

# 由 deepin 视角，看 RISC-V 桌面生态的未来

当我们在谈论桌面生态时，我们在谈论什么？

deepin 开源社区 / deepin-ports SIG

RVSC 2024.08

# 目录

00 简介

---

01 DEEPIN RISC-V 的主线化之路

---

02 跨越嵌入式与桌面的鸿沟

---

03 DEEPIN 与 RISC-V 桌面的未来

---

# 00 简介



(常用头像)

## 杨畅

统信软件 / deepin 开源社区研发工程师

当前工作：

- V23 主线多架构支持 (RISC-V)
- V23 RISC-V 设备适配、镜像发布
- V23 多架构软件包适配和维护

<https://ewe.moe>

GitHub/常用 ID: @YukariChiba

*a.k.a. 羊*

# 01 deepin RISC-V 的主线化之路

deepin-ports SIG 如何完成它的使命

2023 年 10 月，为推进 RISC-V 和 LoongArch 进入 deepin 主线，deepin-riscv 更名 deepin-ports

## deepin-ports 的目标

为 deepin (V23) 操作系统适配 riscv64/loong64 架构，  
为 RISC-V/LoongArch 架构芯片提供更多的桌面应用。

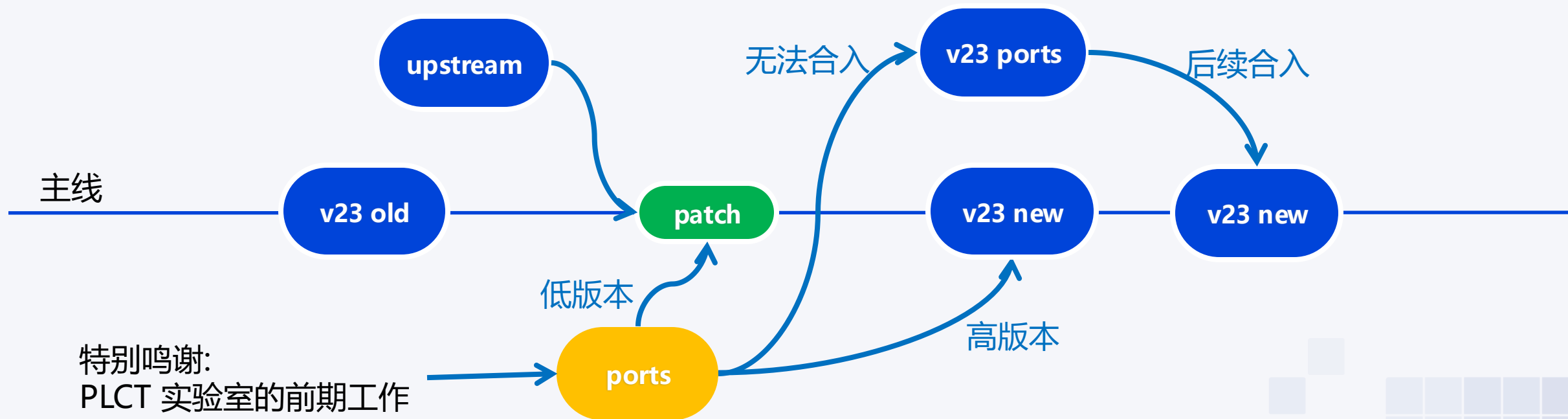
## deepin-ports 的成果

适配市面上大部分可运行桌面环境的 RISC-V 设备和开发板。  
deepin V23 主线支持 RISC-V 架构，并进入 Preview 支持阶段。  
构建、打包、镜像生成基本实现自动化，建立了 ports 附加仓库提供良好打包的驱动和内核。

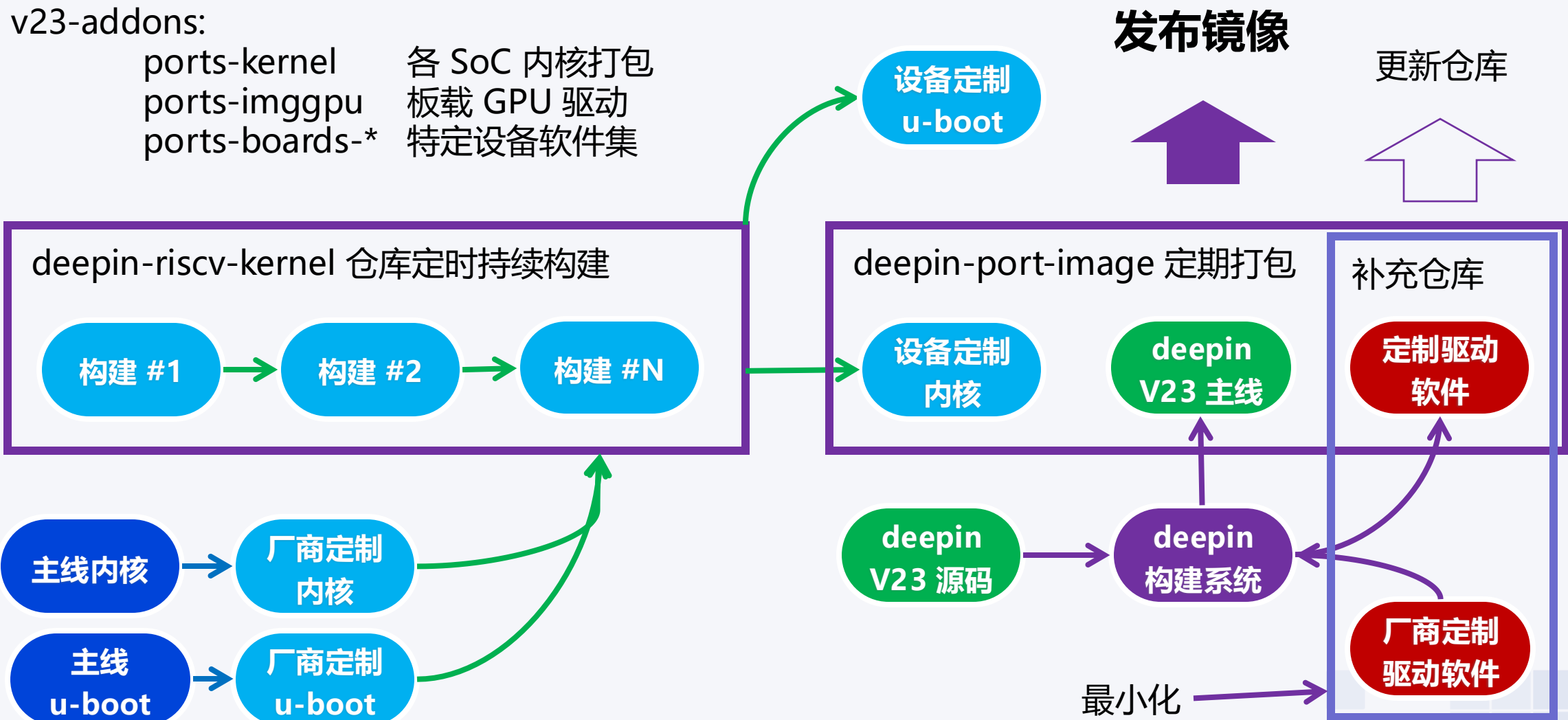
# DEVELOPMENT PATH



0. 在 deepin 主线构建出 RV64 的基本工具链
1. 将 deepin-ports 的仓库的 patch (几乎) 完全合并入主线
2. 将 deepin-ports 必需的高版本软件包在主线中 (几乎) 完全升级
3. 将 deepin-ports 中无法合入主线的小部分软件包在主线的分支中良好地维护, 最终合入主线







## RISC-V新进展!

deepin成功适配VisionFive2

## 2023 第三届 RISC-V 中国峰会 RISC-V Summit China

2023.08.23-25 | August 23-25, 2023  
中国·北京 | Beijing, China

## deepin V23 成功适配 SpacemiT MUSE™ Box

## RISC-V 新成果!

deepin 成功适配 LicheePi 4A

## S<sup>2</sup> 开源操作系统 年度技术会议 OS2ATC

deepin受邀参加主题演讲

deepin

## deepin加入甲辰计划

共建RISC-V繁荣生态

## mesa LLVMpipe ORCJIT上游化的 台前幕后

立即查阅

deepin

## deepin V23 新进展!

成功适配 MilkV Mars

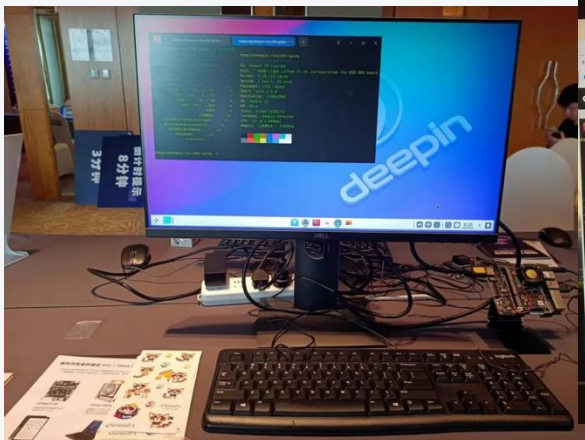
deepin

## deepin V23 成功适配 奕斯伟计算EIC7700X

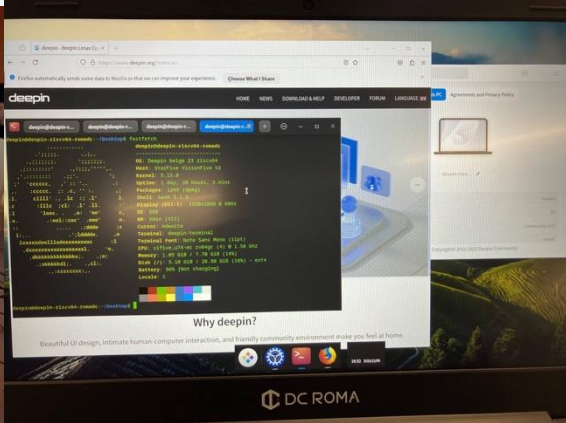


# 01.06: 2023-2024 合订本：截图和照片

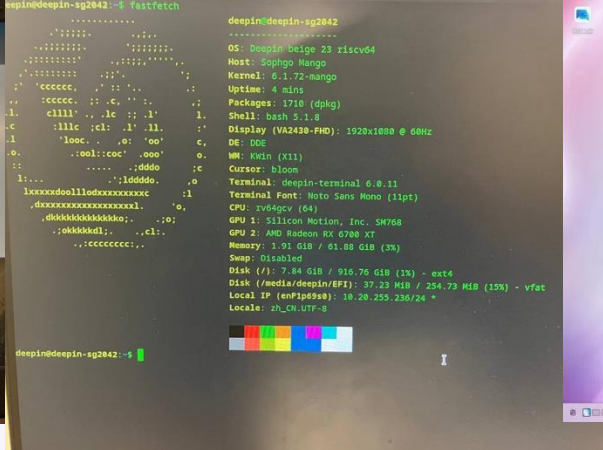
deepin



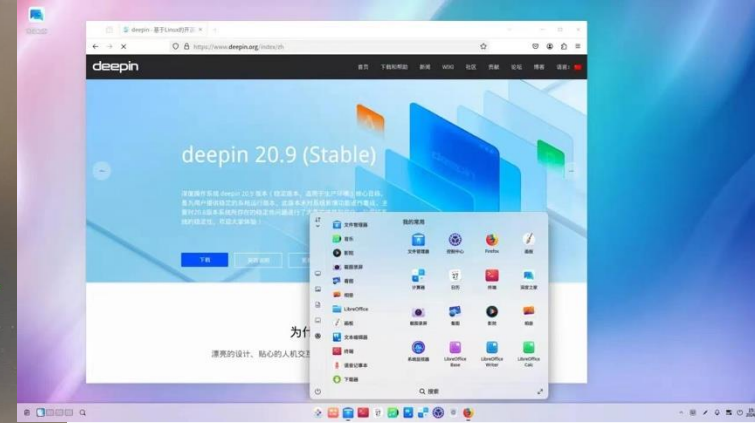
deepin V23 在 RISC-V 中国峰会 (2023)



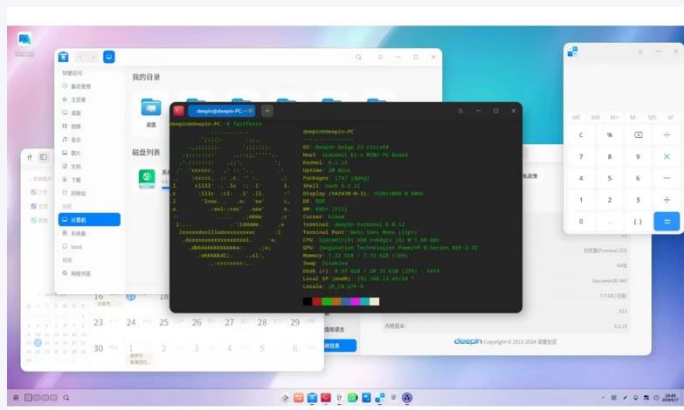
deepin V23 on DC ROMA (2023)



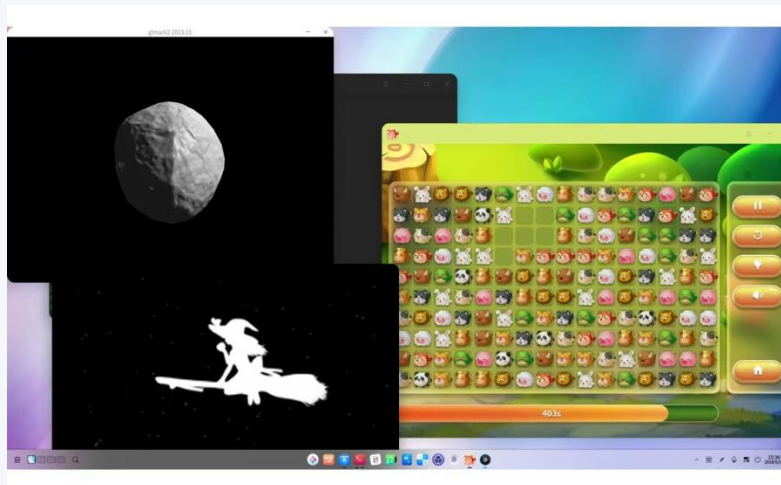
deepin V23 on SG2042 (2023)



deepin V23 on EIC7700 (2024)



deepin V23 on Spacemit muse box (2024)



deepin V23 on MilkV Mars (2024)



我 at OS2ATC (2024)

由 deepin 视角，看 RISC-V 桌面生态的未来：当我们在谈论桌面生态时，我们在谈论什么？

ESWIN



苦芽

milkv

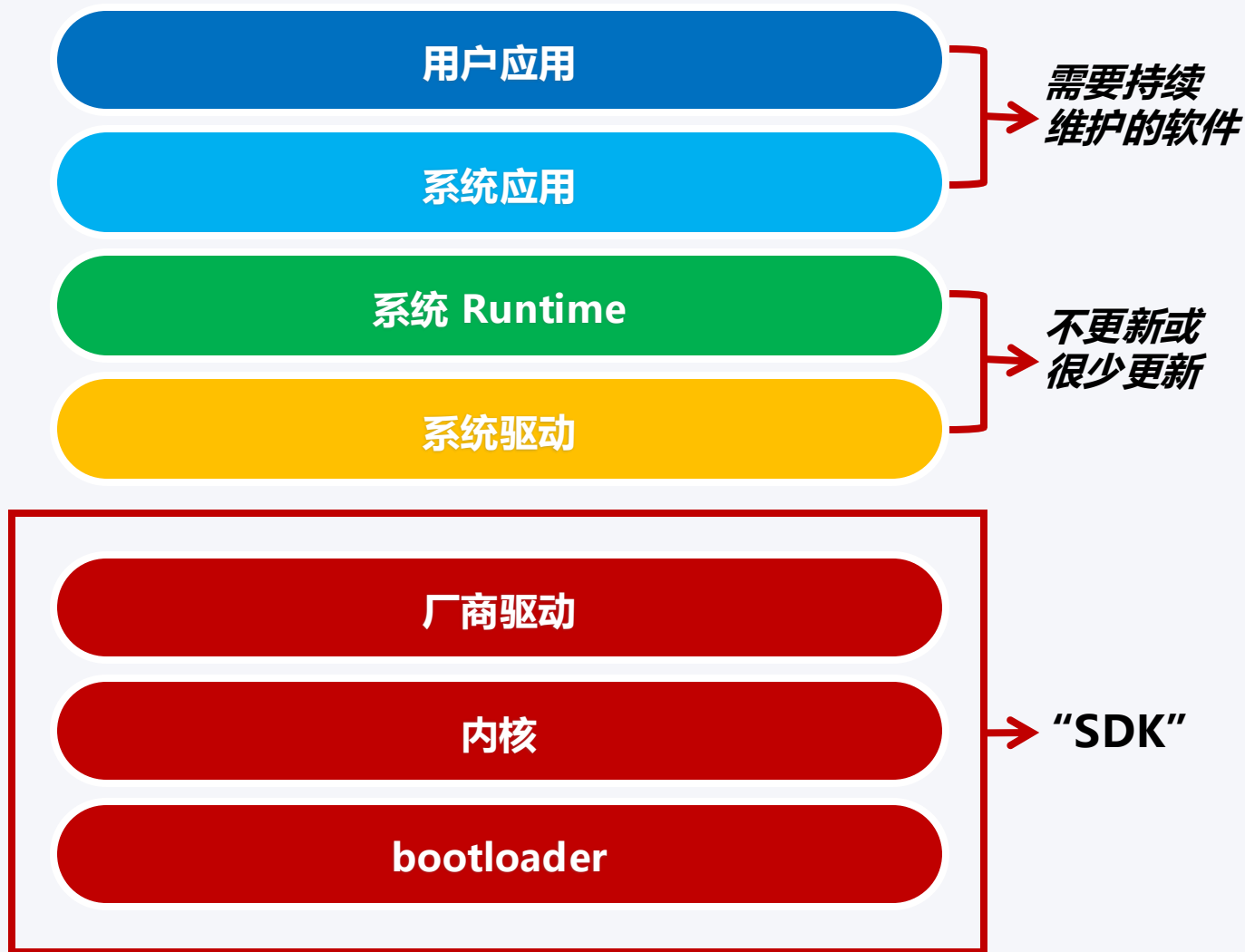


DeepComputing



## 02 跨越嵌入式与桌面的鸿沟

RISC-V 的嵌入式原罪及其影响



## 嵌入式系统架构

- 假定存在类似 HAL 的兼容层，分离操作系统上层与底层驱动
- 操作系统不关心底层驱动和内核
- 驱动和运行时固化，一般不更新
- 甚至操作系统本身没有包管理机制
- 底层被打包为单一的“SDK”，不需要开源或开源意义不大
- “烧录”系统

图例

- 主线化开源软件
- 非主线化开源软件
- 可能闭源的软件
- 闭源软件

既不能支撑复杂的桌面应用，也不是完整操作系统，更没有真正的桌面生态

由 deepin 视角，看 RISC-V 桌面生态的未来：当我们在谈论桌面生态时，我们在谈论什么？





## 桌面系统架构

- 包管理器的（几乎）完全控制，良好的拆分打包和严谨的依赖管理
- 尽可能使用主线软件包，更少的非主线软件包，开源意义很大
- 内核和底层驱动也由操作系统管理
- 较为频繁的系统更新和维护
- 整合内核驱动，拆解厂商“SDK”
- “安装”系统

### 图例

- 主线化开源软件
- 非主线化开源软件
- 可能闭源的软件
- 闭源软件

**boot分区 extlinux.conf vmlinuz dtb initrd 缺一不可**

- 内核使用 FIT Image 格式
- 没有或嵌入 FIT Image 的 initrd
- u-boot 写死启动参数
  - 没有启动菜单 (extlinux.conf)
  - (bootm / booti)
- boot 分区简单甚至没有文件系统
- 分区基本写死，无法调整
- 分区形式嵌入式化
  - vendor root\_a root\_b userdata

*/fitimage*



操作系统友好的设计：

<i>/(boot)</i>	<i>&lt;- 这是 fat32/ext4 分区</i>
<i>/vmlinuz-6.6-riscv</i>	<i>&lt;- 这是内核，是软件包装的</i>
<i>/initrd.gz-6.6-riscv</i>	<i>&lt;- 这是 initrd，是系统生成的</i>
<i>/dtbs</i>	<i>&lt;- 这里放 dtb</i>
<i>/linux-image-6.6-riscv</i>	<i>&lt;- 这是软件包装的</i>
<i>/nya-nya-nya.dtb</i>	
<i>/extlinux</i>	
<i>/extlinux.conf</i>	<i>&lt;- 这是引导菜单，是系统生成的</i>

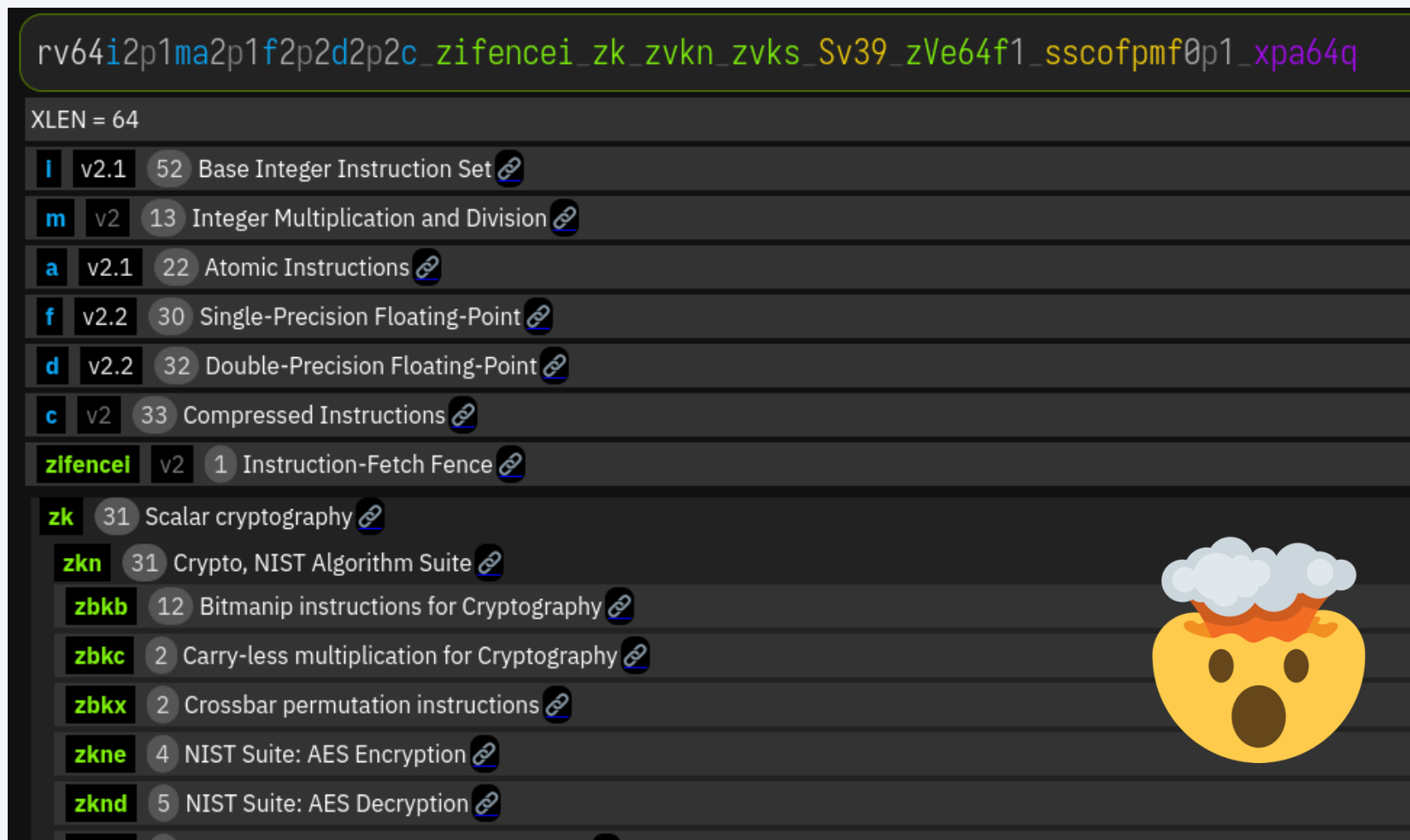
UEFI: When? (non-uboot) ACPI: When?  
(Ref: BRS-I)



# 我们需要能用\*的 PCI-E 接口！

显卡、网卡、存储的板载设备替代方案和扩展能力

\* 指不丢中断、一致性正确、带宽足够



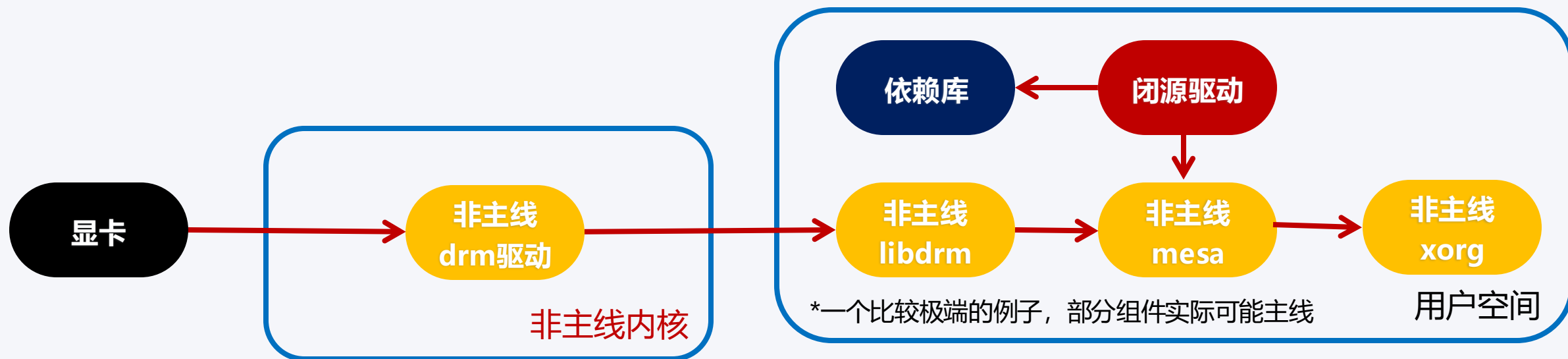
- 巨长的 ISA String
- 厂商私有拓展
- 扩展支持不一
- 分裂生态的修改
- 基线、设备迭代缓慢

*deepin 基线: rv64gc*  
(取最小值, 短板效应)

- 与主流工具链保持一致
- 一次构建同架构到处运行
- 观望, 等待设备的迭代



*RISC-V 的指令集架构有些松弛，  
但是嵌入式生态又弥补了这一点，  
现在如果要推进桌面平台的话，  
就会显示出生态上的短板，  
可能出现扩展特性用不上的情况，  
现在最好的办法就是在推出带更多拓展的 IP 的同时，  
顺便做一个基线的调整。*



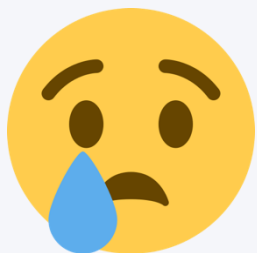
代价:

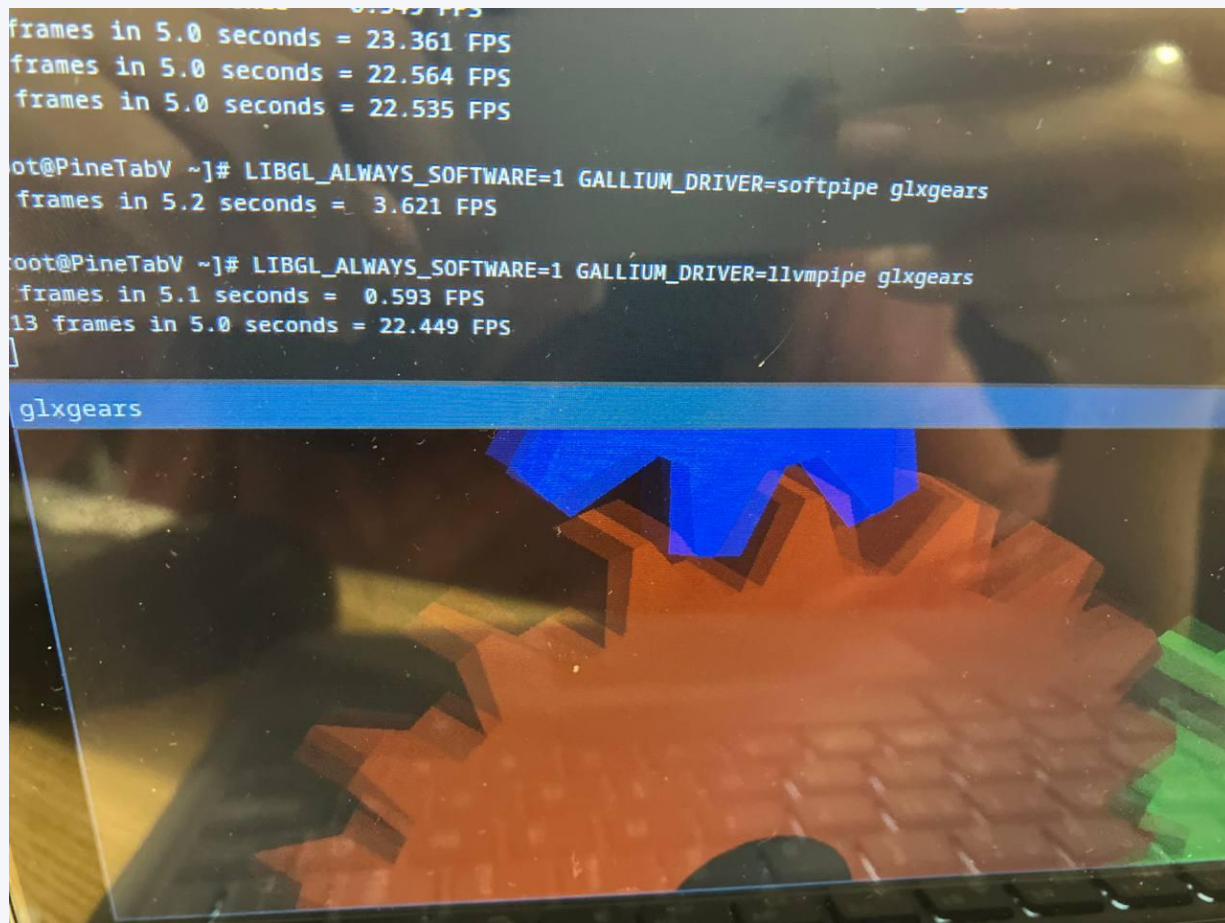
内核驱动难以主线化  
内核不统一影响发布

代价:

需要额外仓库、分发闭源软件包  
主线依赖库版本被锁死  
主线化难度大

部分厂商特有: 不同型号 GPU 驱动不通用





## ORCJIT Patch 的（再次）主线化尝试

内置 GPU：闭源驱动版本落后，Patch 无法主线  
• 弃用！

glxgears: 3.5 FPS --> 22.5 FPS : near 6.5x

特别感谢：

Origin PR            by        @alexfanqi  
Shader Cache        by        @icenowy

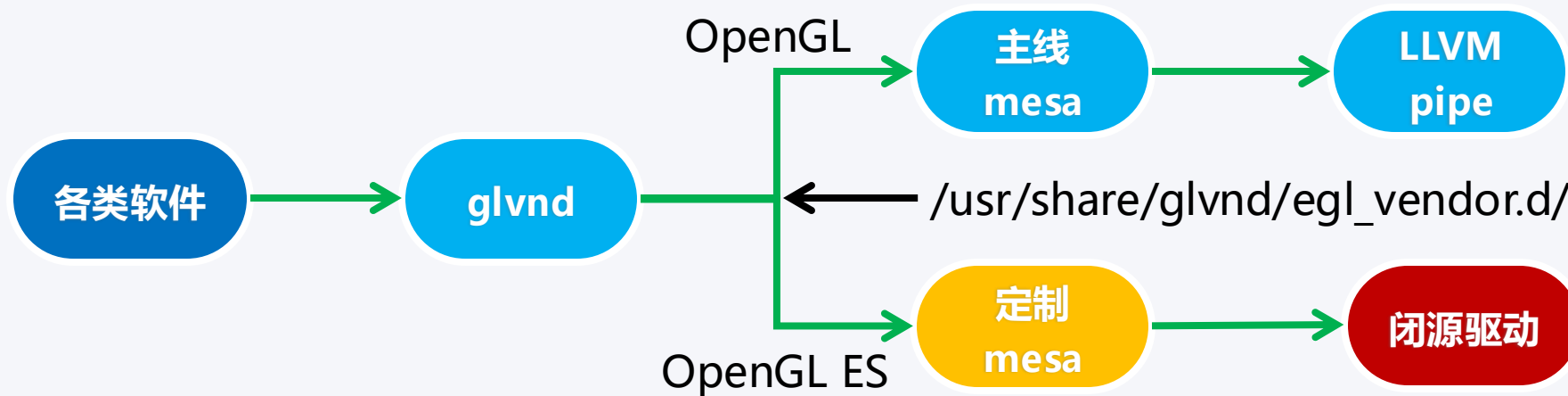
mesa !26018 Merged: 2024.07.16  
在 mesa 24.2 可用

大部分 RISC-V 设备只支持 OpenGL ES:

wayland 不等于 GLES only

deepin 打包策略:

不允许低版本 mesa 覆盖系统 mesa



特别感谢: @icenowy

已实装于 v23-addon 仓库  
并默认在具有内置 GPU 的设备镜像上启用  
大大加速了驱动适配速度

图例

- 主线化开源软件
- 可能闭源的软件
- 闭源软件



### 主动开源

- 各型号下统一的驱动
- 内核态驱动进主线内核
- 用户态主线化进主线mesa
- 开心的用户和开发者



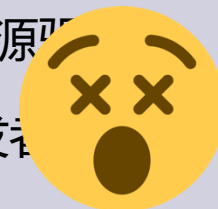
### 被动开源

- 来自社区的逆向
  - asahi
  - lima
  - nouveau
- 兼容的开源方案
- 魔法



### 不开源

- 桌面用户大多是社区用户，开源社区不会为驱动买单
- 桌面平台给予用户自由选择平台组件的权力
  - PCI-E 与跨架构开源驱动
- 不开心的用户和开发者



### 闭源的内核模块等同无法使用

- 重编译的 localversion: 避免包命名冲突
- 启用或关闭的 config: 启用桌面需要的功能 (cgroup, ipv6, 桌面 GPU)
- 增加或删除的 patch

### 闭源的/过时的非主线用户态驱动难以使用

- 软件包的版本依赖
- 非主线、不可合并的 patch

### 开源社区没有必要不惜代价地去兼容某一闭源硬件

**保持主线，宁缺毋滥。**



# 03 DEEPIN 与 RISC-V 桌面化的未来

RISC-V 如何拥抱真正的桌面生态

### deepin 及其它开源 Linux 发行版：保持更新

- 版本选型充分考虑 RISC-V 等新架构的优化
- 更新策略适当激进，适配新特性，尽量使用最新工具链
- 积极采纳上游补丁
- 尽快迁移过时依赖，降低升级成本

建立 RISC-V 设备适配的生命周期，及时淘汰过时硬件，卸掉历史包袱。

所谓“适配”  
究竟是什么意思？

坚持原则  
增加复杂度  
更长维护周期

VS

放弃原则  
快速发布  
较短维护周期

适配九宫格	必须是 主线化的系统	不一定主线化 可以稍微落后/有区别	什么是主线？
必须是以 维护良好的软件包 形式安装的系统	主线仓库 良好维护的定制仓库 自动化的镜像打包	fork 到 port 分支 改一改打镜像也叫适配	只要依赖兼容 把自己系统软件包 放到隔壁系统上也算适配
只要构建过程 可复现就行了	在主线镜像里直接塞入 vendor 提供的二进制 就可以适配啦	把厂商整个 SDK 往系统里覆盖， 补上依赖就好啦	buildroot/yocto 加上自己的 系统组件就算是适配了
镜像不是只要刷上 能跑就行了吗？	手动配置完系统 直接导出分区镜像 脏一点没事	现场在机器上 sudo make install 覆盖适配出来的 系统也能用	亲手从源码打出来 的系统怎么能不叫适配

**以“和”为贵： 与操作系统原有生态和睦相处**

**AI、NPU、大模型： 先做好桌面体验再谈有效利用**

- 桌面产品是 RISC-V 不可避免的一环
- 没有良好的桌面生态，AI 是无法落地的空中楼阁

**专有软硬件： 使用紧跟主线、非侵入式的方式对接系统软件**

- ffmpeg codec

A faint, pixelated world map in a lighter blue shade, centered on the Atlantic Ocean, serving as a background for the text.

# Thanks for listening

愿未来所有的开源指令集架构都能平等地共享能用、好用的桌面生态！

(此处可提问)