2021/03/22_x86逆向_第13课_数组

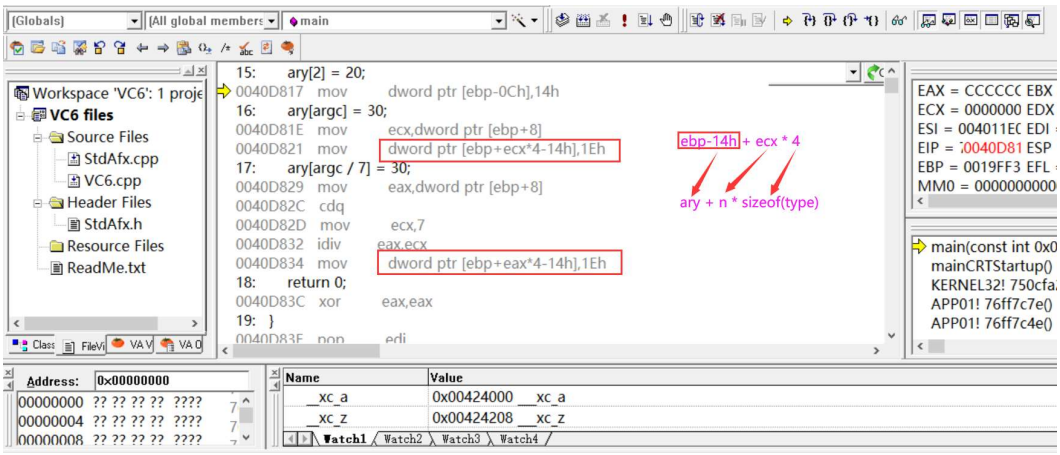
笔记本： x86逆向-C
创建时间： 2021/3/22 星期一 11:05
作者： ileemi

- 数组
- 一维数组
- 二维数组

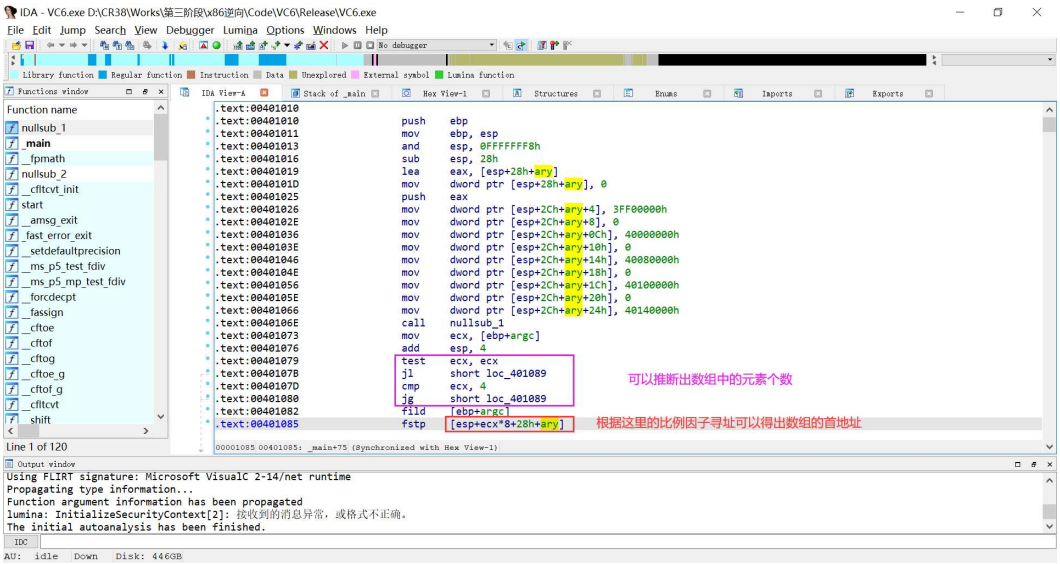
数组

一维数组

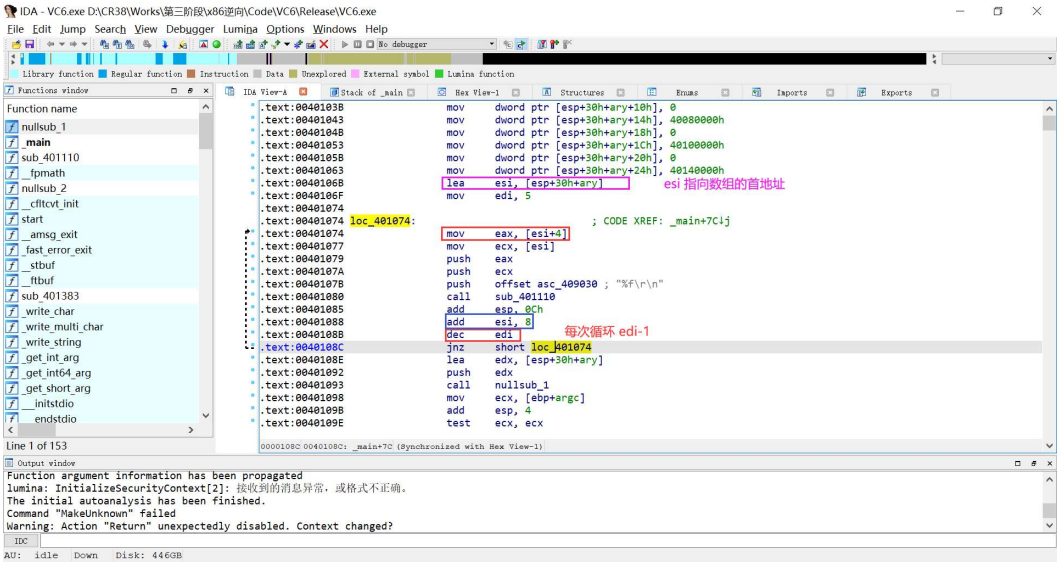
数组的初始化，在汇编代码中定位数组的首地址：



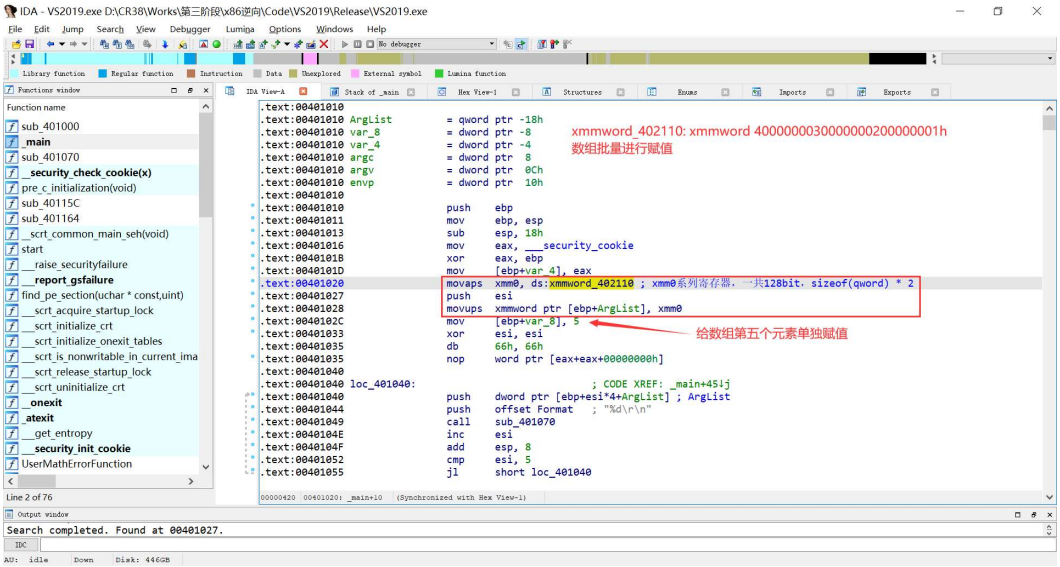
满足连续性（元素在内存中地址）、一致性（类型和逻辑意义一致）的特点就可以还原成数组。证明逻辑意义一致可以通过比例因子寻址（和数组的下标运算紧密相关，在其可能的取值范围内就证明这段内存是连续且一致的）或者循环迭代（对流程结构的识别和算法尤其是乘法的识别）。连续、不一致还原成结构体，不连续，就还原成变量。



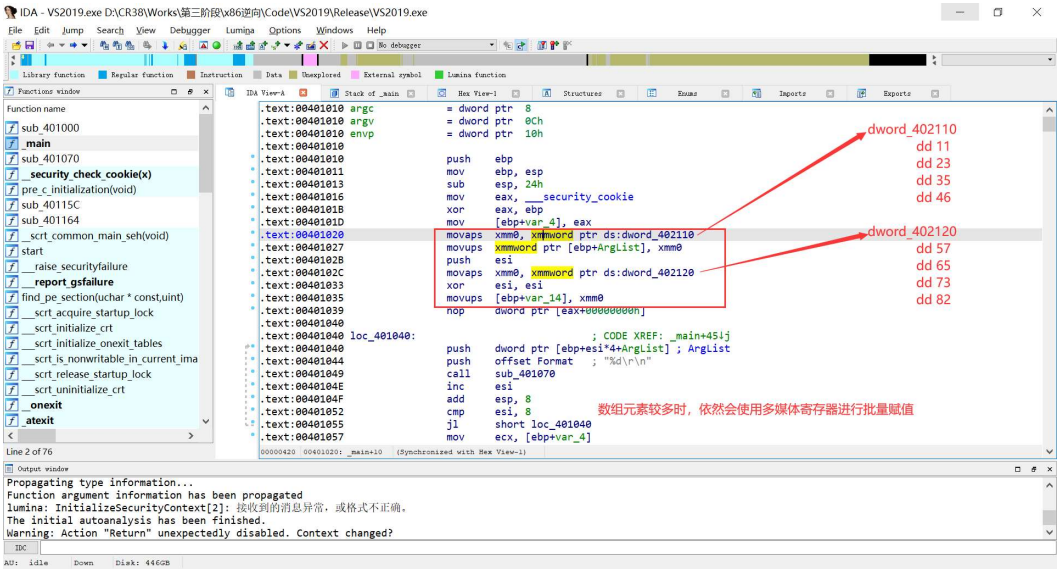
在循环中确定数组：



VS2019在数组初始化时就可以进行确定：



为了使数组的元素快速进行初始化，编译器使用了多媒体寄存器进行批量赋值，而对于连续定义的局部变量，编译器并不会对其进行批量赋值。



> 一个有正常目标需求的程序，如果存在数组，必然有下标运算或者循环处理。使用变量作为数组的下标进行访问时，没有进行范围检查，对应的代码就会存在安全问题。

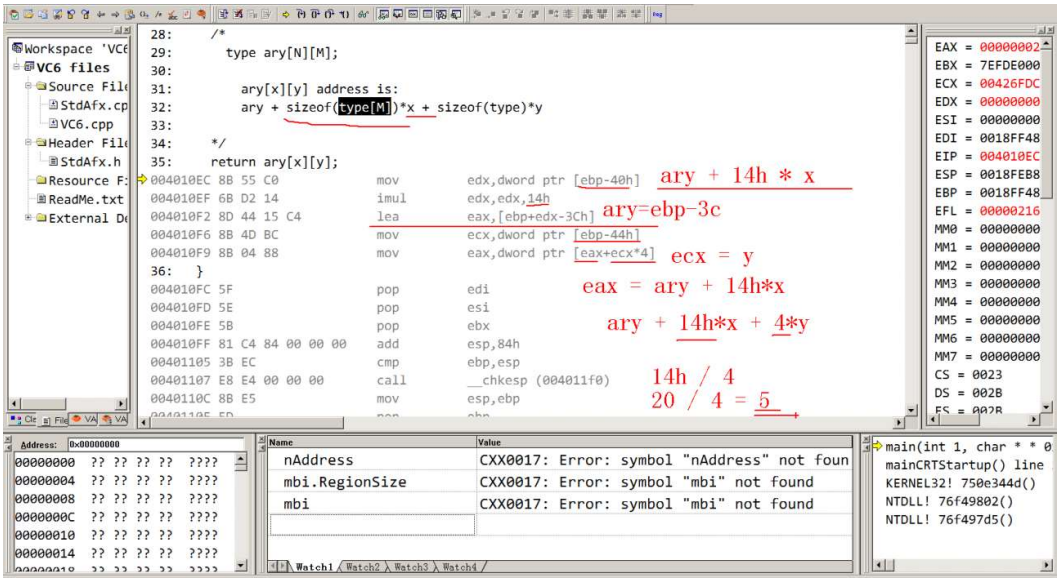
二维数组

> type ary[N][M]
> ary[x][y] addr:
> ary + szieof(type[M]) * x + sizeof(type) * y

代码示例：

```
int main(const int argc, char* argv[])
{
  int ary[3][5] = {
    {1, 2, 3, 4, 5},
    {12, 22, 32, 42, 52},
    {13, 23, 33, 43, 53}
  };
  int x, y;
  scanf("%d%d", &x, &y);
  return ary[x][y];
}
```

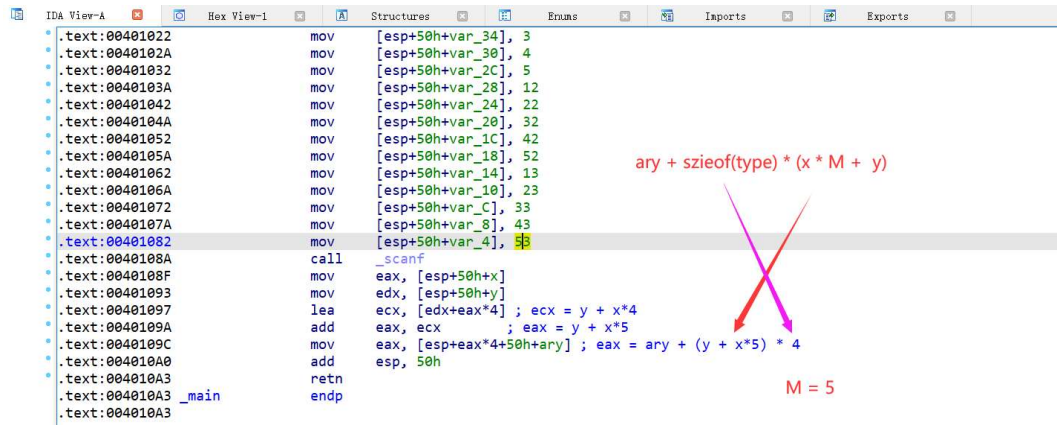
VC++ 6.0 Debug：



多维数组行为更明显，Release版的优化：

> ary + szieof(type[M]) * x + sizeof(type) * y
> ary + szieof(type) * x * M + sizeof(type) * y
> ary + szieof(type) * (x * M + y)

Release：



> VS2019 Release：