**c语言综合研究1-11核心机制总结**

变量：

变量是一种存放数据方式，与常量不同，它的内容是可以改变的。它可以分为全局变量和局部变量，它们的本质区别是存储的位置不同，全局变量是在内存中存储的，而局部变量是在栈段中存储的，这个差别导致了它们的一系列区别：全局变量存储的内存空间是没有内存对齐的情况的，而局部变量有；全局变量作为参数传递是直接用地址调用，而局部变量是入栈的方式；全局变量的生命周期是整个程序，而局部变量的生命周期是当前函数；全局变量的段地址在ds寄存器里，局部变量的段地址在ss寄存器里；全局变量定义是自动清零的，而局部变量定义时在栈中的空间还是原来的数据。比较特别的静态局部变量的存储位置和生命周期都和全局变量一样，只是静态局部变量只能在定义的函数中使用。

比较重要的变量类型有char、int、long、double和结构体，它们分别占的大小为1字节、2字节、4字节、8字节，结构体的大小是结构体中数据项之和。结构体也存在内存对齐的情况，结构体中各数据项存储位置是相邻的。结构体作为参数传递和返回比一般变量要复杂，一般变量都是直接入栈，而结构体必须创建一个临时变量，用块搬移函数将结构体的各数据项复制到临时变量里，在子函数里再将临时变量的值搬移到栈段里面，返回的原理也是相同的。

数组：

数组是利用一段连续的内存空间存放一系列相同类型的数据。一维数组是存储的数据按照线性的顺序来排列，二维数组是存储的数据按照类似围棋棋盘的顺序来排列，多维数组是存储的数据以多维的形式存储，我们可以通过当前一组不断向下查找到某一个元素。数组也可以根据存放的元素类型不同来分类：整型数组的元素是int型数据、指针型数组的元素是指针型数据、结构体数组的元素是结构体数据。数组中的元素是连续存放的。数组名相当于数组的首地址，也是数组第一个元素的地址，它的使用和指针有相似之处，如果p是一个指针，那么p[n]等同于\*（p+n），即跳到下一个元素就相当于在当前地址上加上数组元素的类型大小。数组还有函数指针数组，存放的元素是函数指针，指向函数的指针，函数指针数组可以将要运行的程序以数据的形式写入并对函数进行调用。

函数：

函数是一段指令的集合。函数名相当于一个函数指针，存储函数的入口地址，程序由这个入口地址跳转到当前函数。函数的参数是局部变量，在调用该函数的函数中将参数压入栈中，在子函数里用bp寄存器找到参数的地址进行调用。函数可以有返回值，void函数没有返回值，函数的返回值一般是存储在寄存器中，如果返回值为结构体，则将结构体的内容传递到一个临时变量里。函数是一段数据，它同样存储在内存空间里，这样我们可以以数据的形式将一个函数写到内存中执行。

指针：

指针存储的数据是一个地址，我们可以通过“\*”来取得指针存储的这个地址处的内容，通过“&”来取得一个内存空间的地址赋给指针。指针加减一个数并不是以它的值加减一个数字，而是加减它所指向的存储空间的数据类型的大小，即如果它指向的是int型数据，那么加1就是在当前地址上加上2个字节。我们可以将一个地址赋给一个整形变量，但我们不能对一个整形变量使用“\*”取得它所存储的地址处的值，因为它不是一个指针，同时如果一个指针是一级指针，即定义成\*p，那么只能用“\*”对它取一次值，如果一个指针是二级指针，可以用“\*”对它取两次值，总之，一个指针是几级指针，就可以对它取几次值。对于指针的使用我们一定要注意它和其他的变量的类型匹配问题，近指针占2个字节，存储偏移地址，远指针占4个字节，存储段地址加偏移地址。虽然指针的值的大小是固定的，但是指针指向的值的大小和指针的定义有关。指针在地址和内容之间建立了一条联系，这种联系是c语言最重要的基础，我们可以用它来实现多种数据结构。指针可以指向任意的数据类型，结构体指针指向的是一个结构体，它可以以->符号调用结构体的数据项。函数指针是指向一个函数入口的指针，当定义一个函数指针时要指明函数的类型和参数类型和个数，通过函数指针可以调用指定位置的函数。