

1. 存储系统的层次化主要为了提高性价比。由于不同存储技术在技术指标和成本上的差异,存储采用多层的体系,其中最底层为持久化存储层,速度最慢,容量最大,成本最低;越往上速度越快,容量越低,成本越高。这样使得整体成本接近底层,性能接近最上层,容量和持久性等于持久化存储层。

2. 过大的页:当进程只需要使用很小内存时,使用大页会造成大量内存空间的浪费,使页内碎片增加。

过小的页:虽然减小了内存碎片,但造成每个进程占用较多的页面,从而会导致页表过长,内存利用率降低。

3. 1) D: Dirty, 表明该页是否被写;为0时为未被写;1为已被写,若此时仍对该页进行写入则会触发Page Fault (Store)异常

A: accessed, 表示此页是否被访问过;

G: global, 全局页面标识;为1时当前页面可供多进程共享;为0时为非共享,进程ASID私有。

U: user, 表示用户模式能否访问: 为1时用户模式可访问: 为0时不可访问。

X: executable, 表示此页是否可执行;

W: 表示此页是否可写; R: 表示此页是否可读;

V: valid, 表示物理页在内存中是否被分配好, 为1时表明已分配好。

(2) 如果用户能自由修改页表可能使一些如用户模式不可访问或不能写的页表产生 Page Fault 异常。

(3) X, W, R 全为0表示指针指向下一级页表。

4.1) 判断当前工作模式下 CPU 对特定物理地址的可执行, 可写, 可读权限

X: 表明 PMP 表项匹配地址是否可执行; W 为是否可写; R 表示是否可读。

2) L: Lock 使能位; 为0则机器模式访问都成功, 系统/用户模式则由 X/W/R 决定; 为1则表项被锁住, 无法对相关表项进行修改, 当配置 TOR 模式, 其前一个表项的地址寄存器也无法修改, 所有模式根据 R/W/X 判定是否可以访问。

A: 表示表项的地址匹配模式: 00: OFF, 无效表项;

01: TOR, 使用相邻表项地址作为匹配区间;

10: NA4, 区间大小为4字节的匹配

11: NAPOT, 匹配大小为2的幂次字节的匹配, 大小至少为4KB

5. 1) 页内索引占12位, 剩余52位, 所以需要 $2^{52} \times 8 = 2^{55}$ 字节空间

2) 页内索引占12位, 剩余48-12=36位, 需要 $2^{36} \times 8 = 2^{37}$ 字节空间

3) 因为多级页表是一对多的关系, 每一级索引出的内容作为索引下一级的基地址, 且省却了大量未映射的页表项, 只保留如第一级页表,

1个第二级或第三级在内存中, 从而大幅减少空间。