

1. 简述现代计算机系统需要存储层级的原因。

- 1) 处理器速度与存储速度差异：处理器的运算速度远远快于主存储器的访问速度。处理器需要在较短的时间内获取指令和数据，而主存储器的访问速度相对较慢。为了缓解处理器和主存储器之间的速度差异，引入多级存储层级可以提供更快速的存储访问。
- 2) 提高存储系统性能：存储层级可以通过提供更接近处理器的快速存储来加速数据的访问。较小但更快速的高级缓存（Cache）存储器可以暂存处理器频繁访问的数据和指令，降低对主存储器的访问次数，从而提高整个存储系统的性能。
- 3) 节约成本：存储器的成本通常与速度成反比。通过引入多级存储层级，可以使用较小但更快速的存储器来代替更大但较慢的存储器，以达到更好的性价比。高速缓存的成本相对较高，但其规模相对较小，而主存储器的成本相对较低，但规模较大。通过合理设计存储层级，可以在保持性能的同时节约成本。
- 4) 提高性能：高速缓存作为存储层级中较小且更接近处理器的存储器，能够减少处理器对主存储器的访问次数，从而降低能耗。高速缓存可以存储最频繁使用的数据和指令，避免频繁地访问主存储器，从而减少功耗。

通过存储层级的设计，可以在提供更接近处理器的快速存储访问的同时，兼顾成本、性能和能效等方面的考虑，从而提高整个计算机系统的性能和效率。

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) 过大的页大小：

内部碎片：大页大小意味着每个页可能包含的数据量更多，但进程实际需要的内存可能较小。这会导致在每个页中有大量未使用的空间，从而增加了内部碎片的数量。

内存浪费：当某个页面中的一小部分数据被修改时，整个页面都需要重新写入到磁盘上的交换空间中。这会导致额外的内存开销和磁盘I/O操作，增加了内存和磁盘的负载。

2) 过小的页大小：

外部碎片：小页大小意味着每个页包含的数据量较少，当进程需要的内存超过一个页的大小时，会出现较多的外部碎片。这会导致内存无法有效利用，增加了内存碎片化的风险。

上下文切换开销：较小的页大小意味着更多的页表项需要在内存中保存，从而增加了页表的大小。这会增加上下文切换时的页表切换开销。

因此，页大小的选择需要在内部碎片、外部碎片、内存利用率、磁盘I/O开销和上下文切换开销之间进行权衡。

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：

31	20 19	10 9	8	7	6	5	4	3	2	1	0
PPN[1]	PPN[0]	RSW	D	A	G	U	X	W	R	V	
12	10	2	1	1	1	1	1	1	1	1	1

- 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) 位7至位0的功能如下：

V (Valid/Invalid) : 指示表项是否有效。如果为1，则表示表项有效；如果为0，则表示表项无效。
R (Read) : 指示页表条目是否允许读取访问。
W (Write) : 指示页表条目是否允许写入访问。
X (Execute) : 指示页表条目是否允许执行访问。
U (User) : 指示页表条目是否允许用户态访问。
G (Global) : 指示页表条目是否是全局共享的。
A (Accessed) : 指示页面是否被访问过。
D (Dirty) : 指示页面是否被写入过。

如果用户进程能够自由修改自己的页表，会引发以下问题：

- 2)
安全性问题：用户进程可以修改自己的页表，可能导致越权访问其他进程的内存或内核内存，破坏系统的安全性。
内存管理问题：用户进程可能会错误地修改页表条目，导致内存泄漏、内存碎片化或者无效的内存访问。
- 3) 在 RISC-V 的虚拟内存管理中，一个 X/W/R 位全部为 0 的有效页表条目表示该页面禁止读取、写入和执行访问。这通常用于实现内存保护，以防止某些关键数据或代码被访问、修改或执行。

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：

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

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) 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 位用于进一步限制对物理内存区域的访问权限。页表中的 X/W/R 位指示页表条目是否允许执行/写入/读取访问，而 PMP 控制寄存器中的 X/W/R 位可以进一步限制或放宽对相应物理内存区域的访问权限。通过设置 PMP 控制寄存器中的 X/W/R 位，可以对特定的物理内存区域进行更细粒度的访问权限控制，覆盖页表条目中的权限设置。

2) PMP 配置寄存器中的 L 位（Lock 位）用于锁定对应的 PMP 配置项。当 L 位被设置为 1 时，对应的 PMP 配置项将被锁定，禁止对其进行修改。这可以用于保护关键的内存区域，防止意外或恶意的修改。

PMP 配置寄存器中的 A 位（Address 位）用于启用 PMP 配置项。当 A 位被设置为 1 时，对应的 PMP 配置项将生效，限制相应的物理内存区域的访问权限。当 A 位被设置为 0 时，对应的 PMP 配置项将被忽略，不会对内存访问产生影响。通过设置 A 位，可以选择性地启用或禁用特定的 PMP 配置项。

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)

每个页表条目代表一个4KB的页，因此一个单级页表系统可以映射的页数为 $2^{64} / 4KB = 2^{52}$ 。每个页表条目使用8字节空间，所以总共需要的存储空间为 $2^{52} * 8\text{字节} = 2^{55}\text{字节}$ 。

2)

对于48位的虚拟地址空间，可以映射的页数为 $2^{48} / 4KB = 2^{36}$ 。每个页表条目仍然使用8字节空间，所以总共需要的存储空间为 $2^{36} * 8\text{字节} = 2^{39}\text{字节}$ 。

3)

多级页表可以降低虚拟内存系统的实际页表存储开销的原因是它采用了分级结构的方式来组织页表。多级页表将整个虚拟地址空间划分为多个较小的区域，并为每个区域分配一个页表。每个页表只需要存储当前区域的页表条目，而不需要存储整个地址空间的所有页表条目。这样可以有效地减少页表的大小和存储开销。

通过多级页表，系统只需要为当前活动的页表分配存储空间，而不需要为整个地址空间的所有页表分配存储空间。当发生页表切换时，只需要切换到对应的页表，减少了不必要的内存开销。此外，多级页表还可以实现地址空间的分段管理和按需加载，提高了内存的利用率和系统的性能。