

rCore ch4 lab report

Contents

1. 实现功能	1
2. 问答题	1
2.1. 第一题	1
2.2. 第二题	1
2.2.1. 一	1
2.2.2. 缺页	2
2.2.3. 双页表与单页表	2
3. 荣誉守则	2

1. 实现功能

通过类比 `sys_write` 系统调用中的 `translate_byte_buffer` 函数，实现 `translate_pointer_mut` 函数以获得一个虚拟地址的物理地址的可变引用，来达到在内核态对来自用户态的虚拟内存的修改。以实现启动分页机制后的 `sys_get_time` 和 `sys_task_info`。在 `mmap` 系统调用中，首先获取到当前任务的 `MemorySet` 然后调用其 `insert_framed_area` 来进行申请内存，而在 `munmap` 系统调用中，则通过寻找并移除在 `MemorySet` 结构体中 `areas` 向量中的目标逻辑块，并执行其相应的 `unmap` 方法来取消映射。

2. 问答题

2.1. 第一题

63	54 53	28 27	19 18	10 9	8	7	6	5	4	3	2	1	0
<i>Reserved</i>	PPN[2]	PPN[1]	PPN[0]	RSW	D	A	G	U	X	W	R	V	
10	26	9	9	2	1	1	1	1	1	1	1	1	

1. V(Valid):仅当位 V 为 1 时，页表项才是合法的;
2. R(Read)/W(Write)/X(eXecute):分别控制索引到这个页表项的对应虚拟页面是否允许读/写/执行;
3. U(User):控制索引到这个页表项的对应虚拟页面是否在 CPU 处于 U 特权级的情况下是否被允许访问;
4. A(Accessed):处理器记录自从页表项上的这一位被清零之后，页表项的对应虚拟页面是否被访问过;
5. D(Dirty):处理器记录自从页表项上的这一位被清零之后，页表项的对应虚拟页面是否被修改过。

2.2. 第二题

2.2.1. 一

1. 缺页会导致缺页异常，段错误异常，页面置换等。

2.2.2. 缺页

1. `scause` 寄存器存储了异常原因码，它可以用于确定发生异常的具体原因，包括是否是缺页异常。`stval` 寄存器存储了导致异常的附加信息，对于缺页异常，它通常包含导致异常的虚拟地址。`sepc` 寄存器存储了导致异常的指令的地址。`sstatus` 寄存器包含了有关异常处理的信息，包括中断使能位、用户/核心模式等。
2. 按需分页(Demand Paging), 即只有在任务确实需要的时候，才把所需数据/代码以页为单位逐步从存储设备换入到内存中。其可以优化程序的内存占用，更好的利用计算机资源。
3. 20M
4. 当用户申请内存空间的时候，先只声明分配了 10G 但是不实际分配，只有当用户实际使用的时候再调用 `mmap`。缺页时考虑使用交换内存。
5. 其页表项 `V` 标志位为 0。

2.2.3. 双页表与单页表

1. 在任务切换的时候，通过先切换到本页表的内核态，然后切换到相应个下一个任务。
2. 控制页表项中的 `U` 标志位，置 0 则不可以在 `U` 特权级下访问。
3. 在内核和用户态之间转换时不需要更换页表，也就不需要跳板，可以像之前一样直接切换上下文。
1. 在处理陷入的时候。

3. 荣誉守则

1. 在完成本次实验的过程（含此前学习的过程）中，我曾分别与 以下各位 就（与本次实验相关的）以下方面做过交流，还在代码中对应的位置以注释形式记录了具体的交流对象及内容：无。
2. 此外，我也参考了 以下资料，还在代码中对应的位置以注释形式记录了具体的参考来源及内容：无。
3. 我独立完成了本次实验除以上方面之外的所有工作，包括代码与文档。我清楚地知道，从以上方面获得的信息在一定程度上降低了实验难度，可能会影响起评分。
4. 我从未使用过他人的代码，不管是原封不动地复制，还是经过了某些等价转换。我未曾也不会向他人（含此后各届同学）复制或公开我的实验代码，我有义务妥善保管好它们。我提交至本实验的评测系统的代码，均无意于破坏或妨碍任何计算机系统的正常运转。我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按“-100”分计。