

lab2

实现功能总结

由于用户态调用系统调用，传入的参数是虚拟地址，所以 `sys_get_time` 和 `sys_task_info` 函数之前的实现无效，所以需要在系统调用处理函数中根据不同的任务页表获得其真实物理地址，然后再将数据写入

`mmap` 实现

首先判断参数的合法性，参数不合法直接退出，之后判断传入的虚拟地址页号是否存在已经被映射的虚拟地址页号，如果存在直接退出，在所有判断都通过之后，调用 `insert_framed_area` 函数，创建虚拟地址页号的映射。

`munmap` 实现

和 `mmap` 实现方式同理，首先参数合法性，之后判断传入的虚拟地址页号范围是否存在未被映射的页号，在所有判断都通过之后，由于规定 `munmap` 传入参数对应唯一的 `mmap` 区间，所以直接将之前创造的 `maparea` 删除即可。

简答作业

第一题

请列举 SV39 页表页表项的组成，描述其中的标志位有何作用？

63	62	61	60	54	53	28	27	19	18	10	9	8	7	6	5	4	3	2	1	0	
N	PBMT	Reserved				PPN[2]		PPN[1]		PPN[0]		RSW		D	A	G	U	X	W	R	V
1	2	7				26		9		9		2		1	1	1	1	1	1	1	1

- 仅当 V(Valid) 位为 1 时，页表项才是合法的；
- R/W/X 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/取指；
- U 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；
- G 表示该映射是否适用于所有虚拟地址空间。通常，这个位仅用于操作系统页。
- A(Accessed) 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；
- D(Dirty) 则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表是否被修改过。

第二题

请问哪些异常可能是缺页导致的？

0	12	Instruction page fault
0	13	Load page fault
0	15	Store/AMO page fault
0	20	Instruction guest-page fault
0	21	Load guest-page fault
0	23	Store/AMO guest-page fault

发生缺页时，描述相关重要寄存器的值，上次实验描述过的可以简略。

发生缺页就是通常的异常处理，只有在 `cause` 寄存器上，保存不同的异常原因号，操作系统根据不同的异常原因号，进入不同的异常处理程序，处理异常。

这样做有哪些好处？

减小内存占用，当程序被调用时，程序才占用内存。

处理 10G 连续的内存页面，对应的 SV39 页表大致占用多少内存 (估算数量级即可)？

20MB

请简单思考如何才能实现 Lazy 策略，缺页时又如何处理？描述合理即可，不需要考虑实现。

在任务开始执行时，才创建地址空间，从硬盘导入elf文件，发生缺页时，处理缺页异常，从硬盘加载数据到内存。

此时页面失效如何表现在页表项(PTE)上？

页表项有效位无效

在单页表情况下，如何更换页表？

切换任务时，需要更换页表

单页表情况下，如何控制用户态无法访问内核页面？（tips:看看上一题最后一问）

设置页表项的U位，标志用户态无法访问的内存空间

单页表有何优势？（回答合理即可）

减小不必要的页表切换带来的性能消耗

双页表实现下，何时需要更换页表？假设你写一个单页表操作系统，你会选择何时更换页表（回答合理即可）？

在进入异常处理函数时，缺换到任务的对应内核线程。在退出异常处理函数，切换到下一任务的用户页表

在切换下一任务的地址空间时，更换页表

荣誉准则

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：

|
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：

|
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。