

## 第十二周 chapter 4

### 1. 简述现代计算机系统需要存储层次的原因。

为了提高存储器访问速度和效率。计算机的存储器层级结构由多层组成，每个层级都有不同的容量、访问速度和成本，通常，靠近处理器的层级速度较快、容量较小、成本较高，而离处理器较远的层级速度较慢、容量较大、成本较低。处理器会优先从最快的层级开始查找所需数据，如果未找到，就向下一层级查找，直到找到为止。这样，存储器的访问速度就能够得到大幅提升。

### 2. 在页式虚拟存储中，过大或过小的页分别会引起什么问题？

页式虚拟存储中过大的页会导致内部碎片过多，浪费内存空间，而过小的页则会导致外部碎片过多，使得内存无法充分利用。具体来说，过大的页会导致每个进程需要的内存空间变大，导致内存的使用效率降低。过小的页会增加页表的大小，增加页表的访问开销，同时也会增加缺页中断的次数，影响系统性能。

### 3. 页表条目除了保存物理页号外，一般还包含各种状态和权限信息。它因为内存访问提供了多种细粒度的控制。例如：RISC-V 指令集的 SV32 页表条目具有如下形式：

31	20 19	10 9	8 7	6 5	4 3	2 1	0
PPN[1]	PPN[0]	RSW	D	A	G	UXWRV	
12	10	2	1	1	1	1	1

查阅 RISC-V 规范，回答以下问题：

(1) 查阅 RISC-V 规范，简要描述上述条目中的位 7 到 0 具有什么功能。

位 0CV) 表示页表条目是否有效, 为 1 表示有效, 为 0 表示无效。

位 1(R) 表示读取权限, 为 1 表示允许读取, 为 0 表示禁止读取。

位 2(W) 表示写入权限, 为 1 表示允许写入, 为 0 表示禁止写入。

位 3(X) 表示执行权限, 为 1 表示允许执行, 为 0 表示禁止执行。

位 4 到 6 (保留) 保留供将来使用。

位 7(D) 表示页表条目是否被修改过, 为 1 表示被修改过, 为 0 表示未被修改。

(2) 如果用户进程能够自由修改自己的页表, 可能会出现什么问题?

1. 安全问题: 用户进程可能会修改页表权限, 从而读取、写入或执行本不应该访问的内存区域, 导致系统崩溃、数据泄露或安全漏洞。

2. 稳定性问题: 用户进程可能会错误地修改页表, 导致内存访问异常、程序崩溃或系统不稳定。

为了避免这些问题, 操作系统通常会限制用户进程对页表的访问权限, 只有内核态代码才能够修改页表。

(3) 在 RISC-V 的虚拟内存管理中, 一个 V/R 位全部为 0 的有效页表条目有什么含义?

一个 V/R 位全部为 0 的有效页表条目表示该页表条目所指向的物理页没有被读取、写入 or 执行过。这意味着该页表条目对应的物理页是空闲的, 可以被重新分配给其它虚拟页使用。在操作系统中, 通常会将空闲物理页保存在一个空闲页连串中, 以便于高效地管理内存。

4. RISC-V 的物理内存保护 (PMP) 机制允许硬件线程为特定的物理内存区域指定访问权限, 其配置寄存器如下形式:

7	6	5 4	3	2	1	0
L(WARL)	O(WARL)	A(WARL)	X(WARL)	W(WARL)	R(WARL)	
1	2		1	1	1	1

查阅RISC-V规范，回答以下问题：

(1) 在页表中已经存在 X/W/L 位的情况下，PMP 控制寄存器中的 X/W/R 位有什么作用？

PMP 中的 X/W/R 位可以用来进一步限制对物理内存区域的访问权限。页表中的 X/W/R 位是针对虚拟地址的，而 PMP 控制寄存器中的 X/W/R 位是针对物理地址的。当 PMP 的权限设置和页表中的权限设置冲突时，以权限更为严格为准。

(2) 说明 PMP 配置寄存器中的 L 和 A 位有什么作用？

L 位：表示该 PMP 区域是否被锁定 (Locked)，为 1 表示锁定，为 0 表示未锁定。当一个 PMP 区域被锁定后，就不再被修改或删除，直到下一次重置 PMP 寄存器或者硬件重启。

A 位：Address Match 表示 PMP 区域的地地址匹配模式。当 A 位为 0 时，PMP 区域的地地址范围由 PMPADDR 寄存器锁定；当 A 位为 1 时，PMP 区域的地地址范围由 PMPADDR 寄存器和 PMADDR2 寄存器指定，即采用 2 个寄存器进行地址匹配，以支持更为灵活的地地址映射。

5. 回答以下问题：

(1) 如果页大小为 4KB，每个页表项使用 8 字节空间，内存系统按字节寻址。那么使用完整的 64 位虚拟地址时，一个单级页表系统需要多大空间用于存储页表？

$$2^{64} \text{ bytes} / 4\text{KB} \times 8\text{bytes} = 2^{44} \text{ bytes} / 4 \times 2^{10} \text{ bytes} \times 8\text{bytes}$$

非常已  
得生活：单级页表系统需要的空间

$$= 2^{64-12+3} \text{ bytes} = 2^{55} \text{ bytes}$$

(2) 实际上，多数真实系统仅限制使用 64 位系统的一部分作为有效的内存空间。例如从 4GB 仅仅使用 48 位的虚拟地址空间，则保持其它假设不变时，一个单

$$1 \text{ bytes} = 8 \text{ bit} \quad 1 \text{ KB} = 1024 \text{ Bytes}$$

两级页表浪费存储空间的量为多少？

$$2^{48} \text{ bytes} / 4 \text{ KB} \times 8 \text{ Bytes} = 2^{48-12+3} \text{ bytes} = 2^{39} \text{ bytes}$$

虚拟地址被限制为48位后可以大幅降低单级页表系统所需要的空闲，从而使得页表可以适应实际内存容量。

(3) 多级页表为什么可以降低虚拟内存系统的实际页表开销？

因为多级页表将大的虚拟地址空间划分为多个小的虚拟地址空间，每个小的虚拟地址空间需要的页表也更小。例如：在一个二级页表系统中，第一级页表用于将虚拟地址空间划分为若干个虚拟地址空间，每个小的虚拟地址空间对应一个第二级页表，第二级页表中只需要存储对应的虚拟地址空间中的页表项。因此，多级页表可以减少每个页表的大小，从而降低实际的页表存储开销。