实验报告

一. 功能简介

1. 实现 map

a. 判断能否 map

初始地址按页对齐，否则不能 map。检查 port 是否合法，合法才能进行 map。检查 port 的 1、2、3 位是否为 0，以此来决定是否具有读、写、执行的权限。得到当前任务的地址空间，检测在[start, start + len)是否存在已被映射的页，存在则不能继续 map。

b. 实现映射

当判断可以进行映射时，得到当前任务的地址空间。对于从 start 到 end 的每一个 vpn，检测在当前地址空间的每一个 area，如果包含 vpn 则进行 map_one 映射。当处理完所有 vpn 之后，插入 frame_area。

2. 实现 unmap

首先检查初始地址是否按页对齐，否则不能 unmap。对于从 start 到 end 的每一个 vpn，检测在当前地址空间的每一个 area，如果包含 vpn，则能进行 unmap，直接调用 unmap_one。最后再移除前地址空间中 data_frame 为空的每一个 area，移除即可。

3. Task_info 与 get_time

通过 taskinfo 得到虚拟地址，再通过虚拟地址得到 ppn，ppn 左移十二位拼接上 offset 得到 taskinfo，再像 lab1 写入信息即可。

按照上述相同方式先得到得到 ppn，再得到 TimeVal，最后按照 lab1 写入 time 即可。

二. 思考题

1. 请列举 SV39 页表页表项的组成，描述其中的标志位有何作用？

答：

63 54	53 28	27 19	18 10	9 8	7	6	5	4	3	2	1	0
Reserved	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

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；

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。

2. 缺页指的是进程访问页面时页面不在页表中或在页表中无效的现象，此时 MMU 将会返回一个中断， 告知 os 进程内存访问出了问题。os 选择填补页表并重新执行异常指令或者杀死进程。

a. 请问哪些异常可能是缺页导致的？
   答：illegal instruction，environment call，instruction page fault 等
b. 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
   答：保存了 sstatus,sepc,satp 寄存器的值，切换 sp 和 satp 寄存器的值。

3. 缺页有两个常见的原因，其一是 Lazy 策略，也就是直到内存页面被访问才实际进行页表操作。 比如，一个程序被执行时，进程的代码段理论上需要从磁盘加载到内存。但是 os 并不会马上这样做， 而是会保存 .text 段在磁盘的位置信息，在这些代码第一次被执行时才完成从磁盘的加载操作
   a. 这样做有哪些好处？
      答：可以提升效率，节省时间和内存。
4. 其实，我们的 mmap 也可以采取 Lazy 策略，比如：一个用户进程先后申请了 10G 的内存空间， 然后用了其中 1M 就直接退出了。按照现在的做法，我们显然亏大了，进行了很多没有意义的页表操作。
   a. 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)？
      答：20M
   b. 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。
      答：调用 mmap 时不分配 frame,只在地址空间添加部分虚拟地址的范围。通过 alloc 等方法分配物理页帧。
   c. 此时页面失效如何表现在页表项(PTE)上？
      答：pte 的 valid 位为 0
5. 为了防范侧信道攻击，我们的 os 使用了双页表。但是传统的设计一直是单页表的，也就是说， 用户线程和对应的内核线程共用同一张页表，只不过内核对应的地址只允许在内核态访问。
   a. 在单页表情况下，如何更换页表？
      答：切换 satp
   b. 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）
      答：将页表项的 U 位置为 0.
   c. 单页表有何优势？（回答合理即可）
      答：节约空间，减少页表占用的内存。
   d. 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）？
      答：双页表：用户态到内核态时
      单页表：switch 时