- 中心主题
  - 如何理解地址空间
    - 什么是划分区域：
      - 地址空间的本质是内核的一个 struct 结构体！内部有很多的属性都是表示 start，end
    - 地址空间的理解：
      - 将多个区域通过页表映射到物理空间
  - 为什么要有地址空间，而不直接使用为物理地址
    - 在实际的物理内存中，不同功能的区域是分散的比如代码区在一个区域，数据区在一个区域，堆区栈区又在一个区域等
    - 地址空间将无序变为有序，让进程以统一的视角看待物理内存以及自己的各个区域
    - 将你申请但没有立即使用的的堆上空间，将申请的信息先放入页表而不立即申请，提高内存使用率
    - 拦截对物理空间不正常的非法请求
  - 进一步理解页表和写时拷贝
    - 页表不只是包含地址映射的信息，还包括：
      - 该区域的读写权限，防止非法操作
    - 写时拷贝：
      - 将变量标记为只读的；
      - 当修改时，识别到错误，因为是只读的
      - 是不是不在物理内存中；是不是需要写时拷贝；如果都不是，才进行异常处理；
  - Linux真正的如何实现调度的？
    - Linux系统中每一个 CPU 都有一个运行队列（大小为140），使用的范围是 100 - 139；
    - Linux系统中每一个 CPU 都有一个运行队列（大小为140），使用的范围是 100 - 139；
      - 将优先级 + 40 作为索引，放入该队列对应索引的尾部
    - 利用位图来搜索正在执行进程的位置
  - 当子进程修改数据时，会分配一个新的空间放入修改后的数据，并将该数据的地址修改到子进程的页表中（写时拷贝）
  - 如果父子进程不进行修改呢？该数据默认被父子共享，代码是共享的（只读）
  - 利用写实拷贝的原因： 按需申请，提高效率，稳赚不亏。
  - 在同一个物理地址上，不可能存储两个不同的信息
  - 地址空间的本质就是内核中的一个结构体对象
  - 操作系统会为执行的程序创建虚拟空间
    - 包含页表的结构映射虚拟空间的地址到物理空间
    - 子进程会将父进程的很多内核数据结构全部拷贝一份（包括页表）