

1. 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, 形成了内存墙的问题, 因此现代计算机需要多级分层存储结构, 利用数据的时间局部性和空间局部性来改善访存性能。

2. 过小的页面会造成每个进程占用较多页面, 页表过长, 占用内存过大。
过大的页面会使页内碎片增多。

3. / 1) V位: 表明该页表项是否合法

RWX位: 权限位, 表明是否可读、写、执行。三者均为0时是一个指向下级页表的指针。

U位: 是否可为U态使用

G位: 全局映射

A位: 获取位, 虚拟地址被读写或匹配时置位

D位: 当虚拟地址被写时置位。

2) 如果用户可以修改自己的页表, 就可以访问任何内存, 甚至操作硬件, 那么就无法隔离保护内核和进程。

3) X/W/R位全部为0, 表明该页表条目是一个指向下级页表的指针。

4. / 1) 为了限制运行在 hart 上的低权限上下文访问的物理地址, 可用 PMP 单元控制寄存器允许为每个物理内存区域指定物理内存访问权限。(X/W/R)

2) A位可以决定 PMP entry 控制的物理地址范围, L位表示 PMP entry 处于锁定状态, 忽略对配置寄存器和对应地址寄存器的写入。



No.

Date

10. 共 $\frac{2^{64}}{4 \times 2^{12}} = 2^{52}$ 页

5/1) 一页大小为4KB, 故页位偏移12位, 页号位数即 $64 - 12 = 52$ 位.

$\frac{4KB}{8B} = 2^9$, 即一个页面能装下 2^9 个页表条目.

故须 $\frac{2^{52}}{2^9} = 2^{43}$ 页存储页表, 即 $2^{43} \cdot 4KB = 2^{25} GB$.

2) 页号位数为 $48 - 12 = 36$ 位.

故空间须 $2^{36} \cdot 8B = 2^9 GB$

3) 单级页表在进程创建时为可能用到的所有页表项分配空间, 而多级页表可以在使用时根据内存的占用为进程分配页表空间, 可以实现按需分配, 从而减少内存占用.

