

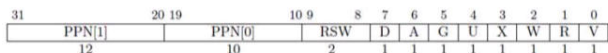
## 1. 简述现代计算机系统需要存储层级的原因。

- ① 访问速度: 不同层级的存储介质访问速度不同. 将数据存储在层级不同的存储介质中, 可以根据数据的访问频率和速度的要求, 选择合适的存储层级, 提高速度.
- ② 存储容量: 高速缓存容量最小, 而外层的存储容量大. 可以选择合适的存储介质, 提高存储容量.
- ③ 成本: 不同层级的存储成本不同. 高速缓存成本最高. 可以根据数据的重要性和成本要求, 降低成本.

## 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?

- ① 过大: 若页过大, 则每个进程所需的页数减少, 导致内存浪费. 如果进程所需的内存不足一次, 则会出现内部碎片, 被浪费掉.
- ② 过小: 若页过小, 则每个进程所需的页数增加, 导致内存中出现很多小的空闲块, 这些空闲块无法被利用, 导致外部碎片的产生. 同时每个进程所需页表变大, 增加了页表的维护成本.

## 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标记位. 它们为内存访问提供了各种细粒度的控制. 例如, RISC-V 指令集的 Sv32 页表条目具有如下的形式:

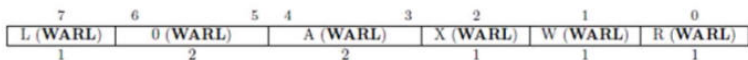


- 1) 查阅 RISC-V 规范, 简要描述上述条目中的位 7 至位 0 具有什么功能.
- 2) 结合上述功能讨论: 如果用户进程能够自由修改自己的页表, 会发生什么问题?
- 3) 在 RISC-V 的虚拟内存管理中, 一个 X/W/R 位全部为 0 的有效页表条目有什么含义?

- 1) V: 有效位, 表示该页表条目是否有效. R: 读取权限位. W: 写权限位. X: 执行权限位. U: 用户权限位, 如果为 0, 则只能被内核访问. G: 全局位, 若为 1, 则对该页表条目是全局页表条目. A: 访问位, 若为 1, 表示该页表条目被访问过. D: 表示该页表条目是否被修改过.

- 2) 用户进程若修改页表来获取系统中其他进程的内存数据, 将导致系统的安全性受到威胁. 同时若将自己页表指向内核区域, 将破坏系统的稳定性.
- 3) 不可执行, 不可写, 不可读. 通常出现在操作系统内核区域的页表中.

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, 其配置寄存器有如下的形式:



查阅 RISC-V 规范, 回答以下问题:

- 1) 在页表条目中已经存在 X/W/R 位的情况下, PMP 控制寄存器中的 X/W/R 位有什么作用?
- 2) 说明 PMP 配置寄存器中的 L 和 A 位有什么作用。

- 1) 用来进一步限制对该页表条目的访问权限, 从而提高系统因安全稳定性。同时还可以用来实现内存保护和隔离, 防止不同进程之间的内存相互干扰。
- 2) L 位决定了是否启用对应的物理内存保护。  
A 位可以通过不同的取值来实现不同的内存保护策略。

5. 回答以下问题:

- 1) 如果页大小为 4KB, 每个页表条目使用 8 字节空间, 内存系统按字节寻址。则使用完整的 64 位虚拟地址时, 一个单级页表系统需要多大的空间用于存储页表?
- 2) 实际上, 多数真实系统仅限制使用 64 位系统的一部分位作为有效的访存空间, 例如 Sv48 即仅使用 48 位的虚拟地址空间, 则保持其他假设不变时, 一个单级页表系统存储页表所需要的空间被降低到多少?
- 3) 多级页表为什么可以降低虚拟内存系统的实际页表存储开销?

- 1) 页大小为 4KB, 则页内索引位数为 12 位, 剩余  $64 - 12 = 52$  位。  
至少需要  $2^{52}$  个表项, 每个表项使用 8 字节空间。  
 $2^{52} \times 8 \text{ byte} = 2^{55} \text{ byte} = 2^{45} \text{ KB} = 2^{15} \text{ MB} = 2^{15} \text{ GB}$

- 2) 只使用 48 位的虚拟地址空间。  
则至少需要  $2^{36}$  个表项。  
则  $2^{36} \times 8 \text{ byte} = 2^{39} \text{ byte} = 2^9 \text{ GB}$

- 3) 因为它采用了分层递进的方式进行地址映射, 每级页表仅需要存储当前级别的页表所占用的内存块指针和标志位, 而不需要存储整个虚拟地址所对应的页表项。  
同时还可以减小页表大小对页表开销的影响。