

1. 为解决计算机访问数据时的速度瓶颈和成本问题。随着计算机处理速度的提高和数据量的增加,单一的存储介质往往不能满足计算机对存储容量和访问速度的要求。而存储层级能通过多个层次的存储介质来提供不同的存储容量和访问速度,以满足不同场景下的存储需求。通过这种层级结构,计算机可以满足存储容量需求,提高数据访问的速度和效率,还可以降低计算机系统的成本。

2. 过大的页：导致内存浪费和访问冗余。当访问页面的一小部分时，也会将整个页面从磁盘加载到内存中，导致内存中有大量未使用的数据。此外，页面过大 会增加页面置换的开销。

过小的页：导致内存利用率低和内存碎片化。进程需要访问的数据可能被分散到多个页面中，导致需加载更多页面，导致内存利用率低；若这些页面并不连续，会产生大量内存碎片，浪费内存资源。

3. (1) 位 0: 有效位 (1) 表示该页组是否有效, (1 有效 0 无效)

位1: 读取权限位 (R) 表示该页表条目对应的虚拟地址是否允许读取操作。(允许/禁止)

位2. 写权限位 (w) 表示该页表条目对应的虚拟地址是否允许写入操作。(允许0禁止)

位3: 执行权限位(x)表示该条件语句 - - - - - 执行操作。(1分) (4分)

注4: 用户权限位 (U) 表示 —— 用户态访问。(15# 0 为内核态访问)

位与全局位 (G) 表示该变量由谁所代表的物理地址是否是全局共享的。(是 0 否 1)

位6: 已使用位(A) 类似条件目是记被~~用~~ C1已被使用 味被使用

在7: 已修改位 (D) 表示该表项所对应的虚拟地址所对应的物理页是否被修改 (1是 0否)

(2) ①安全问题: 普通用户进程可通过修改位1至位4来将原本只读/可执行的内存页设置成可写/可执行, 从而绕过操作系统的安全机制, 获取系统权限, 系统安全风险增加。

②内存问题: 用户进程可通过修改位0有效位, 将某些内存页置为无效, 导致内存泄漏或程序崩溃!



③性能问题: 用户进程可以通过修改页表中的位5全局位, 将某些内存页设置为全局共享, 导致TLB缓存失效, 降低系统性能。

④稳定性问题: 用户进程可以通过修改页表中的位7已修改位, 将某些内存页设置为已修改, 导致操作系统在页表项被替换时需要写回大量的内存页, 从而影响系统的稳定性、可靠性。

(3) 含义: 该页表项对应的虚拟地址是一个合法的虚拟地址, 但其对应的物理页不允许读取、写入或执行操作。可能发生在, 该虚拟地址对应的物理页是一个空白页/是被保留的。

4. (1) 页表项中的X/W/R用于描述虚拟地址与物理内存页的访问权限和属性, 而PMP控制寄存器中的X/W/R位用于进一步限制对物理内存页的访问。当页表项中已经存在X/W/R位时, PMP控制寄存器中的X/W/R位将会与页表项中的对应位进行逻辑与运算, 以确定是否允许对物理内存页进行读取、写入或执行操作。

(2) PMP配置寄存器中的L和A位用来指定PMP匹配时的地址匹配模式和地址限制模式。  
L位用于指定地址匹配模式。当L位被设置为1时, PMP匹配时将使用基地址进行匹配, 而不使用地址范围。  
A位用于指定地址限制模式。当A位被置0时, PMP匹配时将使用最大地址限制作为地址范围的限制; A位被置1时, PMP匹配时将使用地址范围作为地址范围的限制。

5. (1) 4KB即 $2^{12}$ 字节, 即页内偏移量需占用12位。而虚拟地址64位, 故页号需占用52位。

$$\therefore \text{共需 } 2^{52} \times 8 = 2^{55} \text{ 字节}$$

(2) 虚拟地址共48位, 故页号需占用36位

$$\text{共需 } 2^{36} \times 8 = 2^{39} \text{ 字节}$$

(3) 因为多级页表分散存储页表信息, 一个虚拟地址经多级映射得到对应物理地址, 每级页表只存储部分虚拟地址到物理地址的映射关系, 每级页表比单级页表小很多, 节约实际的存储空间。且避免了为所有虚拟页都分配物理页框, 从而节约内存空间。

