

# 操作系统教学与能力培养改进： 基于OpenHarmony的操作系统教学实践

杜冬冬、夏虞斌

上海交通大学并行与分布式系统研究所

<https://ipads.se.sjtu.edu.cn>

2023.12.1 (上海)

# 课程信息：计算机系统设计与实现

- 课程名：计算机系统设计与实现 (Computer System Design & Implementation, 简称CSDI)
  - 面向研究生 (主要是研一和部分大四的学生)
- 从23年开始, CSDI关注在以OpenHarmony为基础的终端操作系统进行课程调整和改进

# 背景：计算机系统

Software



Hardware



# 背景：计算机系统

Users



User Applications



System Software  
(OS, compiler, VM...)



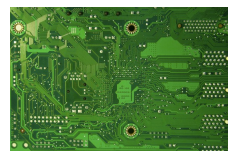
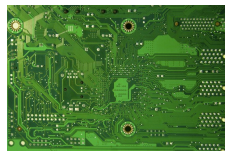
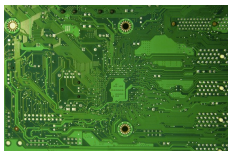
## Software

## Hardware

CPU, Memory, Disk



Logical Circuits,  
Flip-Flops, Gates



Transistors

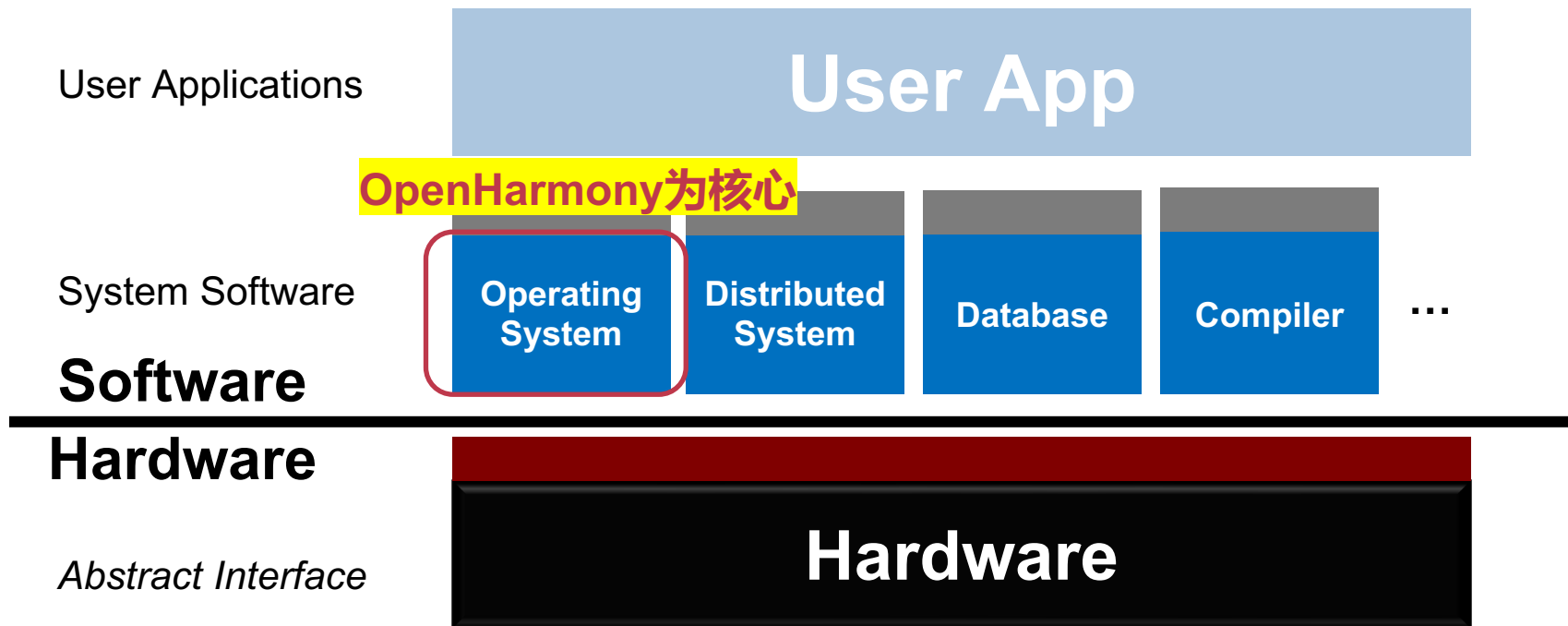


Diodes



Resistors

# 背景：计算机系统



# 研究生阶段的操作系统怎么上?

- 研究生 → 还有大量的科研任务、基础差别大
- 操作系统 → 复杂、“不生动不形象”、容易陷入太简单或者太难的两个极端

什么是OpenHarmony?

为什么是OPENHARMONY (OH) ?

# 为什么是OpenHarmony?

- **什么是OpenHarmony?**

- 一个（开源）操作系统，从华为到开发原子开源基金会
- 主要面向万物互联、万物智联场景

- **现状是，我们已经有了非常多很成熟的操作系统**

- 实践中：Linux, Windows, Android, etc.



- 为什么需要一个新的操作系统？（学生们真实的疑惑）

- **机会：越来越多的智能设备、万物智联呼吁新的OS**

- 现有的操作系统难以适应巨大的异构与差异所带来的复杂性



# 为什么是OpenHarmony?

- 机会: 越来越多的智能设备、万物智联呼吁新的OS
  - 现有的操作系统难以适应巨大的异构与差异所带来的复杂性

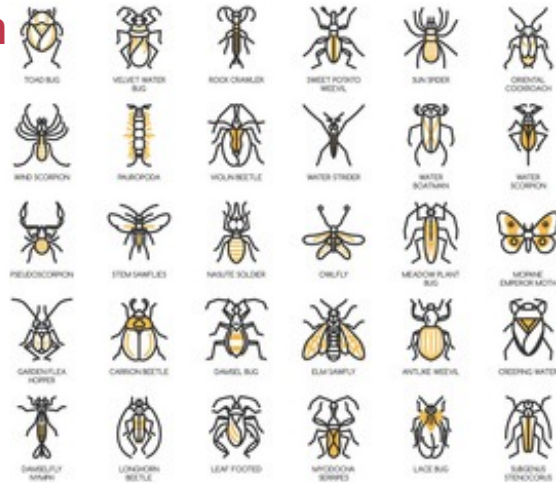
## 昆虫纲悖论



~5 million

100x

~50K



上海交通大学并行与分布式系统研究所 (IPADS@SJTU)





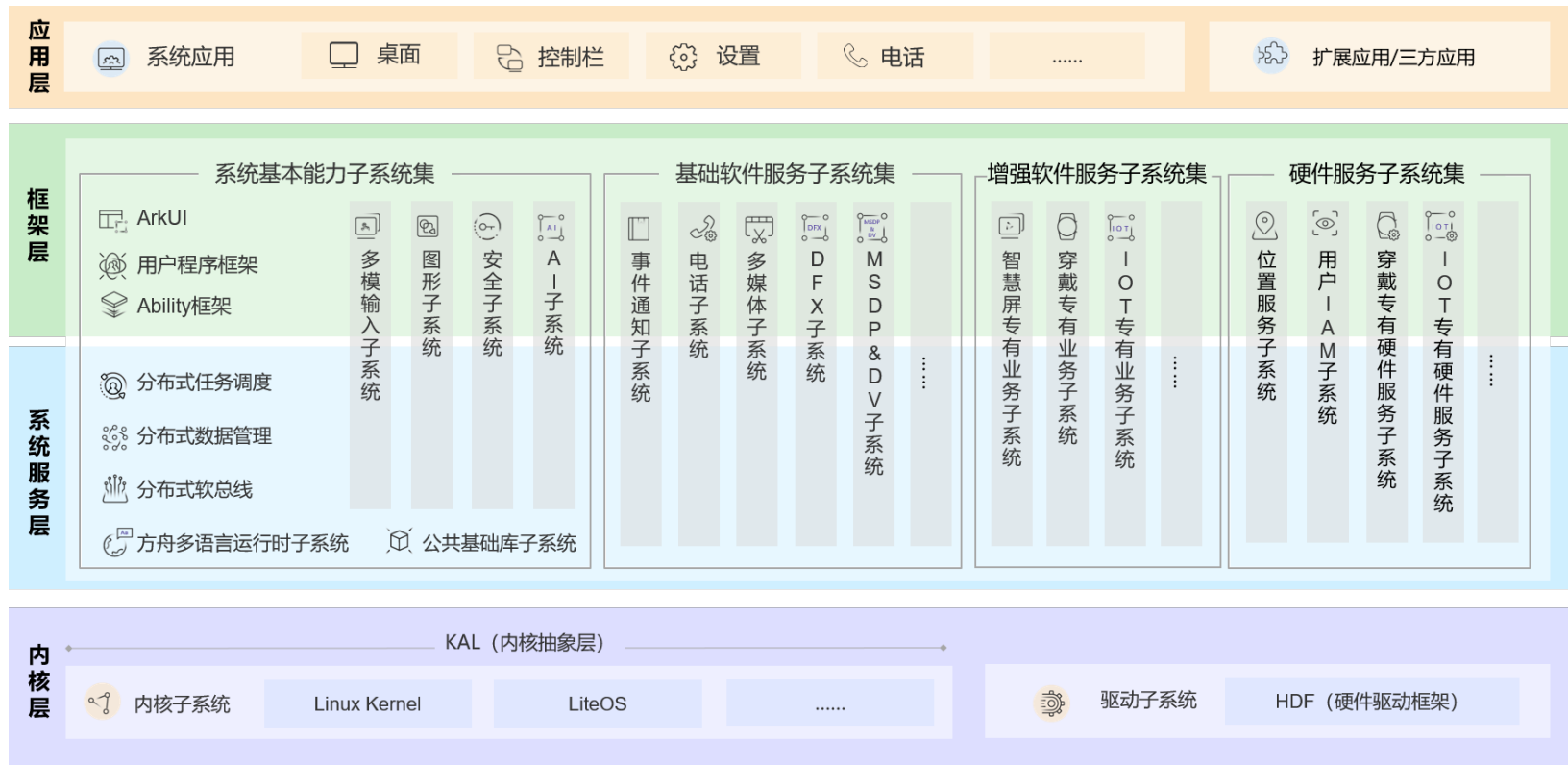
# 准备阶段

# 怎么上? 先学习

- **Learn by?**

- 用OpenHarmony (上板子)、读文档、看设计

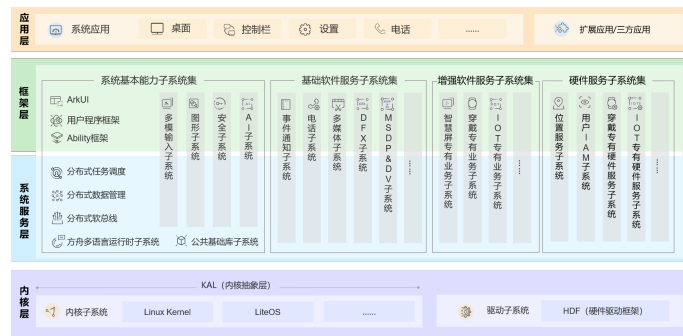
# OpenHarmony的4层架构



# OpenHarmony的4层架构

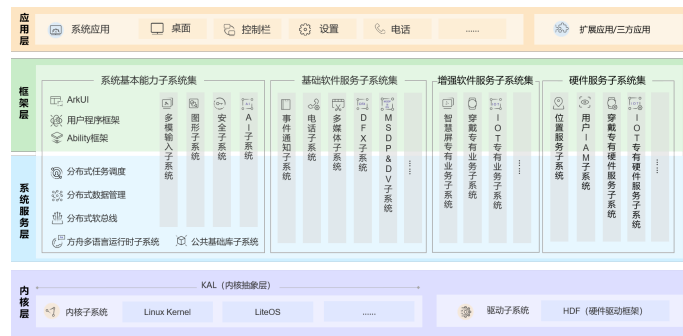
- **Layer 1: Kernel**

- 为应用提供基础能力和支撑 (e.g., 调度等)
- 操作系统  $\geq$  内核



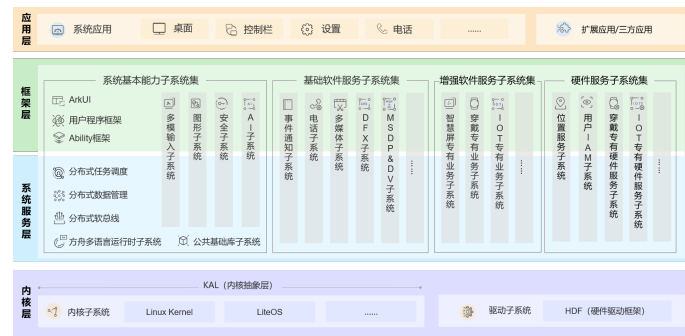
# OpenHarmony的4层架构

- Layer 1: Kernel
- Layer 2: System services
  - 为终端场景的应用提供关键支撑服务
  - E.g., DSoftBus (分布式软总线)



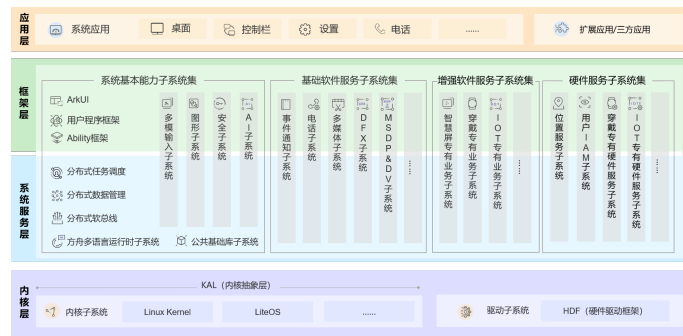
# OpenHarmony的4层架构

- Layer 1: Kernel
- Layer 2: System services
- Layer 3: Framework
  - 直接支撑应用，包括开发和部署
  - E.g., UI development



# OpenHarmony的4层架构

- Layer 1: Kernel
- Layer 2: System services
- Layer 3: Framework
- Layer 4: (Distributed) Applications





# 关键原则: Componentization

- 将一个大规模的系统切分为小的、可重用、可组合的小组件们

- 可以被用来定制化各种各样的场景和需求

- HarmonyOS/OpenHarmony

HarmonyOS DevEco Service

定制发行版



# 怎么上？先学习

- Learn by?

- 用OpenHarmony（上板子）、读文档、看设计
- 文档仍然存在不少缺失，很多时候只是解释**是什么**，而不是**为什么**？
- 是**开发**文档而不是**学习**文档
  
- 更深入？

# 怎么上? 先学习

- Learn by?

- 用OpenHarmony (上板子)
- 文档仍然存在不少缺失, 很
- 是开发文档而不是学习文档
- 更深入? **Coding!**

排名	共建者邮箱	共建者ID
179	xuzijay@163.com	xuzijay@163.com
180	dd_nirvana@sjtu.e...	dongduResearcher
181	wuyawei1@huawe...	blancwu
182	luowei@iscas.ac.cn	openharmy-lab...
183	xiaofan@iscas.ac.cn	xfan1024
184	wangweichao2@h...	weichaox
185	quizkid@163.com	cleverchap
186	hanjixiao@huawei...	chuxuezhe111
187	jianglili2@huawei.c...	lily0525
188	yanshuifeng@hua...	yan-shuifeng
189	zhanghaibo0@hua...	zhanghaibo0

为什么?



# 教学实践

# 课程整体情况

- **教学内容**

- 共包含**6**个主题，涵盖智能终端操作系统各个方面

- **邀请报告**

- 主题内容教学结束后，邀请华为专家进行相关主题的扩展介绍
- 共安排**4**次邀请报告

- **课程实验**

- 发布**3**个课程实验，学生自选其一、分组完成
- 基于真实设备进行OH开发和扩展

# 课程整体情况

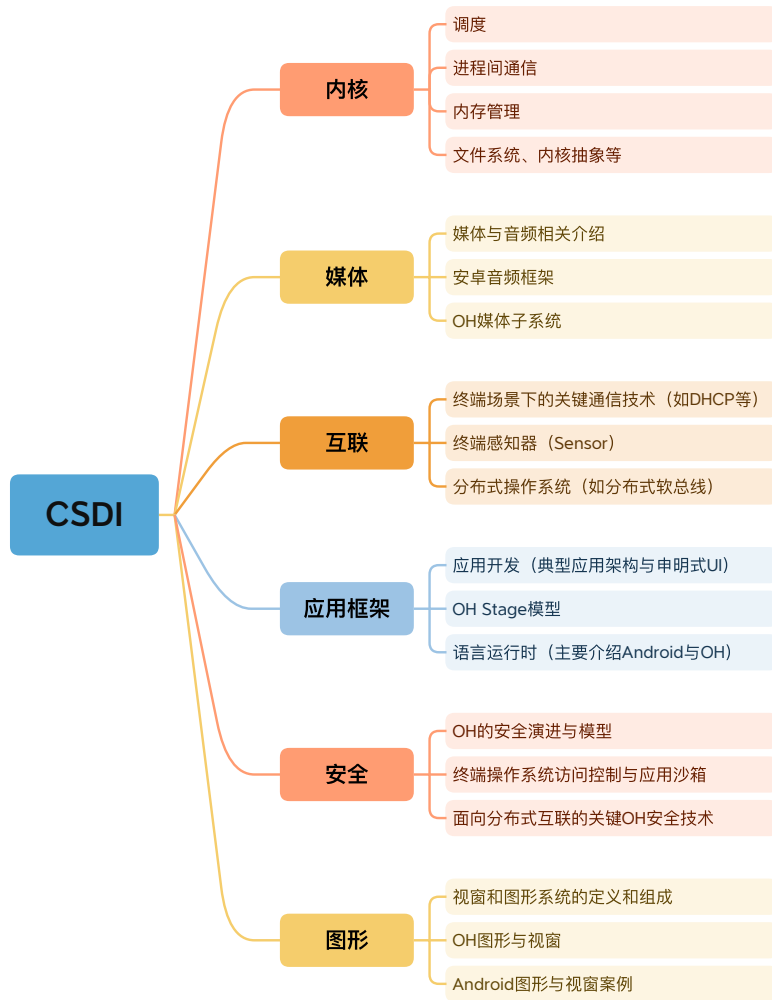
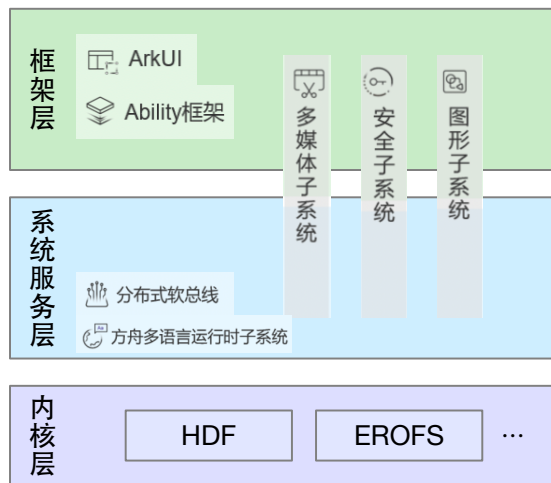
## • 课程安排

教学周	周一	内容	周三	内容
1	2023/2/13	绪论和OH介绍 (WMY)		没课
2	2023/2/20	智能终端内核-1 (DD)	2023/2/22	智能终端内核-2 (DD)
3	2023/2/27	智能终端内核-3 (DD)		没课
4	2023/3/6	媒体系统-1 (WMY)	2023/3/8	媒体系统-2 (WMY)
5	2023/3/13	通信与互联-1 (DD)		没课
6	2023/3/20	通信与互联-2 (DD)	2023/3/22	通信与互联-3 (DD)
7	2023/3/27	应用框架-1 (WMY)		没课
8	2023/4/3	通信与互联-华为分享 (分布式软总线)	2023/4/5	放假
9	2023/4/10	应用框架-2 (WMY)		没课
10	2023/4/17	应用框架-3 (WMY)	2023/4/19	应用框架-华为分享 (arkui、方舟、ability等)
11	2023/4/24	安全架构-1 (DD)		没课
12	2023/5/1	五一放假	2023/5/6	安全架构-2 (DD)
13	2023/5/8	安全架构-3 (DD)		没课
14	2023/5/15	安全架构-华为分享	2023/5/17	图形系统-1 (DD)
15	2023/5/22	图形系统-2 (DD)		没课
16	2023/5/29	总结课	2023/5/31	图形系统-华为分享

# 教学内容介绍

## • 内容全面

- 涵盖内核、媒体、互联、应用框架、安全、图形六个主题



Learn by 臧斌宇老师



# 教学实践#1：原理 + 实现，聚焦OH， 不限于OH



# 系统原理 + OpenHarmony实现的课程内容

- **问题与挑战**

- 传统操作系统往往只停留在理论阶段，或者结合简单的操作系统（例如MIT xv6等）进行系统的介绍然而
- 和真实生产环境中的系统仍然具有较大的鸿沟

- **实施方法：**

- 建设操作系统原理和OpenHarmony系统实现相结合的课程内容，围绕**终端操作系统**，介绍操作系统内核、分布式、媒体、应用框架、**终端安全**等多方面的内容
- 在每部分的课程内容，均采用基本原理结合真实系统（即OpenHarmony）实现的方式来介绍，一方面让学生掌握核心方法，另一方面也能对应到真实系统中

# 教学内容介绍

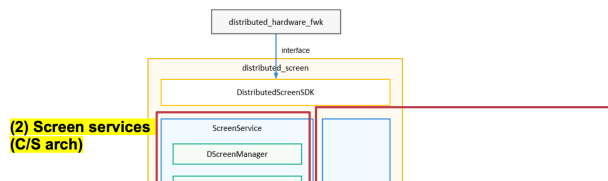
- 聚焦OH，但不限于OH

- 围绕OH关键技术进行介绍（如Eswap、ArkUI、HDF等）
- 重点介绍智能终端场景需求，突出OH特色

Mini  
主要需求：音频播放



Small  
主要需求：音频播放+视频播放



Device 1

Device 2

# 教学内容介绍

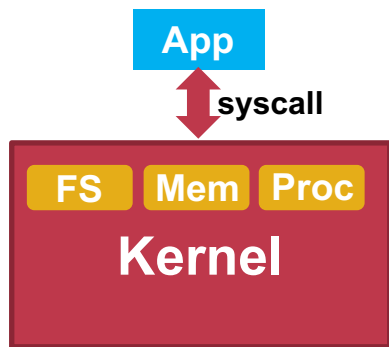
- 聚焦OH，但不限于OH
  - 围绕OH关键技术进行介绍（如Eswap、ArkUI、HDF等）
  - 重点介绍智能终端场景需求，突出OH特色
  - 扩展介绍其他操作系统案例，对比设计优劣



# OH关键技术：多种内核支持

- **Monolithic kernel (宏内核)**

- 一个大型的内核，集成需要支持的各种能力
- 应用通过系统调用来和内核进行“交互”



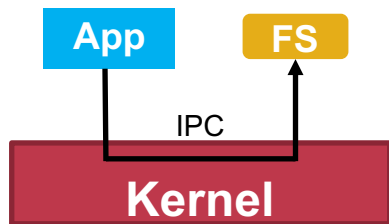
## 缺点:

- Lack of isolation
- Large trust code base (TCB)
- Lack of support for heterogenous hardware
- ...

# OH关键技术：多种内核支持

- **Micro-kernel (微内核)**

- 最小化内核中的部分
- 系统服务以*Service*的方式运行在用户态
- 应用通常通过进程间通信 (inter-process call , IPC)方式来和这些用户态系统服务交互



## **Core issue: IPC!**

- IPC's performance is critical to micro-kernel

# OH关键技术：多种内核支持

- OH怎么选？面临不同场景，通过支持多种内核来适应不同需求
  - 普通设备（标准设备）：Linux
  - 资源受限（轻量）设备：LiteOS
  - 在未来可能会支持更多的其他内核
- **问题与挑战：如何保持跨多个内核的兼容性？**



# OH关键技术：多种内核支持

- **内核抽象层：Kernel Abstract Layer (KAL)**
- **通过提供一套统一的抽象层来解耦上下**
  - 包括我们比较熟悉的接口，如POSIX
- **内核需要适应和支持这些接口**

# OH关键技术：在“超级设备”上的分布式执行

- 在特定场景下，多设备能够被当做单一“超级设备”
  - 案例: 智能家庭
  - 不同设备存在不同的特性和能力

Music  
Phone



Time  
Health

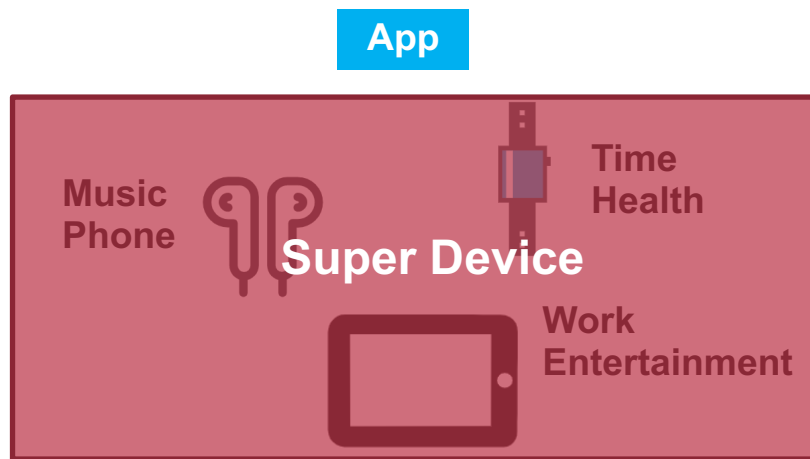


Work  
Entertainment



# OH关键技术：在“超级设备”上的分布式执行

- 在特定场景下，多设备能够被当做单一“超级设备”
- 超级设备：将多个物理设备融合为一个大型的、虚拟的设备



# OH关键技术：分布式执行的支持

- **分布式软总线 (Distributed soft bus, DSoftBus)**
  - 一个基于软件支持构建的虚拟总线
  - 基于虚拟总线, 使得设备发现、数据传输、互联更加容易
  - 支持其他通信方式和机制, 例如RPC



# OH关键技术：分布式执行的支持

- **分布式数据管理**

- 数据通常不是和一个单一固定的设备绑定的
- 在“超级设备”内部流转和移动
- 基于分布式软总线构建

- **抽象：分布式data store**

- 能够像本地的数据对象一样使用 (with CRUD interfaces)
- 使用 DataShare 来实现设备之间的数据共享

# OH关键技术：其他分布式系统服务

- **分布式硬件管理**

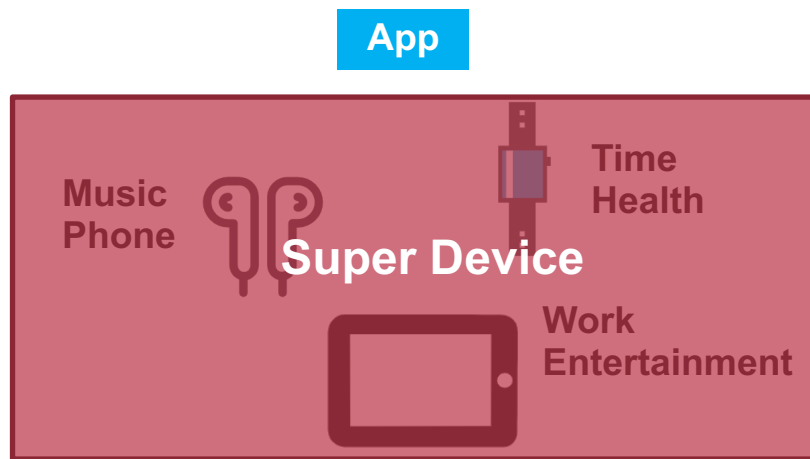
- 管理设备池，使他们形成一个整体
  - E.g., 分布式摄像头 (远端摄像头)

- **其他共用服务和库**

- E.g., synchronizations, files, data serialization

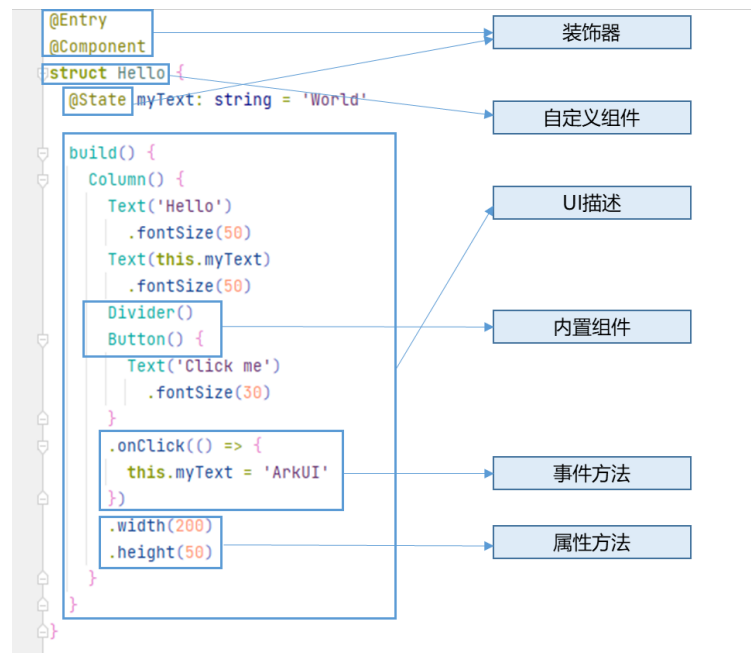
# OH关键技术：应用框架

- “超级设备”已经将多个单一设备进行连接
  - 但是应用开发没有（而且也不应该）有太大的变动和影响
    - 依然是UI-based, interactive apps



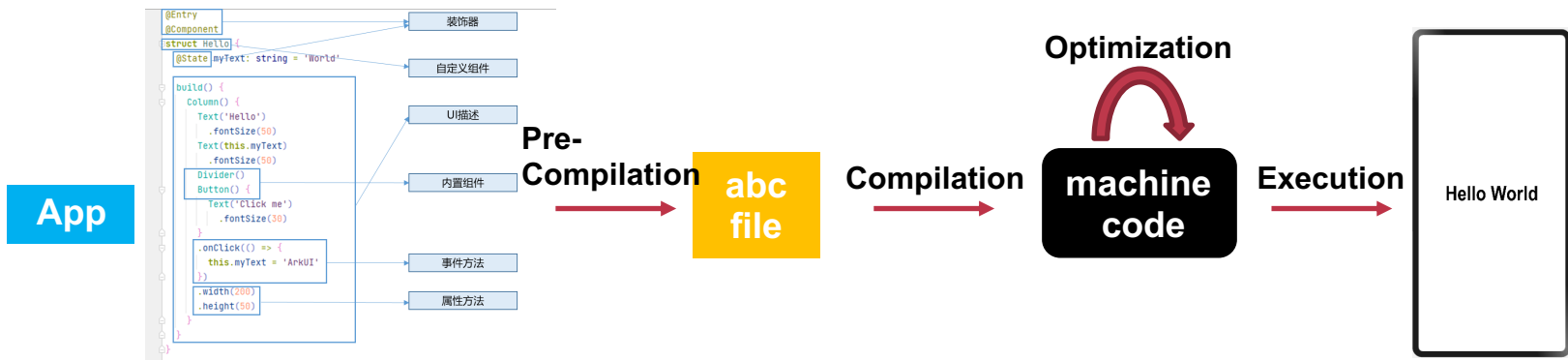
# OH关键技术: ARKUI: Interfaces for Building UIs

- 基于声明式方法来定于UI接口
  - Language: ArkTS
- Compiled and executed on Ark runtime



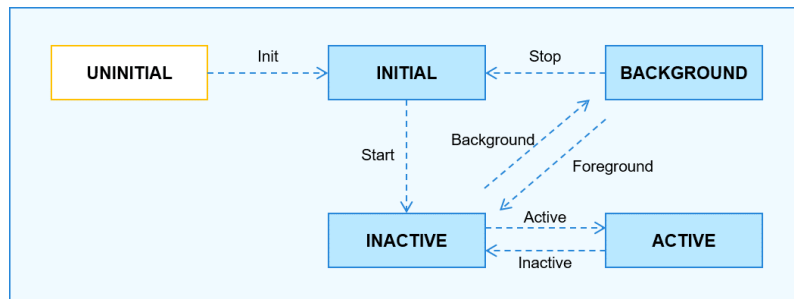
# OH关键技术：如何运行应用？

- OH的选择：一个执行运行时，结合字节码（bytecode support）
  - 类似Java/C# 语言虚拟机
  - 在多种差异化的设备上实现兼容性、易移植性



# OH关键技术：Abilities (元能力)

- 最小的一个系统调度单元
  - 一个应用可以拥有多个元能力
- OH提供了一个元能力子系统来管理和运行元能力





# ▶ 教学实践#2：要有趣！

# 方法论：要有趣，系统和现实生活不是脱钩的

- 一个长期以来困扰的问题

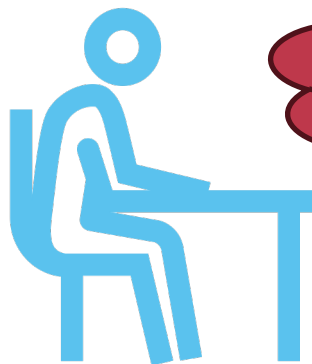
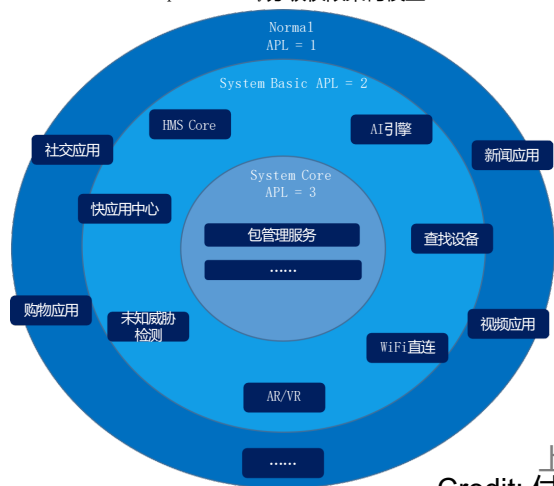
- 当我们讲内核的时候，学生们真的知道什么是内核吗？
- 当我们讲系统安全的时候，学生们是怎么理解的？
  
- 看不到、摸不着，演示是比较困难的

# 方法论：要有趣，系统和现实生活不是脱钩的

## • 案例：终端操作系统的权限管理

- OpenHarmony：分级权限（APL）、分布式设备等
- Android：IAM (Identity and Access Management)
- 其他

OpenHarmony分级权限架构模型



学生：对，这个设计很合理，但是和我有什么关系？

# 案例：Why Permission Matters?

@网上冲浪的男孩子们：不要luǒ聊！

江宁公安在线 平安交大 2021-01-28 18:00

建议转给身边上网冲浪的男孩子们都看一下，没开玩笑👉

防范电信网络诈骗系列

不要裸聊

# 案例：Why Permission Matters?

目前比套路的裸聊诈骗流程一般是这样的：



受害男：（打开某个交友APP，看看有没有我喜欢的女孩子）



美女：小哥哥，你好帅啊，聊天吗很开放的那种哦！



受害男：（居然有美女主动加我？先聊聊看）你好美女！



美女：小哥哥，我是做主播的，刚开始做这行业绩也不大好，能不能帮我点个关注？



受害男：（怪不得这么好看原来是主播啊，只是让加个关注，我还以为是骗子呢）没问题啊小姐姐！




美女：小哥哥那我把我们的直播app发你，关注我顺便点个赞就好，谢谢你啦么么哒！（发送app）




受害男：（安装，注册手机号，找到主播小姐姐，关注，点赞）妥了

博取同情，请求  
下载应用点赞

# 案例：Why Permission Matters?



美女：太谢谢你了小哥哥你真是好人，我刚做主播还不知道怎么吸引粉丝，多亏了你！我要报答你！给你独家福利！我们来视频吧！



受害男：好，好啊（独！家！福！利！好激动哦！）


受害男：好，好的呢~！




美女：小哥哥，这些人都是你的亲戚朋友同事上司吧？（发来一份通讯录）




受害男：你，你怎么知道的？



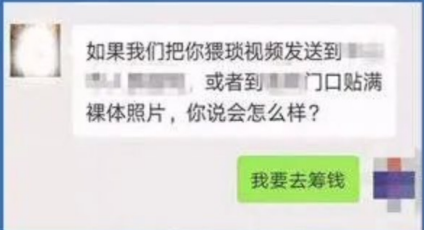
美女：懒得和你废话，给你个账号马上打5万块钱来，不然我就把你的蘑菇照发给你所有亲朋好友。



受害男：我，我没有那么多钱！



美女：



泄露个人信息

图穷匕见

## 案例：Why Permission Matters?

这种裸聊诈骗套路基本上就是这样的。聪明的同学应该已经看出来，骗子获得受害人通讯录上人员信息的方法很简单，就是一开始那个所谓“直播app”，获取你手机通讯录信息易如反掌。

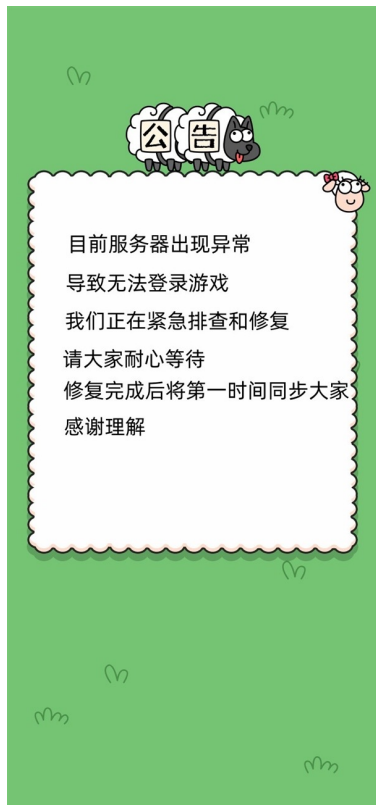
而一旦你选择给骗子打钱，则开启了一个无底洞，他们会无限制的问你要钱。

# 案例：Why Permission Matters?

- **当下非常重要的话题：反诈**
- **吸引学生的兴趣：**
  - 即使猜到故事内的主人公的解决，学生们还是很感兴趣
  - 发生在自己学校
- **从故事回到问题**
  - 权限滥用（通讯录权限）→ 被诈骗（实际问题）
  - 终端操作系统的权限管理设计，对我们每个人都息息相关



# 细节设计：有趣、真实



## 紧急提示！

四川天府健康通官微 2022-11-24 08:23 发表于四川



“四川天府健康通”由于流量过大，部分用户暂时出现登录困难，现正逐步恢复。

核酸检测登记时，若遇无法扫码，可采用扫身份证方式。

# 细节设计：有趣、真实

- **Heterogenous devices**
  - Used in different scenarios, with different characteristics
  - Brand new hardware technology with new opportunities
- **New applications with unexplored requirements**
  - AI: large data size & models



Welcome to ChatGPT

Learn from 夏虞斌老师



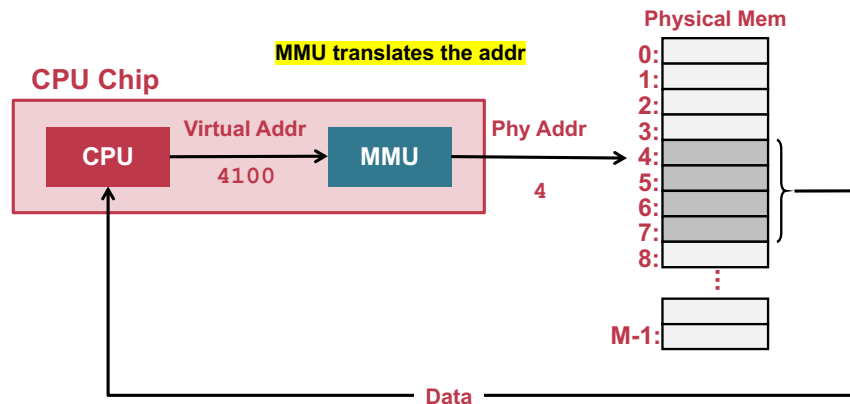
# 教学实践#3: SOMETHING OLD, SOMETHING NEW

# 方法论：Something new, something old

- 大家熟悉的OS → 大家不熟悉的
  - 案例：内核，内存管理

## ▶ Review: OS Memory Management (2)

- Address translation



# 方法论：Something new, something old

- 大家熟悉的OS → 大家不熟悉的
  - 案例：内核，内存管理



## ▶ Case Study: Android

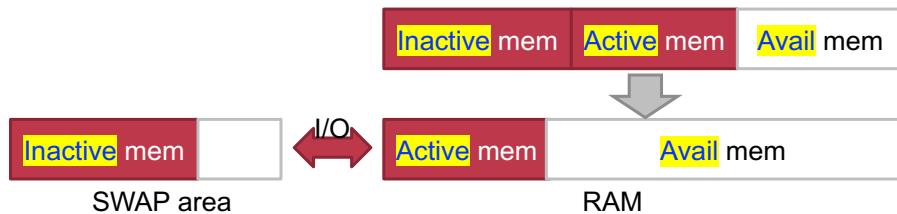
- “The Android platform runs on the premise that **free memory is wasted memory.**”
- **A case: application startup latency**
  - Android keeps apps in memory **after they've been closed** so the user can quickly switch back to them
  - 1<sup>st</sup>: seconds
  - 2<sup>nd</sup>: milliseconds

# 方法论：Something new, something old

- 大家熟悉的OS → 大家不熟悉的
  - 案例：内核，内存管理

## ▶ Swapping

- **Basic idea**
  - Utilizing disks to extend the physical memory size
  - Real capacity: physical memory + swap area
- **How?**
  - Swap area on disk
  - Page fault handling: on-demand swap-in/swap-out



# ▶ 教学实践#4：多视角

# 教学铺垫 + 专家分享的教学模式

- **问题与挑战**

- 传统课程中，往往由1-2位校内老师，对课程内容进行介绍
- 课内的**教学内容**和介绍往往经过一定的总结和凝练，和**真实系统**及其最新进展之间仍然具有一定的差距

- **实施方法：**

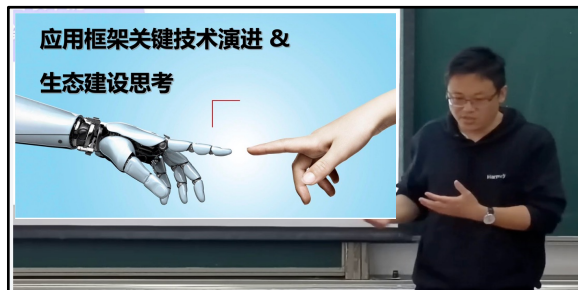
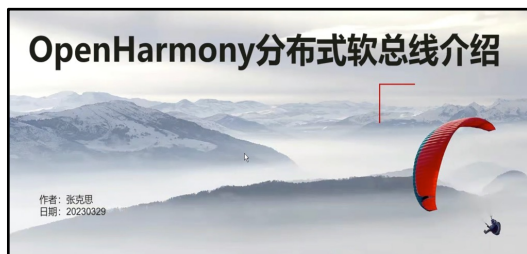
- “教学铺垫+专家分享”的教学模式，针对每部分大块内容（例如操作系统分布式能力），邀请来自OpenHarmony一线开发的专家参与教学过程
- 简单地邀请专家来为学生上课可能导致内容过易或过难
- 需要在前期教学中进行一定的**铺垫**，并且提前和社区/业界专家沟通分享材料，让学生能够从课内知识贯通到前沿进展



# 邀请报告介绍

- 共安排4次邀请报告

- 涵盖互联、应用框架、安全、图形主题
- 来自不同部门的专家围绕主题进行了深入介绍



# ▶ 教学实践#5：构建动手的实验体系

# OpenHarmony特色的实验体系

- **问题与挑战**

- 传统操作系统课程往往基于xv6等小型系统进行开发，导致学生难以深入理解真实复杂系统

- **实施方法：**

- 基于OpenHarmony关键特性，例如分布式能力、ARK框架（Stage模型等）、图形等，设计实现体系
- 为学生提供可以运行OpenHarmony的开发版或设备（例如可编程小车）
- 不限制学生的设计和实现路线，充分调动学生的主观能动性
- 实现将所学得到应用的目的

# 课程实验介绍

- **围绕OH内容，要求学生组队完成课程大作业**
  - 熟悉OH开发基本流程，实现简单App
  - 了解OH关键特性，并能用于开发
  - 熟悉OH某一（些）模块内部实现，并能进行针对性优化
- **华为专家全程参与**
  - 筹备阶段：命题讨论
  - 过程管理：相关主题介绍、答疑
  - 结题阶段：参与答辩和评分

# 课程实验介绍

## • 实验题目概览

- 基于OH的分布式应用 (关键词: stage模型、分布式协同)
- 基于OH一碰配网的智能小车 (关键词: 互联、端云协同)
- 基于OH的高效自动化截图 (关键词: 虚拟屏)



### AutoScreenShot

持续时间 40 秒

间隔时间 1000 毫秒

截图

暂停

重置

00:00:39.99

# 存在的一些问题

- **OH本身是一个非常庞大、复杂的系统，并且在不断演进**
  - 如何总结和提炼内容对教师而言，是一个非常大的挑战
- **课程实验难度较高**
  - 开发板调试本身就存在一定难度
  - 大规模代码阅读有挑战性
- **与社区互动比较有限**

# 未来展望

- 进一步增进对OH系统的了解，提升授课质量
- 完善课程实验
  - 提供更多引导，使同学们专注于关键功能实现
- 从社区汲取优质课题，希望能回馈社区

**请各位专家老师批评指正！**

