

- ① 优化性能。随着计算机技术不断发展, CPU处理速度不断提高, 而内存和硬盘等存储设备的速度较慢。存储层级可以加快数据访问速度。
- ② 减小成本。硬盘、内存、高速缓存的成本依次增加, 存储层级设计可合理分配存储设备以达到最优成本效益。
- ③ 提高数据一致性和可靠性。存储层级可以通过一些协议来维护数据的一致性, 从而保证多处理系统正常运行, 并通过数据的冗余和备份提高可靠性。

2. 如果页^过太大, 会导致:

- ① 内部碎片: 浪费一些页面上未被使用的空间。
- ② 外部碎片: 一些未被使用的空间散布在物理内存中, 但无法被利用。
- ③ 页面置换效率低下: 每个页面都需要读取或写入更多的数据, 导致性能下降。

如果页过小, 会导致:

- ① 进程地址空间浪费: 进程地址空间中有过多未使用的空间。
- ② 段表或页表过大: 需要更多的条目来映射整个地址空间, 从而浪费更多的内存。
- ③ 页面错误率高: 一个进程可能需要访问很多页面才能完成一个操作, 缺页中断增加。

3. V位决定了该页表的其余部分是否有效

R, W和X位分别表示此页是否可以读取、写入和执行

U位表示该页是否是用户界面, G位表示这个映射是否对所有虚拟地址空间有效

A位表示自上次A位被清除以来, 该页面是否被访问过

D位表示自上次清除D位以来页面是否被修改

2) 如果用户进程能够自行修改自己的页表, 那么该用户就可以访问任何地址, 从而产生安全问题。

3) 如果这三个位是0, 那么这个页表项是指向下一级页表的指针, 否则它是页表树的一个叶节点。

4. 1) 当处于U模式的处理器尝试取指或执行 load 或 store 操作时, 将地址和所有的PMP地址寄存器比较。如果地址大于PMP地址*i*但小于*i+1*, 则*i+1*的配置寄存器决定该访问是否可以继续。如果不能将会引发访问异常。

2) 2锁锁定了PMP和对应的地址寄存器, A设置是否启用此PMP。

5. 1) 需要 $\frac{2^{64} \times 8}{4 \times 2^{10}} = 2^{53}$ 字节用于存储页表, 即 512TB

2) 需要 $\frac{2^{48} \times 8}{4 \times 2^{10}} = 2^{37}$ 字节用于存储页表, 即 128GB

3) 多级页表通过将整个虚拟地址空间分成多个较小的区域, 从而降低了实际页表存储开销。另外, 多级页表还允许操作系统对于物理内存的分配和回收的分配和回收进行更细粒度的控制, 使得内存管理更加灵活和高效。