所有参与openKylin社区贡献的单位和个人，你们的贡献不仅为社区带来了丰富的资源和先进的技术，更为社区发展注入了强大的活力。期待未来，我们能继续携手合作，共创openKylin开源操作系统根社区的新篇章，为推动国产操作系统技术创新和生态繁荣贡献更多的力量。感谢以下在openKylin社区做出过贡献的单位：

麒麟软件有限公司
联想开天科技有限公司
北京技德系统技术有限公司
沐曦集成电路（上海）股份有限公司
北京中科通量科技有限公司
飞腾信息技术有限公司
国防科技大学
国家工业信息安全发展研究中心
清华大学
中科方德软件有限公司
先进操作系统创新中心
湖南麒麟信安科技股份有限公司
上海兆芯集成电路股份有限公司
海光信息技术股份有限公司
中电科申泰信息科技有限公司
北京面壁智能科技有限责任公司
北京麟卓信息科技有限公司
长沙景美集成电路设计有限公司
先进计算与关键软件海河实验室
深度数智科技（珠海）有限公司
上海鉴释科技发展有限公司
阿里巴巴达摩院（上海）科技有限公司
广东赛昉科技有限公司
北京算能科技有限公司
进迭时空（杭州）科技有限公司
深圳矽速科技有限公司
龙芯中科技术股份有限公司
此芯科技集团有限公司
英特尔亚太研究有限公司
武汉双椒派信息技术有限责任公司

深圳市研易科技有限公司
格兰菲智能科技有限公司
上海涛略信息技术有限公司
科大讯飞股份有限公司
深圳市亿图软件有限公司
北京海泰方圆科技股份有限公司
北京儒特科技有限公司
元心信息科技集团有限公司
北京数科网维科技有限责任公司
广州海格通信集团股份有限公司
四川川大智胜系统集成有限公司
普华基础软件股份有限公司
广州视睿电子科技有限公司
永中软件股份有限公司
江苏北弓智能科技有限公司
福建升腾资讯有限公司
天津光电通信技术有限公司
北京金山办公软件股份有限公司
北京高德品创科技有限公司
北京航空航天大学
上海交通大学
北京邮电大学
南开大学
东南大学
天津科技大学
武汉大学
.....
(以上排名不分先后)

商标声明

在本案例集中出现的商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。

免责声明

本案例集内容可能含有预测信息，包括但不限于有关openKylin社区未来的运营、版本规划、新技术发展等信息。由于实践中存在很多不确定因素，可能导致实际结果与预测信息有较大差别。因此，本案例集提供的信息仅供参考，不构成任何要约或承诺，不对您基于本案例集内容做出的任何行为承担责任。可能不经通知修改案例集内容，述不另行通知。

未经书面同意，任何单位和个人不得擅自摘抄、复制本案例集内容的部分或全部，并不得以任何形式传播。