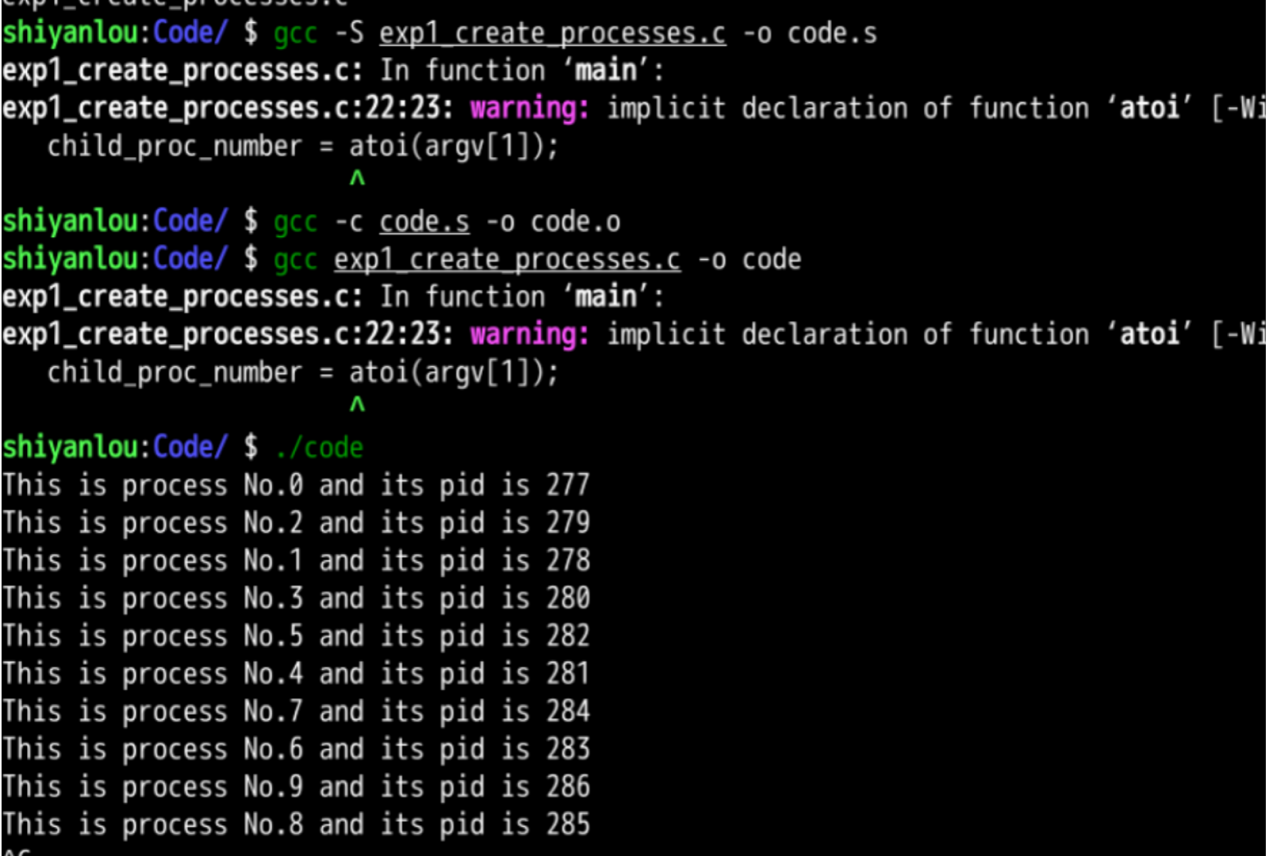
1.

1）都是虚拟空间，编译器和链接器都是在虚拟地址空间进行操作，只有在程序加载到内存并运行时才会被映射到物理地址空间。



2）因为编译和链接阶段，不同的代码会被分配到合适的虚拟地址，这是一种静态重定位，因为这个过程发生在程序加载到内存运行之前。

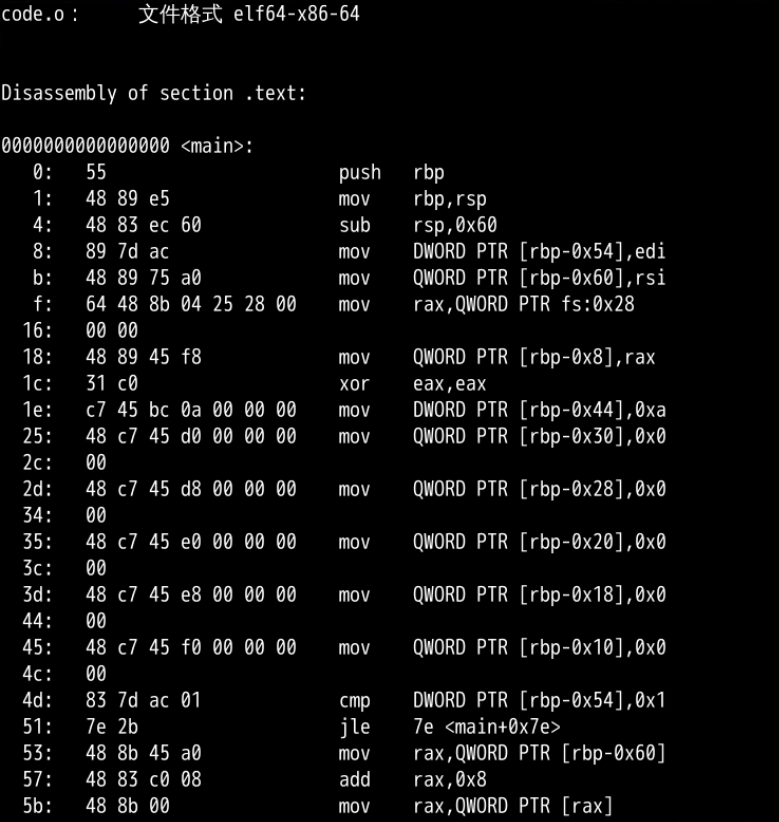
3）属于静态链接。

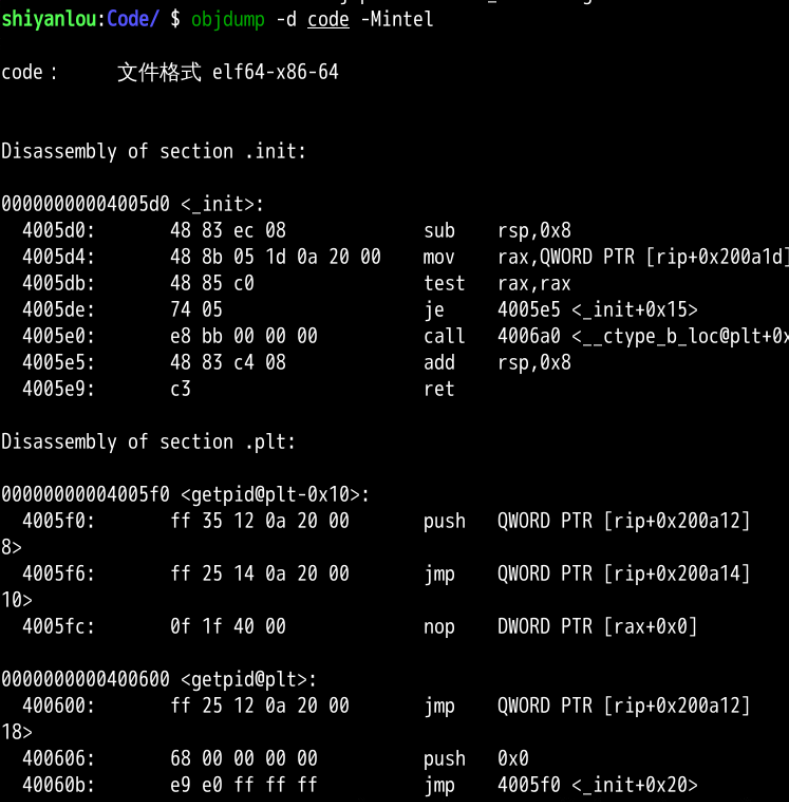
静态链接是指把所有需要的目标文件和库文件都链接到可执行文件中

动态链接是指在程序运行时，动态地将需要的库文件加载到内存中并进行链接。

4）会被装入到物理内存。操作系统会设置程序计数器以及其他寄存器的值，以准备运行程序。

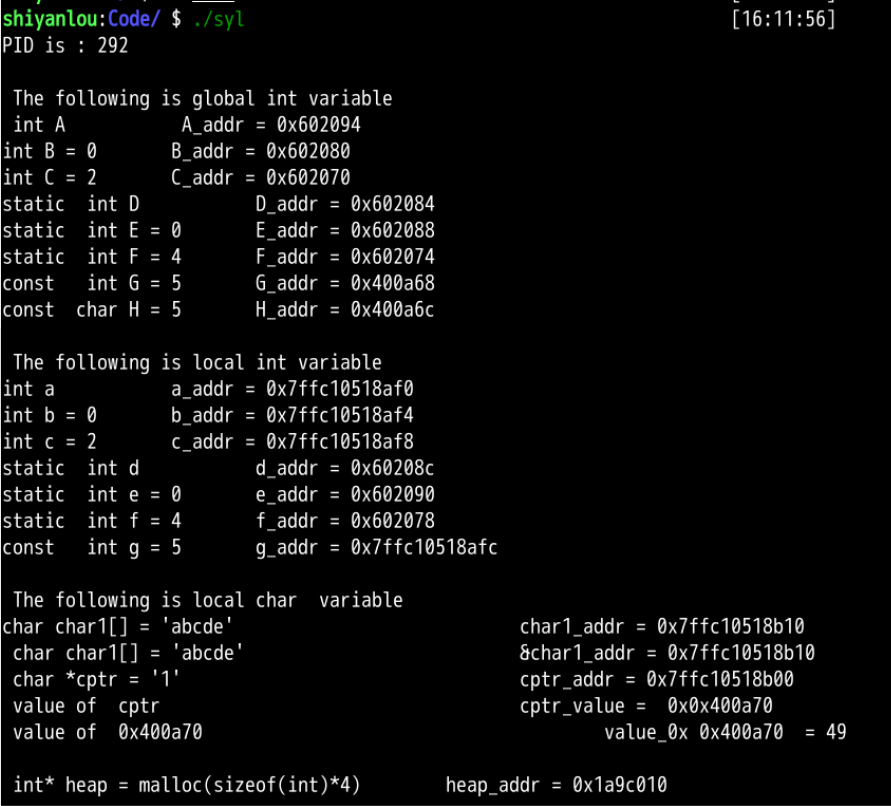
**code.o文件:**



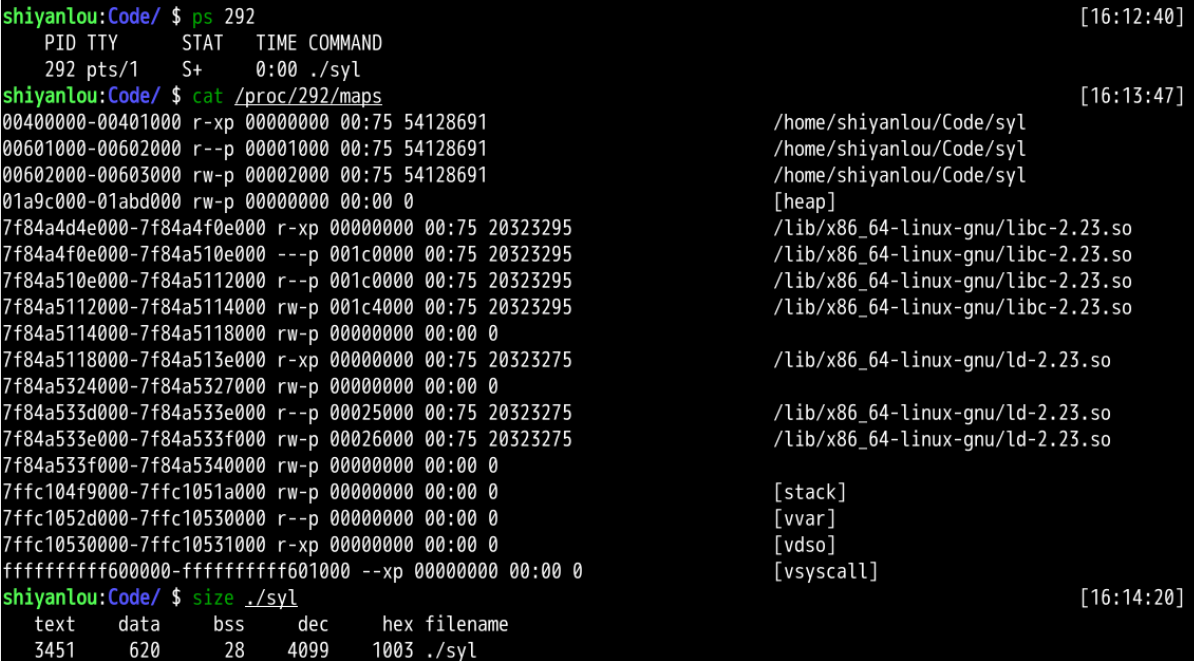
**code文件：**

2.

运行截图：



对比图：



可由进程的地址一一对应，因此无需箭头

理解：

* 代码段：存储程序的可执行代码。这部分内存是**只读**的，以防止程序在运行时修改自己的代码。
* 数据段：存储程序的全局变量和静态变量。数据段分为已初始化数据段和未初始化数据段。已初始化数据段存储程序中已赋初值的全局变量和静态变量，而未初始化数据段存储未赋初值的全局变量和静态变量。
* 堆：用于存储程序在运行时动态分配的内存。堆从**低地址**向高地址增长。程序可以通过内存分配函数（如C语言中的malloc和C++中的new）在堆中分配内存，并在不再需要时通过释放函数（如C语言中的free和C++中的delete）释放内存。
* 栈：用于存储程序的局部变量、函数调用的参数和返回地址等。栈从**高地址**向低地址增长。函数调用时，栈会自动分配所需的内存空间，函数返回时，栈会自动释放这部分内存。

text 表示代码段（即可执行指令），data 表示数据段（包括初始化数据和非初始化数据），bss 表示未初始化数据段，dec 表示总大小（单位为字节），hex 表示十六进制的总大小。