

第4章

1. 因为随着现代处理器性能的不断攀升，处理器和内存之间的速度差距不断扩大，形成了“内存墙”。而寄存器与缓存虽然速度快，但成本高、容量小，折中考虑性价比与速度，因而使用存储器层级。

2. 过大的页会使页内偏移位数过长，过小的页会使页号位数过长、页表所占空间过大。

3. (1) D - Dirty 该页是否被改写

A - Accessed 该页是否可访问

G - Global 全局页面标识，该页是否可供多个进程共享

U - User 用户模式是否可访问

X WR 可执行、可写、可读

V - Valid 物理页在内存中是否分配好

(2) 一旦将用户进程的页表项修改为指向其他进程的内存位置，权限机制将形同虚设。一旦用户进程自由修改页表项中的状态与权限标志位，这些标记将毫无意义，无法控制内存访问。

(3) 若 XWR 全为 0，则其含义为指向页表下一级的指针。

4. (1) 启用分页后，访问虚拟内存的指令可能会导致多次物理内存访问，包括对页表的隐式引用，PMP 检查适用于所有这些访问。
最底层的安全保护机制。

(2) L: 表项的 Lock 使能位

A: 表项的地址匹配模式

5. (1) 页大小为 4KB，则页内索引位数为 12 位

则虚拟页号占 $64 - 12 = 52$ 位，即 2^{52} 个页表项

而每个页表项使用8个字节空间

$$\text{共 } 2^{52} \times 8 = 2^{55} \text{ 个字节}$$

即 32 PB.

(2) 虚拟页号减少为 $48 - 12 = 36$ 位 即 2^{36} 个页表项

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

即 512 GB.

(3) 因为多级页表在进程占用内存空间较小时, 可对应地减少页表数目。也就是说多级页表的页表存储是灵活可变的, 而单级页表的页表存储不可变, 只能取最大值。