

1. 为使存储空间足够大且又能满足CPU的存取速度要求而且价格适中。
2. 页面过小，会造成每个进程占用较多的页面，导致进程的页表过长，占用大量内存。此外还会降低页面换进换出的效率。
页面过大，会使页内碎片增大。
3. 1) V bit：说明此页表条目是否有效；
permission bits (R, W, X)：说明此页的读、写、执行权限；
U bit：说明此页是否能在用户模式下被占用；
G bit：说明该页表项后续级别中的映射是否为全局的；
D bit：说明此页是否被写脏；
A bit：说明此页是否被访问过。
2) 可能使进程中的虚拟地址空间被映射到错误的物理地址空间；
也可能会修改内存中相应位置的读、写、执行权限，最终导致进程发生错误。
3) 当 X/W/R 全为 0 时，表示该页表条目是下一级页表条目的索引。
4. 1) 页表条目中的访问类型只有在不违背 PMP 寄存器对该页的访问权限规定时才有效，否则会出现访问错误报告。
2) A 字段对该 PMP 配置寄存器相关联的 PMP 地址寄存器的地址匹配模式进行编码；
L 位用于说明该 PMP 条目是否被锁定。被锁定时，对于该 pmpcfg 与 pmpaddr 的写入将被忽略，且 R/W/X 权限将 ~~被相~~ 在机器模式中生效。



扫描全能王 创建

5. 1) 4KB 对应 12 位地址，因此参与映射的虚拟地址位数为 $(64 - 12) = 52$ 位。

因此，对于单级页表系统，需要的页表条目数

为 2^{52} 。又由于一个页表条目占用 8 字节空间，

故存储该单级页表系统所需的空间大小为 $(2^{52} \times 8)B = 4PB$

2) 同 1) 理，存储该单级页表系统所需的空间大小

降低到 $(2^{(48-21)} \times 8)B = 512GB$

3) 因为多级页表可降低每个页表条目占据的空间 X

因为多级页表的后级页表可以只在需要时被创建，并始终由首级页表达到覆盖全部虚拟空间的要求；而单级表的每一条目在任何时候都必须存在，才能达到覆盖全部虚拟空间的要求。



扫描全能王 创建