§5.动态数据区与静态数据区的研究与思考

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 测试项 | VS2017 x86 | VS2017 x64 | CodeBlocks  (32bit) | Dev C++  (32bit) | Dev C++  (64bit) | Linux C++  (64bit) | 思考与心得体会 |
| 动态数据区大小（精确到KB） | 1000K | 995K | 1M 1002K | 1M 1002K | 1M 995K | 15M 991K | Linux的动态数据区最大，vs2017的最小 |
| 极限定义下动态数据区的上限  （起始分配地址） | 0x00EE5A4C | 0x00000013DF8D63D0 | 0x4b5300 | 0x4e52a0 | 0x506e50 | 0x7ffed65e7880 | X64（64bit）的地址为16位，x86（32bit）为8位，dev是例外 |
| 极限定义下动态数据区的下限  （结束分配地址） | 0x00FDFE4B | 0x00000013DF9CF3CF | 0x6afeff | 0x6dfe9f | 0x6ffe4f | 0x7ffed75df87f |  |
| 静态数据区大小（精确到KB） | 1G760M 131K | 1G782M 1023K | 1G782M 1023K | 1G782M 1023K | 1G782M 1023K | 2G244M 1023K | 静态数据区普遍远大于动态数据区，同样linux的最大 |
| 极限定义下动态数据区的上限  （起始分配地址） | 0x00D7A138 | 0x00007FF75AD8C160 | 0x00489080 | 0x00491080 | 0x004a7080 | 0x6011c0 | 数组内，无论动态或者静态，其地址都是由高到低 |
| 极限定义下动态数据区的下限  （结束分配地址） | 0x71C7A137 | 0x00007FF7CBC8C15F | 0x7138907f | 0x7139107f | 0x713a707f | 0x8fb011bf |  |
| 如果动态数据区和静态数据区都极限定义，两者地址最近相差多少？ | 0x2459275 | 0x7f2adc37ccb1 | 0x172c281 | 0x1154221 | 0x5b5fdd1 | 0x7fff2ecd81f1 |  |
| 动态数据区，char x,y,z; 观察xyz间的地址间隔 | &x-&y=12  &y-&z=12 | &y-&x=32  &z-&y=32 | &x-&y=1  &y-&z=1 | &x-&y=1  &y-&z=1 | &x-&y=1  &y-&z=1 | &x-&y=1  &y-&z=1 | Vs两种模式相同，其他三编译器此项相同 |
| 动态数据区，int x,y,z; 观察xyz间的地址间隔 | &x-&y=12  &y-&z=12 | &y-&x=32  &z-&y=32 | &x-&y=4  &y-&z=4 | &x-&y=4  &y-&z=4 | &x-&y=4  &y-&z=4 | &x-&y=4  &y-&z=4 | 动态数据区变量的存储地址由高到低 |
| 动态数据区，char x; int y; char z; 观察xyz间的地址间隔 | &x-&y=15  &y-&z=9 | &y-&x=32  &z-&y=32 | &x-&y=7  &y-&z=1 | &x-&y=7  &y-&z=1 | &x-&y=7  &y-&z=1 | &x-&y=7  &y-&z=1 |  |
| 动态数据区，int x; double y; int z; 观察xyz间的地址间隔 | &x-&y=16  &y-&z=12 | &y-&x=36  &z-&y=28 | &x-&y=12  &y-&z=4 | &x-&y=12  &y-&z=4 | &x-&y=12  &y-&z=4 | &x-&y=8  &y-&z=4 |  |
| 动态数据区，int k, a[10];若要使a[x]就是k的地址，x是几 | X=12 | X=-9 | X=10 | X=10 | X=10 | X=26 | 数组的溢出部分会覆盖其他数据 |
| 静态数据区，char x,y,z; 观察xyz间的地址间隔 | &y-&x=1  &z-&y=1 | &y-&x=1  &z-&y=1 | &y-&x=1  &z-&y=1 | &y-&x=1  &z-&y=1 | &y-&x=1  &z-&y=1 | &y-&x=1  &z-&y=1 | 静态数据区之中存储的变量地址紧邻的 |
| 静态数据区，int x,y,z; 观察xyz间的地址间隔 | &y-&x=4  &z-&y=4 | &y-&x=4  &z-&y=4 | &y-&x=4  &z-&y=4 | &y-&x=4  &z-&y=4 | &y-&x=4  &z-&y=4 | &y-&x=4  &z-&y=4 | 静态数据区变量的存储地址由低到高 |
| 静态数据区，char x; int y; char z; 观察xyz间的地址间隔 | &y-&x=4  &y-&z=3 | &y-&x=4  &z-&x=1 | &y-&x=3  &z-&x=1 | &y-&x=3  &z-&y=4 | &y-&x=3  &z-&y=4 | &y-&x=4  &z-&y=4 | 静态数据区中相同类型的变量会存储在相邻的区域 |
| 静态数据区，int x; double y; int z; 观察xyz间的地址间隔 | &y-&x=8  &y-&z=4 | &y-&x=8  &z-&x=3 | &y-&x=4  &z-&y=8 | &y-&x=4  &z-&y=2 | &y-&x=4  &z-&y=2 | &y-&x=4  &z-&y=8 |  |
| 静态数据区，int a[10], k;若要使a[x]就是k的地址，x是几 | X=10 | X=-2 | X=-7 | X=-7 | X=-7 | X=13 | 数组的溢出部分会覆盖其他数据,可能朝高位或者低位溢出 |

* 地址用16进制
* 地址之间的差值用10进制，转换为K/M/G等容易识别的单位即可
* 某些编译器每次执行时具体地址不同，给出某一次的具体地址即可
* 思考与心得体会中写出你的一些认识即可