

3.7虚拟存储器

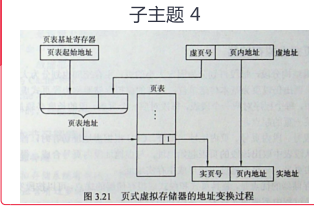
基本概念

- 将主存或者辅存的地址空间统一编址
- 实地址对应的是主存地址空间
- 使用虚地址需要辅助硬件找出虚地址和实地址之间的关系，并对其对应存储单元装入状态进行判断
- 实际情况：需要使用的先送入主存，暂时不用的放在磁盘中

页式虚拟存储器

- 以页为基本单位的虚拟存储器称为页式虚拟存储器
- 虚拟地址 = 虚页号 + 页内地址
- 虚页号 + 页表起始地址 = 页表地址      根据页表地址查找实页号
- 实页号 + 页内地址 = 实地址

计算过程

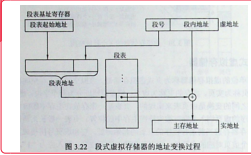


优点：页面长度固定 调入方便  
缺点：零头浪费 对页的处理、保护、共享不是特别方便

段式虚拟存储器

- 按照程序的逻辑结构划分
- 虚地址 = 段号 + 段内地址
- 段号 + 段表起始地址 = 段表地址      查询段表数据
- 实地址 = 段表所得数据 + 段内地址

计算过程



优点：段分界与程序分界相对应 易于编译、管理、修改、保护、共享  
缺点：段长分配不便 存在段间碎片

段页式存储器

- 先将程序按逻辑分段，再将每段分为固定大小页
- 段长必须是页的整数倍
- 虚地址 = 段号 + 段内页号 + 页内地址

优点：可以按段实现共享和保护，同时也有着页的调用方便  
缺点：地址变换要两次查表，开销较大

快表 ( TLB )

- 根据局部性原理，将一些经常访问的页放入高速缓冲器中构成快表，可以极大提高查询的效率
- 采用相联存储器构成，可以按照内容查询
- 访问顺序：TLB--->页表--->Cache--->主存

命中情况

- Cache命中，page必然命中，TLB不一定命中
- Cache不命中，无法退出TLB与page命中情况
- Page不命中，Cache和主存不会命中，此时要执行调页策略

只要抓住存储数据的来源就可以推导出命中情况