## 请简述 head.s的工作原理

当引导Linux 0.00内核时，head.s文件扮演了至关重要的角色，确保系统能够启动和执行操作系统的核心功能。以下是Linux 0.00内核的引导过程的完整阐述：

（1）段寄存器和栈指针的设置：首先，head.s文件会设置段寄存器和栈指针，以确保程序可以正确地执行和访问内存。这是引导加载程序的一部分，它确保系统处于适当的状态。

（2）读取内核映像文件的头部信息：接下来，head.s文件将读取内核映像文件的头部信息，以获取内核程序的大小、入口点和其他关键信息。这些信息对于将内核加载到内存的正确位置至关重要。

（3）加载内核程序到内存：此后，head.s文件的任务是将内核程序从磁盘加载到内存中，将其复制到正确的内存位置，以便系统能够访问它。这确保了内核在内存中正确初始化。

（4）设置系统调用向量表：内核的初始化过程包括设置正确的系统调用向量表，以确保内核程序能够正确响应系统调用。这涉及建立系统调用处理程序的入口点和相关数据结构。

（5）跳转到内核程序的入口点：最后，引导加载程序或内核启动代码将跳转到内核程序的入口点，将控制权转交给内核。从这一刻开始，内核将开始执行操作系统的核心功能，例如进程管理、内存管理和文件系统操作

## 请记录 head.s 的内存分布状况，写明每个数据段，代码段，栈段的起始与终止的内存地址

## 简述 head.s 57 至 62 行在做什么？

## 简述 iret 执行后， pc 如何找到下一条指令？

## 记录 iret 执行前后，栈是如何变化的？

## 当任务进行系统调用时，即 int 0x80 时，记录栈的变化情况。