

第6次编程作业：寻找互素数对

$$4 \perp 9, \quad 34 \perp 35 \quad (1)$$

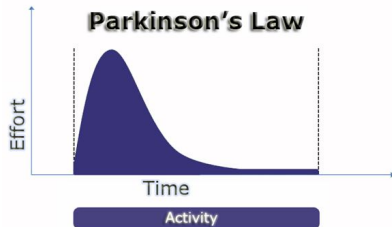
$$\Pr(m \perp n) = \frac{6}{\pi^2} \approx 0.607927102 \quad (2)$$

第6次编程作业：寻找互素数对

- ▶ 硬件平台：Intel Core2 (Duo)
- ▶ 计算100000以内互素的数对的个数
- ▶ 共涉及100亿个数对
- ▶ 计算结果：6079301507
- ▶ 理论预测：6079271020
- ▶ 相对误差：万分之0.05
- ▶ 耗时：18分钟
- ▶ 本次作业：用多线程编程技术加速

Parkinson's law

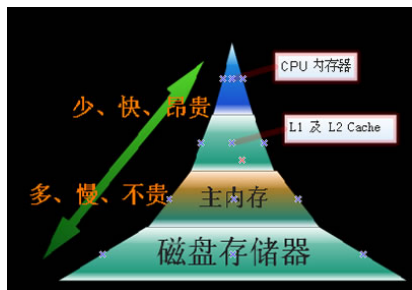
1. Work expands so as to fill the time available for its completion.
2. Data expands to fill the space available for storage.



程序员希望的内存

联系上次作业

私有的、无穷大、无穷快、便宜、持久性



内存管理器(Memory Manager)的任务

- ▶ 提供内存抽象界面
- ▶ 分配物理内存，回收物理内存
- ▶ 记录内存使用情况等

没有内存抽象：程序员直接操作物理内存

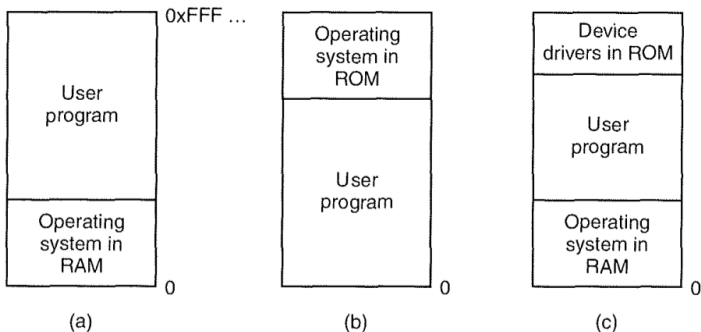


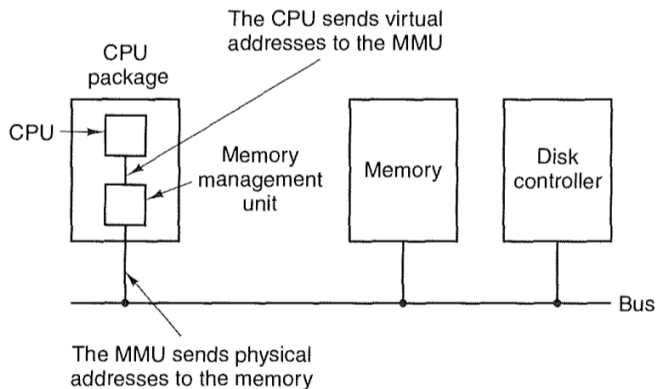
Figure 3-1. Three simple ways of organizing memory with an operating system and one user process. Other possibilities also exist.

- a. 内存中一次只能驻留一个程序: **MOV REGISTER1, 1000**
- b. 操作系统自身代码难以保护(没有地址空间概念)

虚拟内存技术

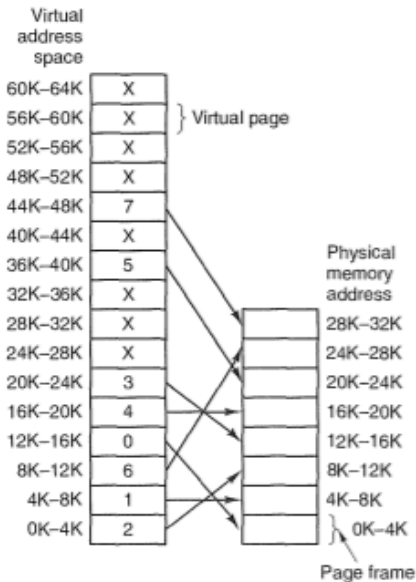
- ▶ 问题一：如何让多个程序驻留内存？
 - ▶ 每个程序有专属地址空间
 - ▶ 程序不能非法访问其他程序的地址空间
- ▶ 问题二：如何满足程序对内存的无限需求？
 - ▶ 地址空间分成若干页面
 - ▶ 地址空间的页面映射到物理内存的页框内
 - ▶ 利用页表实现页面号到页框号的映射
 - ▶ 不是所有的页面都需要放到物理内存中
 - ▶ 缺页中断技术

虚拟地址、物理地址及内存管理单元



虚拟地址、物理地址及内存管理单元

- ▶ 虚拟地址空间: 64K (16 bit)
- ▶ 物理地址空间: 32K (15 bit)
- ▶ 页面大小: 4K
- ▶ 共16个(虚拟)页面, 8个(物理)页框
- ▶ 对于大于32K的程序, 只能有32K驻留物理内存(右图数字部分)



虚拟地址、物理地址及内存管理单元

MOV REG, 20500

$20500 = 20K + 20$

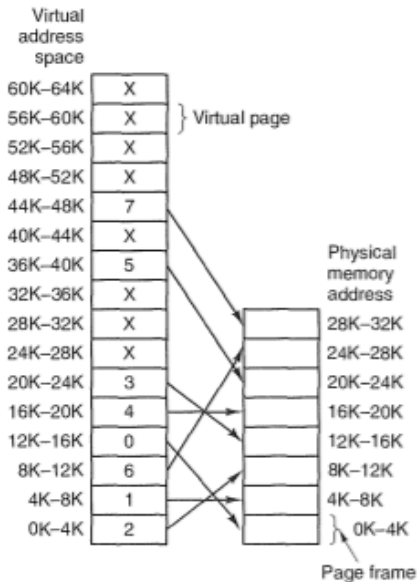
页面5 → 页框3

$12K + 20 = 12308$ (物理地址)

MOV REG, 32780

对应虚拟页面8 (缺页)

what next?



虚拟地址、物理地址及内存管理单元

虚拟地址映射到物理地址，考虑右图例子：

- ▶ 页面大小：4K
- ▶ 页内地址为12位
- ▶ 对于16位机器而言，有4位用于页表索引
- ▶ 因此共有16个虚拟页面
- ▶ 8个物理页框（需3位）

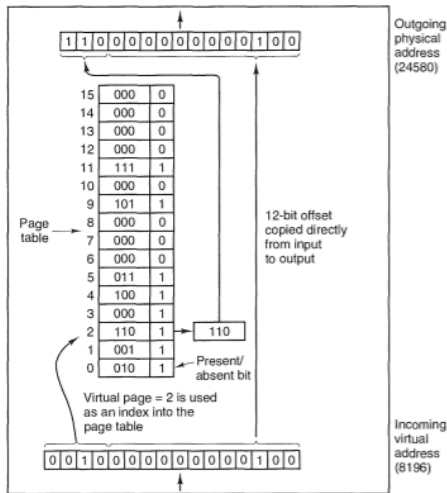
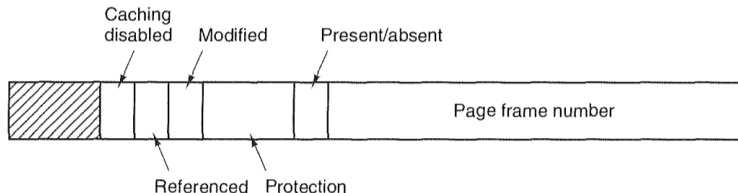


Figure 3-10. The internal operation of the MMU with 16 4-KB pages.

页表项



page frame number 描述该页表项对应的页框编号

present / absent 表示该页表是否在内存中

protection 含读、写、执行等权限信息

modified 该页表内容是否被修改过

referenced 该页表内容是否被用过（读写）

caching disabled 用于memory mapped I/O（后续）

虚拟内存技术与分页技术面临的两大问题

1. 从虚拟地址到物理地址的映射必须快
 - ▶ 每条指令都需从内存取出
 - ▶ 大量指令涉及读写内存(CISC机器)
2. 如果虚拟地址空间很大，则页表规模会特别大
考虑页面大小**4K**的虚拟内存系统：
 - ▶ **32位虚拟地址**: **100万个**页表项
 - ▶ **64位虚拟地址**: ??

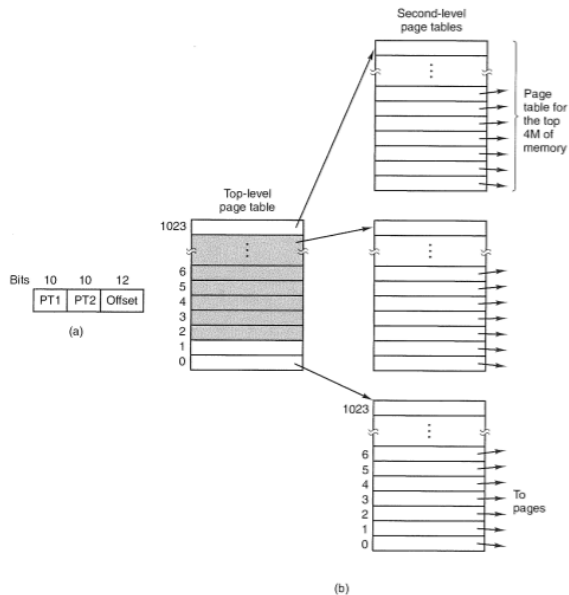
从虚拟地址到物理地址的快速映射: TLB(联想式存储)

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

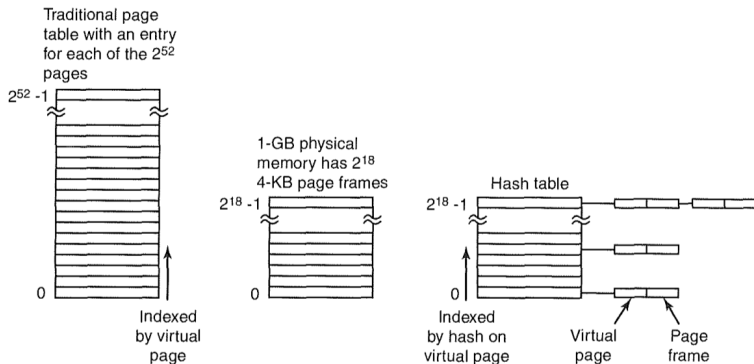
放置在MMU里面。来虚拟地址，先查TLB。

- ▶ 若能在TLB中查到该虚拟地址，则直接输出物理页框号码
- ▶ 否则，去内存中的页表中查找对应页框号码，并将其调入TLB

处理大规模地址空间的方法一：多级页表



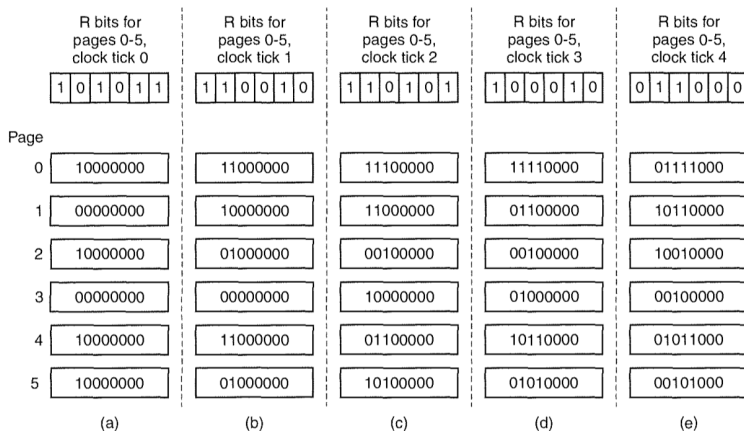
处理大规模地址空间的方法二：倒排页表



页面置换算法

当发生缺页时，OS需要选择一个页面将其从物理内存转移至硬盘，然后从硬盘调入所缺页面。

老化(aging)页面置换算法



工作集(working set)页面置换算法

- ▶ 访问的**局部性**：在任一时间段内，程序仅仅访问其所有页面的一小部分。
- ▶ 我们将程序在某时间段内密集访问的页面集合成为**工作集**

工作集(working set)页面置换算法

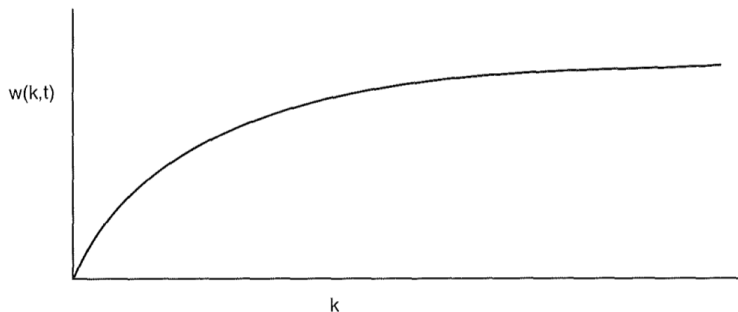
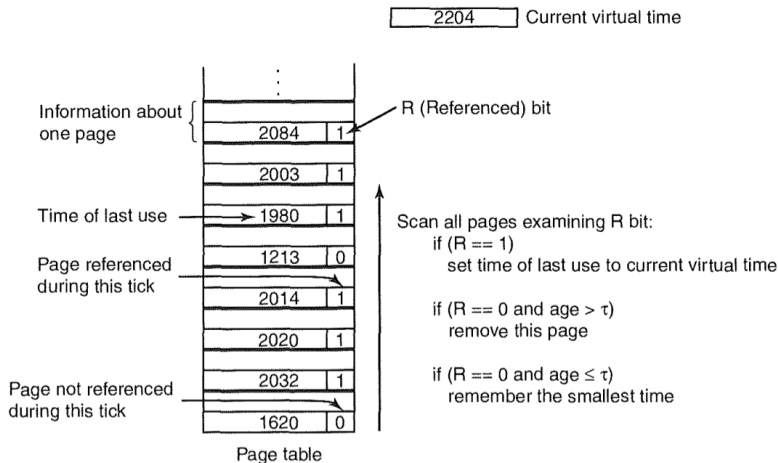


Figure 3-19. The working set is the set of pages used by the k most recent memory references. The function $w(k, t)$ is the size of the working set at time t .

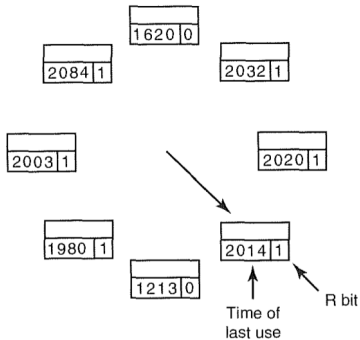
工作集(working set)页面置换算法



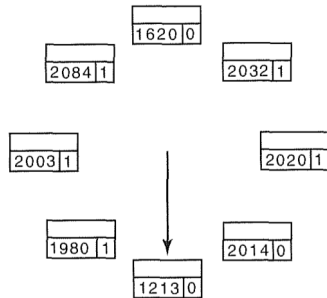
WSClock页面置换算法

2204

 Current virtual time



(a)



(b)

WSClock页面置换算法

