

# os lab2

计算机系 2019011312 姚建竹

## 编程作业

### sys\_mmap

首先检查start是否按照页大小对齐，以及port是否合法。然后生成对应的MapPermission，在给定的每个vpn range里面，检查是否已经有页面被映射且合法，如果有，那么返回-1。最后调用memory set的insert\_framed\_area对这个虚拟地址空间进行映射。

### sys\_munmap

首先检查start是否按照页大小对齐。然后检查对应的虚拟地址空间是否有已经被释放过的页面，如果有返回-1。如果页面本身不合法，也返回-1。最后使用memory set的page table对每个虚拟地址进行unmap

### sys\_get\_time

首先使用translated\_byte\_buffer将当前task对应的地址空间内的TimeVal指针对应的一个字节片段取出来，再将一个新的TimeVal变量逐字节写进去就可以了

### sys\_task\_info

逻辑与sys\_get\_time基本一致

## 问答作业

### 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	

标志位含义：

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. G(Global): 全局Mapping
5. 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;
6. 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。

## 第二题

1. 指令缺页异常、store/AMO缺页异常、load缺页异常
2. 发生缺页异常时, `sstatus`记录CPU之前的特权级, 比如User, `scause`里面记录是page fault的异常原因, `stval`写入了包含错误的虚拟地址, `sepc`记录之前执行的最后一条指令的地址, `stvec`给出控制处理代码的入口地址。
3. 好处: 节约内存空间, 考虑到加载的程序可能并不会被执行, 也有可能节约从磁盘加载到内存的时间
4. 每个页面大小为4KB, 所以一共有页面  $\frac{10 \times 2^{30}}{4 \times 2^{10}} = 2.5 \times 2^{20}$  个, 每个页面对应的页表项大小为64位, 即8Bytes, 所以页表大概占用空间为  $2.5 \times 2^{20} \times 8 = 20M$
5. 比如可以将想要申请的虚拟空间分成几块, 先不一次性申请完毕, 而是当用户想要访问某块时, 再帮忙申请这个块对应的空间。
6. 页表项的Valid位为0

## 第三题

1. 在多级页表中, 使用不同的页表目录。
2. 将内核页面的U标志位置为0
3. 在内核和用户态切换时不需要切换页表, 可以直接切换上下文。
4. 双页表在切换上下文时需要更换页表(即用户程序和内核转换时, 用户程序转换时)。单页表中我会选择当切换任务的时候更换页表, 因为即使多个应用都在用户态, 他们也不应该能够获得其他应用的内存空间布局, 否则会有安全问题。