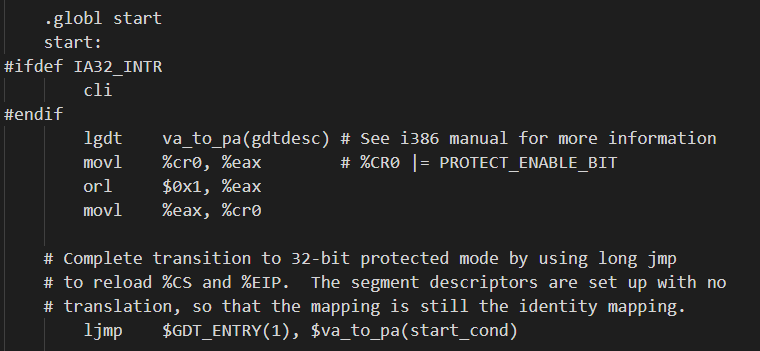
PA3-2

1.

观察kernel/start/start.S



在执行完上述五条汇编指令后(lgdt开始的)nemu即进入了保护模式.

这五条指令装载了gdtr寄存器, 并将cr0的pe位置为1, 然后通过ljmp跳转执行.

2.

物理地址. 因为在执行程序时, 我们需要借助gdtr寄存器将虚拟地址转化成物理地址去访问, 若gdtr中的段表首地址不为物理地址, 则会陷入死循环中.

PA3-3

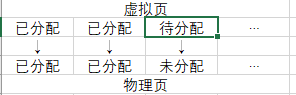
1.

由kernel/src/memory/kvm.c中的init\_page()函数代码可知(因为截图太长, 此处不再贴图), kernel中的虚拟页和物理页是一一对应的映射关系(具体见下图).



2.

用户进程的虚拟页和物理页也是一一对应的映射关系, 只是映射方式和kernel不同, 此处是将当前待分配的虚拟页映射到地址最低的可分配物理页(具体见下图).



能分配给用户进程的虚拟页的物理页的起始地址是0x10000000.

kernel映射到物理地址为0x0 – 0x10000000的内存中.

3.

理论上正确. 因为全局变量的地址是虚拟地址, 在页表初始化未完成时, 无法将虚拟地址正确映射到物理地址, 所以不能正确地访问全局变量.

但在我们所用的nemu的kernel中, 可以通过va\_to\_pa函数绕过地址转换过程, 强行访问全局变量, 在init\_page函数中也正是这么做的.