

11 How to DEBUG C++ in Visual Studio

Breakpoints and *reading memory*

设断点来读取内存

1. Breakpoints

断点是程序中调试器会**break**(中断)的一个点，当执行到断点所在行时，执行线程会悬停

Visual Studio断点快捷键：F9

Debug(调试)时，要确保自己在“Debug”模式下，配置可见Configuration

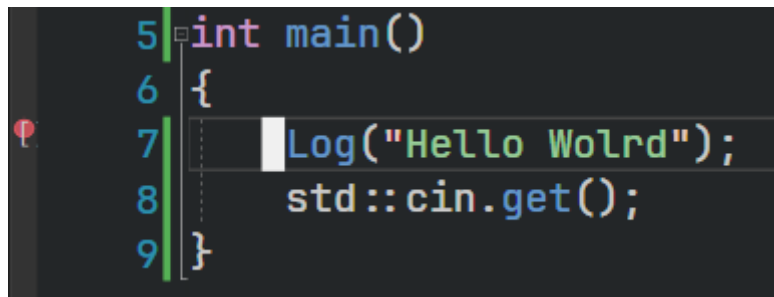
“Release”模式下代码会被更改，你的断点可能永远不会执行

调试

Step Into : F11

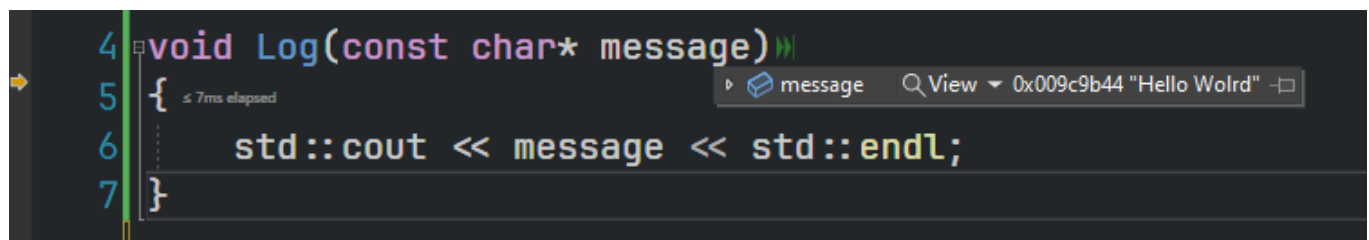
Step Over : F10

Step Out : Shift + F11



```
5 int main()
6 {
7     Log("Hello Wolrd");
8     std::cin.get();
9 }
```

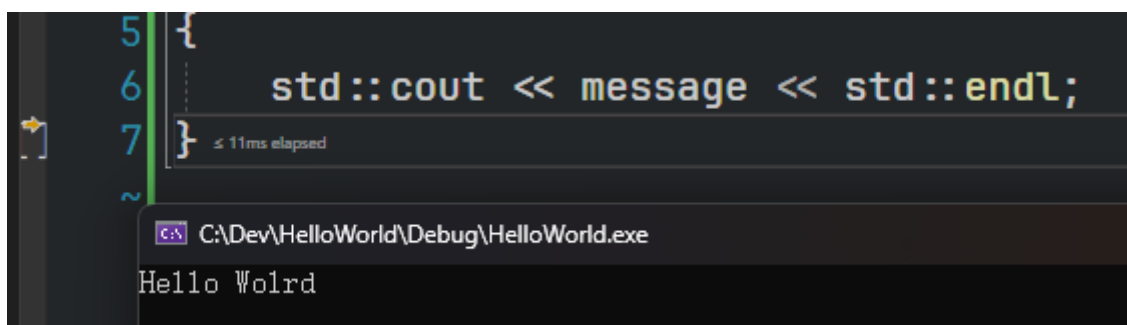
现在对main函数的第七行进行Step into



```
4 void Log(const char* message)
5 {
6     std::cout << message << std::endl;
7 }
```

黄色箭头表明还没执行当前行，按F10箭头到第六行

再次F10后，字符串才被输出



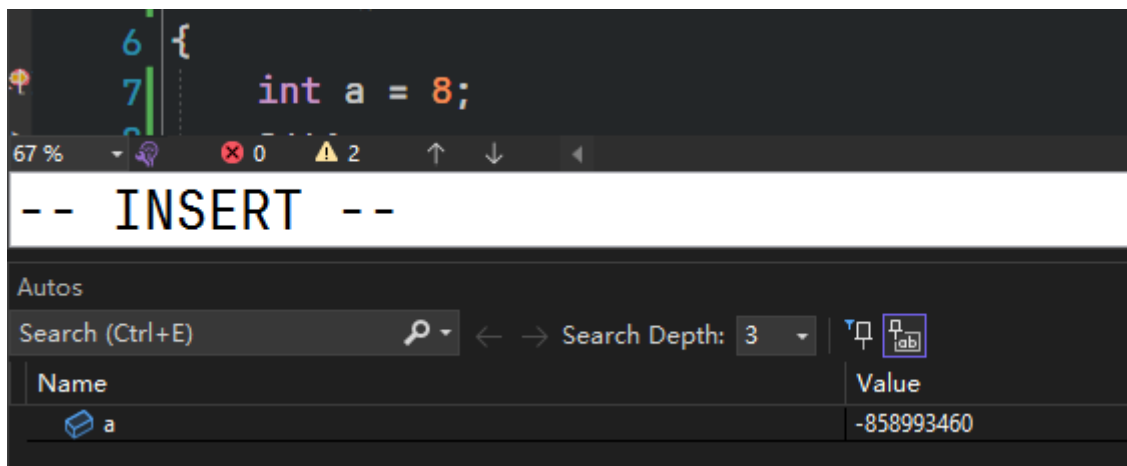
```
5 {
6     std::cout << message << std::endl;
7 }
```

```
C:\Dev\HelloWorld\Debug\HelloWorld.exe
Hello Wolrd
```

看这段程序：

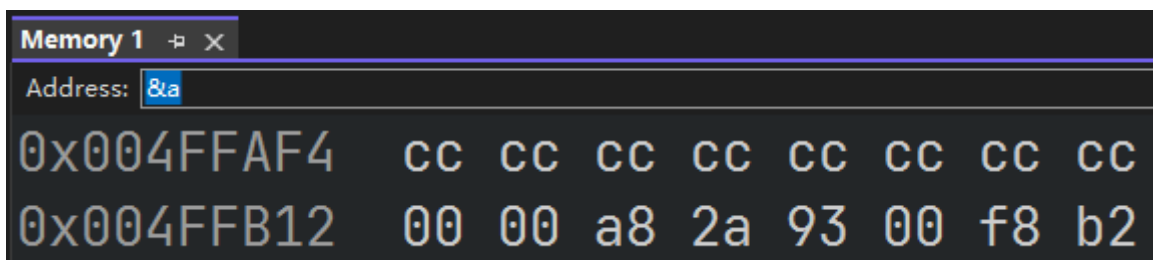
```
C++  
  
int main()  
{  
    int a = 8;    //断点  
    a++;  
    const char* string = "Hello";  
  
    for (int i = 0; i < 5; i++)  
    {  
        const char c = string[i];  
        std::cout << c << std::endl;  
    }  
  
    Log("Hello Wolrd");  
    std::cin.get();  
}
```

此时 `int a=8` 还没有被执行，还是没初始化的内存



Memory View (内存视图)

Debug过程中，上方导航栏Debug→Windows→Memory→Memory1即可调出内存视图

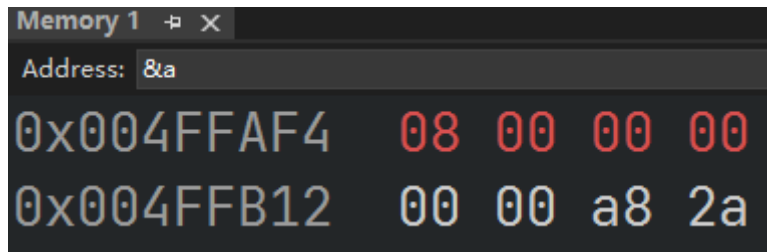


输入 `&a` 即可访问变量 `a` 的内存地址，其中都是16进制数。

为什么都是cc?

说明这是一个未初始化的堆栈内存

按下F10



可以看到内存中存储的数据发生改变。其中每两个十六进制数字代表一个字节。

如果我想跳出循环，按Step Out会跳出整个函数。所以只需要再设一个断点，再按F5即可运行到断点处。

