

4-1 原因: 存储层级可以在性能和成本之间取得平衡, 利用不同存储技术在速度, 容量和价格上的差异, 构成一个高效经济的存储系统; 且利用程序的局部性原理, 将最常用的数据放在最快的存储设备中; 可以提供更大的虚拟存储空间。

4-2 过大: 内部碎片增多, 造成空间浪费; 同时, 过大的页也会增加页表的大小
过小: 外部碎片增多, 内存中未被分配的空间浪费, 增加页表项的数量和页换出的频率, 性能降低。

4-3 (1) PPN[1]: 物理页的第1位, 用于支持超页

PPN[0]: 第1位, 用于支持超页

RSW: 保留的软件位, 用于操作系统自定义使用

D: 位为1时, 表明该页是否被改写

A: 是否可访问, 位1为可

G: 全局页面标识, 当前页可供多个进程共享

U: 用户模式可访问

XWR: 可执行, 可写, 可读

V: 表明物理页在内存中是否分配好

(2) 破坏内存的隔离和保护, 可能会修改权限位, 执行非法操作; 可能修改状态位, 影响操作系统内存管理

(3) 指向下一级页表, 不允许任何模式的指令执行, 只能用于建立虚拟地址空间的结构

4-4 (1) 表明是否可执行 / 可写 / 可读

(2) L: 表项的Lock使能位, 1: 锁位 A: 表项的地址匹配模式

00: OFF; 01: TOR; 10: NA4; 11: NAD07

$$+5(1) \quad \frac{260}{2^{12}} \cdot 64 = 2^{58}$$

$$(2) \quad \frac{268}{2^{12}} \cdot 64 = 2^{42}$$

(3) 可实现按需分配, 不连续分配, 实现动态调整, 结合 TLB 提高地址转换效率

页地址寄存器, 主1号页寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号: [0, 1, 2, 3, 4, 5, 6, 7]

物理地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]

页地址寄存器, 主1号寄存器: [0, 1, 2, 3, 4, 5, 6, 7]