



COSCon'25

第十届中国开源年会

众智开源 | Open Source, Open Intelligence

AOSP和Linux的深度融合方案探索

Open Fusion Desktop Environment-开源Linux桌面

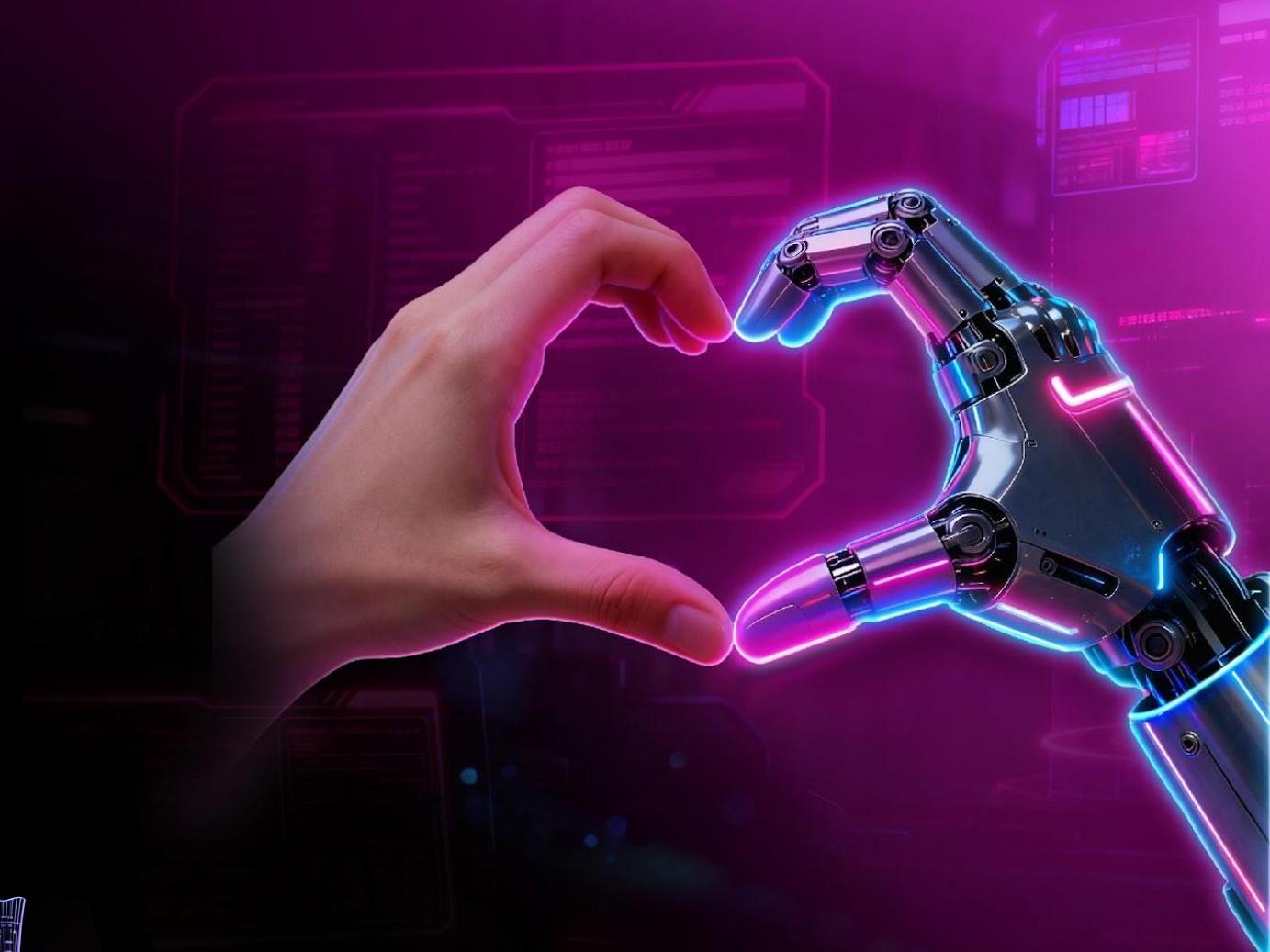
龚勇 社区负责人

gongyong1734@phytium.com.cn



CONTENTS

- 01 背景初衷
- 02 全新升级
- 03 未来规划
- 04 社区介绍



PART 01 背景初衷



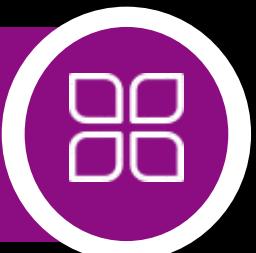


显性问题



- 商业软件‘不新’
- 游戏软件缺失
- 关键的‘生产力’软件缺失
- 外设驱动支持少

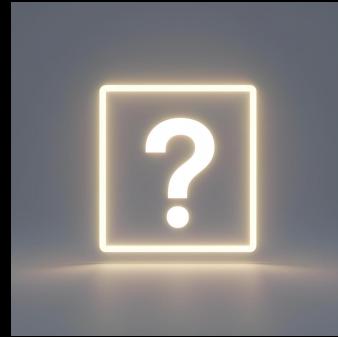
隐性问题



- 桌面软件兼容性问题
- 开发环境构建复杂



一个真正的“生产力”桌面



我们希望做到什么？



IDE 不再东拼西凑

统一的开发IDE环境
(**Android Studio**)，就在
桌面里面原生调试开发桌面应
用



海量应用生态

随时用到**Android** 生态中丰
富且最新的应用程序



开发随心所欲

依托文档齐备的 **AOSP** 框架，接
口标准统一，让开发者想怎么写就
怎么写。



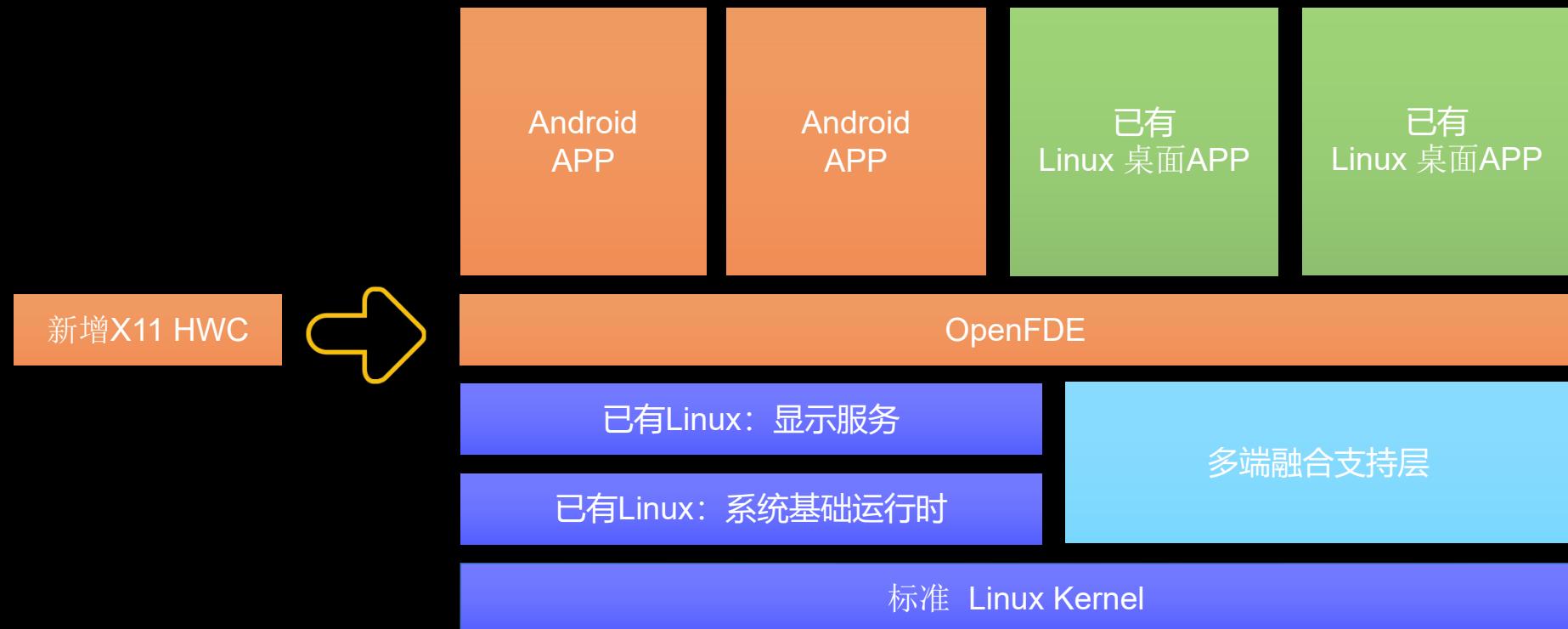
“传统”依旧

原有的**Linux GUI**应用，
Linux下方便的控制台，也是
我们的日常刚需

背景初衷



将AOSP统一的图形与API设计，对接在成熟、安全、可靠的Linux系统基础上，使得Android应用可以原生运行；同时，也能原生运行已有Linux应用



PART 02 全新升级



全新升级



核心升级

我们承诺升级至 AOSP 14，现已完成核心升级，也带来了更好的性能和新功能。



体验重构

解决了“仅仅能用”到“好用”的跨越。打造简洁优雅质感体验，构建桌面风格专属应用，键鼠映射全面优化。



开发生态

打通了工具链。现在，你可以直接在 OpenFDE 上运行 Android Studio 并原生调试应用。

AOSP版本升级问题汇总



FDE-X11核心升级

01

窗口管理器优化和扩展——参照XFWM4, 实现扩展协议(netwmstate, 最大化, 最小化)的支持 (文件复制android和linux之间的复制)

02

优化生命周期(定制android framework), 窗口类型

03

DRI3重构(兼容更多gpu和系统差异 kylin和uos)

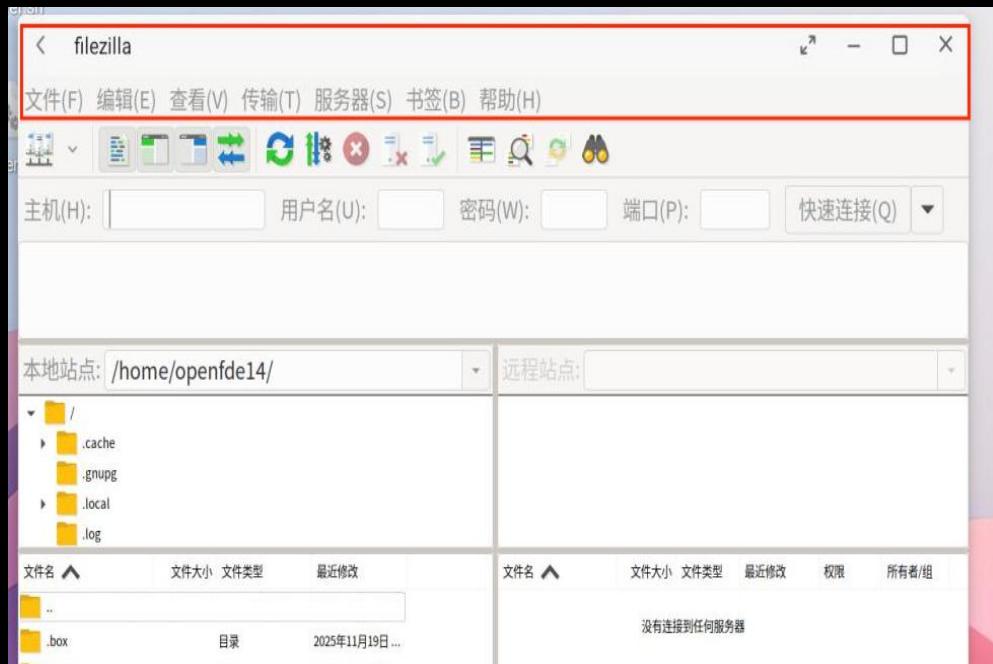
04

服务化(默认随系统启动, 简化使用方式)

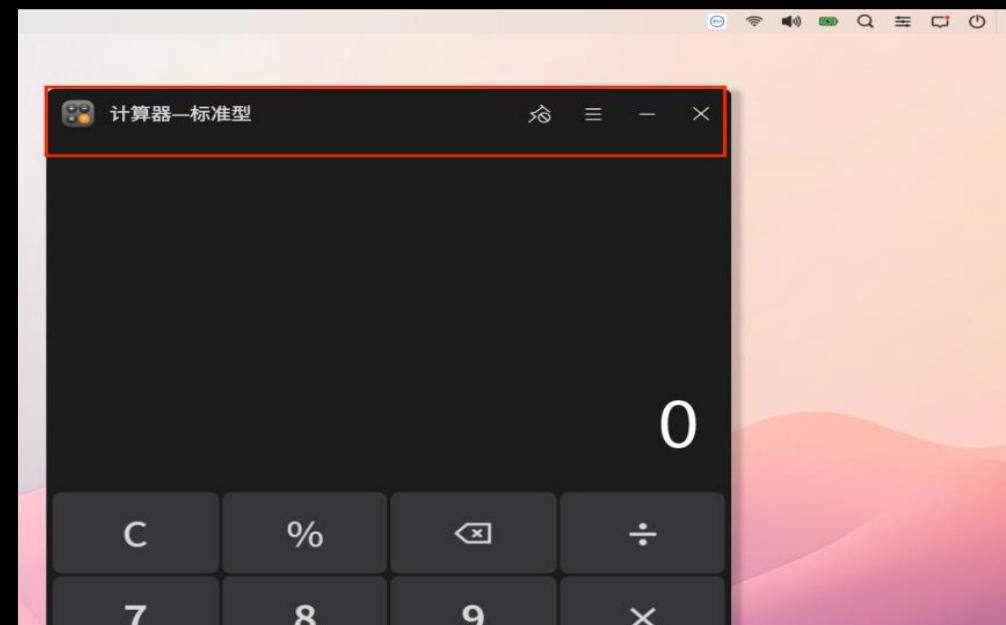


FDE-X11窗口扩展协议支持

之前Android和Linux 标题栏都需要显示



现在通过协议支持，可以只显示Linux标题栏





FDE-X11 窗口扩展协议支持

策略 vs. 机制

X 服务器提供“机制”（绘制窗口），但布局“策略”的执行被委托给窗口管理器（WM）。标准的 X11 并不强制规定窗口行为。

EWMH / NetWM 标准

扩展窗口管理器提示 (EWMH) 定义了现代桌面环境的标准交互：

最大化：客户端发送 `_NET_WM_STATE` ClientMessage 到根窗口，附带如 `_NET_WM_STATE_MAXIMIZED_VERT` 等原子 (Atom)。

最小化：客户端使用 `XIconifyWindow` (触发 `WM_CHANGE_STATE`) 或 `_NET_WM_ACTION_MINIMIZE`。

客户端应用 (Linux 应用)

发起请求 (最大化/最小化)

Xserver (FDE-X11)

路由至 SubstructureRedirect

窗口管理器 (FDE-WM)

计算策略和几何尺寸

Xserver (FDE-X11)

更新窗口状态

客户端应用 (Linux 应用)

接受事件并重绘

FDE-X11 直接渲染支持

利用 DMA-BUF 实现客户端与 X Server 间的高效零拷贝缓冲区共享。

直接硬件访问

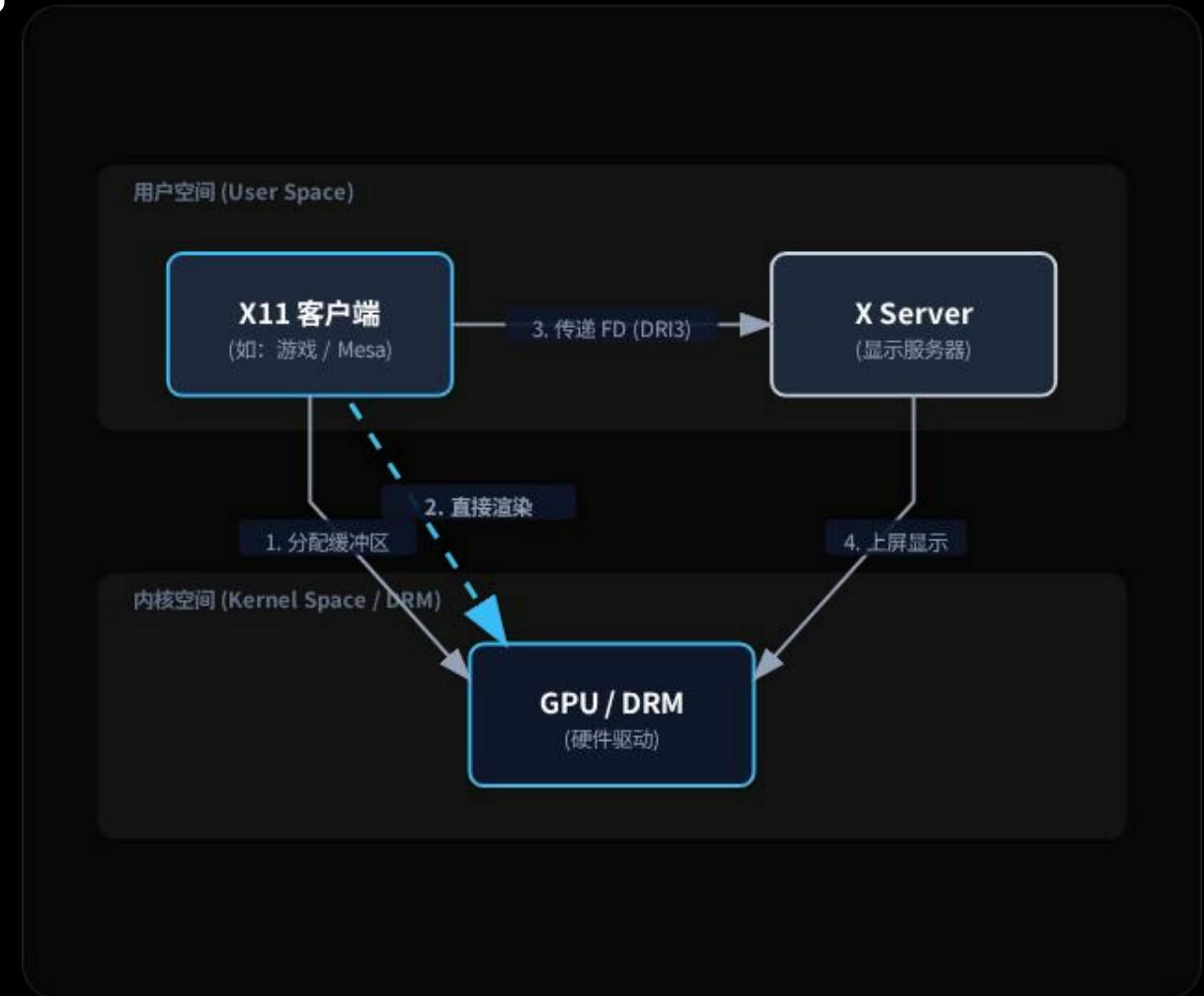
客户端将渲染指令直接发送给 GPU，完全绕过 X Server 这一瓶颈。

零拷贝性能 (Zero-Copy)

通过传递 DMA-BUF 句柄共享显存缓冲区，无需在进程间进行昂贵的像素拷贝。

高效同步 (Sync Fences)

显式同步机制确保了画面无撕裂，且不会阻塞整个显示服务器的运行。



全新升级



开源社
kaiyuanshe

FDE-X11 直接渲染支持



借助飞腾PhyBin + Wine
运行Windows游戏。

FDE-X11通过DRI3模式，
流畅显示游戏画面。

全新升级



打造简洁优雅质感体验



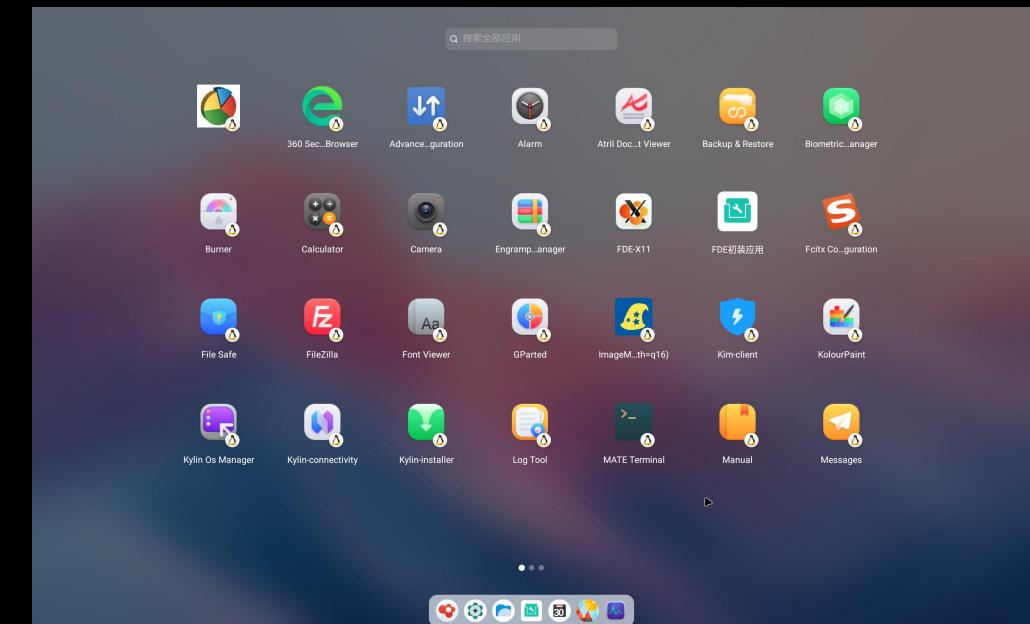
FDE新桌面



FDE老桌面

Status: 重设计图标354个，涉及页面100多页，涉及应用6个（Launcher、Systemui、DocumentUI、Settings、TaskResource、SetupWizard）。

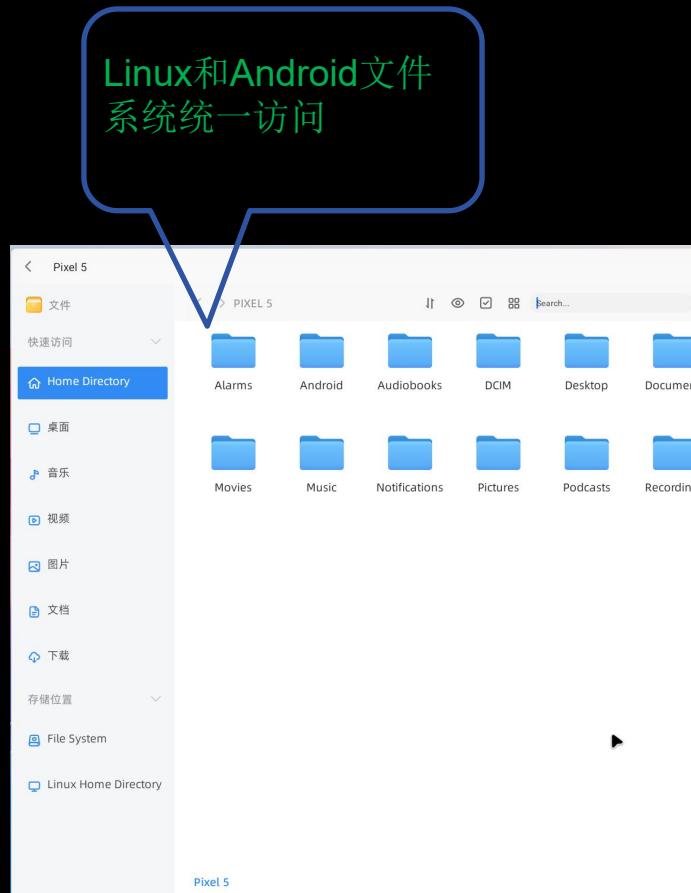
高斯模糊在各系统中已广泛用于UI美化，AOSP从12开始做了接口级支持



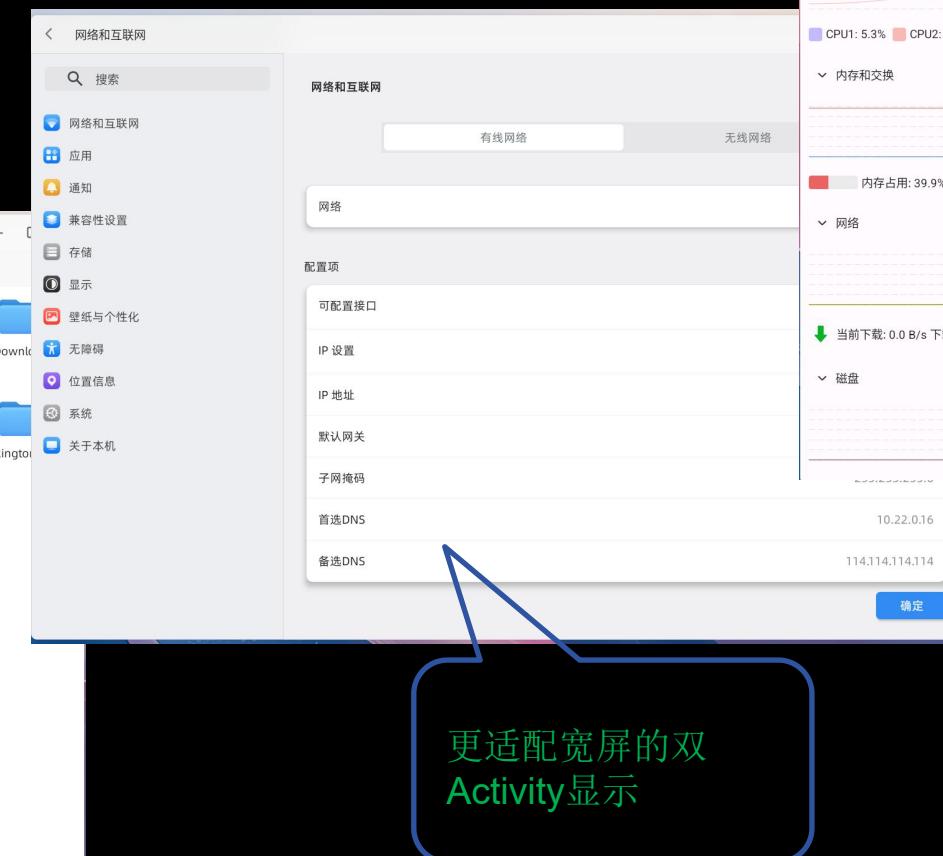
全新升级



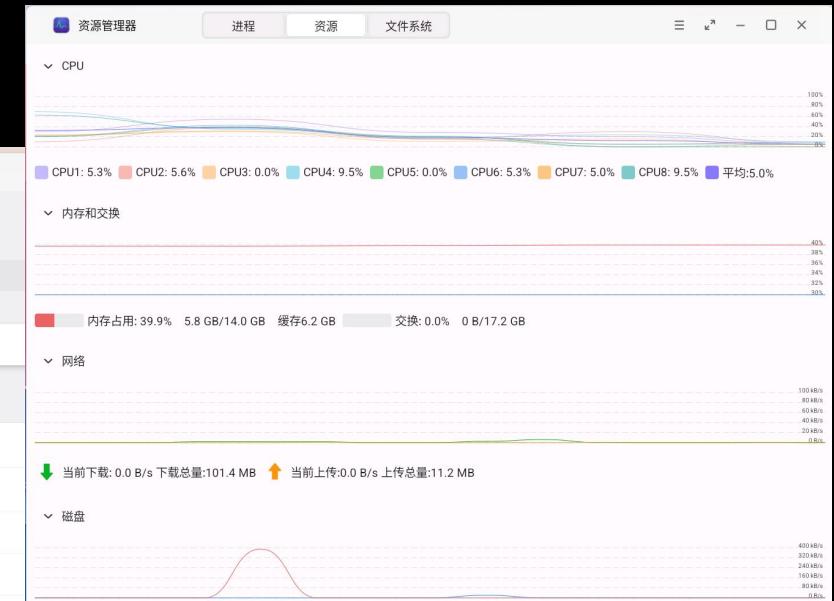
构建OpenFDE桌面专属应用



Linux和Android文件
系统统一访问



更适配宽屏的双
Activity显示



PC更习惯的资源管理器

全新升级

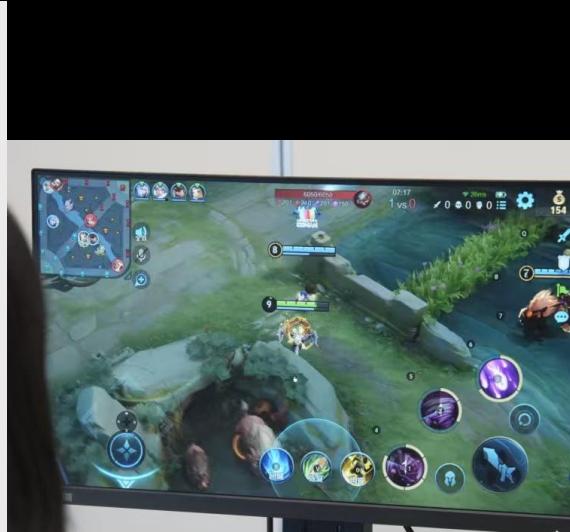


按键助手提升游戏操作体验 助力飞腾工会游戏大赛



腾锐D3000+OpenFDE强势驾驭王者荣耀，
赋能员工精彩生活

国芯护航，峡谷争霸





全网首个移植Android Studio到ARM Linux

已在飞腾D3000/D3000M硬件平台成功编译和运行



编译工具链

- › 实现 Android Studio 完整交叉编译流程
- › 重建编译工具链，彻底替换 X86 依赖
- › 基于交叉编译环境，深度定制 Clang SDK/NDK



核心攻坚：UI 渲染

- › 解决最困难的 UI 预览渲染问题
- › 基于 AOSP 源码进行底层适配
- › 关键依赖库 LayoutLib 深度适配
- › Skia 的Raster模式减少依赖



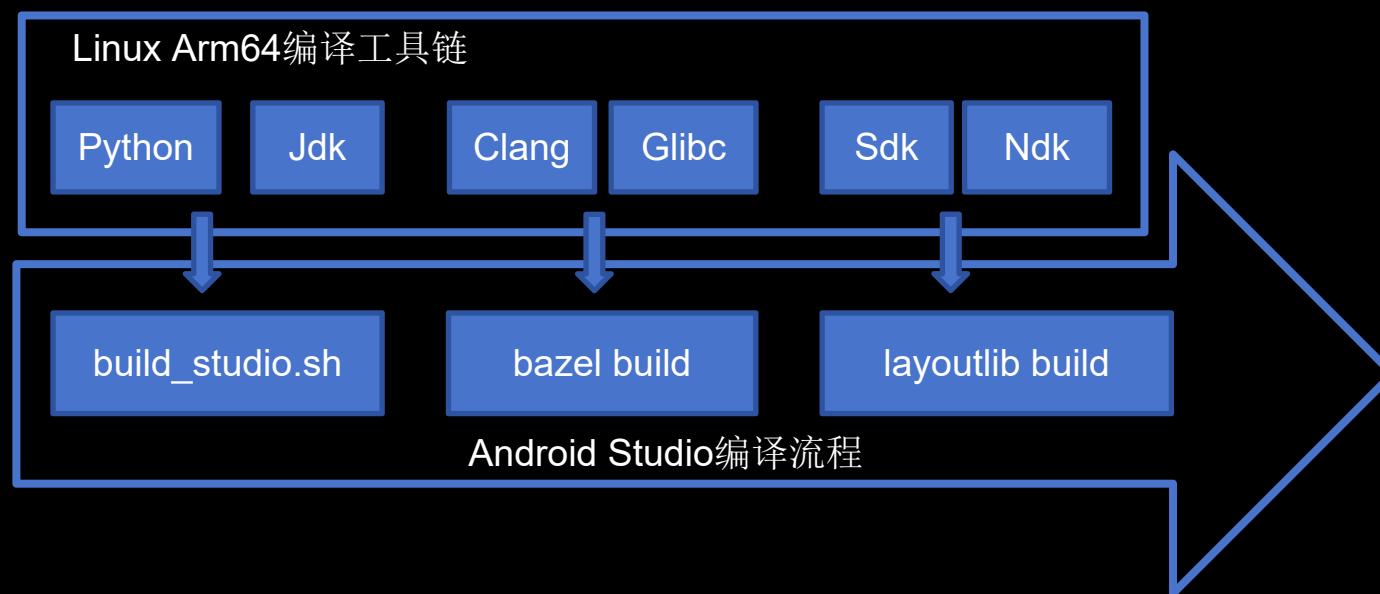
Studio 编译

- › 克服官方文档严重滞后问题
- › 源码工程目录变更
- › 命令行编译效率高于 UI

01 编译工具链准备：

Python/JDK/GCC可直接从官方渠道获取对应的Linux ARM版本完成替换。

Clang/Android-SDK/Android-NDK缺乏官方的ARM架构支持，需通过深入分析构建脚本，基于交叉编译方法进行定制化改造，以实现ARM平台的目标构建。



02 studio和插件编译：

官方资料过时，studio源码目录变更

编写Bash脚本，对源码中的依赖JAR定义进行批处理

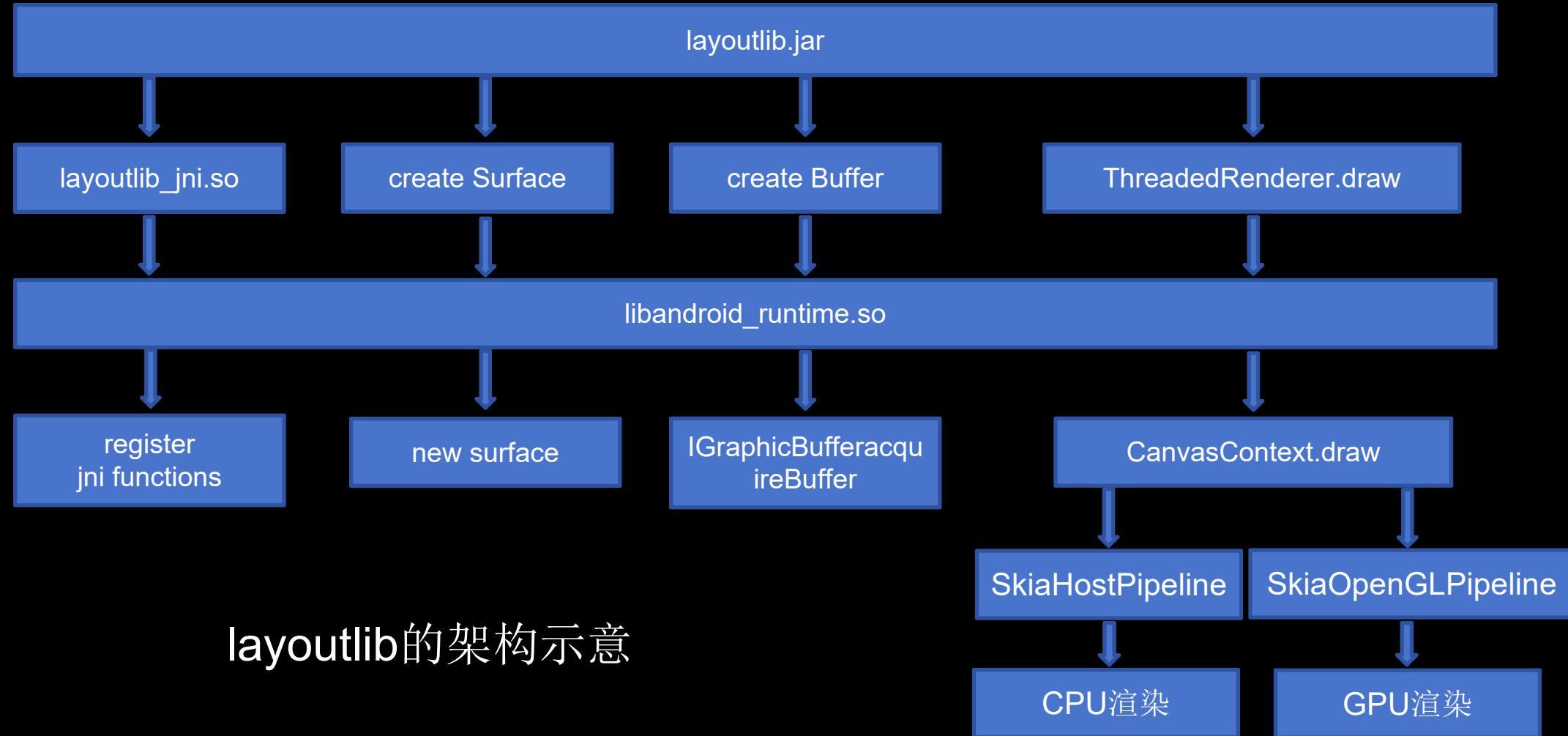
03 layoutlib编译：

找不到layoutlib动态库的源码，网络上也无任何相关的参考资料。

AOSP源码中libandroid_runtime动态库存在深层的依赖链，且无法运行在Linux平台。

通过在Android Studio源码的layoutlib仓库提交记录中追溯，成功定位到layoutlib_jni.so及layoutlib.jar对应的源码仓库与分支。

经过内部技术讨论，确定可行方案是基于AOSP源码，使用Skia的Raster模式在Linux平台上进行CPU渲染绘制UI。



全新升级



打造和Windows连贯的开发流程一样的体验，形成了闭环。

Windows 开发 Windows

```
public class EmojSearch
{
    public string TearsOfJoySearch(List<string> tweetsToSearch)
    {
        string specificEmoji = tweetsToSearch[0];
        return specificEmoji;
    }

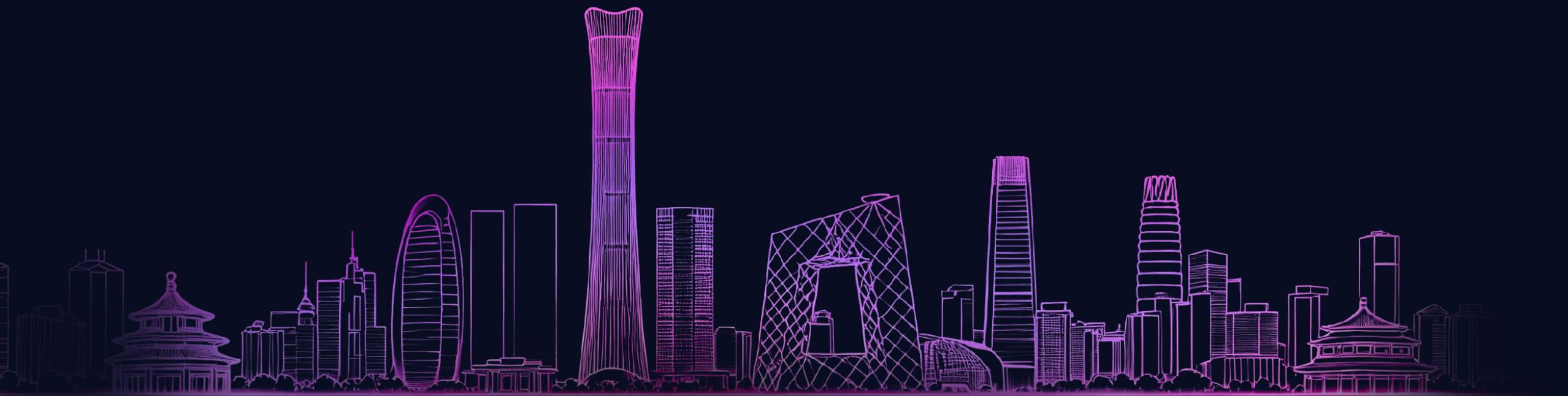
    public int SpecificEmoji(List<string> emojiToSearch)
    {
        if (emojiToSearch == null)
        {
            throw new ArgumentNullException();
        }

        int numberFound = 0;
        for (int i = 0; i < emojiToSearch.Count; i++)
        {
            Match match = new Regex(emojiToSearch[i]).Match();
            if (match.Success)
            {
                numberFound++;
            }
        }
        return numberFound;
    }
}
```

Android 开发 Android

```
public class MainActivity extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

PART 03 未来规划





01

复用Linux图形驱动

基于已实现的libHybris架构，
进一步实现virtgpu和英伟达
gpu的Linux驱动复用



02

支持更多发行版和硬件

Arch Linux
Mac M系列PD虚拟机
树莓派5



03

更多硬件直通

深度复用 Linux 原生硬件驱动，
实现硬件直通（wifi，蓝牙、
usb）

PART 04 社区介绍





社区官网: openfde.com

代码仓库:

gitee.com/openfde

github.com/openfde

欢迎加入!



关注b站
查看实际运行效果



关注公众号
了解更多信息



COSCon'25

第十届中国开源年会

众智开源 | Open Source, Open Intelligence

Thanks

龔勇

OpenFDE 社区负责人

gongyong1734@phytium.com.cn

