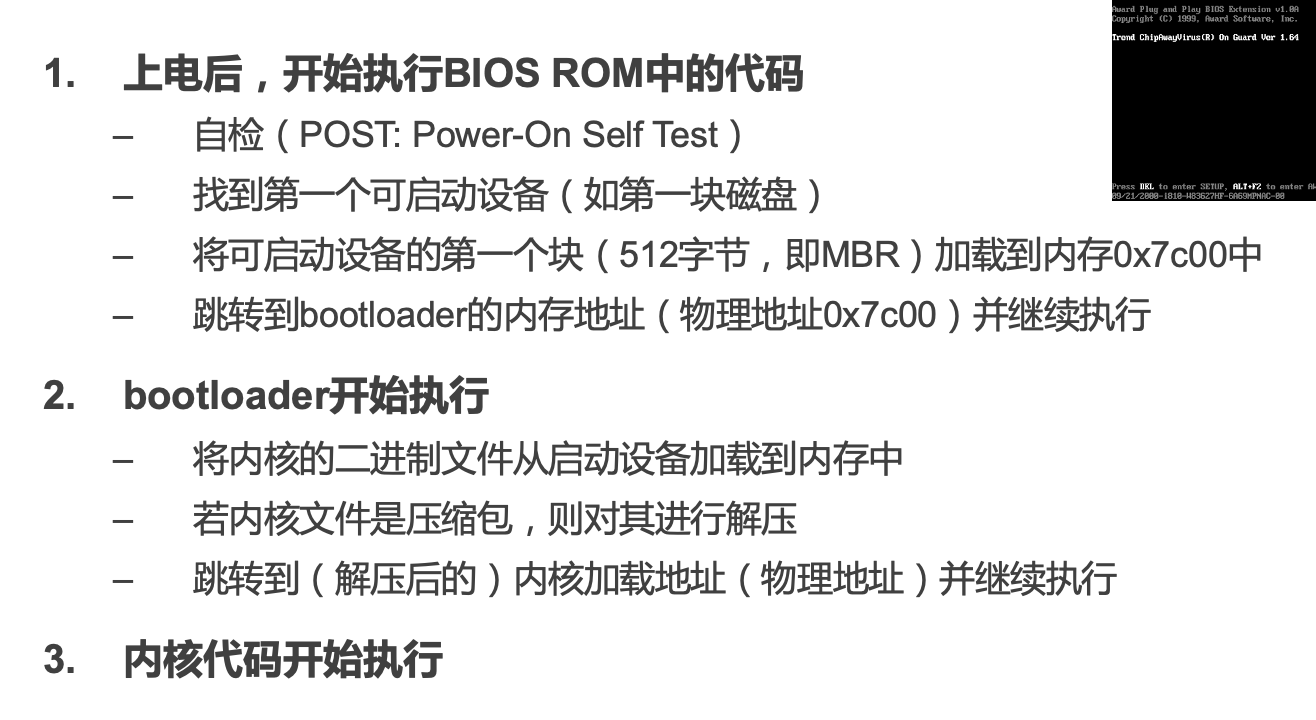
# 5虚拟内存管理

## 内核启动前：BIOS作用



512bytes：磁盘一个块

BootLoader：OS的代码

启动设备：磁盘OR flash

> linux编译完成后是压缩文件包



Why会出现BIOS？

IBM PC兼容机：最早做电脑，（乔布斯传），最早卖电脑，是没有BIOS的，所有东西都是自己拼装的，不用考虑网卡，磁盘跟别人兼容。

IBM采用了另一种模式：它定标准，设备都不是自己造的，告诉别人设备是如何启动的。

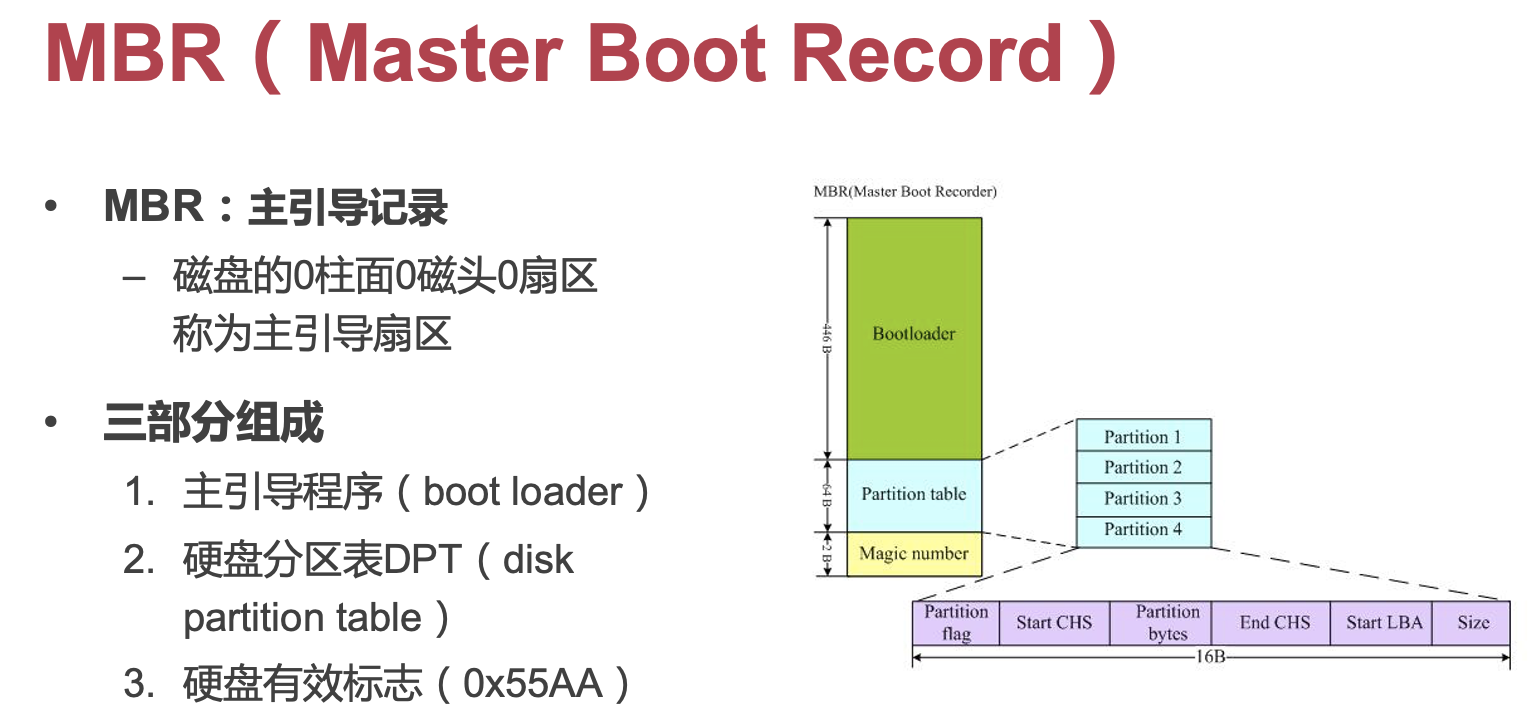
> IBM把标准开放了，比如电脑坏了，能直接替换磁盘，网卡等，换了之后能把设备识别出来。各种厂商的设备，通过BIOS识别，检查。

Q：为什么很多嵌入式设备没有BIOS？

A：场景专用，可以省掉很多组件，不用兼容别人，可以减少成本。——导致千国百治

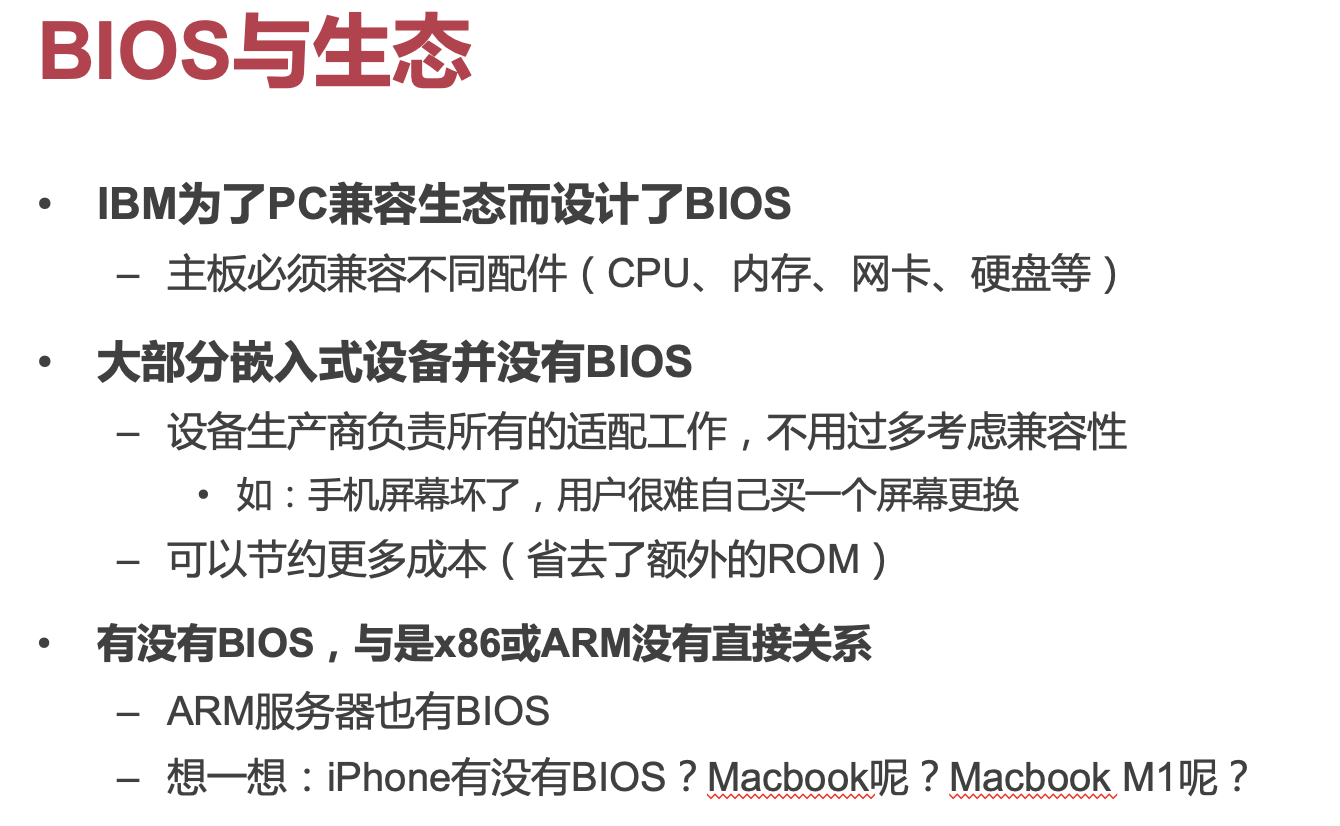


BIOS会看我有几块硬盘，是什么类型的，满足什么标准，识别PCI设备，IRQ（中断号）等

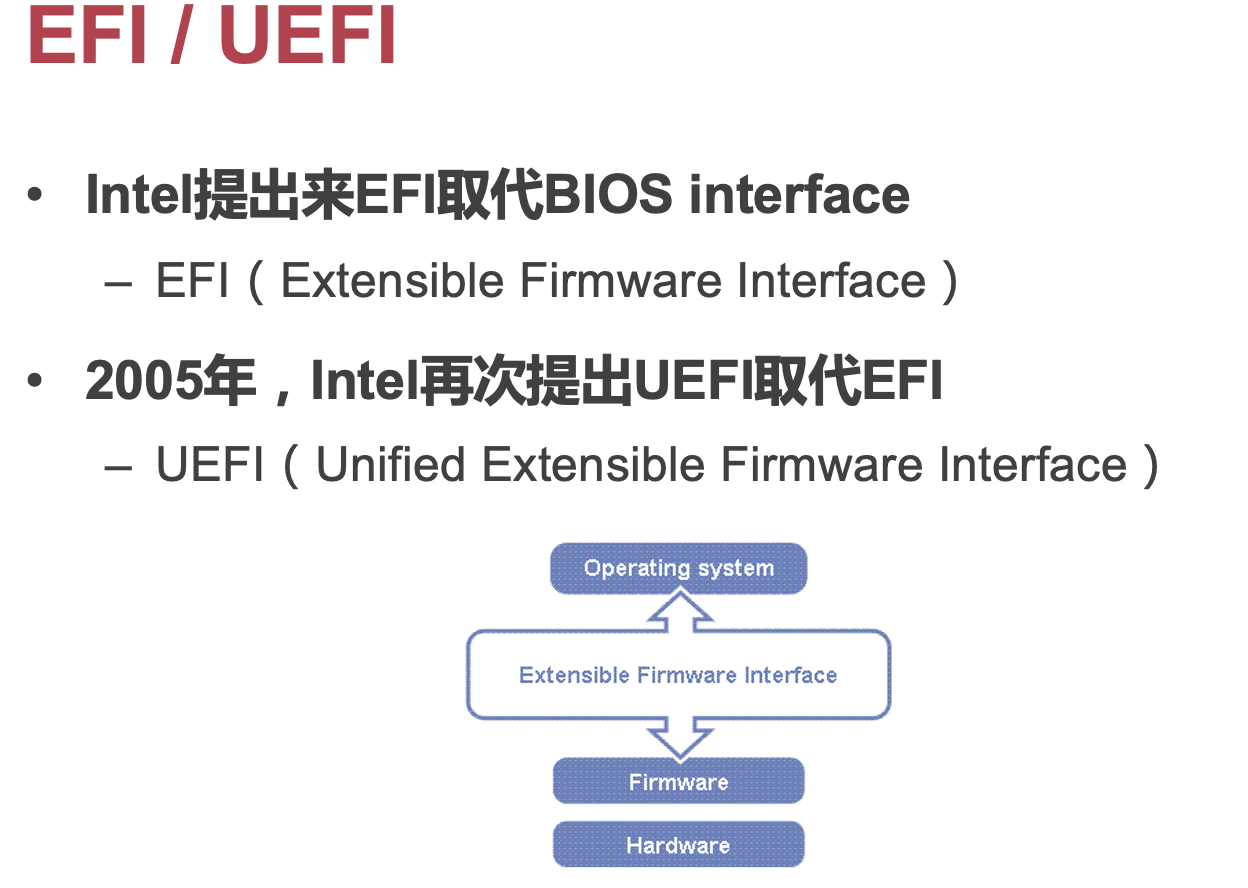


约定：OS代码到底放在哪？——一般：磁盘0柱面0磁头0扇区。

55AA如果变成56AA之类的说明磁盘损坏。

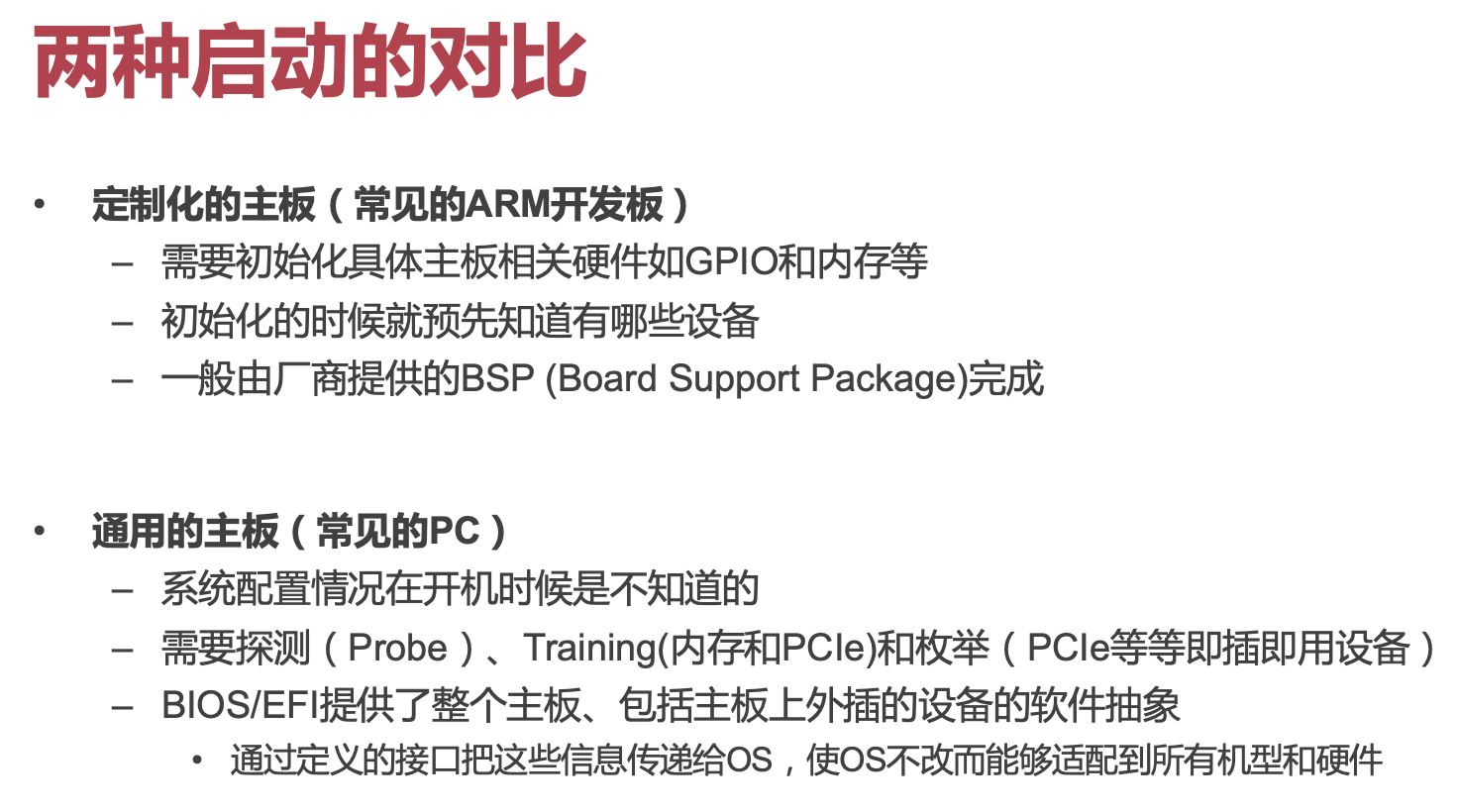


促进业界分工。iPhone mac等等都没有BIOS。

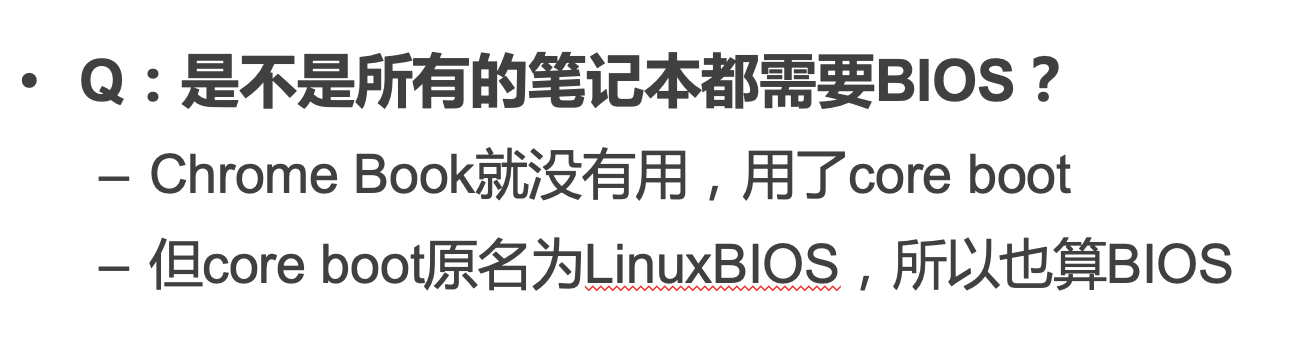


本质上还是做了BIOS要做的事情

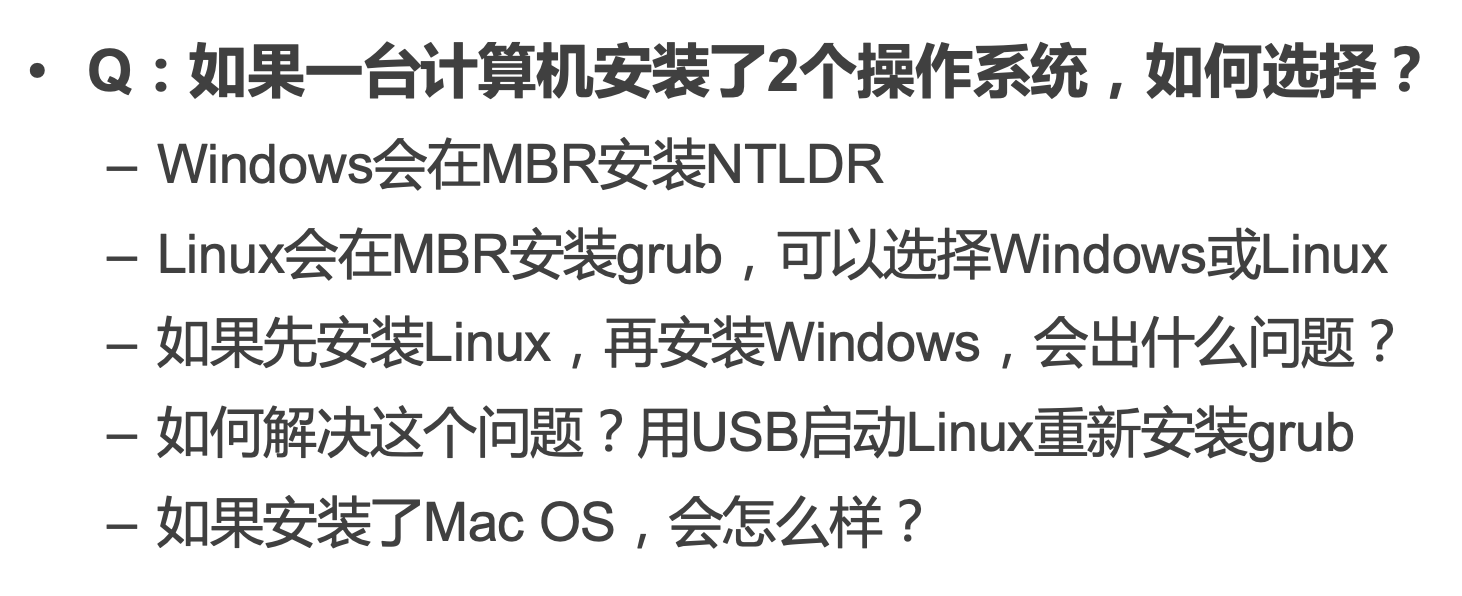
e.g. 可以改变启动扇区，不会出现0号扇区出问题了就没法启动。



没有BIOS的话，用BSP来完成硬件启动。

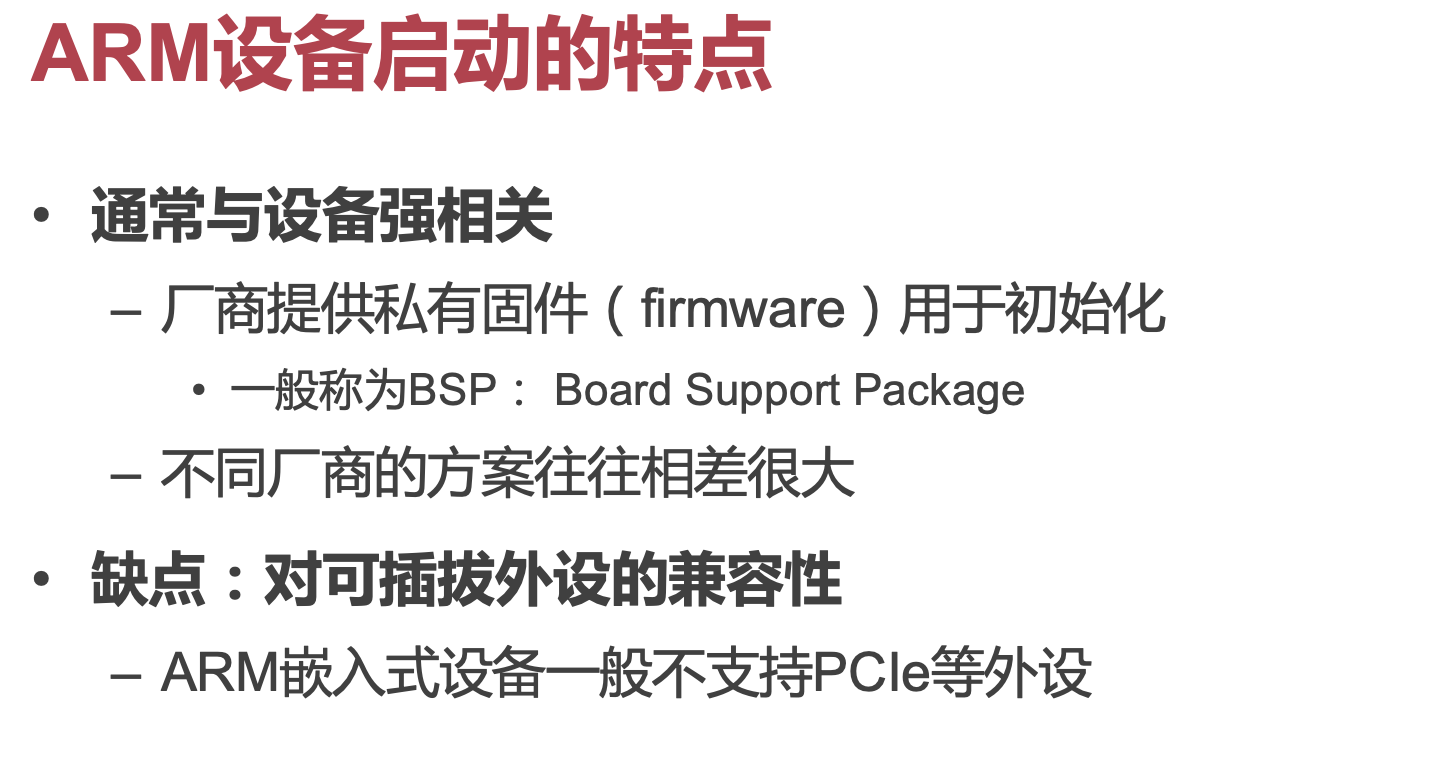


M1也没有。



3. grup就被覆盖了。Linux就不见了。

5. MacOS就得用mac的引导程序。

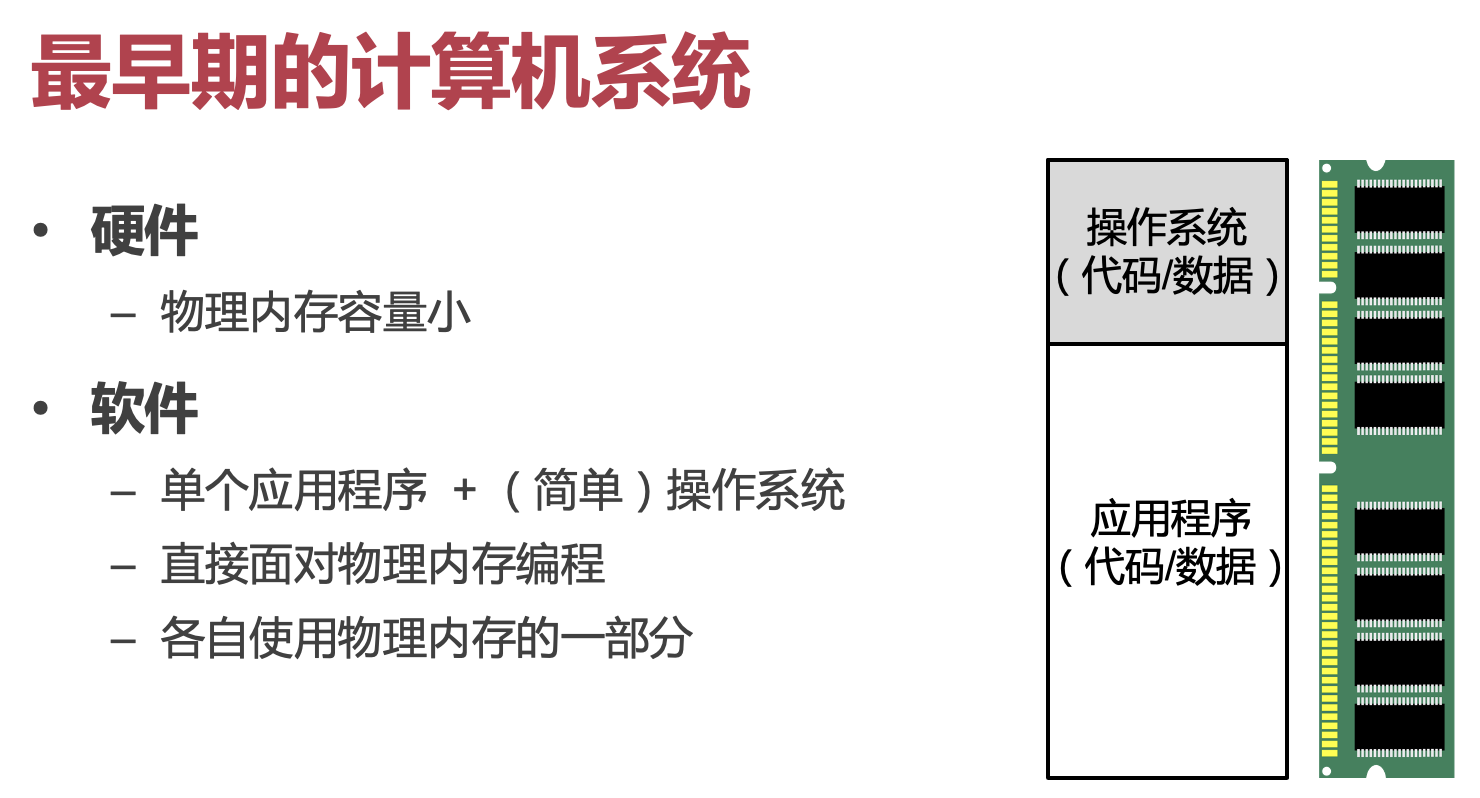


比如树莓派：插兼容外设很困难。

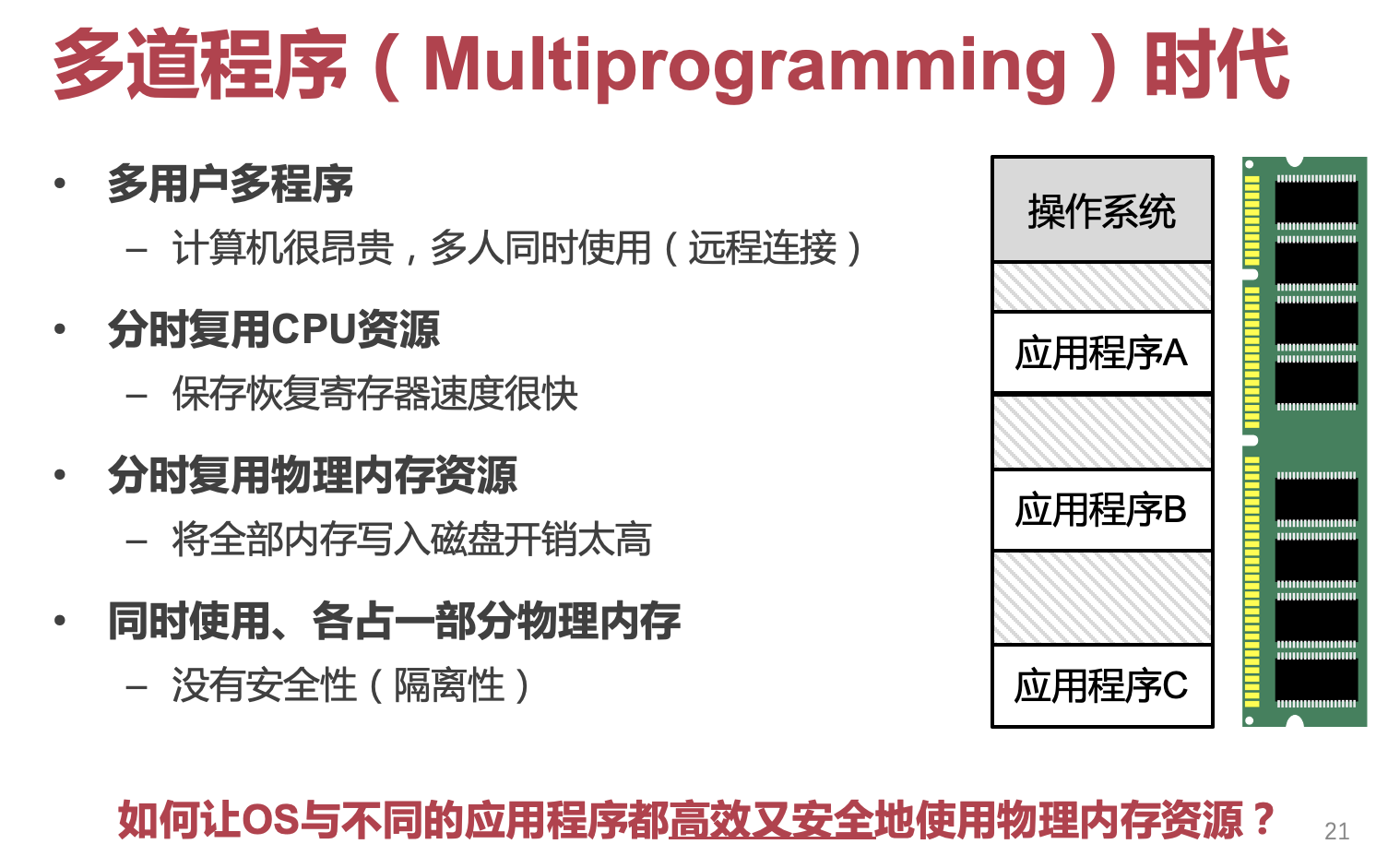
生态碎片化：pros：创造工作机会。

## 虚拟内存

### 硬件做了什么



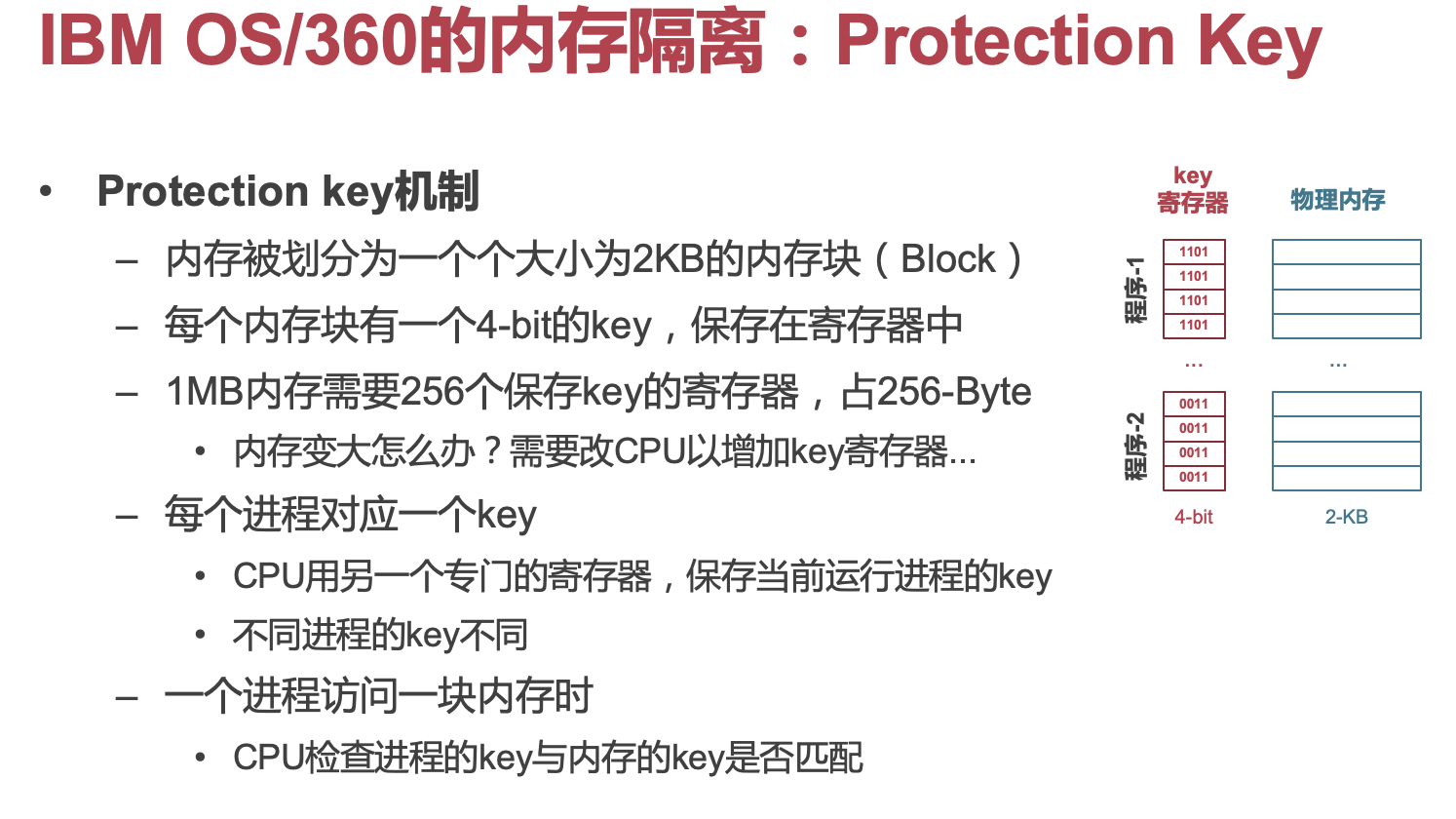
只接受物理地址编程，每个应用使用物理内存一部分，隔离性很差（Win98仍为完全使用VM，程序经常把OS地址覆盖了——只能重启）



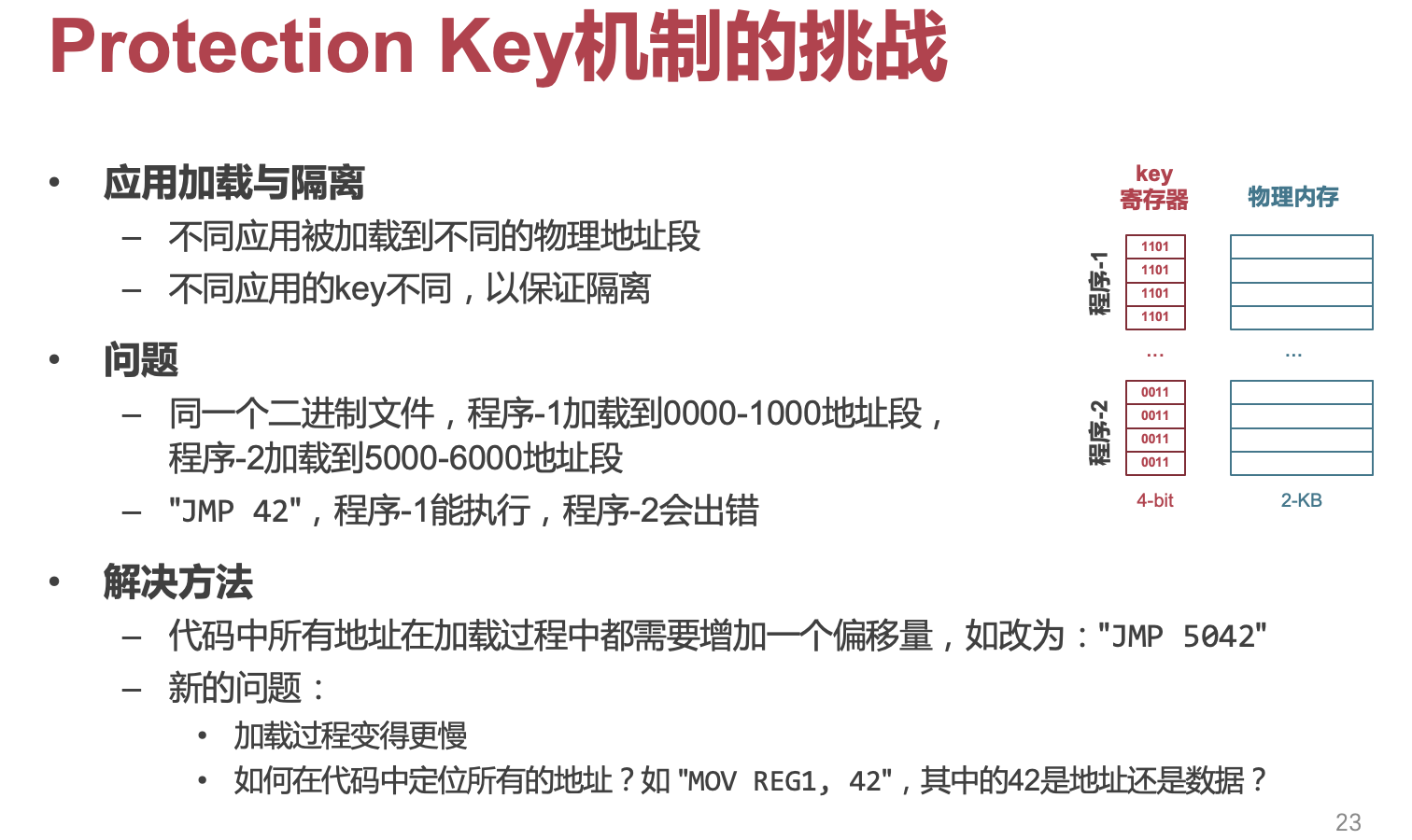
为什么现在还要讲如此古老的设备：现在还有很多设备没有MMU支持。e.g.车上设备（刹车控制器——很简单的OS，仪表等等）

MMU：哺乳类动物——5w种

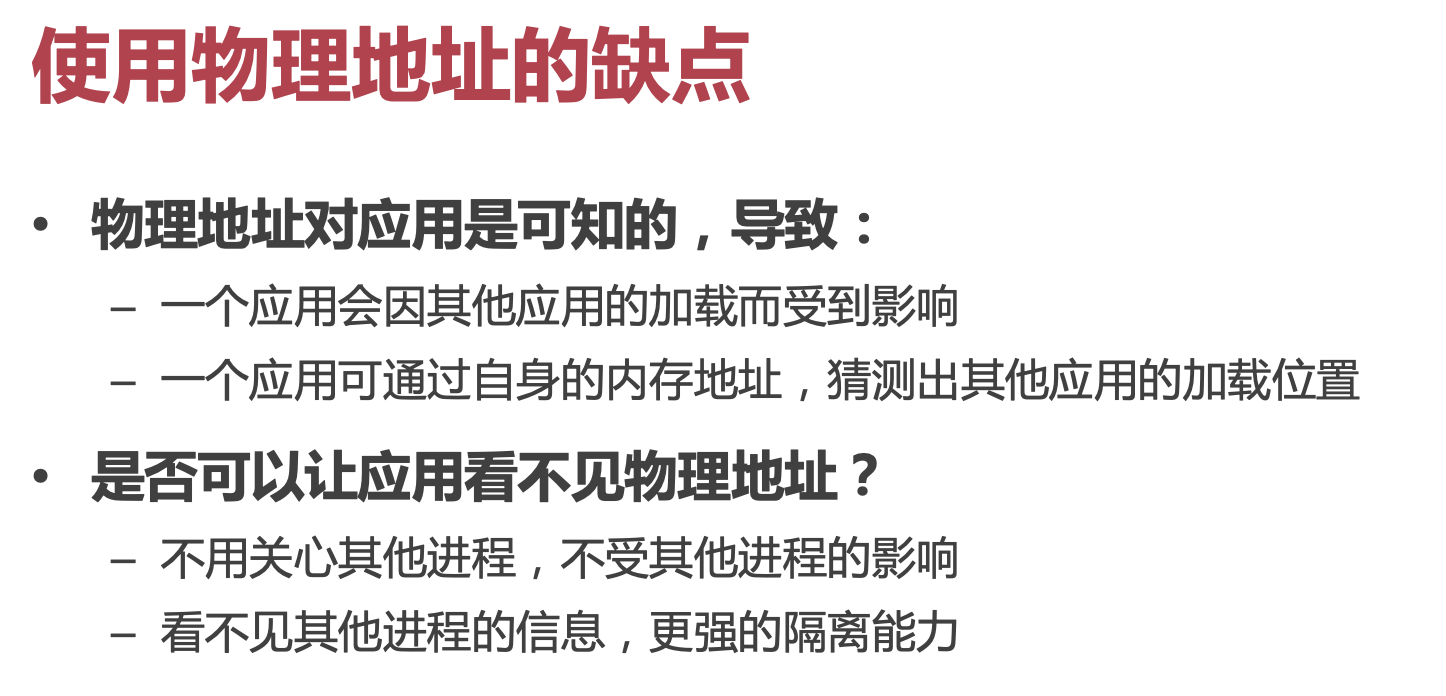
没有MMU：昆虫——500w种



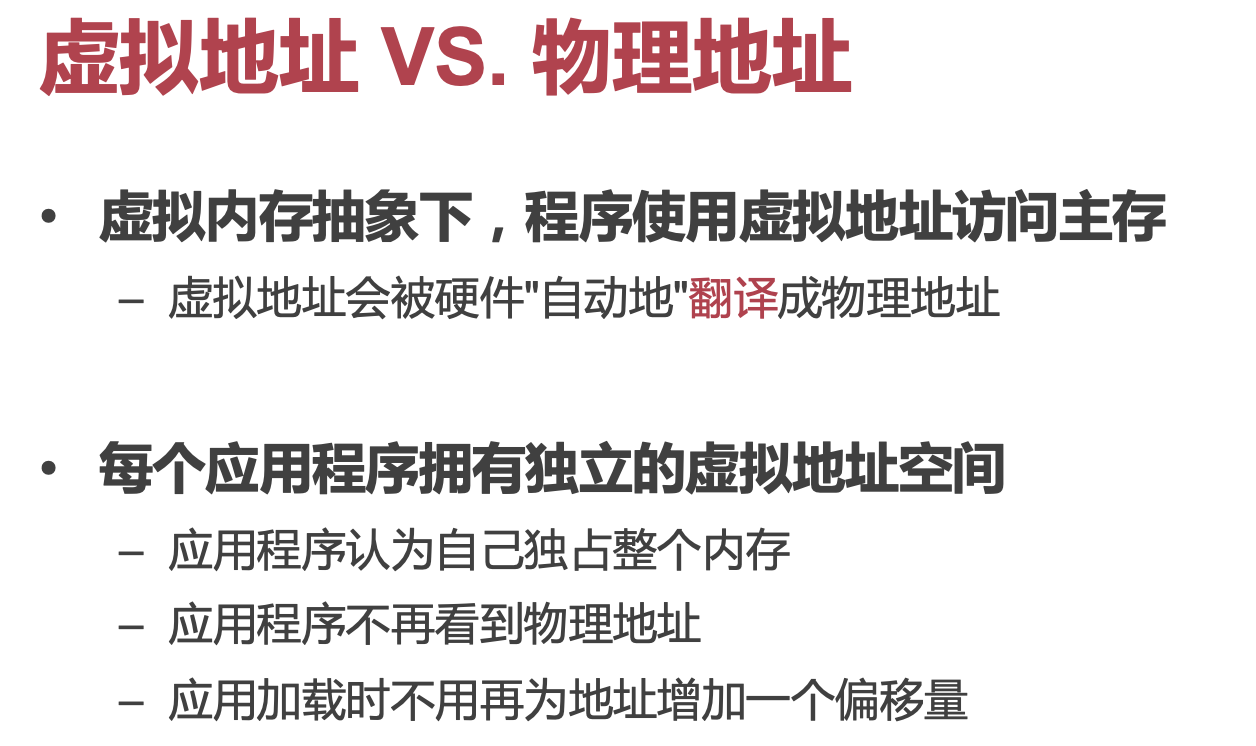
限制：只能支持16个线程，OS再占一个，剩15个。

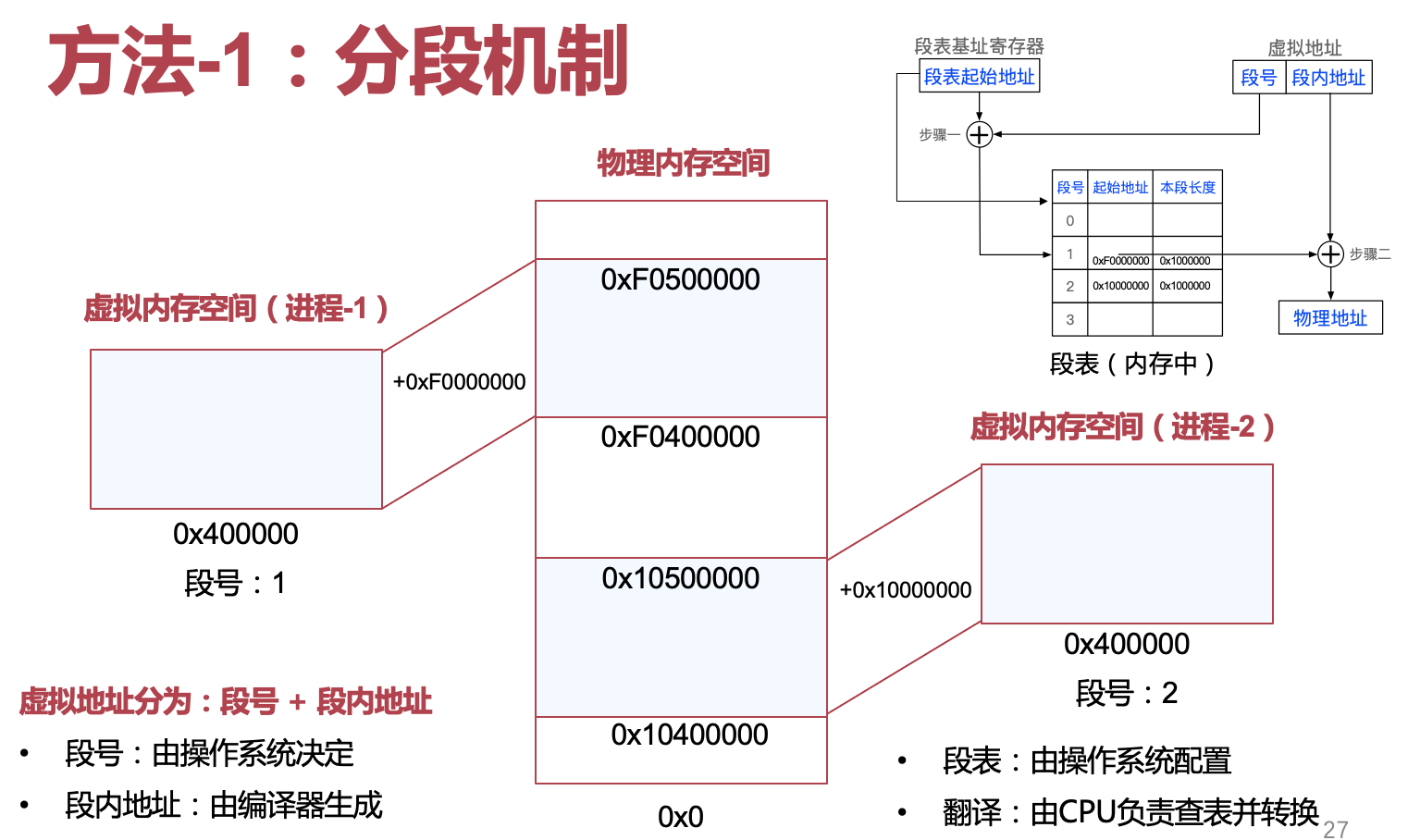


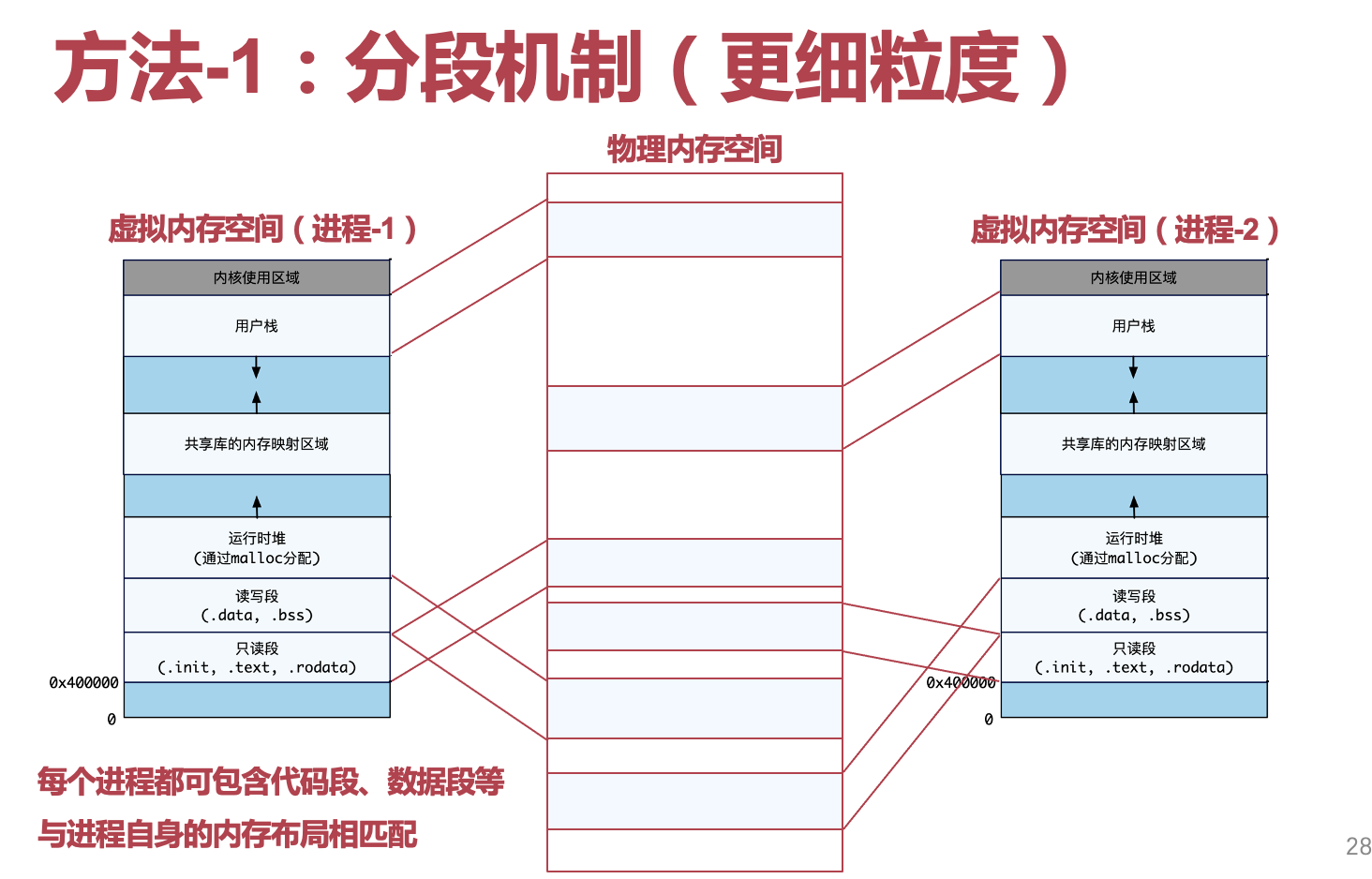
代码加载过程中加bias，变的话要重新加载，很难定位所有地址

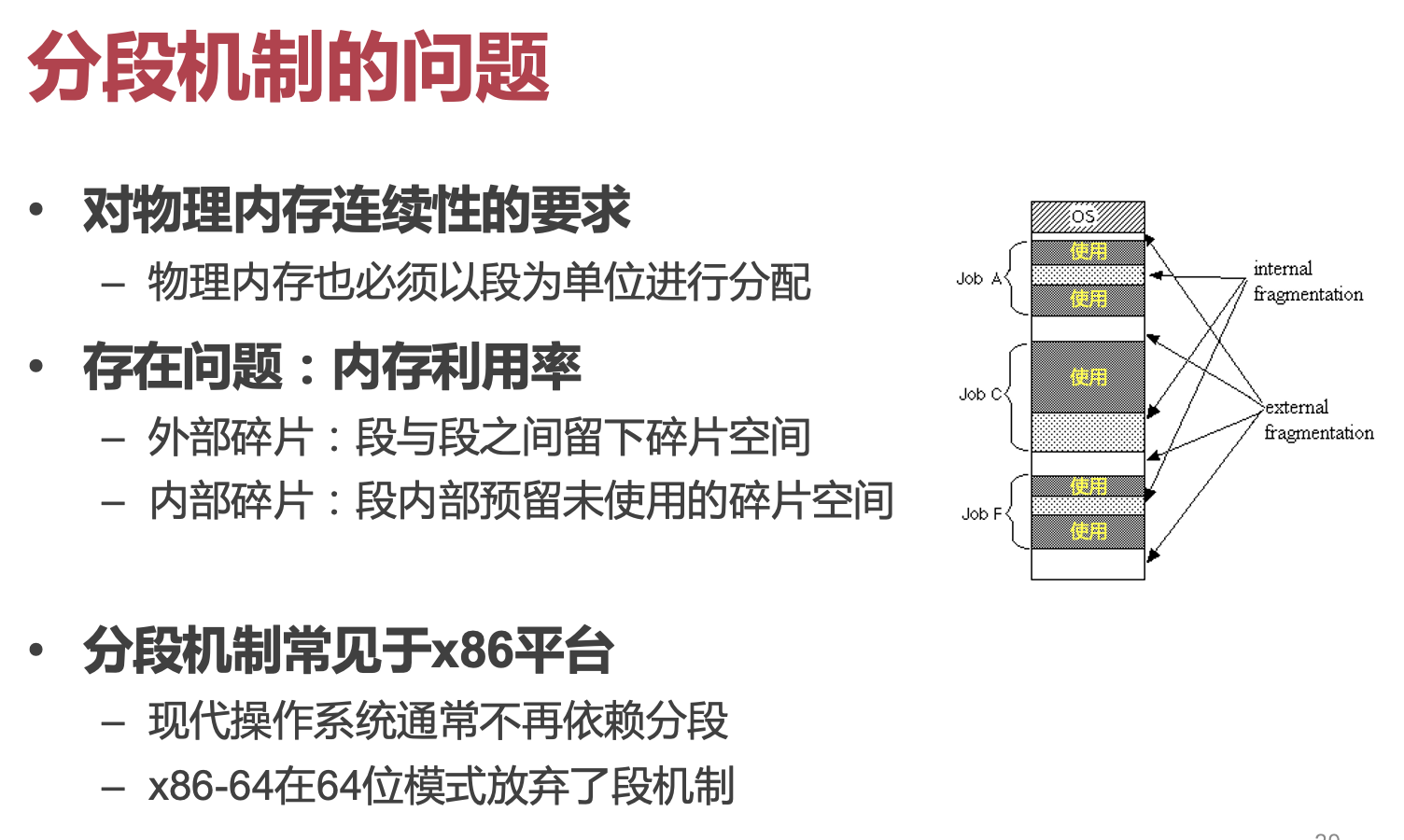


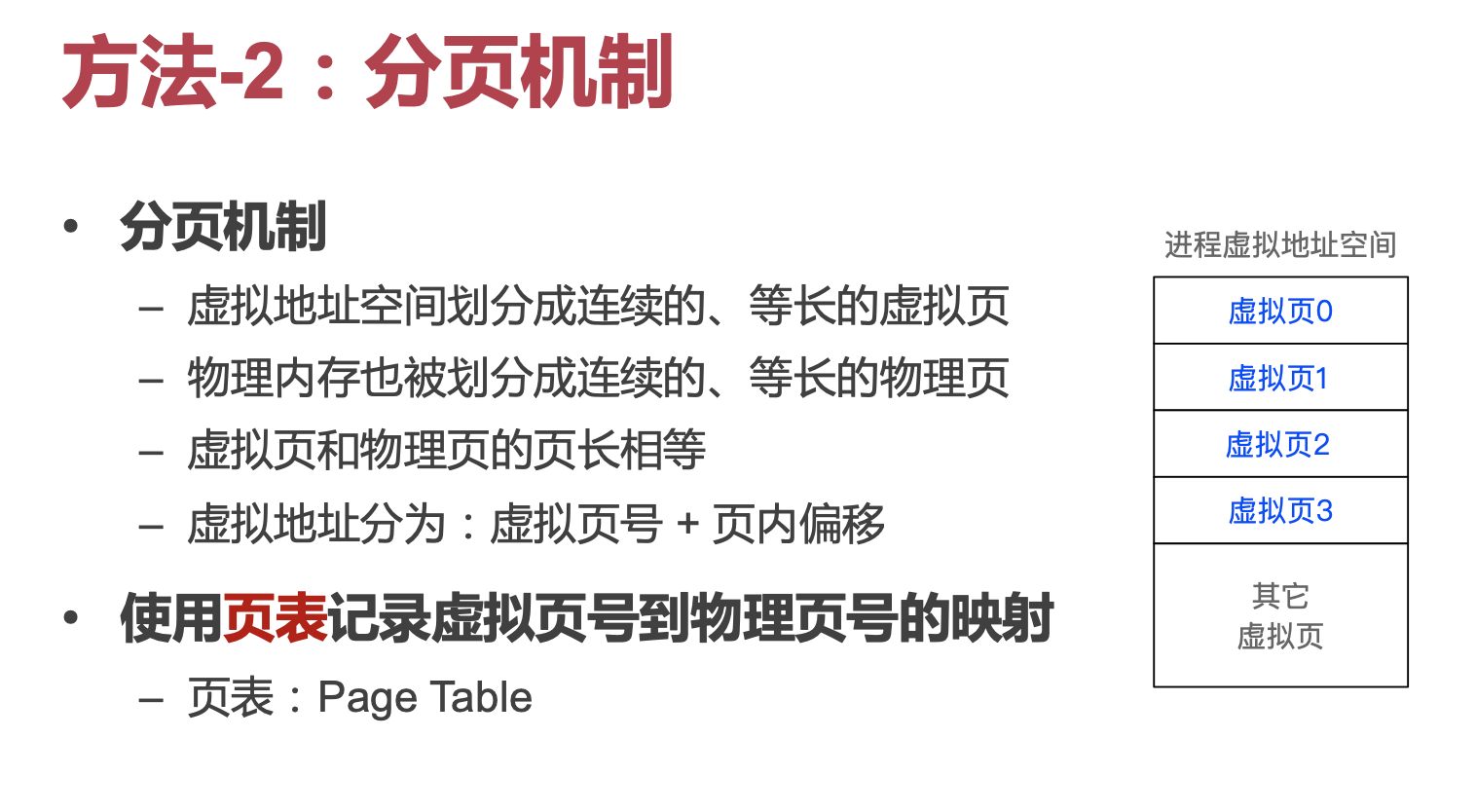
现在protection key仍然有用：比如riscV的处理器，假设只有5个程序。

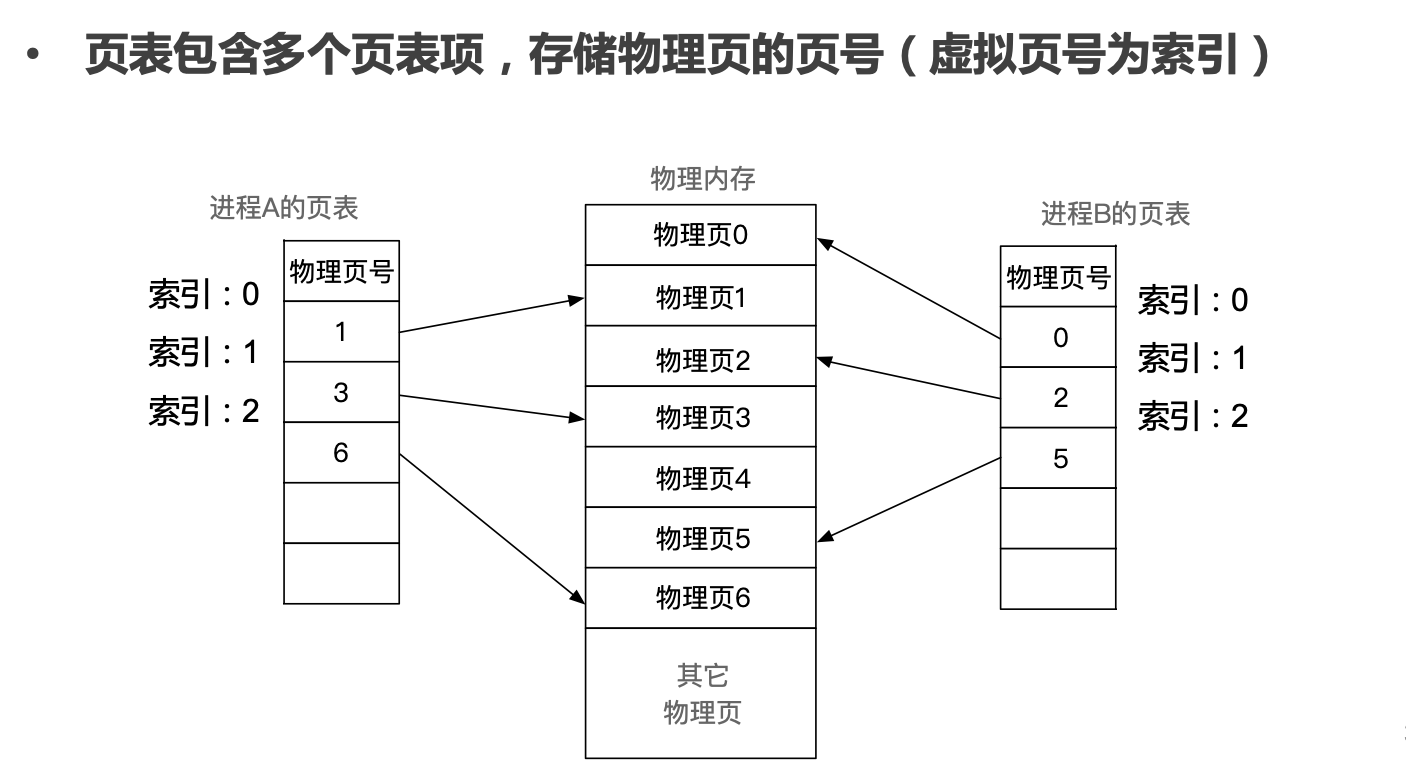


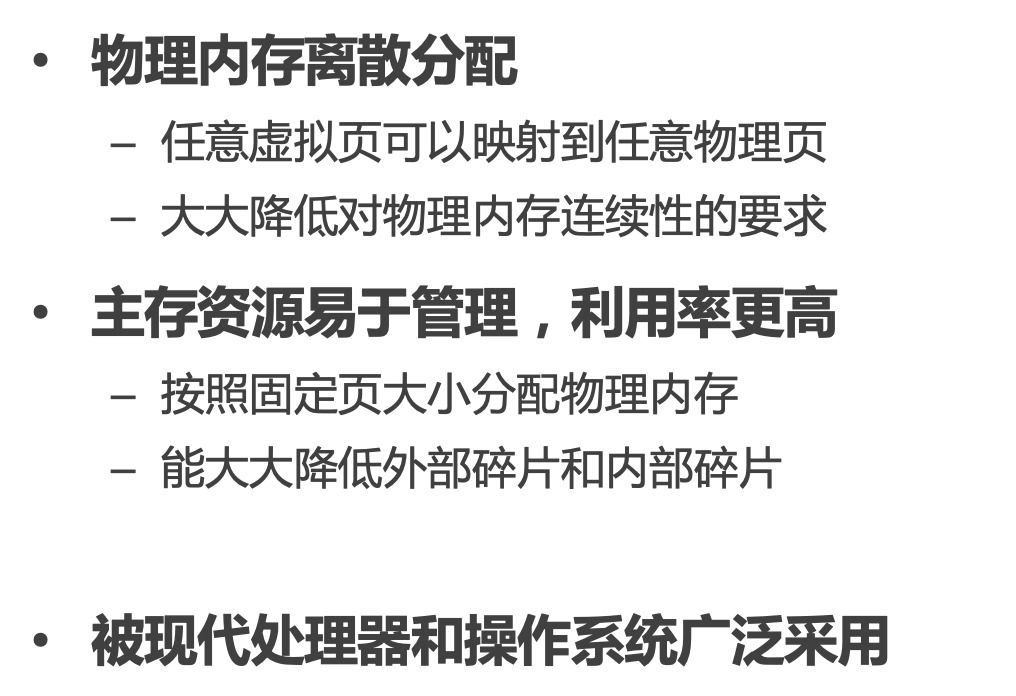




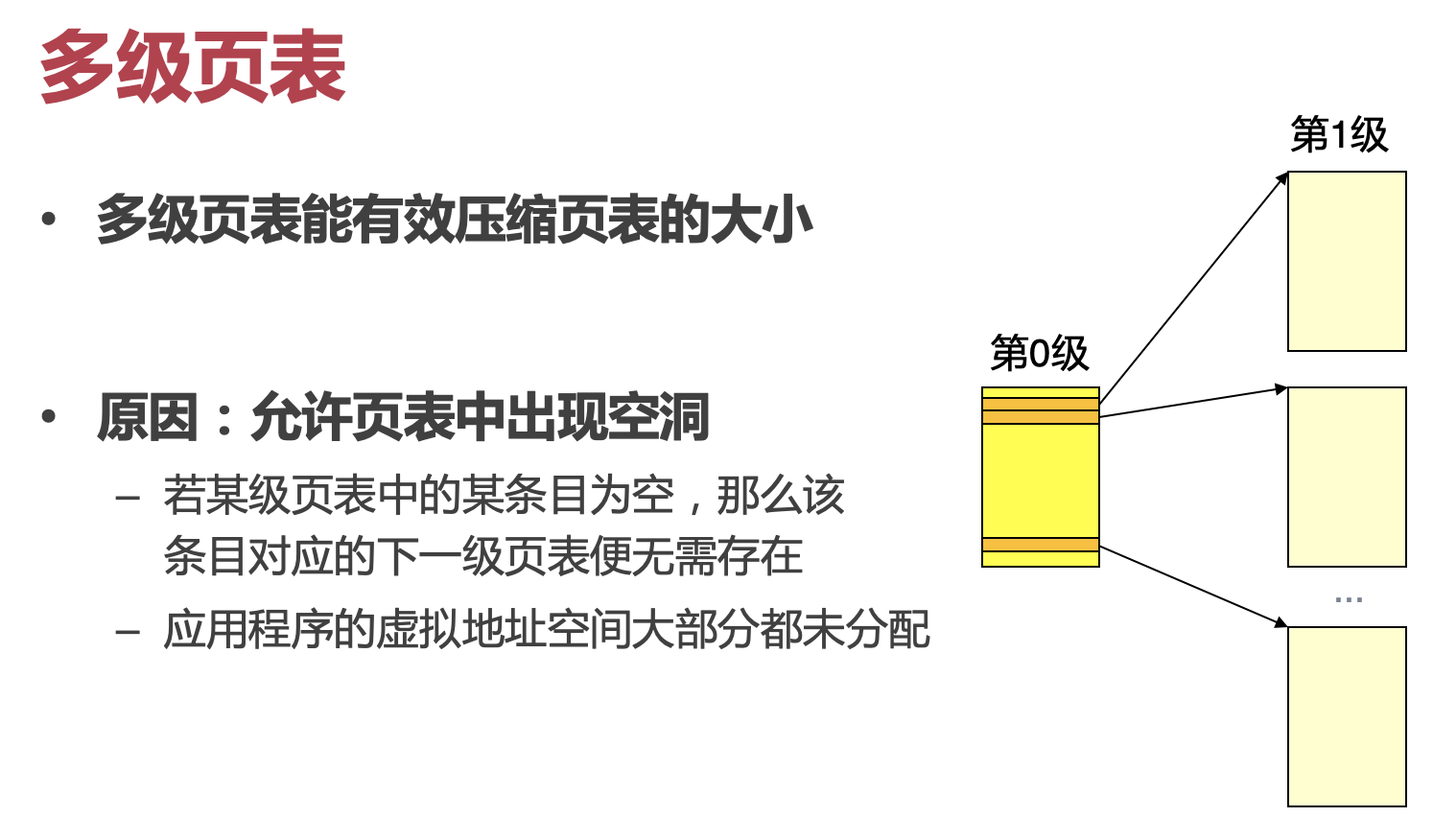








FreeRTOS：没有VM



AArch64



48~63bit：全0（应用程序）或者全1（内核地址）

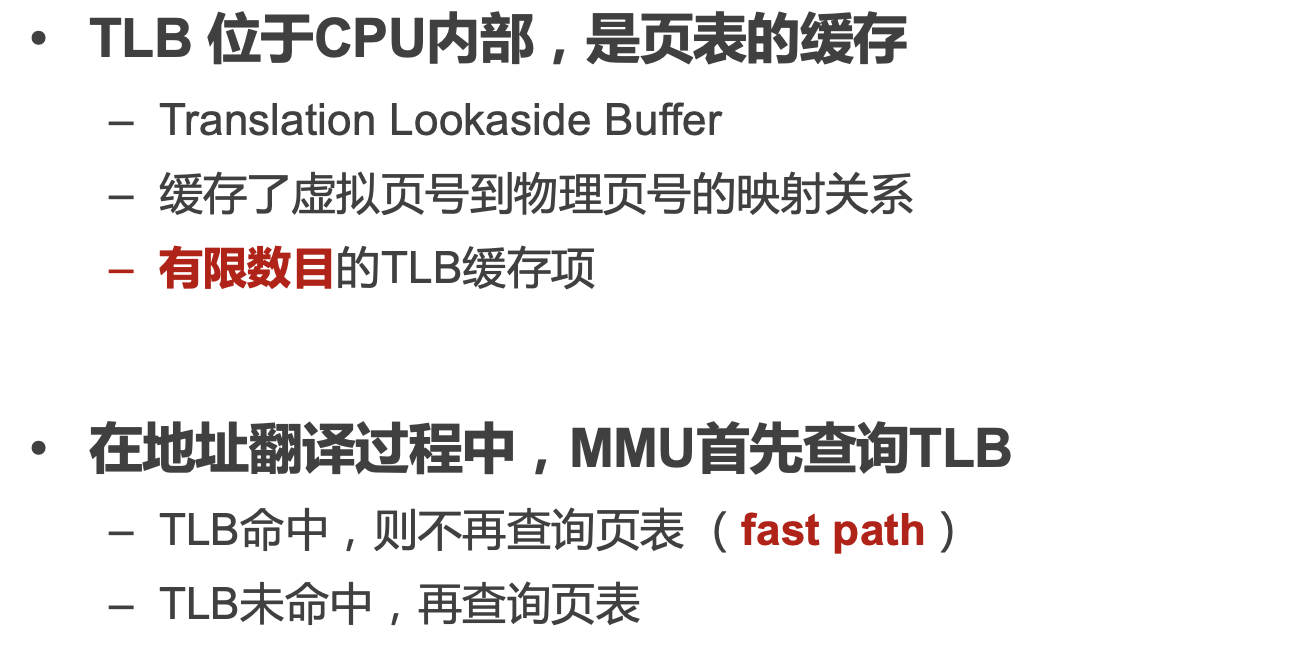
39~47bit：0级页表索引 2^9 下一级有512个

etc

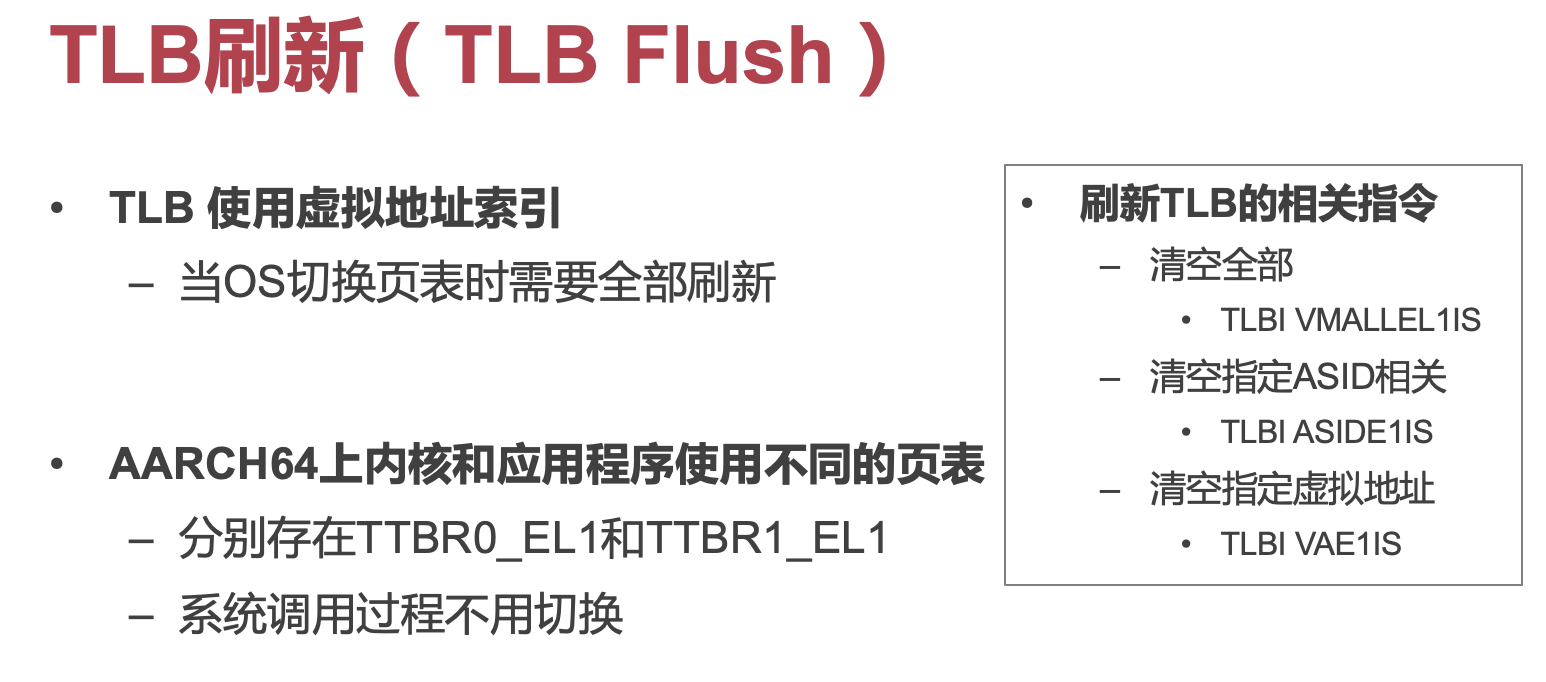


两个寄存器的好处：早期x86不切CR3。18年之前用户态内核态一个CR3。18年：Melt Down漏洞。用户态共享页表，通过硬件中猜测执行读取内核地址。——Arm躺赢……用户态，内核态两个页表。

### TLB：页表cache



Arm等现代处理器里面TLB多级，也有类似L1，L2 cache。



上下文切换不用刷。进程间切换还要。

IS：inner sharable

SID：如果发生进程切换就刷掉TLB，进来再重新加载回来效率低。现代操作系统：TLB加上SID tag，切换进程不用刷TLB。

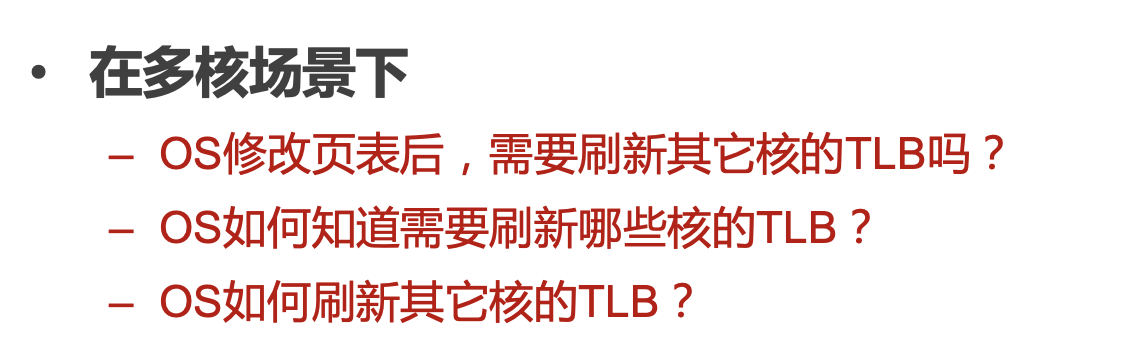


Q：修改页表映射后，仍需刷新TLB？

A：

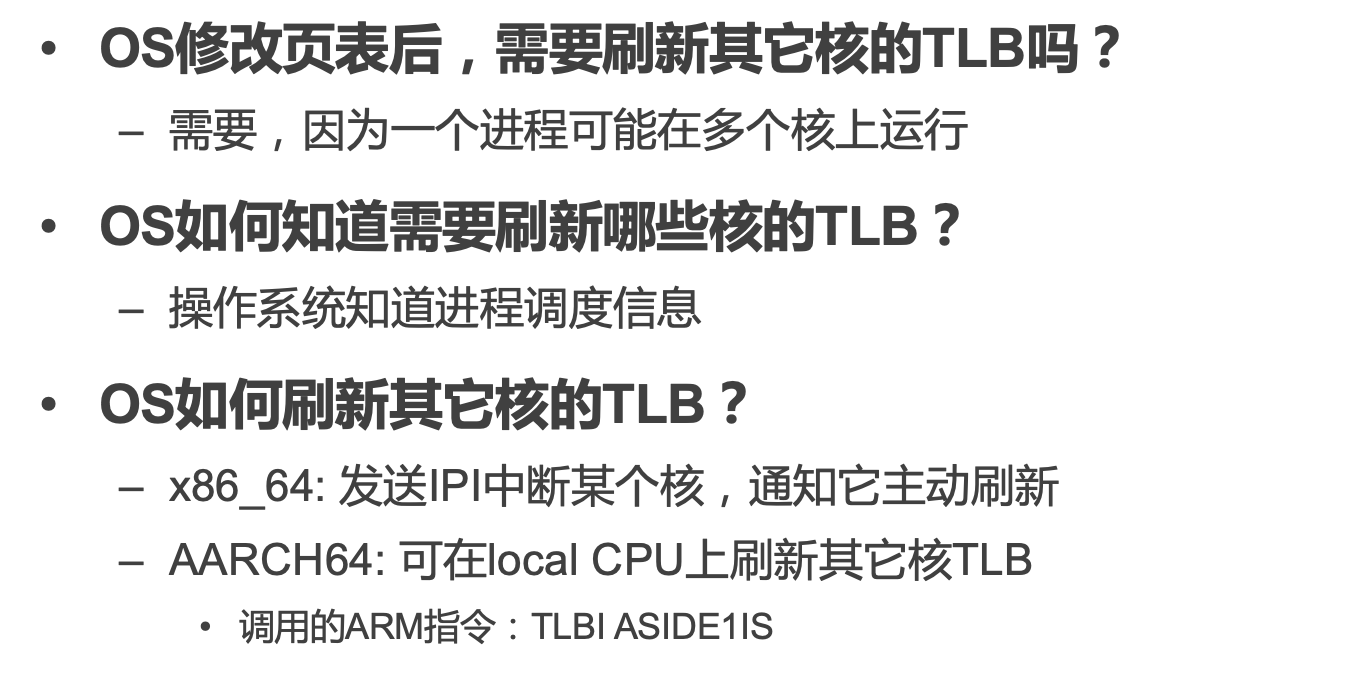
Q：8bit SID只支持256个进程，不够用？

A：手机大概8000个线程，suppose 1w个进程。需要进程和SID的映射，最常用的跟SID对应起来。不常用的就可以共享SID，还有很多问题？如何知道对应关系，需要进程调度等等多做一些工作。



1. YES

2. 怎么知道哪个进程在哪个核？OS知道哪个进程在哪（些）个核上。



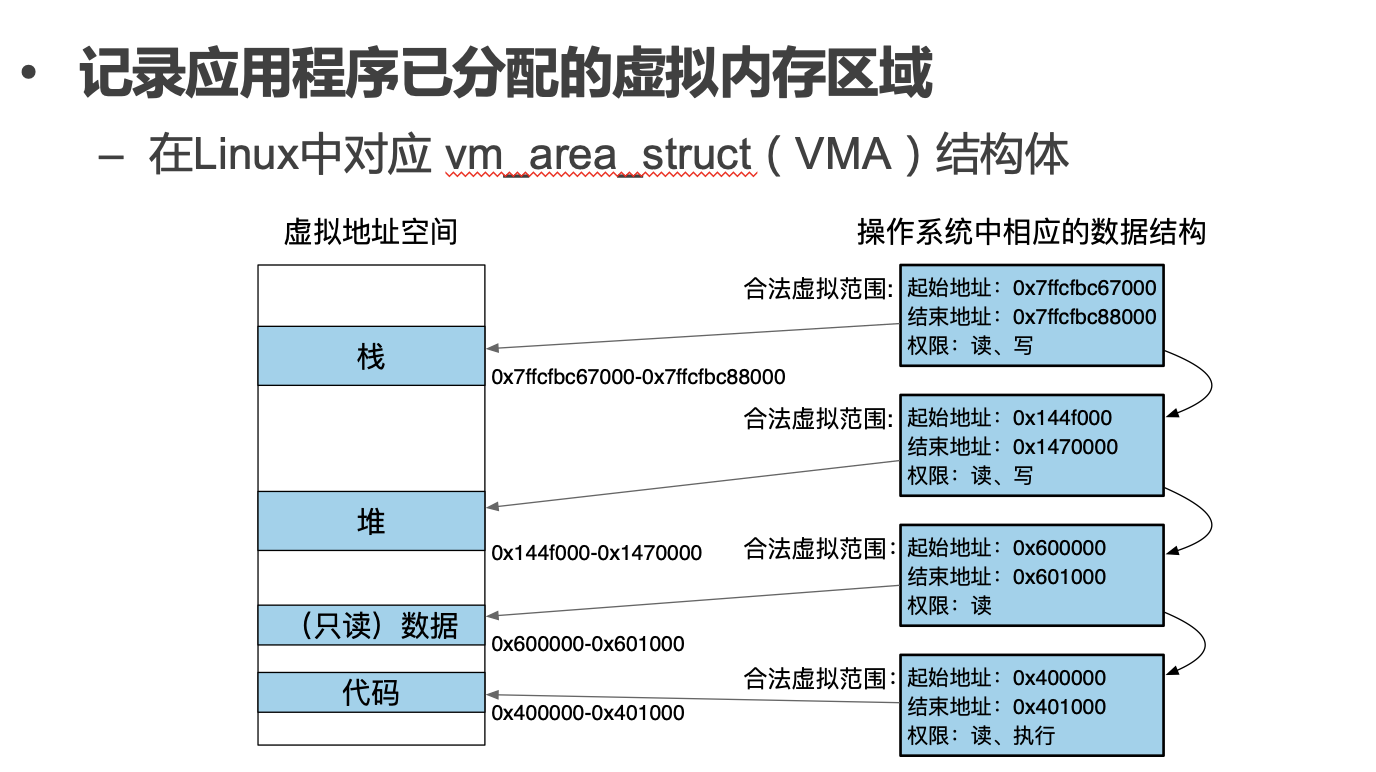
### 段 VMA

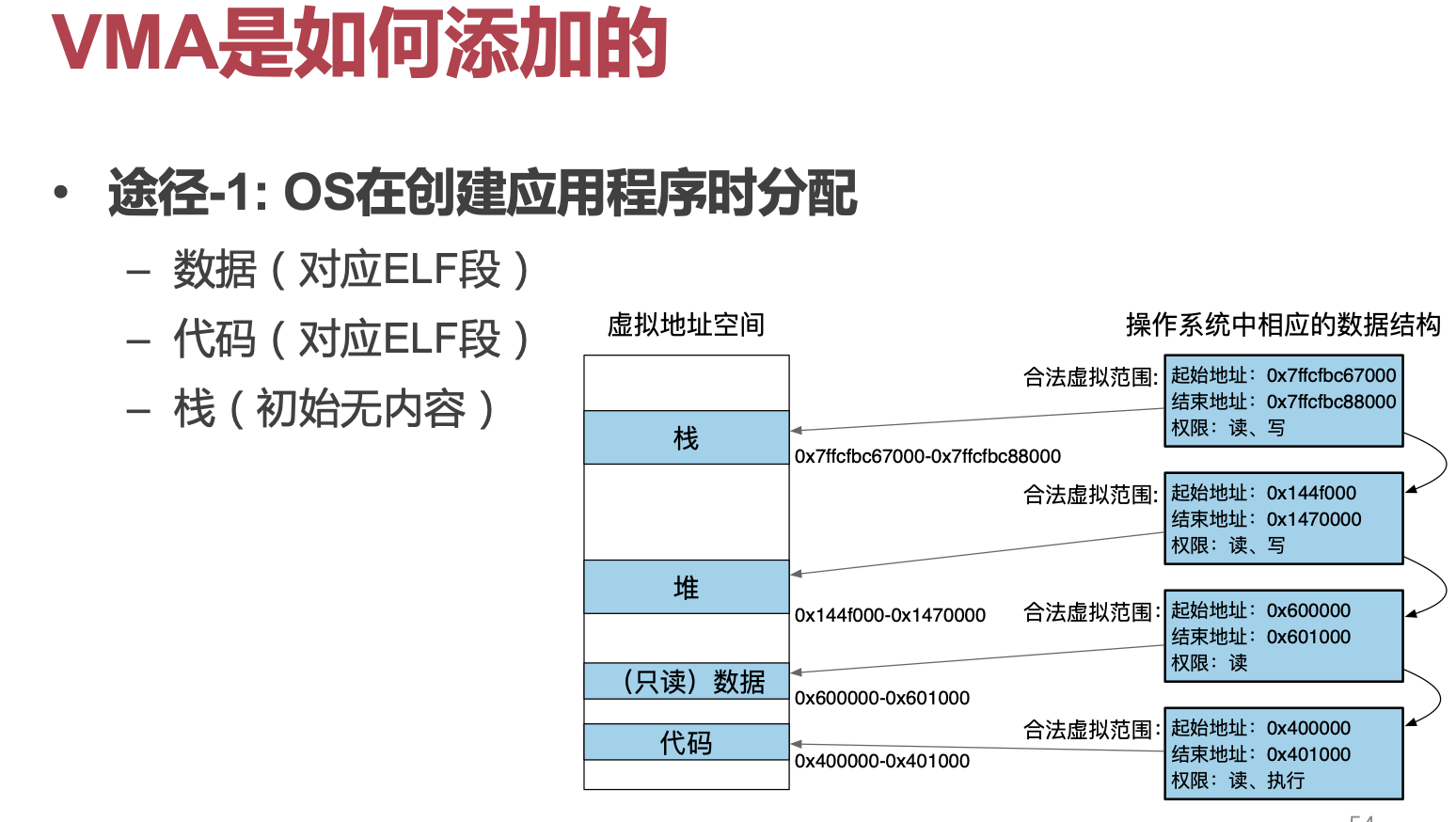
现在没有segment了？为什么还叫segmentation fault？

虽然硬件没有段的机制了，操作系统还是用段来管理虚拟内存的。



用页是否可以？用段简洁，因为相同段内的页往往具有相同的读写权限等等。





还有COW直接复制。

