实现功能：

通过查表将虚拟内存转为物理内存，来避免虚拟内存导致的sys\_get\_time和sys\_task\_info函数失效。

实现mmap和munmap：检查边界，然后找到vpn并使用维护的memory\_set进行插入和删除。

问答题：

1.页表项的组成

V 位决定了该页表项的其余部分是否有效（V = 1 时有效）。若 V = 0，则任何遍历到此页表项的虚址转换操作都会导致页错误。

R、W 和 X 位分别表示此页是否可以读取、写入和执行。如果这三个位都是 0，那么这个页表项是指向下一级页表的指针，否则它是页表树的一个叶节点。

U 位表示该页是否是用户页面。若 U = 0，则 U 模式不能访问此页面，但 S 模式可以。若 U = 1，则 U 模式下能访问这个页面，而 S 模式不能。

G 位表示这个映射是否对所有虚址空间有效，硬件可以用这个信息来提高地址转换的性能。这一位通常只用于属于操作系统的页面。

A 位表示自从上次 A 位被清除以来，该页面访问过。

D 位表示自从上次清除 D 位以来页面是否被弄脏（例如被写入）。

RSW 域留给操作系统使用，它会被硬件忽略。

PPN 域包含物理页号，这是物理地址的一部分。若这个页表项是一个叶节点，那么 PPN 是转换后物理地址的一部分。否则 PPN 给出下一节页表的地址。

1. 缺页

1）缺页导致的异常：

Exception Code = 12: page fault caused by an instruction fetch

Exception Code = 13: page fault caused by a read

Exception Code = 15: page fault caused by a write

1. 发生缺页时，相关重要寄存器的值：

M态下：

mtvec（Machine Trap Vector）它保存发生异常时处理器需要跳转到的地址。

mepc（Machine Exception PC）它指向发生异常的指令。

mcause（Machine Exception Cause）它指示发生异常的种类。

mie（Machine Interrupt Enable）它指出处理器目前能处理和必须忽略的中断。

mip（Machine Interrupt Pending）它列出目前正准备处理的中断。

mtval（Machine Trap Value）它保存了陷入（trap）的附加信息：地址exception中出错的地址、发生非法指令exception的指令本身，对于其他异常，它的值为 0。

mscratch（Machine Scratch）它暂时存放一个字大小的数据。

mstatus（Machine Status）它保存全局中断使能，以及许多其他的状态

S态下：

S 模式有的sepc、stvec、scause、sscratch、stval 和 sstatus，它们执行与 M 模式 CSR 同的功能。监管者异常返回指令 sret 与 mret 的行为相同。S 模式处理例外的行为已和 M 模式非常相似。如果 hart 接受了异常并且把它委派给了S 模式，则硬件会原子地经历几个类似的状态转换，其中用到了 S 模式而不是 M 模式的CSR:

发生例外的指令的 PC 被存入 sepc，且 PC 被设置为 stvec。

scause 根据异常类型设置值，stval 被设置成出错的地址或者其它特定异常的信息字。

把 sstatus CSR 中的 SIE 置零，屏蔽中断，且 SIE 之前的值被保存在 SPIE 中。

发生exception时的权限模式被保存在 sstatus 的 SPP 域，然后设置当前模式为 S 模式。

3）基于此机制, 程序开始运行时并不需要将全部内容加载到内存, 也就是说在程序运行的机器的实际物理内存 小于 进程运行所需总内存时, 程序依然能够通过页面的换入换出机制运行起来

4）10G ；512

5）不考虑局部性的情况下, 使用lazy策略时, 根据发生缺页异常的类型, 去memory\_set的各个段寻找 [缺页的虚拟地址在当前逻辑段 && 逻辑段权限达到缺页的要求], 如果寻找成功, 则将页面加载到物理内存并映射

6）Valid位为0

3.双页表与单页表

1）切换任务后, 返回用户态时重写sntp

2）内核对应的地址只允许在内核态访问, PTE 的 U = 0

3）用户态和内核态切换时不用更换页表

4）用户态和内核态切换时; 在返回用户态时 or 任务切换时, 因为内核部分都是一样的