得分

第六题（15 分）虚拟内存地址转换

为了提升虚拟内存地址的转换效率，降低遍历两级页表结构所带来的地址转换开销，英特尔处理器中引入了大页 TLB，即一个 TLB 项可以涵盖整个 4MB 对齐的地址空间（针对 32 位模式）。只要设置页目录页中页目录项（PDE）的大页标志位，即可让 MMU 识别这是一个大页 PDE，并加载到大页 TLB 项中。大页 PDE 中记录的物理内存页面号必须是 4MB 对齐的，并且整个连续的 4MB 内存均可统一通过该大页 PDE 进行地址转换。

在 32 位的 Linux 系统中，为了方便访问物理内存，内核将地址 0~768MB 间的物理内存映射到虚拟内存地址 3GB~3GB+768MB 上，并通过大页 PDE 进行进行该区间的地址转换。任何 0~768MB 的物理内存地址可以直接通过加 3G（0xC0000000）的方式得到其虚拟内存地址。在内核中，除了该区间的内存外，其他地址的内存通常都通过普通的两级页表结构来进行地址转换。

假设在我们使用的处理器中有 2 个大页 TLB 项，其当前状态如下：

索引号	TLB 标记	页面号	有效位
0	0xC4812	0x04812	1
1	0xC9C33	0x09C33	1

有 4 个普通 TLB 项，当前的状态如下：

索引号	TLB 标记	页面号	有效位
0	0xF8034	0x04812	1
1	0xF8033	0x09812	1
2	0xF4427	0x12137	1
3	0xF44AE	0x17343	1

当前页活跃的目录页（PD）中的部分 PDE 的内容如下：

PDE 索引	页面号	其他标志	大页位	存在位
786	0x04800	…	1	1
807	0x09C00	…	1	1
977	0x09C33	…	0	1
992	0x09078	…	0	1

注：普通页面大小为 4KB，并且 4KB 对齐。每个页面的页面号为其页面起始物理地址除以 4096 得到。大页由连续 1024 个 4KB 小页组成，且 4MB 对齐。

1．分析下面的指令序列，

```
movl $0xC4812024, %ebx
movl $128, (%ebx)
movl $0xF8034000, %ecx
movl $36(%ecx), %eax
```

请问，执行完上述指令后，eax 寄存器中的内容是（）；在执行上述指令过程中，共发生了（）次 TLB miss？同时会发生（）次 page fault？

注：不能确定时填写“--”。

2．请判断下列页面号对应的页面中，哪些一定是页表页？哪些不是？哪些不确定？

页面号	是否为页表页（是/不是/不确定）
0x04800	
0x09C33	
0x09812	

3．下列虚拟地址中哪一个对应着够将虚拟内存地址 0xF4427048 映射到物理内存地址 0x14321048 的页表项（）？

（A）0x09C33027　　（B）0xC9C3309C

（C）0xC9C33027　　（D）0x09C3309C

通过上述虚拟地址，利用 movl 指令修改对应的页表项，完成上述映射，在此过程中，是否会产生 TLB miss？（）（回答：会/不会/不确定）

修改页表项后，是否可以立即直接使用下面的指令序列将物理内存地址 0x14321048 开始的一个 32 位整数清零？为什么？

```
movl $0xF4427048, %ebx
movl $0, (%ebx)
```

答：

答案：

第 1 小题（各 1 分）

（1）128；（2）0；（3）0；

两个虚拟地址映射的是同一块物理内存；因此读出的就是写入的；此过程中全部

TLB 命中，因而既无 TLB miss，也不会有 page fault。

第 2 小题（各 2 分）
（1）不确定；因为是大页，一定不是当前页目录项对应的页表页，但不一定该页面不会用作其他页目录项对应的页表页。
（2）是；当前页目录项（977）对应的页表页。
（3）不确定；任何页面都可能用作页表页。

第 3 小题
B；（2 分）
虚拟地址对应的页表页的页面号（0x09C33）已知，通过其地址直接加 3G（即 0xC0000000），即可得到当前页表页的基地址（0xC9C33000），在加上对应的第 0x27 乘以 4 到页内偏移。

不会；（2 分）
因为地址 0xC9C33000 在大页映射范围内，已经被大页 TLB 项覆盖到了，会直接命中。

不能直接修改。（1 分）
因为 TLB 项中的内容和页表中的内容不一致，需要将对应的 TLB 项设置为失效，然后通过 TLB miss 重新加载页表结构中新的地址映射关系，之后才能访问对应的虚拟地址。（1 分）