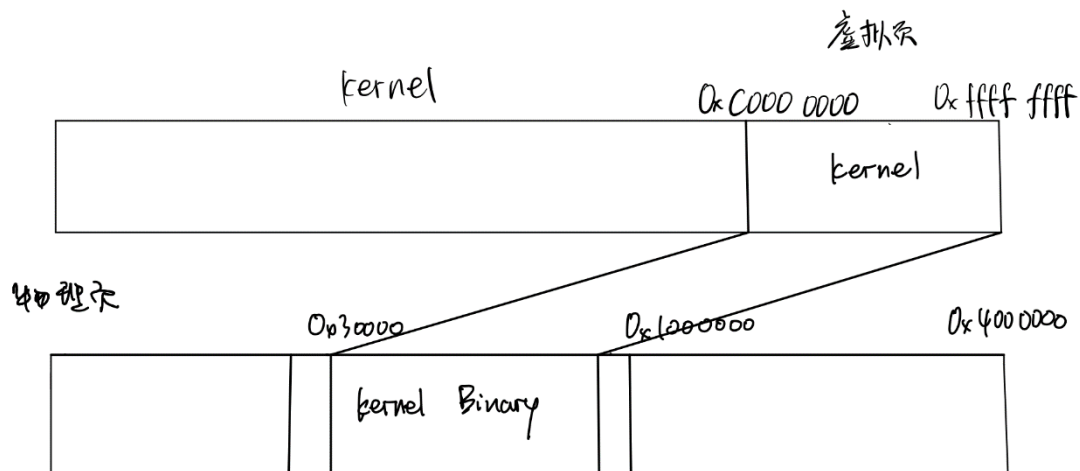


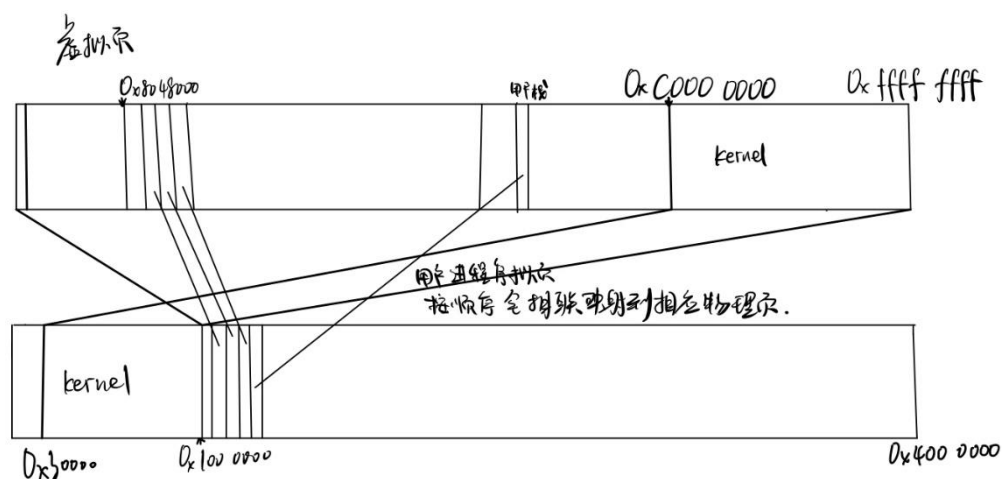
1. Kernel 的虚拟页和物理页的映射关系是什么？请画图说明；
全相联映射



2. 以某一个测试用例为例，画图说明用户进程的虚拟页和物理页间映射关系又是怎样的？
Kernel 映射为哪一段？你可以在 loader() 中通过 Log() 输出 mm_malloc 的结果来查看映射关系，并结合 init_mm() 中的代码绘出内核映射关系。

全相联映射

```
objcopy -O binary kernel kernel.img
make-[1]: Leaving directory '/home/pa191220013/pa2020_fall1/kernel'
pa191220013@b740d52aa5f4:~/pa2020_fall1$ ./nemu/nemu --autorun --kernel --testcase mov-c
NEMU load and execute img: ./kernel/kernel.img elf: ./testcase/bin/mov-c
nemu trap output: [src/main.c,82,init_cond] {kernel} Hello, NEMU world!
nemu trap output: [src/elf/elf.c,29,loader] {kernel} ELF loading from ram disk.
nemu trap output: [src/elf/elf.c,44,loader] {kernel} m_addr:1000000 p_vaddr:0 p_memsz:f4
nemu trap output: [src/elf/elf.c,44,loader] {kernel} m_addr:1001000 p_vaddr:8049000 p_memsz:10f
nemu trap output: [src/elf/elf.c,44,loader] {kernel} m_addr:1002000 p_vaddr:804a000 p_memsz:4c
nemu trap output: [src/elf/elf.c,44,loader] {kernel} m_addr:1003000 p_vaddr:804c000 p_memsz:68
nemu trap output: [src/elf/elf.c,71,loader] {kernel} m_addr:1004000 stack:bff00000 stack_size:100000
nemu: HIT GOOD TRAP at eip = 0x08049103
NEMU2 terminated
pa191220013@b740d52aa5f4:~/pa2020_fall1$
```



3. “在 Kernel 完成页表初始化前，程序无法访问全局变量”这一表述是否正确？在 init_page() 里面我们对全局变量进行了怎样的处理？

这一表述正确，在 init_page() 里面我们进行了内核页表的初始化，并将页表首地址传给 CR3，全局变量使用的是虚拟地址，在完成页表初始化后，通过分页机制才能进一步转化为相应的

物理地址进行访问。而 `init_page()`里我们通过 `va_to_pa` 宏将全局变量的虚拟地址转化为了物理地址。