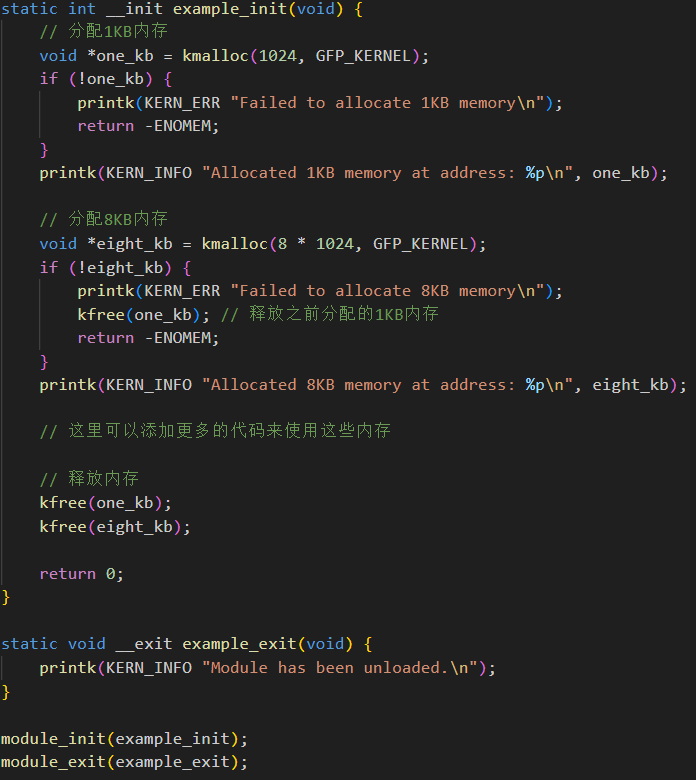
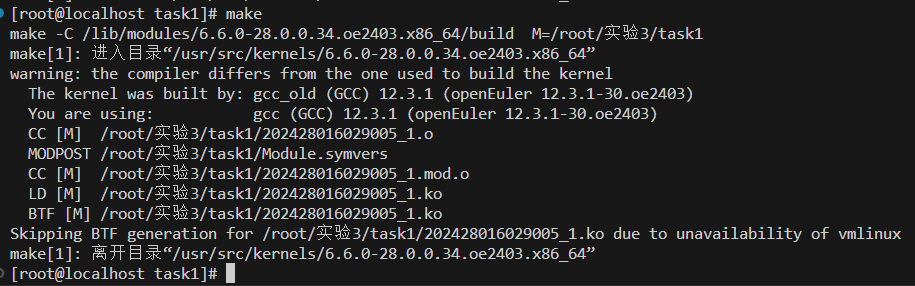
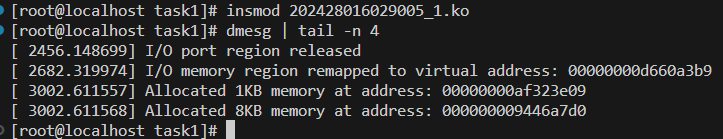
***任务1：***

依照要求编写代码如下：



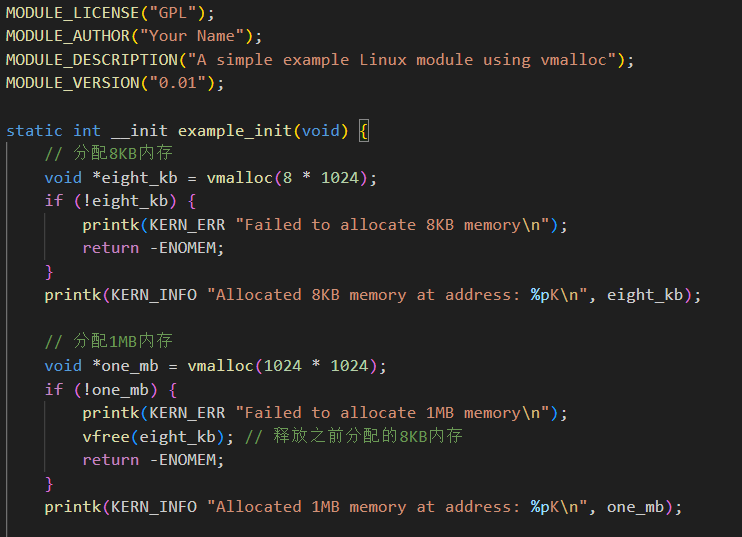
make及其内核输出结果：

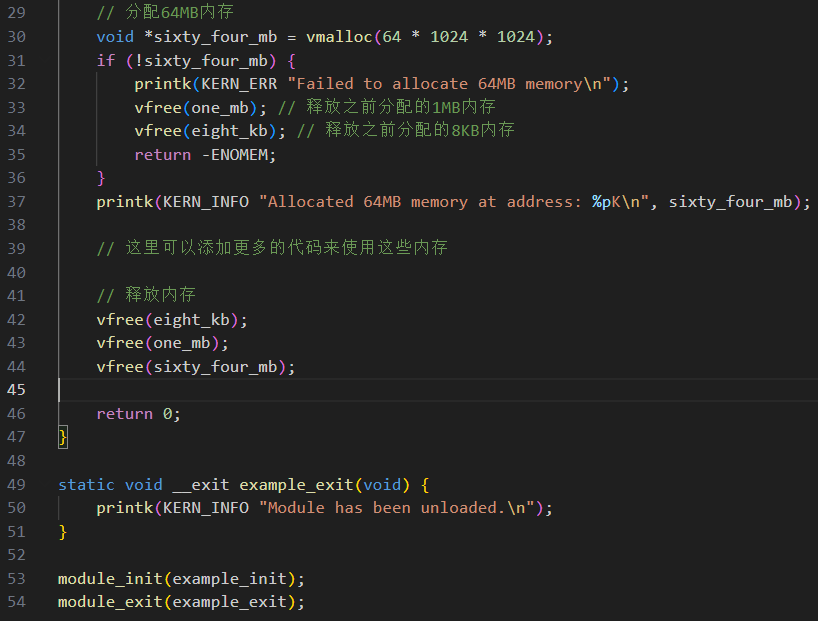




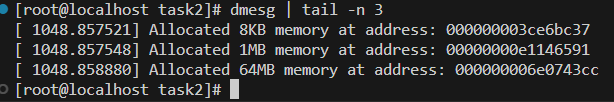
***任务二：***

依照任务要求编写代码如下：





内核输出结果：



分析结果：

在32位系统中，虚拟地址空间通常是4GB。这4GB被分为两部分：

用户空间：通常占用前3GB（0x00000000 - 0x7FFFFFFF）。

内核空间：占用剩下的1GB（0xC0000000 - 0xFFFFFFFF）。

在32位系统中，vmalloc分配的内存通常位于内核空间的高端，即接近0xC0000000的地址。这是因为内核需要保留低地址空间用于其他目的，如内核代码、数据结构和早期的内存分配。

在64位系统中，虚拟地址空间要大得多，通常有16EB（16 Exabytes）或更多。这使得vmalloc可以分配大块内存而不会影响内核的低地址空间。

在64位系统中，vmalloc分配的内存通常位于一个特定的区域，称为虚拟内存映射区。这个区域通常位于内核空间的高端，远高于常规的内核代码和数据结构。

在64位系统中，通常存在两个地址空间：

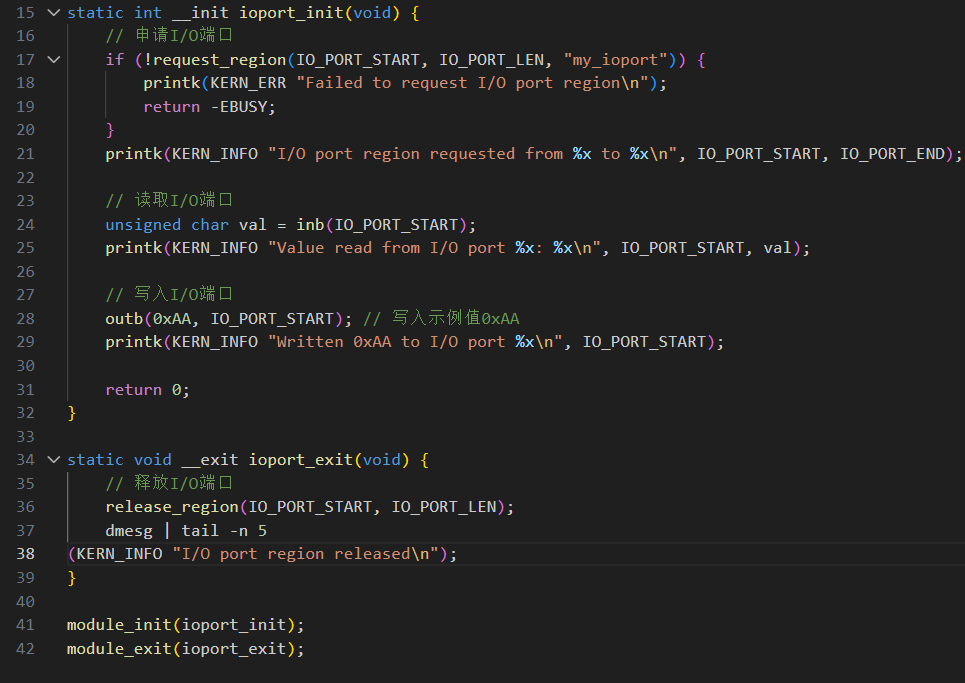
Canonical address space：这是直接映射的地址空间，通常从0x0000000000000000到0x00007FFFFFFFFFFF，这个范围是直接映射的物理内存。

High memory address space：这是通过某种形式的映射机制（如分页）访问的地址空间，通常从0xFFFF800000000000到0xFFFFFFFFFFFFFFFF。

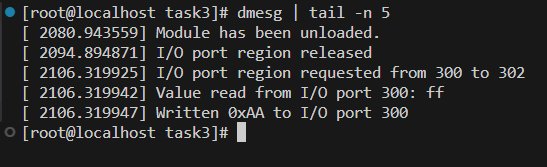
因此我的结果是64位的分配结果

***任务三：***

按照任务要求编写代码如下：

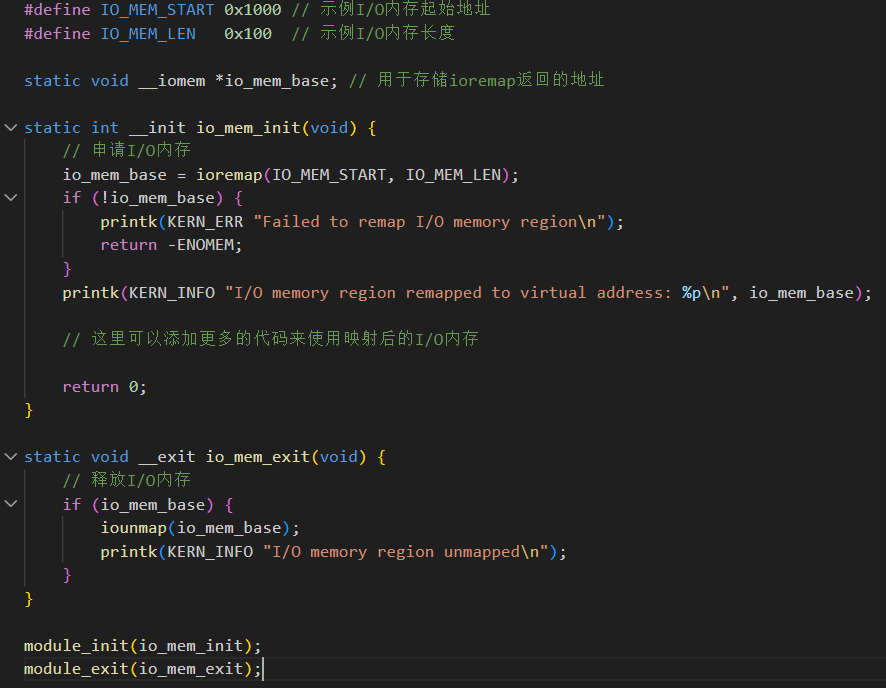


运行结果：



***任务四：***

按照任务要求编写代码如下：



内核模块输出如下：

