OSLAB2 实验报告
141242026 匡亚明学院 刘驭壬
邮箱：141242026@nju.edu.cn

实验进度：

1. 实现了系统调用，所有的对读串口输出，写显存，响应键盘中断等操作都以系统调用的方式进行。
2. 实现了虚拟内存的管理（分页），完成了 pmap.c,写了 init_mm 等函数
3. 实现了 loader
4. 将 game 与 kernel 分离，并且能够正确运行

必答题：

1.

```
Q：如果你参考jos的代码，在entry.S中，有这样2行代码：
_start = RELOC(entry)
...
jmp *%eax
请分别解释这2行代码的意义
```

第一行代码含义：entry 是我们的程序入口，它在链接的时候被链接到高地址，但是由于我们还没有开启分页，我们代码只能执行在物理地址上，RELOC 函数将 entry-kernelbase 的值赋给_start。下面就是 entry.S 和 kernel.ld 的配合,在 kernel.ld 中 entry(_start)将_start 设置为 kernel 的入口函数，所以加载 kernel 后可以从正确地址执行。

第二行代码是为了从低地址空间回到高地址空间（已经开启分页）。

2.

```
首先，因为进程后面需要进行系统调用，系统调用需要执行kernel中的代码，所以你需要将kernel的页目录 拷贝一份作为进程的页目录的模板。

Q：这样做为什么可以，会不会带来什么问题?
```

可以，可能出的问题是改变 kernel 的地址空间。这个可以通过设置保护位解决。

关于实验的部分说明：

系统调用没有参考任何资料。
Pmap.c 部分部分参考了网上的代码。
Loader 采取陈挚同学的建议仿照了 PA 的写法。
Makefile 的修改主要是仿照 Makefile 中 kernel 的部分。

出现的问题：

首先其实每个地方的理解到实现都困难重重，坑数众多。

1. page_init 的链表要从低到高
（boot_map_region 是为了让 kernel 填 game 的页表。）
如果从高到低，会映射到页表还没映射到的地方。（最前面被 0 到 4 兆覆盖，要先访问），所以会访问出错.

```
int i;
for(i=0;i<size/PGSIZE;i++,va+=PGSIZE){
        p=pgdir_walk(pgdir,(void*)va,1);
        *p=(pa+PGSIZE*i)|perm|PTE_P;
}
//printk("f");
```

for 里面的第二个语句会缺页。

2. Makefile 文件中将 game 分离的时候，有的地方改错，依然使用了 kernel 的.o 文件，所以 make 的时候一直错，后改回。

鸣谢
感谢刘志刚同学在 debug 方面提供的帮助，感谢陈挚在代码原理方面的和 loader 编写部分提出的建议（仿 PA），以及指出 page_init 部分的坑。