C语言综合研究三报告

先解决一下这次指导书中提出的问题。

1. 程序运行时全局变量、局部变量以及参数的段地址在哪个寄存器里？

我们可以通过观察程序调用这些变量时的操作来得到答案。比如全局变量，程序对它的寻址是[idata]形式的，即使用ds为默认段地址，所以段地址存在ds中。同理可得局部变量和参数的段地址存在ss中。

1. 全局变量、局部变量以及参数分别存储在什么段，函数返回值存储在什么位置？

既然已经知道了段地址，那么就知道了存储的段。整型返回值的存储位置为ax寄存器，其余的还没测试。

1. 全局变量、局部变量以及参数的内存空间在什么时候分配，什么时候释放？

这个问题我认为问得有点不准确，我一度很疑惑这是否是暗示汇编中有分配内存和释放内存的指令。但是这个程序里明显是没有的。那么从CPU的内存调度功能看，当然是用之前分配，不用的时候释放，那么就是研究变量的有效区间问题。从这个角度来看，很显然，全局变量在程序初始化时就分配内存，程序结束时释放内存。而局部变量就在子程序开始时分配内存，子程序结束时释放内存。参数则是在传递前分配内存，子程序结束时释放内存。

1. 函数f3在调用和返回方式上与f1和f2有何不同？

其实就是汇编的长转移和段转移的区别。前者在转移和返回时都需要修改CS和IP而后者只修改IP。

1. 程序2变量n和a的内存分配方式有何不同？

这个就是静态局部变量和局部变量的区别。前者的保质期要长一些，所以存在数据段中，在程序结束后释放；后者保质期短，存在栈段中，在子程序结束后释放。而所谓释放内存就是指CPU可以将这段内存分配给其他程序或子程序使用，不再保证这段内存内容的安全性。

1. 程序3中所有变量的存储空间相邻吗？tc2.0中，整型、字符型、长整型数据的存储空间分别为多大？

就测试显示的分配而言，这些变量的存储空间都是相邻的，而整型、字符型、长整型数据的存储空间分别为2、1、4字节。

1. 不同的数据类型对数据运算方式有何影响？

从长度来看，不同的长度使用寄存器的数量不同，需要两个及以上寄存器的，就需要进位标志位的参与。而编码规则的不同，如整型和浮点型，意味着不同的计算公式，对同一个运算，也会有不同的操作。

1. 程序4中变量a、b和他们各个数据项的存储空间是如何分配的？

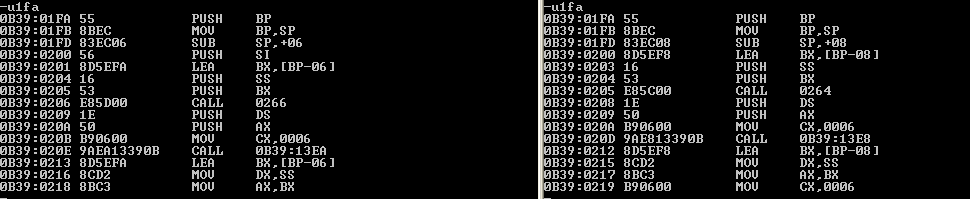
简单的说，就是连续分配。a和b连续，他们中的各个数据项也连续。

1. 向函数传递结构体型数据是如何实现的？

可以通过debug先观察子程序是如何利用传递后的数据的。我发现子程序是直接从栈里调用的。然后再观察程序是如何将数据放入栈段。可以很明显的从debug里看到程序调用了非我们所编写的子程序进行参数的传递。再分析子程序所接受的参数以及实现的功能。

由于没有很深的分析整个实现传递功能的代码，所以我只能大致的认为，程序通过一个接受结构体长度和结构体首地址的子程序，通过repz指令来将结构体入栈。

1. 从函数返回的结构体型数据存储在何处？

和上面的方法一样，我们先观察程序如何调用返回值。通过对比利用返回值赋值和不用返回值赋值的汇编代码发现这样的区别：

主要区别是前者多一步SI入栈，但是这一步操作的实际意义暂时没弄懂。

没办法，就一步一步跟程序，最后发现两者的实际区别。两者都调用了一个非程序员定义的函数，而这个函数还是使用repz指令，将返回值搬到了一段内存c7c:042a里。不同的是，前者在子程序里面调用这个子函数，然后在子程序后面再调用这个子函数将c7c:042a的内容转移到栈里。后者只是在子程序后面调用这个子函数将返回值搬到c7c:042a。

所以可以大致的认为，结构体型的返回值会被放置在程序的一段内存里，我猜想是数据段里。还有需要讨论的则是这段内存是否长期安全，也就是这个返回值是否是安全的?

下面是我的一些疑问与猜想：

1. 全局变量与静态局部变量的区别在于什么？

首先是分配内存的时间不同，其次是能够调用变量的区域不同。可以这么认为，全局变量是整个程序的静态局部变量。

1. 连续存储的好处是什么，坏处是什么？

我猜好处主要是节省时间成本，节省内存空间。坏处就是对于多进程系统来说，可能会出现很多无法利用的内存碎片？除了连续存储，我目前只能想到用链表来保证逻辑的连续性，链表虽然增加了利用率，但是很明显会多一些不必要的内存和指令，又降低了实际的利用率和运行速度。

最后是总结：这次的研究做了很久，我认为这次研究的主要成果

是，前面的问题让我复习了一些C语言知识，最后一

个问题让我对分析程序的方法有了一点感悟，就是从

数据角度入手，追踪数据的流向和改变，可以较为轻

松的理解程序。