

2 0 2 1

# Book OS 介绍

--梦想开始的地方

汇报人：胡自成  
重庆师范大学大三学生  
B站：我叫小骨头

2021.3.21

# 致谢

李石亮老师 陈莉君老师



# 参与开源贡献的小伙伴



hzcx998

633 commits



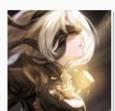
huloves

97 commits



ZhUyU1997

28 commits



NightfallDM

2 commits



st127

1 commits



张萌

孤独



# 目录

CONTENTS

- 01 BookOS的前世今生
- 02 效果演示
- 03 内核及OS功能介绍
- 04 用户态lwip网络协议的封装
- 05 view图形框架



# 01

O N E

## BookOS的前世今生

--有的人生来就是改变世界的





# 发展史

## HISTORY

- ◆ “书是人类进步的阶梯。”--高尔基
- ◆ 基于x86平台的32位PC操作系统
- ◆ 2018.11.10 - 使用BookOS作为操作系统的名字
- ◆ 2019.7.8 - 使用XBook作为操作系统的名字
- ◆ 2020.5.8 - 使用XBook2作为操作系统的名字
- ◆ 2021.2.5 - 使用BookOS作为操作系统的名字，  
xbook2作为操作系统内核的名字，OS和内核  
分离开发。



开源给我们带来了便利，我们开源给别人带来便利。

BookOS和xbook2都是基于MIT协议进行开源  
可以自由使用和修改，方便学习使用

官网：[www.book-os.org](http://www.book-os.org)

<https://gitee.com/hzc1998/BookOS>

<https://gitee.com/hzc1998/xbook2>

<https://github.com/hzcx998/BookOS>

<https://github.com/hzcx998/xbook2>



# 交流群





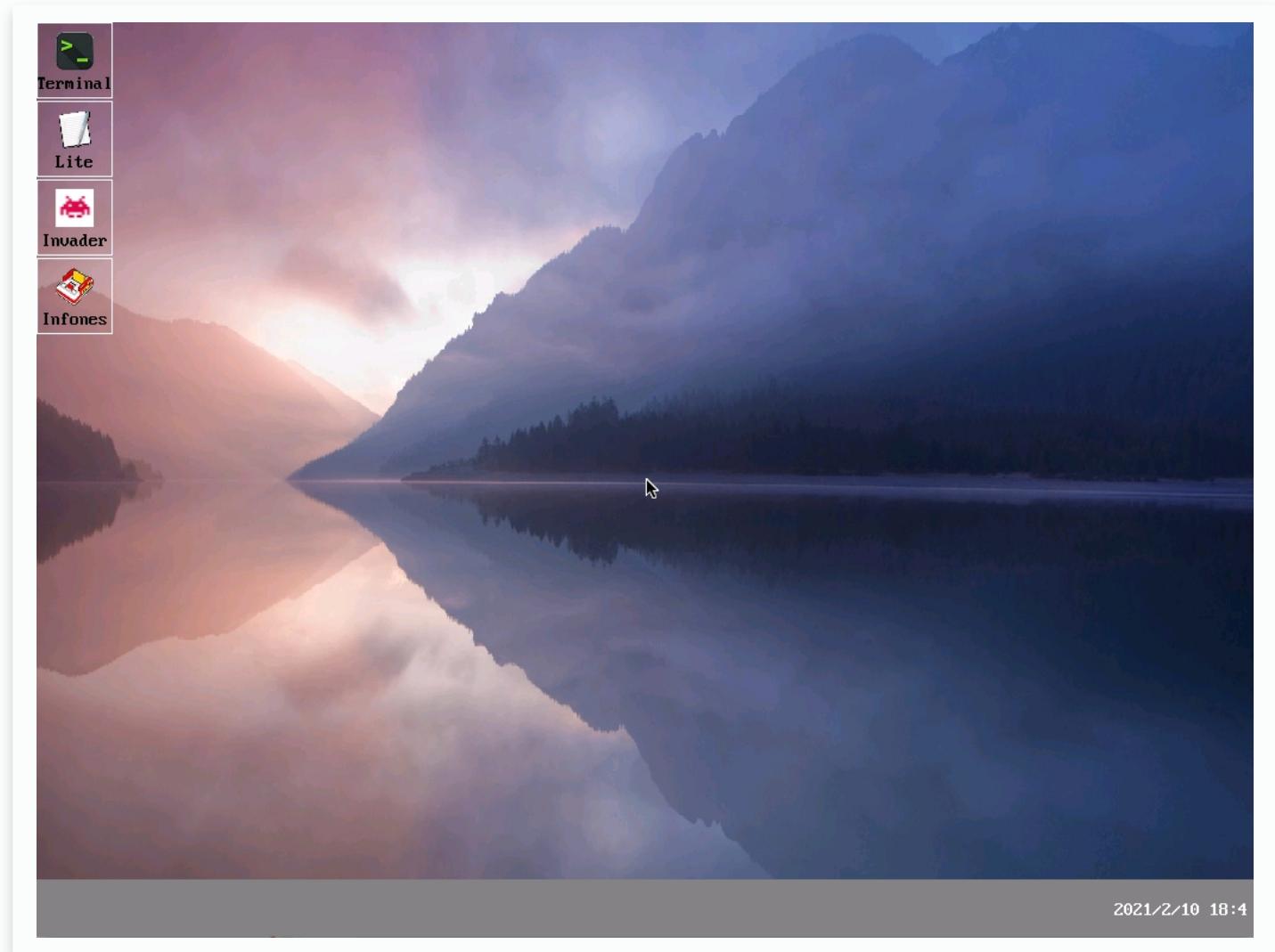
运行时留  
下的美丽  
瞬间。

```
/root/.bash_history  size=29
fs: init done.
account init: OK!
[task]: start user process.
+-----+
| Welcome to xbox2 kernel!
| All rights reserved by xbox2 kernel develop Team.
+-----+
[root /] ls
bin etc home lib sbin var acct .bash_history
[root /] ps
  PID  PPID  STAT    PRO    TICKS  NAME
      5      1  WAITING     1       45  sh
      7      5  RUNNING     1       58  ps
[root /] pwd
/
[root /] uname
xbook2-0.1.7
[root /] cal
      March 2021
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
[root /]
qemu-system-i386.exe  Search + ▾
```

xbook2 运行时，串口输出重定向到终端显示



# 简单才是 真！



BookOS启动界面，简单地显示了几个图标

# 02

T W O

## 效果演示

--视频比图片地动态效果表现得更好



# 虚拟机

Emulator



# 物理机

Real Machine





# 03

T H R E E

## 内核及OS功能介绍

--集跬步以行千里



# 源码组织

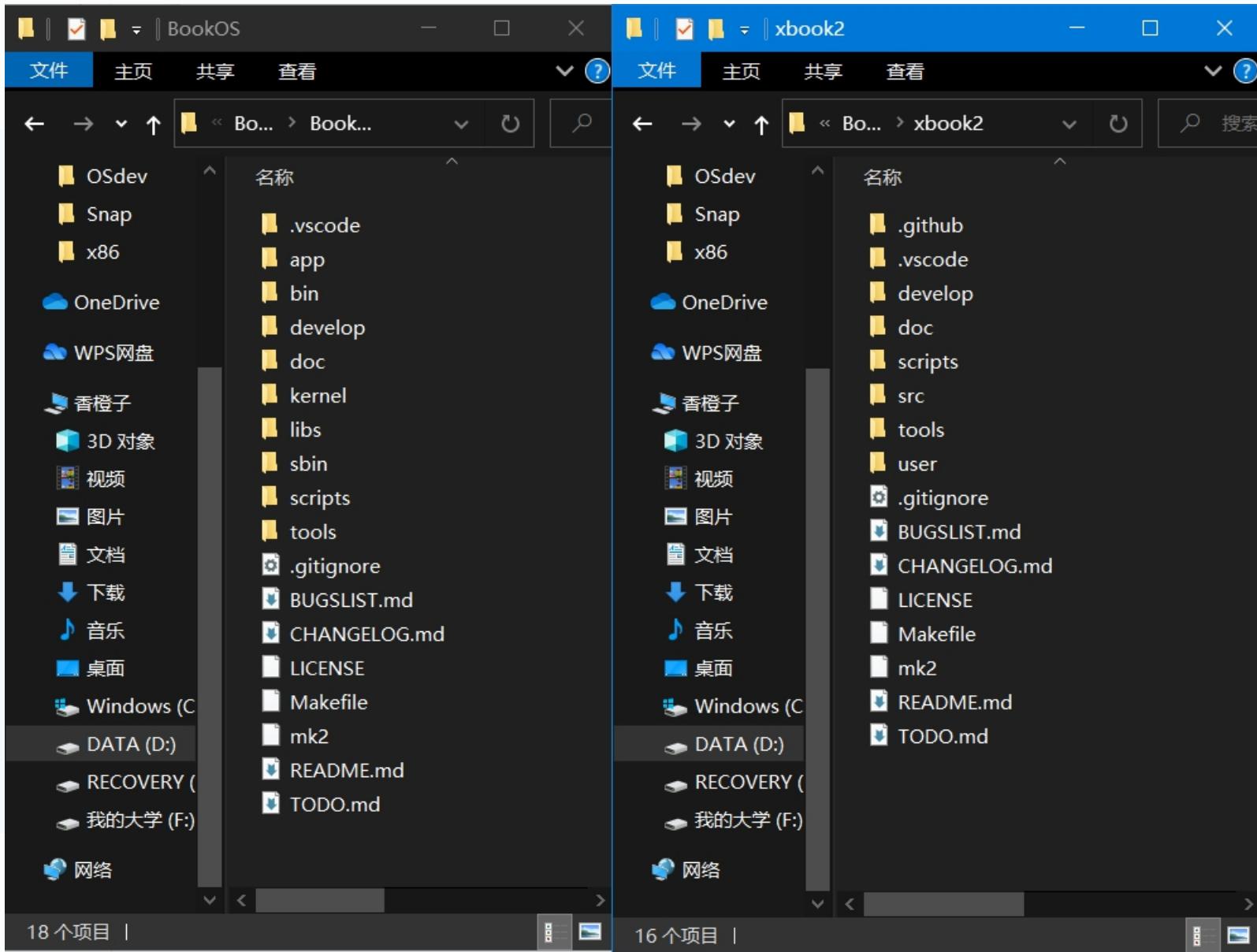
树型构造



# 系统结构

宏内核





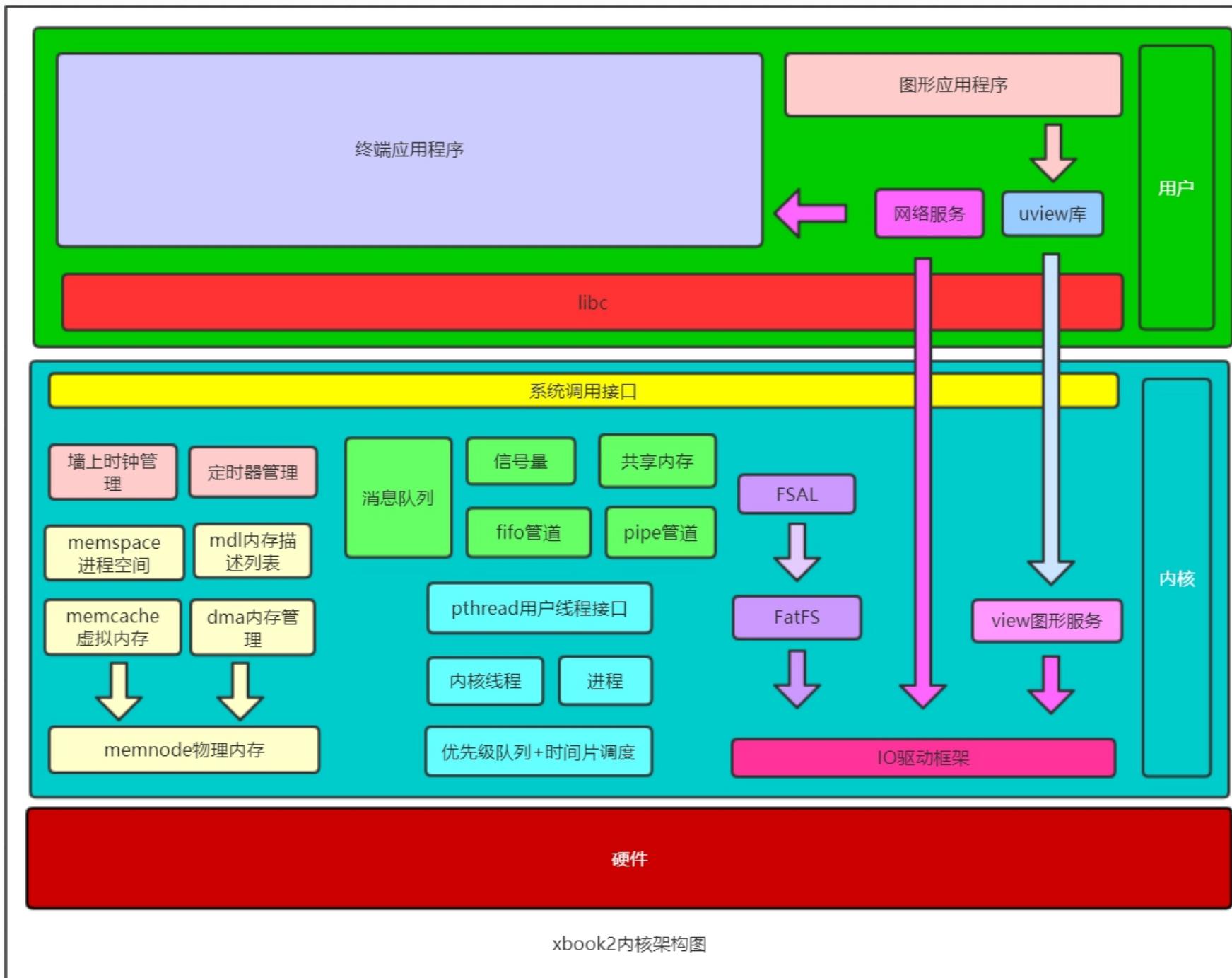
以Xbuild为构建脚本，控制编译内核，应用程序。  
xbuild: <https://github.com/ZhUyU1997/XBuild>

以FATFS作为文件系统，将rom写入到镜像文件，内核便可读取镜像上的文件。  
fatfs: [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)



# BookOS构建与启动流程





什么是 Book OS ? “书是人类进步的阶梯” ——高尔基。因此它叫做 Book OS 。

Book OS 是一个基于x86平台的32位操作系统，其基于 **xbook2** 内核进行开发。

**xbook2** 内核特性：

- > i386处理器平台
- > ELF格式内核和程序
- > 内存管理: 分页内存管理, 物理内存管理, 虚拟内存管理, 进程地址空间管理
- > 多任务: 多进程和内核线程, 用户多线程 (**pthread**)
- > IO驱动框架: 设备对象框架, 高级中断机制, 软中断机制
- > 内核: 时间管理, 定时器管理, 异步异常管理
- > 文件系统: FSAL文件系统抽象层 (类VFS) , FATFS文件系统
- > 进程间通信: 消息队列, 共享内存, 信号量, pipe管道, fifo命名管道
- > 通用驱动: ramdisk, null, zero, tty, ptty, view
- > PC驱动: ahci磁盘, console控制台, e1000网卡, ide磁盘, ps2keyboard键盘, ps2mouse鼠标, rtl8139网卡, sb16声卡, serial串口, vbe视频

**Book OS** 系统功能：

- > 库: **xlibc**标准C库, **pthread**, **netsocket**, **SDL2**, **SDL2\_image**, **SDL2\_ttf**, **cario**, **cjson**, **dotfont**, **freetype**, **gato**, **jpeg**, **libwebp**, **pixman**, **uvview**, **xtk**, **zlib**等
- > 命令行程序: **bash**, **cal**, **cat**, **cp**, **cpuinfo**, **date**, **echo**, **ls**, **lua**, **mem**, **mkdir**, **osver**, **ps**, **rename**, **rm**, **rmdir**, **sh**等
- > 图像界面程序: **infones**游戏模拟器, **invader**游戏, **lite**文本编辑器



## 04

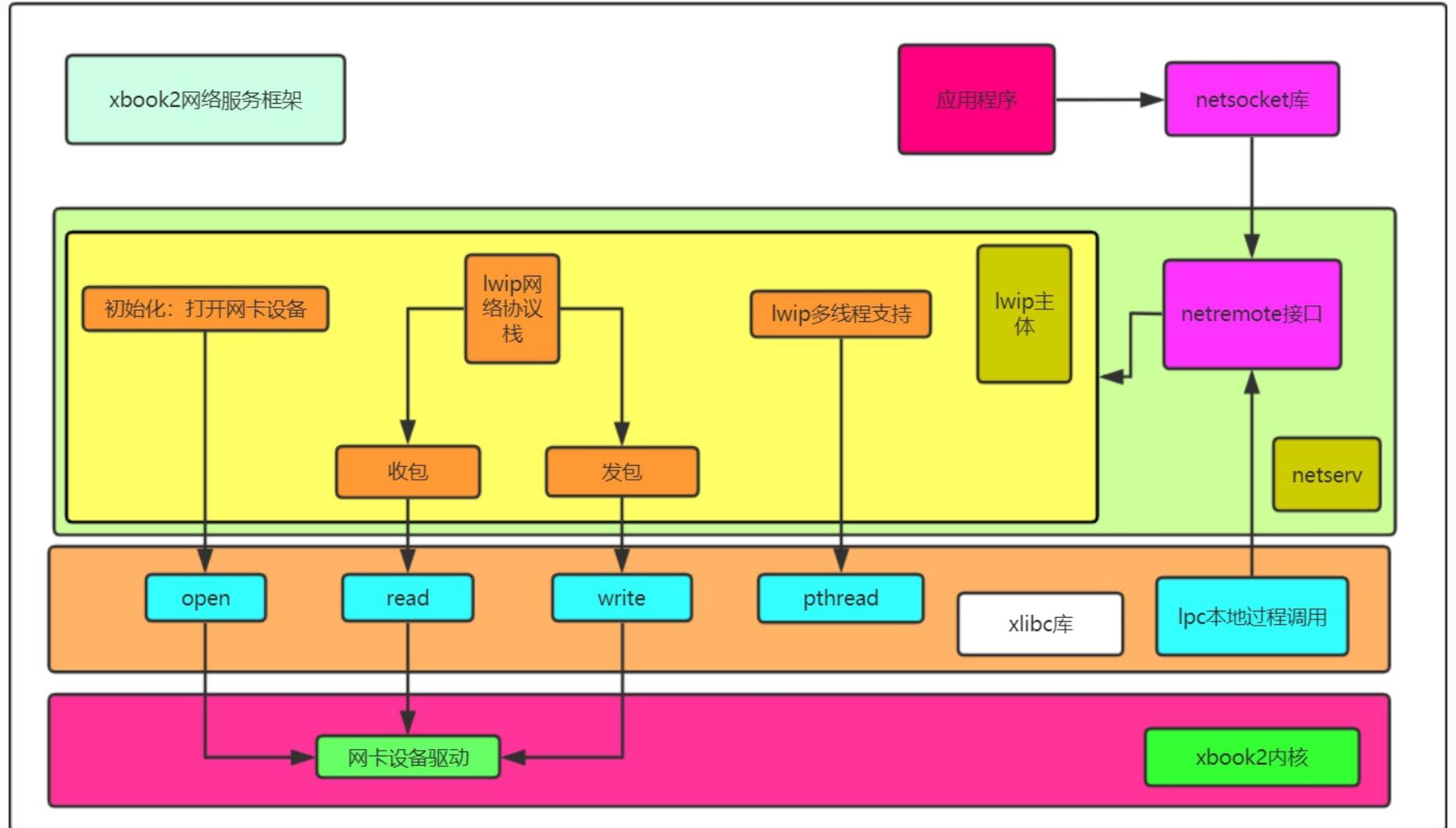
FOUR

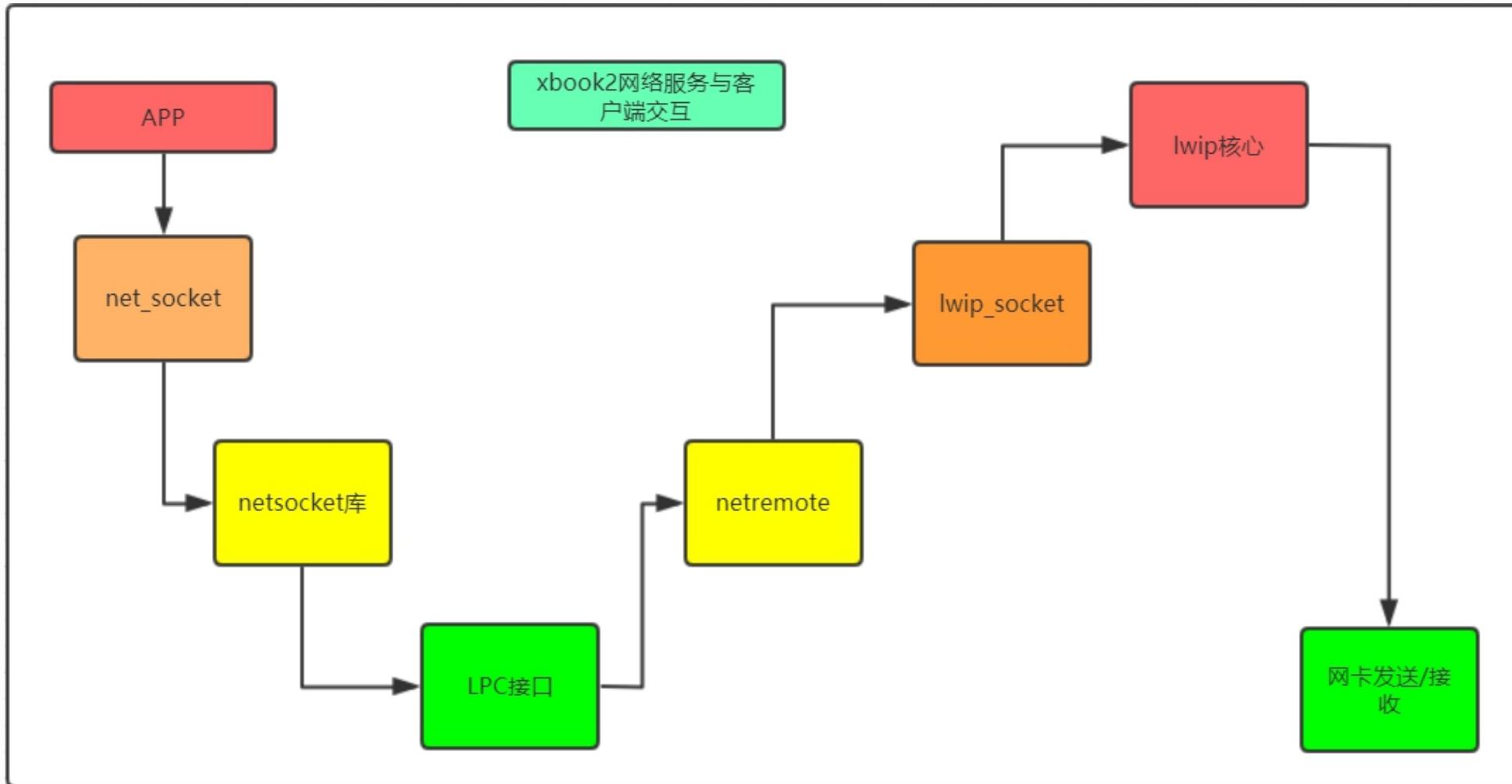
## 用户态lwip网络协议的封装

--移植是一件美好的事情



- > 融入微内核思想
- > 用户态服务器
- > 减少内核代码量，  
模块分离



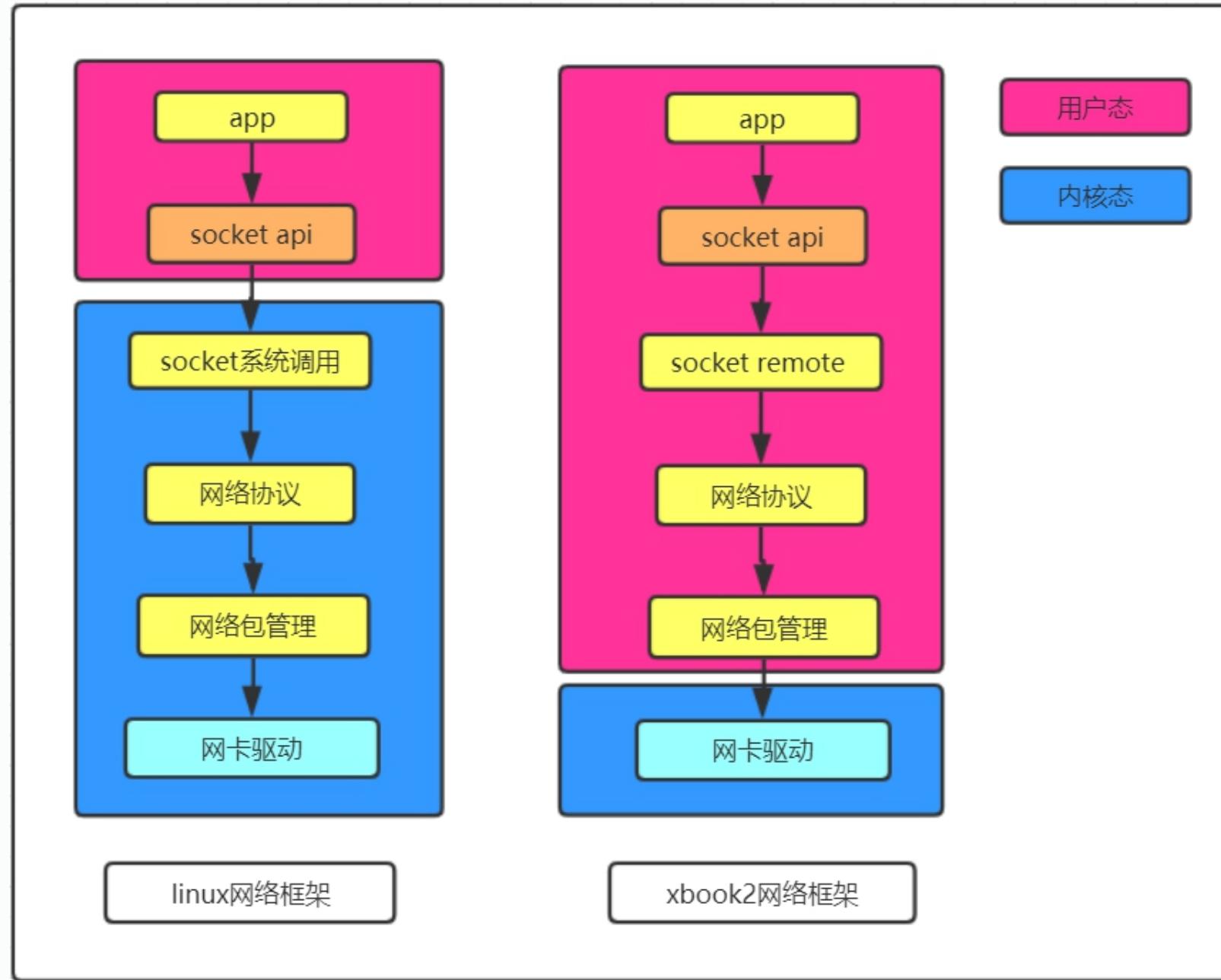


**优点：**

- \* 性能较高
- \* 对多客户端支持方便

**缺点：**

- \* 使内核庞大
- \* 产生问题将使内核崩溃



**优点：**

- \* 易扩展，独立内核实现
- \* 崩溃后不会使内核坠机，可以重新启动

**缺点：**

- \* 性能较低
- \* 对多客户端支持不好
- \* 网络包容易被劫持





# QEMU网 络支持的 流程与步 骤

- ◆ 1. Tap虚拟网卡的安装
- ◆ 2. Tap虚拟网卡ipv4信息设置
- ◆ 3. Qemu虚拟机配置
- ◆ 4. 网卡驱动支持
- ◆ 5. xbook2网络ipv4信息配置
- ◆ 6. 启动网络服务
- ◆ 7. 网络服务测试（浏览器服务、网络连接端口测试）

第二轮提问环节

# 05

F I V E

## view图形框架

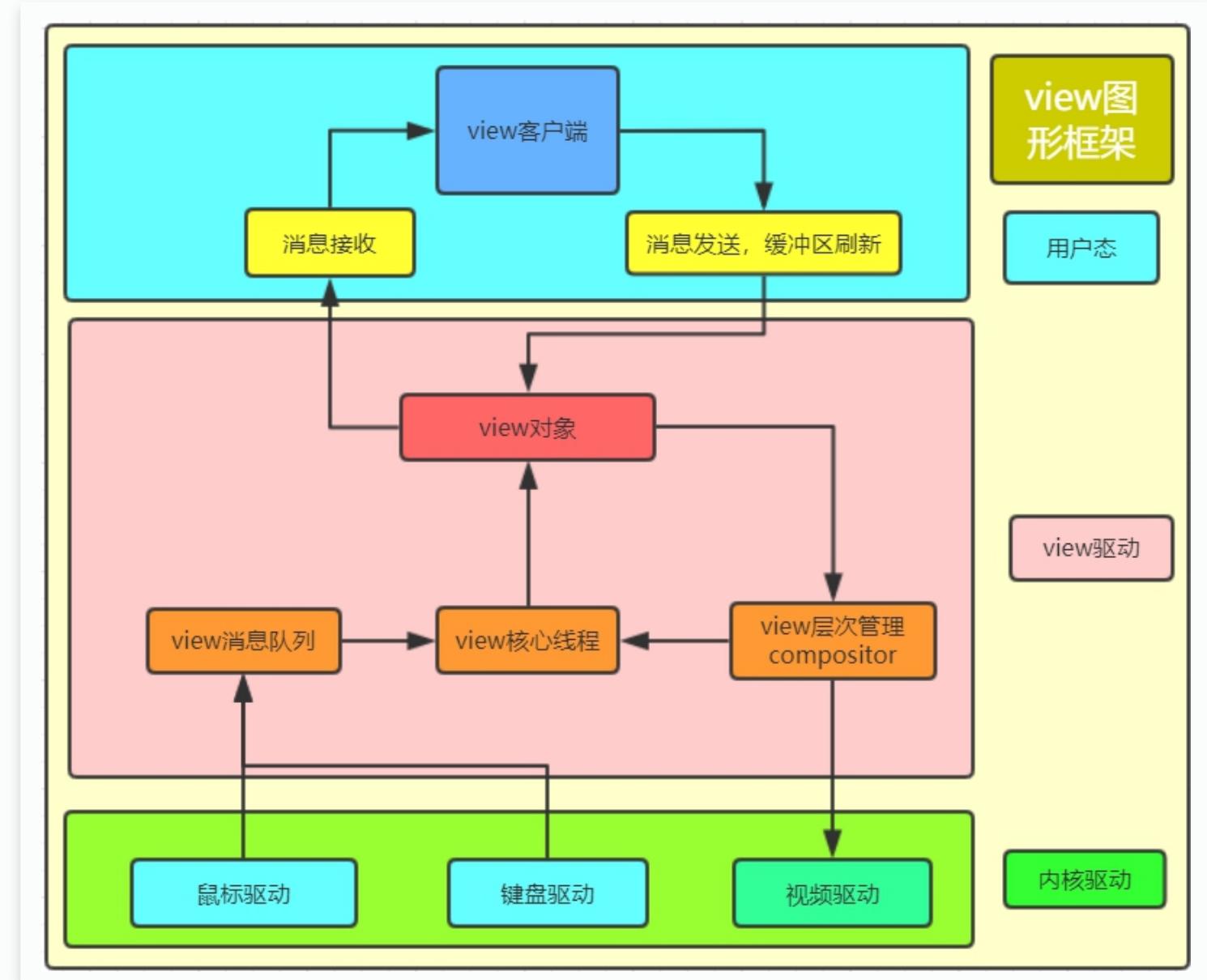
--我很庆幸，能够看到世间美丽的事物！





## xbook2内核 view图形框架

- > 内核态的图形框架，以驱动的形式出现，可有可无
- > 基于消息机制的图形框架
- > 高性能，快速
- > 使内核变得臃肿庞大，提高内核的不稳定性
- > 使用open, close, write, read, ioctl进行编程控制





## Xwindow图形框架 按钮点击

->内核产生鼠标事件

->x server决定发送给谁

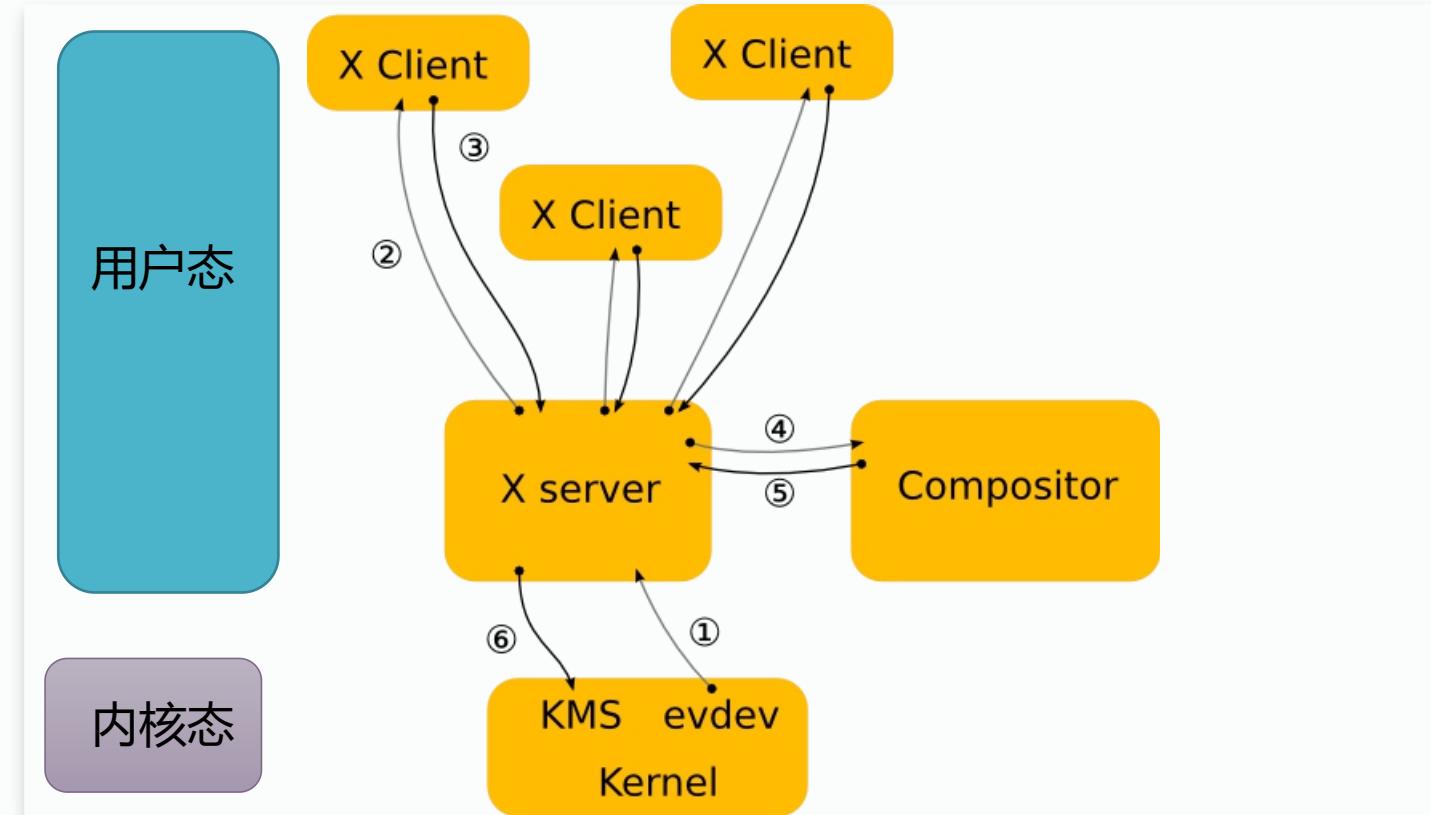
->x client响应后请求x server进行渲染

-> x server渲染请求的内容

-> compositor合成

->compositor响应后请求x server进行渲染

->x server刷新到屏幕显示



xserver->xclient

xclient->xserver

xserver->compositor

compositor->xserver

系统调用产生次数分析:

x server: 2+2+2

x client: 2

compositor: 2

总共: 10次



## Wayland图形框架 按钮点击

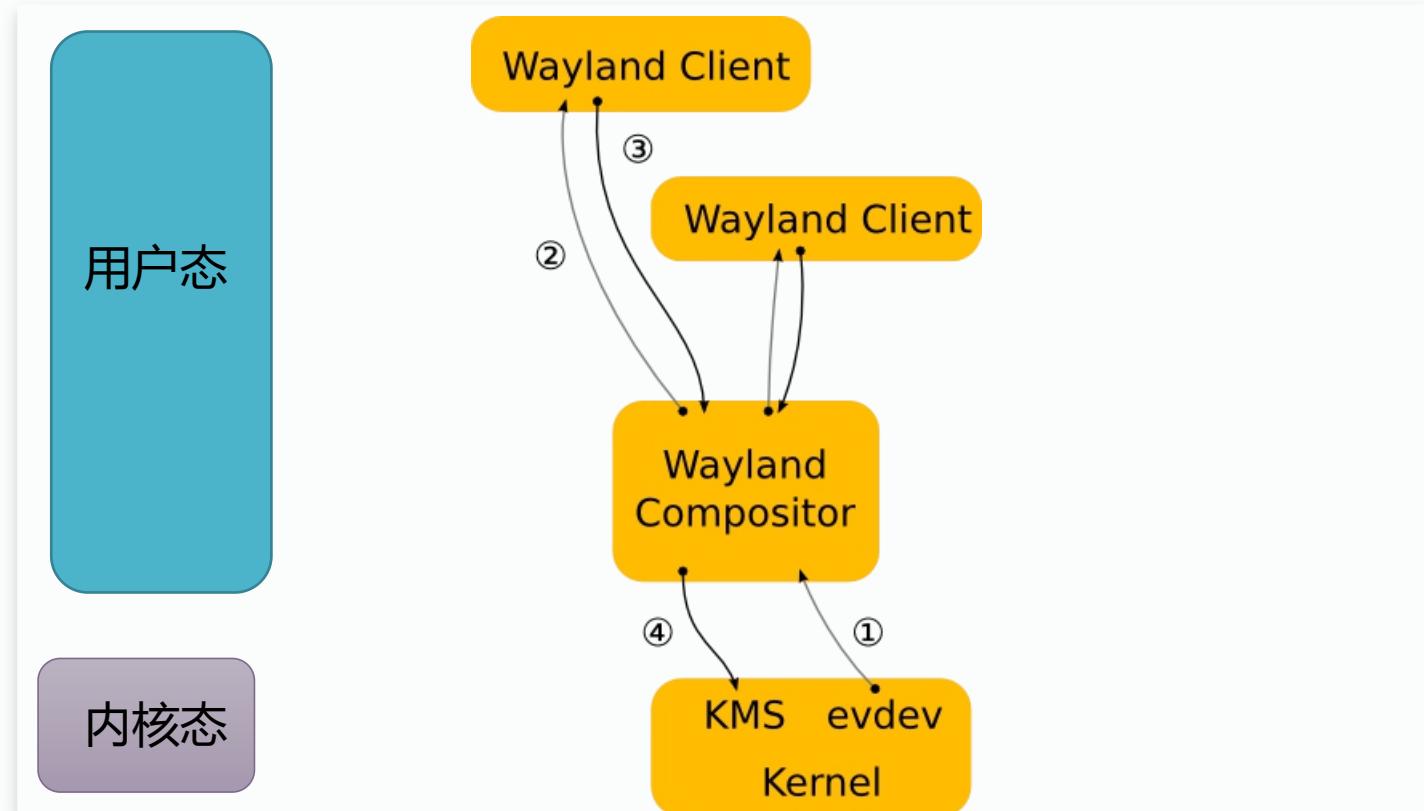
->内核产生鼠标事件

->wayland compositor决定发送给谁

->wayland client响应后自己渲染到buffer

-> compositor直接合成client的buffer

->刷新到屏幕显示



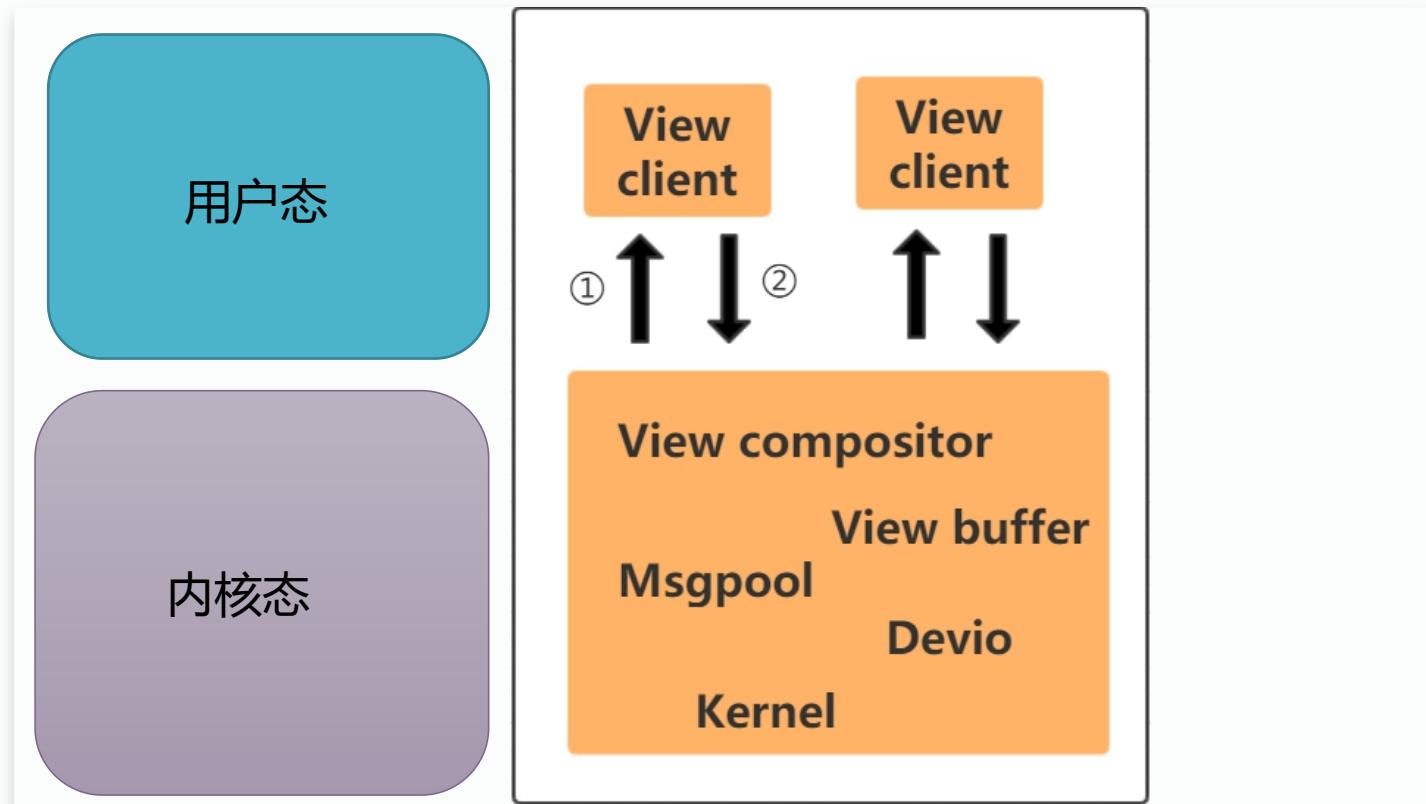
wayland compositor->wayland client  
wayland client->wayland compositor

系统调用产生次数分析：  
wayland compositor: 2+2  
wayland client: 2  
总共: 6次



## View图形框架 按钮点击

- >内核产生鼠标事件
- >View compositor决定发送给谁
- >View client响应后自己渲染到buffer/直接寻到View buffer
- >compositor直接合成client的buffer
- >刷新到屏幕显示



Viewcompositor->View client  
View client->View compositor

系统调用产生次数分析：  
view client: 2  
总共: 2次

## 本质区别

# View

compositor在内核态

- > 响应更快，性能更高
- > 对内核内存占用高，崩溃后影响内核
- > 合成器在内核，统一管理

# Wayland

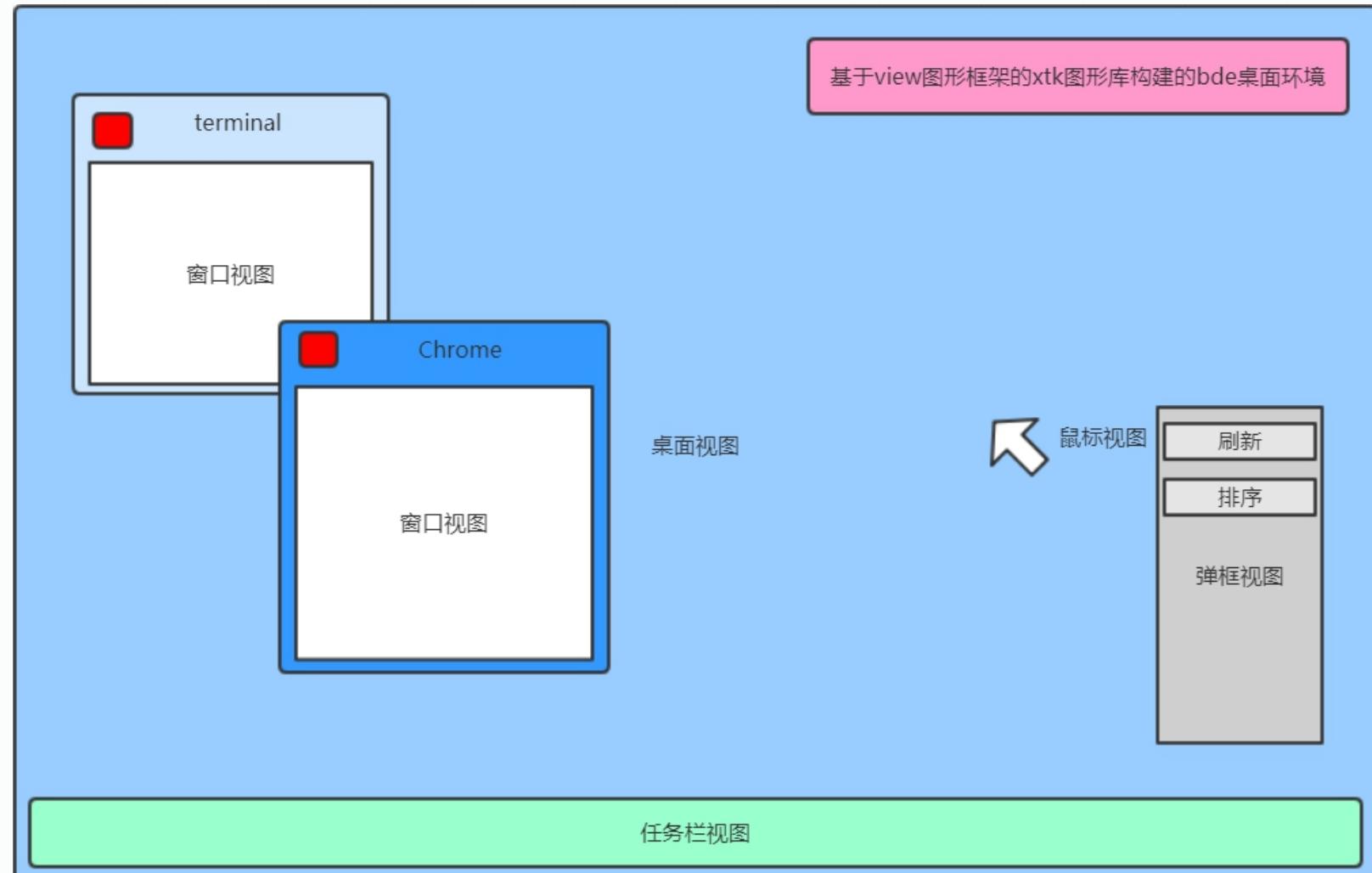
compositor在用户态

- > 独立于内核实现，方便扩展
- > 崩溃后不会影响内核，可重新启动
- > 合成器在用户态，多种多样





# View图形 框架的具体应用



内核view驱动 -> 用户态uview (user view) 图形库 ->  
xtk (xbook tools kit) 图形开发套件库 -> bde(Bookos Desktop  
Environment) 桌面环境



# BookOS 总结



## 优点

简单、轻巧、代码量少、麻雀虽小五脏俱全、自由度高

## 缺点

稳定性差、隐藏BUG比较多、教程文档少



# 未来展望

- 增强内核稳定性，打造一个强健的内核（地基不稳地动山摇）
- 编写更多辅助文档，让更多人参与项目开发
- 能够让大学生或者操作系统爱好者学习操作系统相关知识，推广操作系统领域的知识
- 打造国内独立研发的操作系统，实现系统底层的独立

第三论提问环节

# 彩蛋！

你以为这么快就结束了吗？



# 操作系统开发学习资料推荐





# THANKS

---

感谢大家的观看