计算机操作系统

自映射

课堂练习

假设一个机器有38位的虚拟地址和32位的物理地址。

(1) 与一级页表相比，多级页表的主要优点是什么?

(2) 如果使用二级页表，页面大小为16KB，每个页表项有4个字节。应该为虚拟地址中的第一级和第二级页表域各分配多少位?

课堂练习

3. 假设一个机器有38位的虚拟地址和32位的物理地址。

(1) 与一级页表相比，多级页表的主要优点是什么?

(2) 如果使用二级页表，页面大小为16KB，每个页表项有4个字节。应该为虚拟地址中的第一级和第二级页表域各分配多少位?

答案:

除了顶级页表之外，使得每一个页表都放在一个物理页框中。

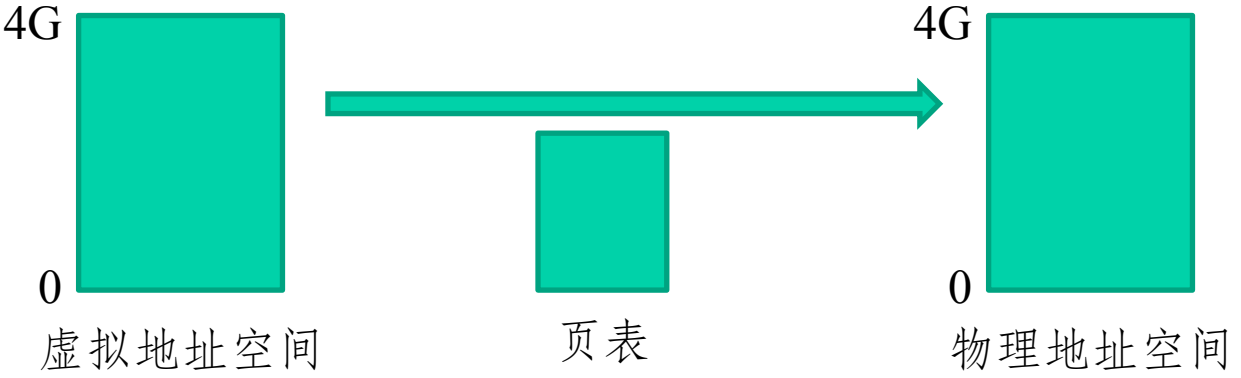
页面16K，页内偏移14位。让二级页表等于页面大小16K。

二级页面页表项数 ＝ 16K/4=4K； 12位。

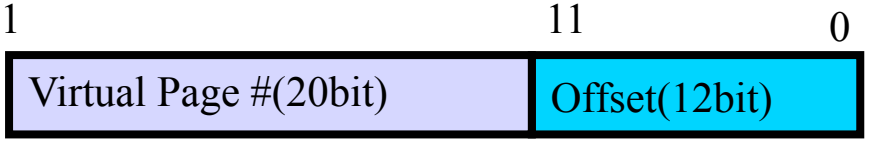
一级和二级页表域分别需要12位，页内偏移量需要14位。

专题：页目录自映射

- 基本事实：
  - 页表的作用是将虚拟地址空间映射到物理地址空间

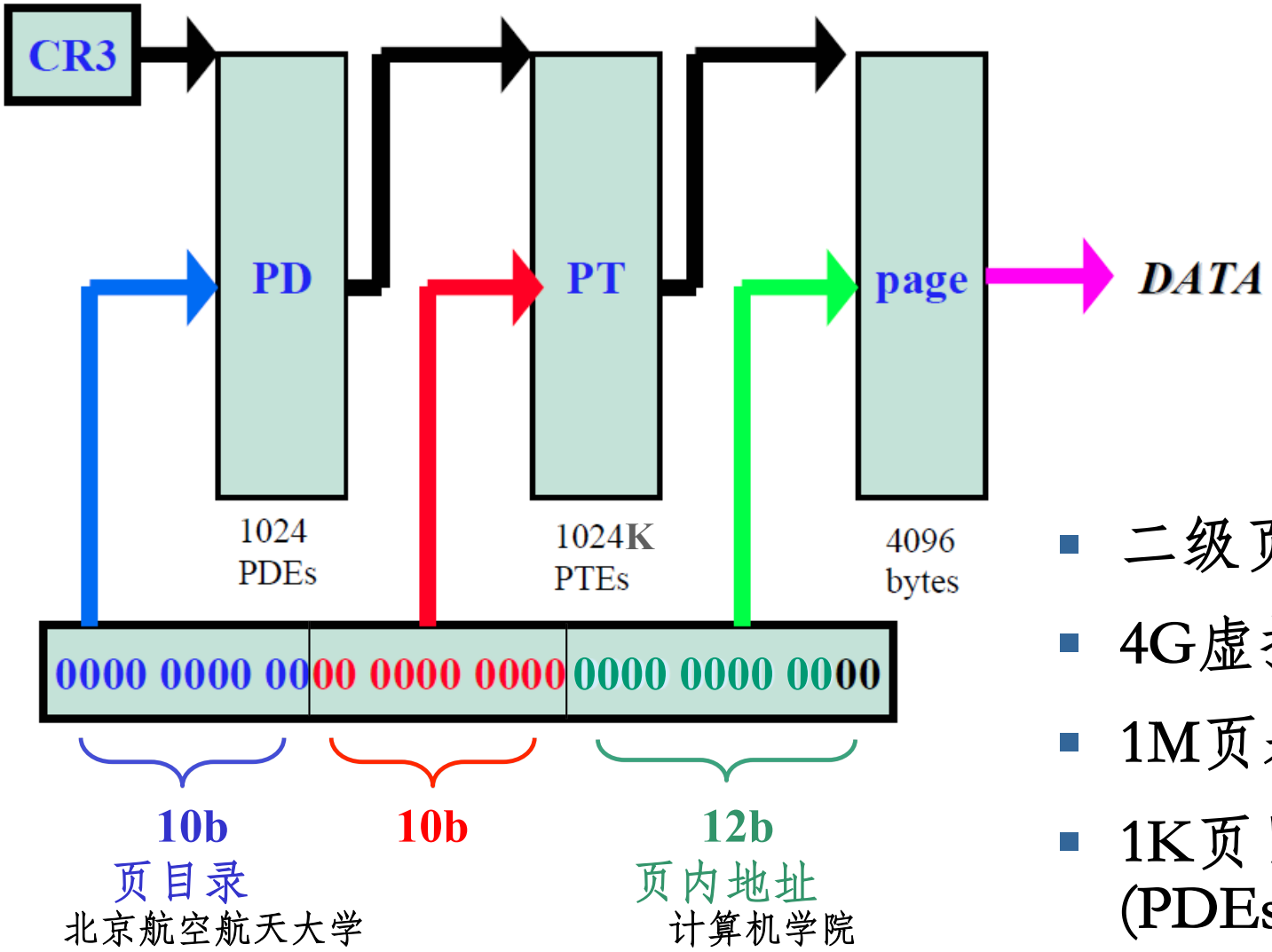


  - 对于32位地址长度，可寻址空间为4GB
  - 采用12位页内偏移，表明内存页大小为4KB
  - 每个页表项负责记录1页（4KB）的地址映射关系
  - 整个4GB地址空间被划分为4GB/4KB=1M页，所以需要1M个页表项来记录逻辑-物理映射关系



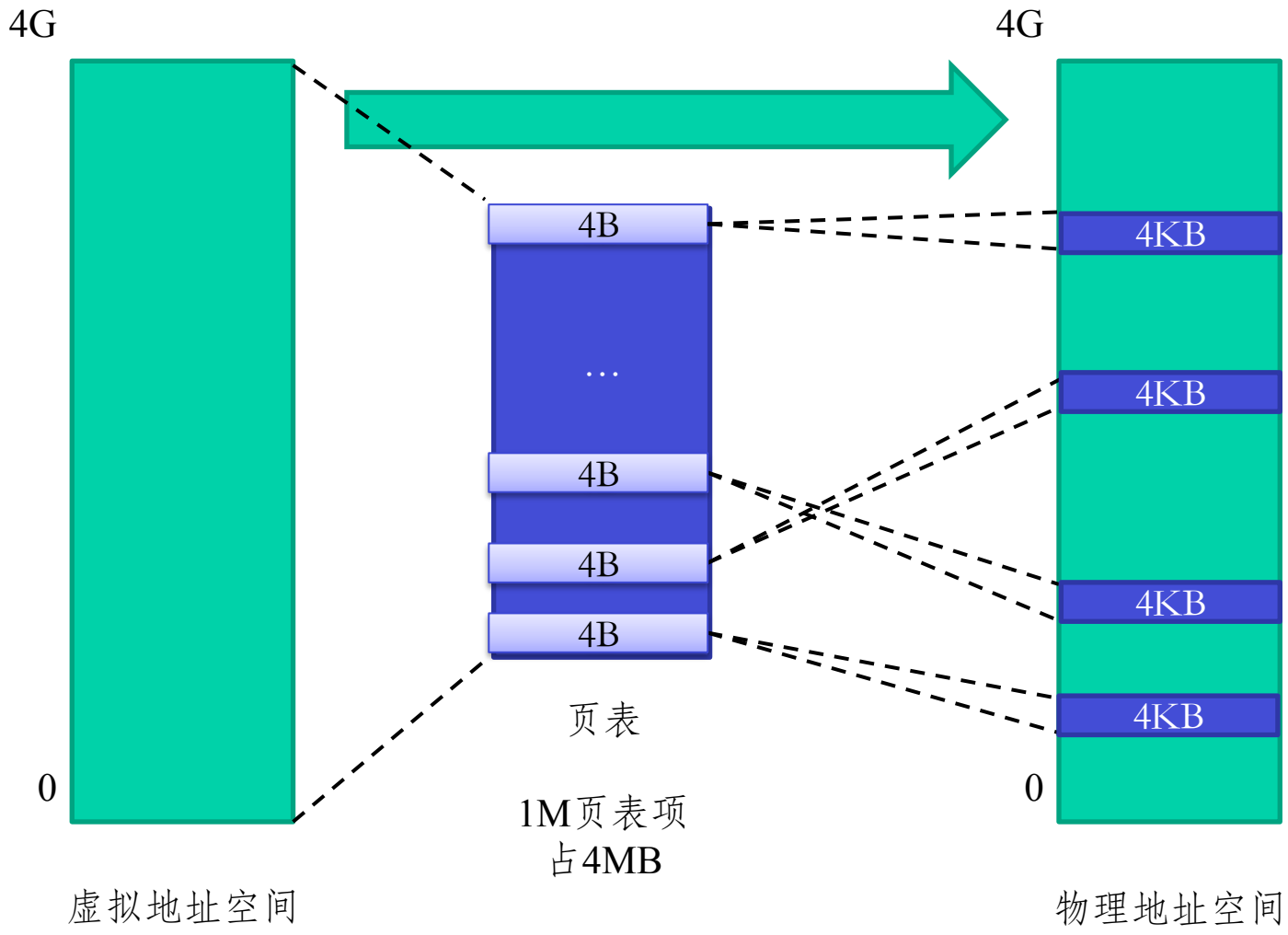
多级页表中，每级页表大小应该正好在一页。一个页表项4Byte，4KB/4B=1K＝1024，所以一级页表10位，剩余二级页表10位，页表合计20位

Virtual Address Translation



- 二级页表
- 4G虚拟地址空间
- 1M页表项(PTEs)
- 1K页目录表项(PDEs)

页目录自映射



页目录自映射

- 页表也要存储在内存中
  - 页目录和页表也被映射到了进程的虚拟地址空间
  - 操作系统对页表访问也通过虚地址
  - 而页表本身维护了完整的虚拟地址到物理地址的映射
  - 所以页表也维护了自身虚拟地址到物理地址的映射（自映射）
  - 每个页表项需要4字节，所以1M个页表项需要4MB字节存储，所以整个页表占用的内存大小就是4MB

页目录自映射

- 4MB页表也要分页存储，共需要4MB/4KB=1024页表项，正好一个页表大小
- 所以1M个页表项中，其中有1个页表（1024个页表项）正好描述了页表本身从虚拟地址到物理地址的映射。大小4K。
- 二级页表结构中，一级页表＝页目录
- 根据一级页表（页目录）的定义：记录这4M页表地址空间到物理地址空间映射关系的，就是页目录。大小4K

页目录自映射

- 4MB页表也要分页存储，共需要4MB/4KB=1024页表项，正好一个页表大小
- 所以1M个页表项中，其中有1个页表（1024个页表项）正好描述了页表本身从虚拟地址到物理地址的映射。大小4K。
- 二级页表结构中，一级页表＝页目录
- 根据一级页表（页目录）的定义：记录这4M页表地址空间到物理地址空间映射关系的，就是页目录。大小4K

所以页目录的内容和某一个页表（1024页表项）内容一致

页目录自映射

- 4MB页表也要分页存储，共需要4MB/4KB=1024页表项
- 所以1M个页表项中，其中有1024个页表项（1个页表）正好描述了页表本身从虚拟地址到物理地址

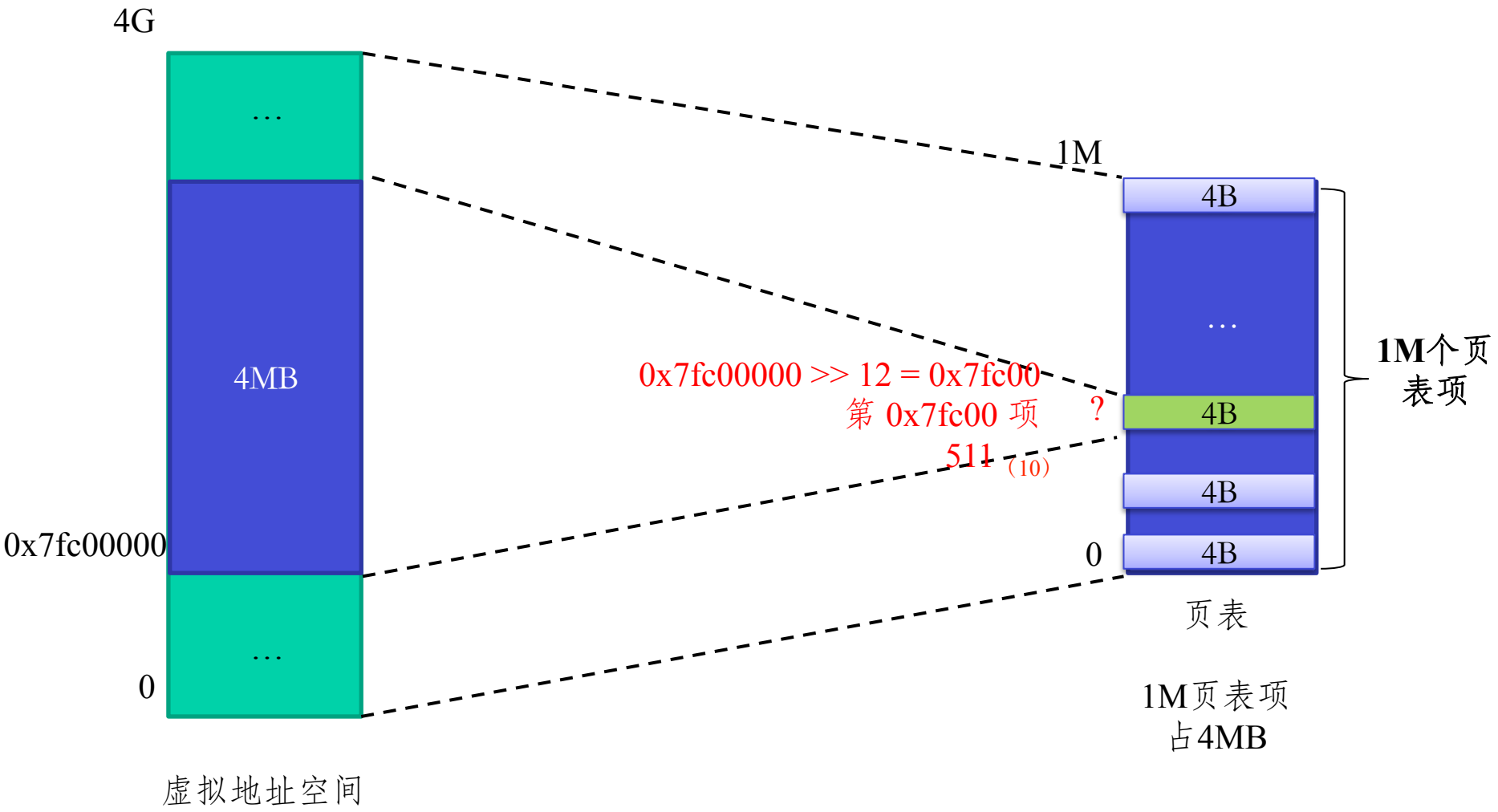
所以页目录的内容和某一个页表（1024页表项）内容一致
→某 一个页表和页目录是重合的!
→ 页目录不需要单独映射

- 址空间到物理地址空间映射关系的，就是页目录。大小4K

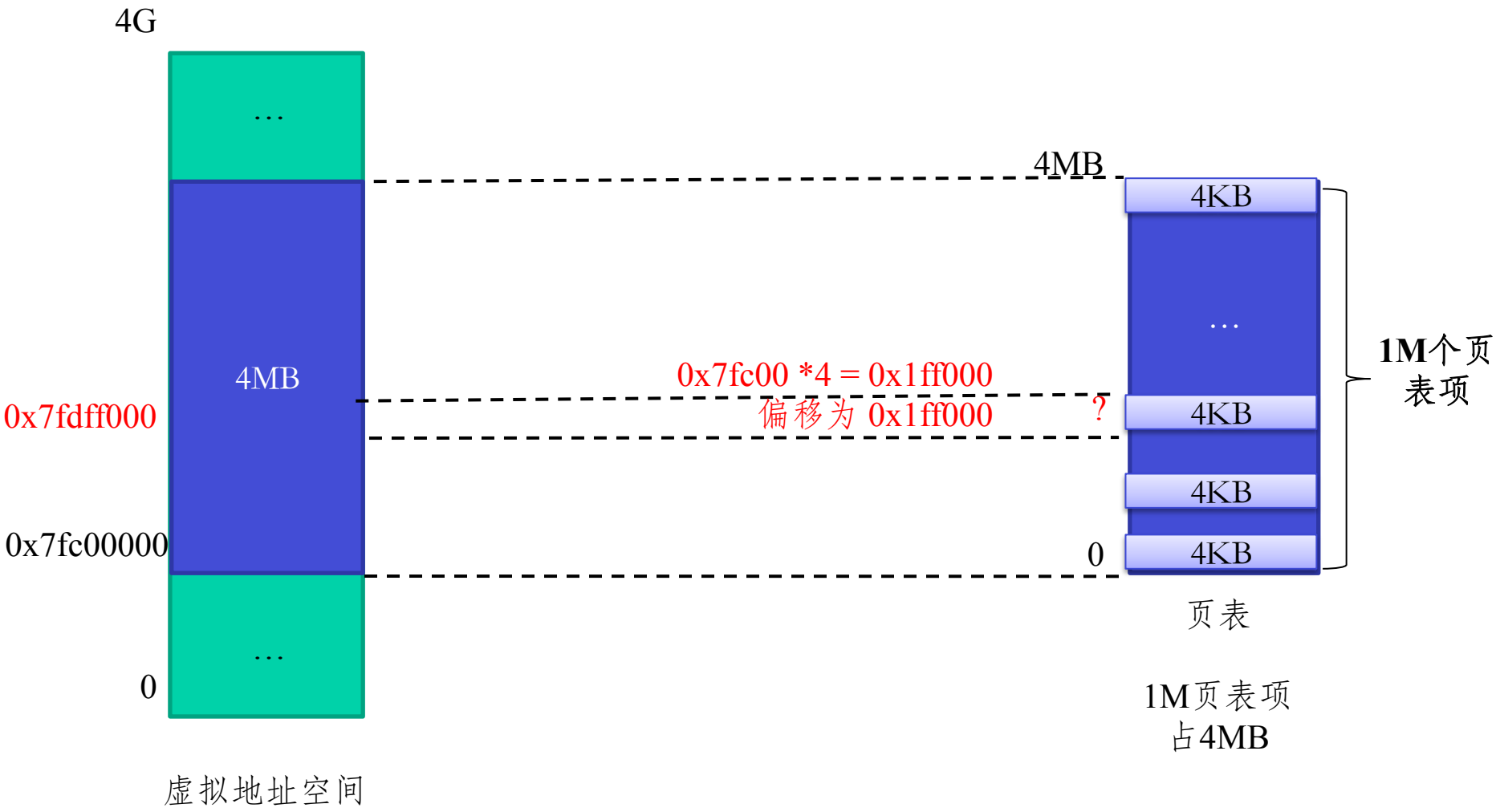
页目录自映射

- 页目录在哪?
  - 给定页表虚拟地址起始位置，例如0x7fc00000
  - 将整个4GB地址空间划分为1MB个4KB页。
  - 由于1M页表项和4G地址空间线性映射，所以页目录首地址对于页表的偏移和页表首地址相对于4G空间的偏移是对应的。
  - 页表首地址地址对应于第（0x7fc00000>>12）个4KB页，因此其逻辑-物理映射关系应该记录在第（0x7fc00000>>12）个页表项中
  - 每个页表项4个字节，所以该页表项对于的页表起始地址地址偏移为（0x7fc00000>>12）*4 = 0x1ff000

页目录自映射

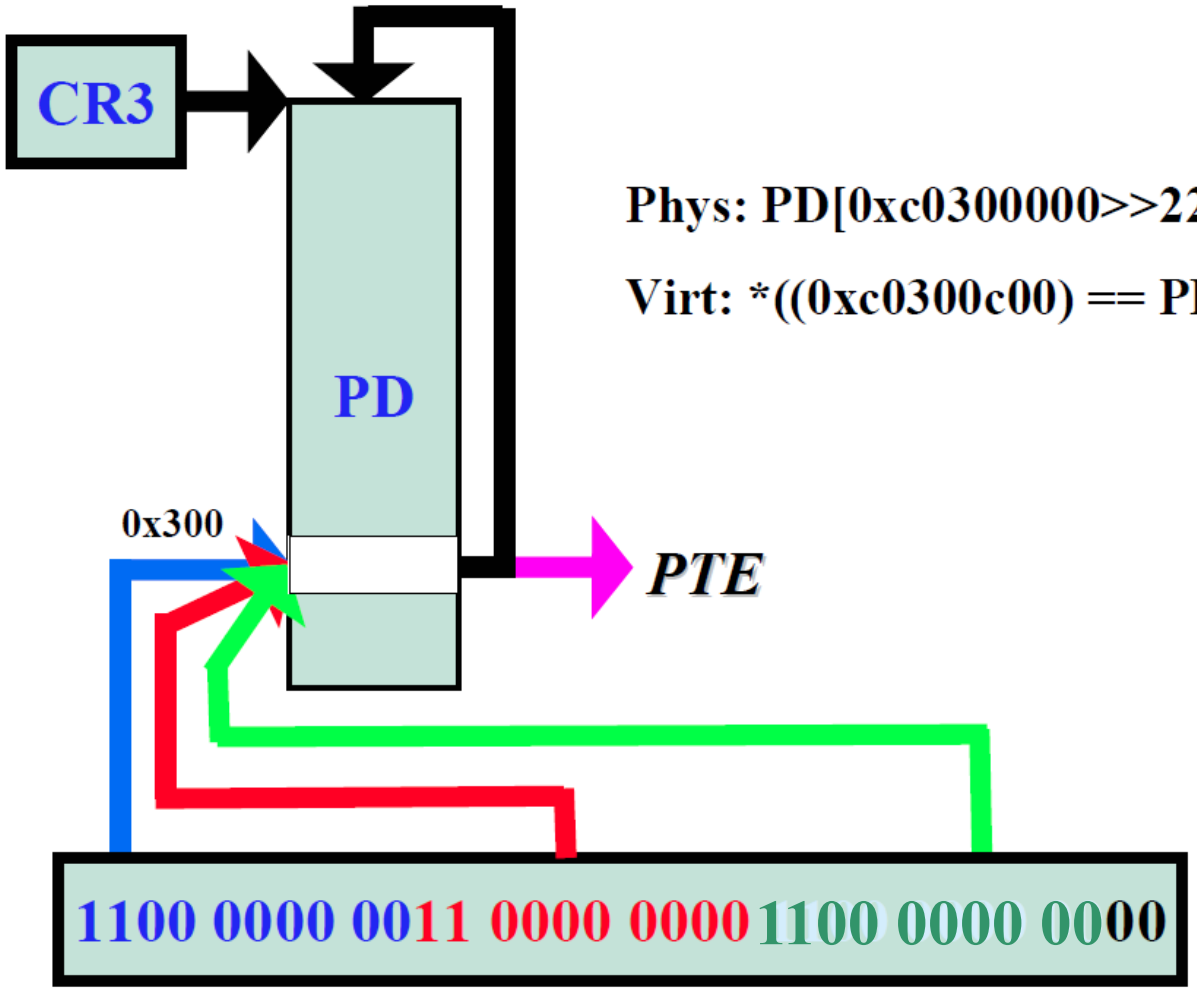

4G
…
4MB
0x7fc00000
…
0
虚拟地址空间
1M
4B
…
4B
4B
4B
0
页表
1M个页表项
0x7fc00000 >> 12 = 0x7fc00
第 0x7fc00 项
?
511 (10)
1M页表项
占4MB

页目录自映射


4G
…
4MB
0x7fdff000
0x7fc00000
0
…
虚拟地址空间
4MB
0x7fc00 *4 = 0x1ff000
偏移为 0x1ff000
?
4KB
…
4KB
4KB
4KB
0
页表
1M页表项
占4MB
1M个页
表项

Windows中的页目录自映射

Virtual Access to PageDirectory[0x300]



Phys: PD[0xc0300000>>22] = PD

Virt: *((0xc0300c00) == PD

页目录自映射

- 简化计算
  - 对于32位地址字长，2级页表，4KB页面大小，虚拟地址va
  - 对应页表PTE起始地址=:
    - ((PMMPTE)(((((ULONG)(va)) >> 12)<< 2) + PDE_BASE))
  - 对应页目录PDE起始地址=:
    - ((PMMPTE)(((((ULONG)(va)) >> 22)<< 2) + PTE_BASE))
- 练习:
  - 页表起始地址0x80000000，页目录起始地址=?
  - 0x80000000 + 0x200000 = 0x80200000
- 反过来：如果给定页目录起始地址，求页表起始地址?
  - E.g. 页目录起始地址0xC0300000，页表起始?