第4章

1. 现代处理器性能不断攀升，导致处理器和内存之间存在巨大的速度差距，出现"内存墙"。如果处理器每次都要访问内存来获取数据，则需要消耗大量周期等待。为此，存储需要分层的结构，~~使处理器首先去访问缓存，这样很快~~？越接近处理器核心的存储器可以与处理器进行更快的数据交互，但其成本也会更高，容量也更小，因而出现了多级缓存

2. 如果页过小，在需要管理的总的内存空间一定的情况下，就要将内存分成更多的页去管理，页号所占位数更多，则页表的长度就会大大增加，但每页表中每一项的大小基本不变，最终导致页表大小过大，占据内存太多；如果页过大，就会导致页内碎片问题，每个页中实际只被使用了一部分，内存使用效率下降

3. 见 riscv-privileged P79 4.3.1

此时为一个32位虚拟地址，4KB页表，页内偏移为12位，多级页表的表项数为 $\frac{4KB}{2^{12}} = 2^{10}$ 所以用到10位的两级页表组成，共20位 VPN (virtual page number)，将被转为22位的 PPN (物理页号)

页表中每一项叫做1个 PTE，是32位的(4B)，除了22位 PPN 外还有其他若干标志位

(1) 第0位：V 该页表项(PTE)是否有效 ~~第1位~~，若为0，该PTE其他项都没用

第1.2.3位 (R.W.X) 说明该页是否可读.写.执行　第4位：U 表明该页在用户模式下是否可访问，U=1时用户可访问　第5位：G 全局映射（在所有地址空间存在）　8.9位：RSW 保留的状态位

第6位：A accessed 表明距离上一次A被清除后该虚拟页是否被读/写/访问过

第7位：D dirty 表明在上一次D被清除后该虚拟页是否被写过

(2) 若用户进程可自行修改页表，则这些状态权限位可以任意更改，无法达到原来的权限保护和页表管理效果，系统的安全性受到严重挑战

(3) 说明该PTE是指向更下一级页表的一个有效指针

4. PMP（物理内存保护）：为特定的物理内存区域指定访问权限　P56

(1) pmp控制寄存器中的X/W/R决定了这块物理内存能不能被正常访问，页表中的X/W/R位只是说明页表这个虚拟地址到物理地址的转换能否正常访问完成，在页表成功转换到物理内存的地址后，还会要去访问物理内存，这里就需要过pmp的check，所以当页表的每一次访存都要过pmp的check，两者结合达到更精细的内存访问控制

(2) L位是lock. Pmp的整体结构是：最基础为pmp entry，至少由8位configuration reg构成，这8位存在物理地址的对应信息和权限信息，然后pmp entry可有64个，64个pmp entries拼一起可构成1个pmp CSR，一共有多个pmp CSR，而pmp CSR区域称为WARL. L位说明pmp entry被锁住(locking)也即不能再对configuration register再作出更改. A位说明地址匹配模式，A为2位，A=0代表无地址匹配，A=1为TOR，A=2为NA4(4 bytes)，A=3为NAPOT(≥8 bytes)

5. (1) 页大小4kB → 12位页内偏移 PTE为8B，则64位地址中虚拟页号占64-12=52位

则对单级页表为 $2^{52}\times 8B = 2^{55}B = 2^{15}TB$ 巨大

(2) 使用48位虚拟地址：页号：48-12=36位　$2^{36}\times 8B = 2^{39}B = 512GB$

(3) 多级页表可按照进程的实际大小去分配已有空间相应的页表，多级页表中后面的每级的叶子页并不一定都会对应分配到内存，支持稀疏的地址空间，最终使页表分配占内存与进程的实际要看前一级对应的页表项是否有效　内存使用量成正比