

1. Cache 和 RAM 成本和速度差很大，存储层次可以提高计算机的速度和响应能力。
原理是处理器直接从高速缓存中读取数据，不必访问 RAM，大大缩短数据访问时间。其次，存储层次还可以大大减少数据传输量，提高计算机效率。
2. 地址过大，会引起内部碎片，交换磁盘工作量过大，引起并发访问冲突时间长，地址过小，会引起外部碎片，操作系统开销增大，地址转换开销增大，降低效率。

3. (1) 表示页表项对应物理页号已被修改 | X: execute 是否允许执行
 A: Accessed 是否被访问过
 G: Global 是否全局有效
 U: user 是否允许用户访问
- (2) 如果用户进程能自由修改自己的页表，就会导致物理内存被恶意读取或写入，有极大隐患。

(3) 表示该页表条目不能执行，写入权限，不能被任何进程使用，称为保留页，
因为为了防止用户进程在访问内存时不小心写入某些关键数据或代码，
导致崩溃，它们从不被使用。

4. (1) PMP内存条的 XWR 位会覆盖页表条目中的 XWR 位，从而提供对物理内存的额外保护。

- (2) L: 指定该条目是否锁定，L 位为 1，PMP 位为 0，无法修改，提供额外的安全保护。
 A: Address match 指令 PMP 条目地址范围，A 位为 0，监视整个物理地址空间
 A 位为 1，监视 PMPV 选择的范围，PMPV=1，监视全局地址范围或局部物理地址范围。
 A 位为 0，监视 PMPV=0，监视当前物理地址范围。

5. 16 位虚拟地址大小 = 2^{16} bit = 2^{16} 位 = 2^{51} KB = 2^{49} × 4 KB。
 对应需要 $2^{49} \times 8B = 2^{52} B$ 大小。

2. 48 位 2^{48} bit = 2^{45} B = 2^{35} KB = 2^{33} × 4 KB
 降低到要 $2^{33} \times 8B = 2^{36} B = 2^6 GB$

- 3) 1. 清除了无用的页表项
 2. 分页访问页表，将一个大的页表分为多个较小的页表，每个页表存储部分地址空间的映射关系，将大的页表分为多个小的页表，从而降低每个进程的页表大小。
 同时，分页访问页表，可以减少每次地址转换时需要访问的页表项，提高速度。

3. 软件内存空间