什么是万能按键call

当我们在游戏中敲击键盘的按键时，游戏会对键盘事件会做出响应，去执行某些功能。

以口袋西游为例，按B键会打开人物背包，通过B键打开背包的这个call就是万能按键call。

只要找到这一个call，就相当于找到了所有的按键call，因为剩下的所有的快捷键功能都会走这一个call。而且这个call实用性也比较强，可以用来替代很多的游戏功能。

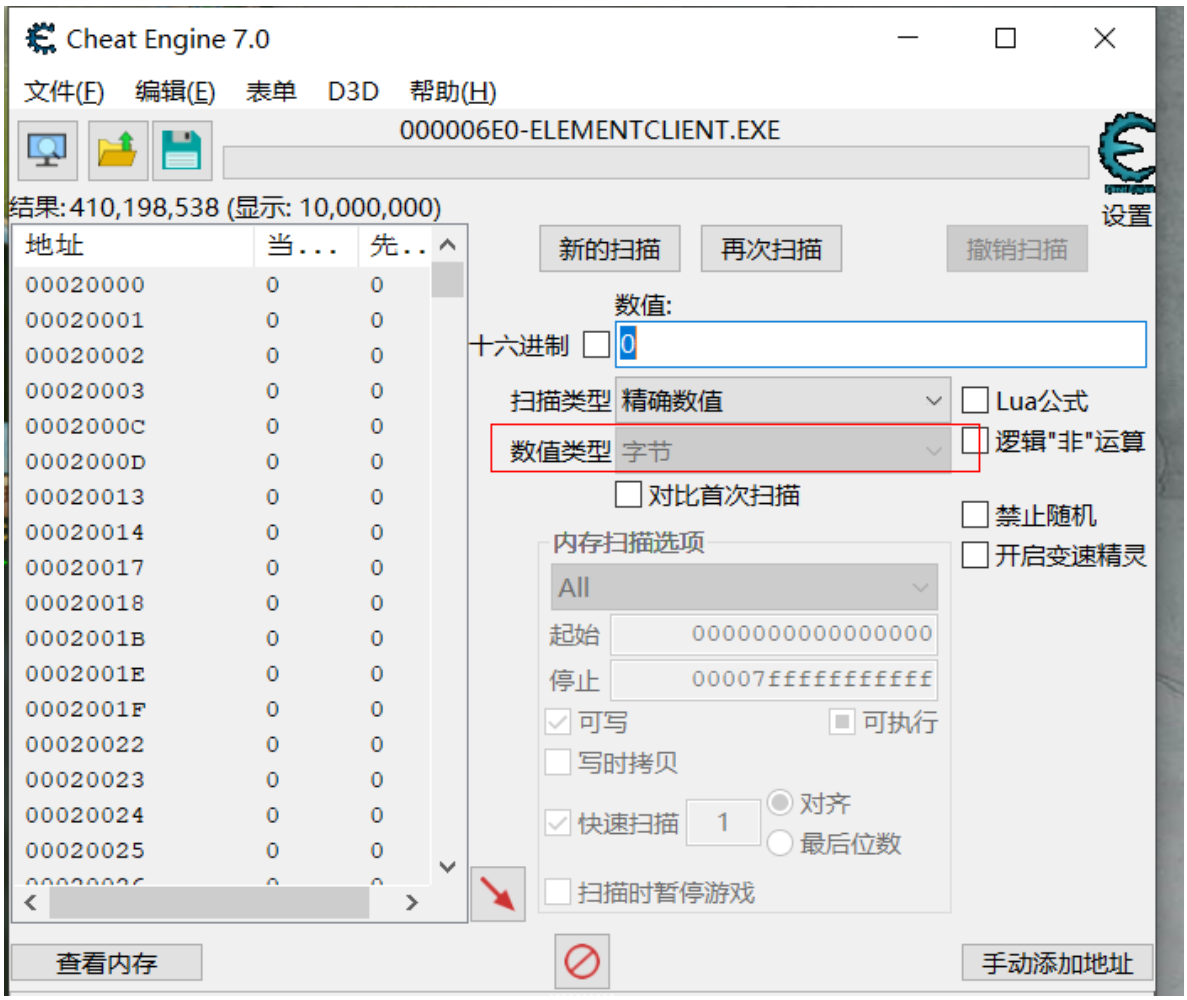
切入点

当我们点击B键时，背包会被打开，再次点击，背包会被关闭。

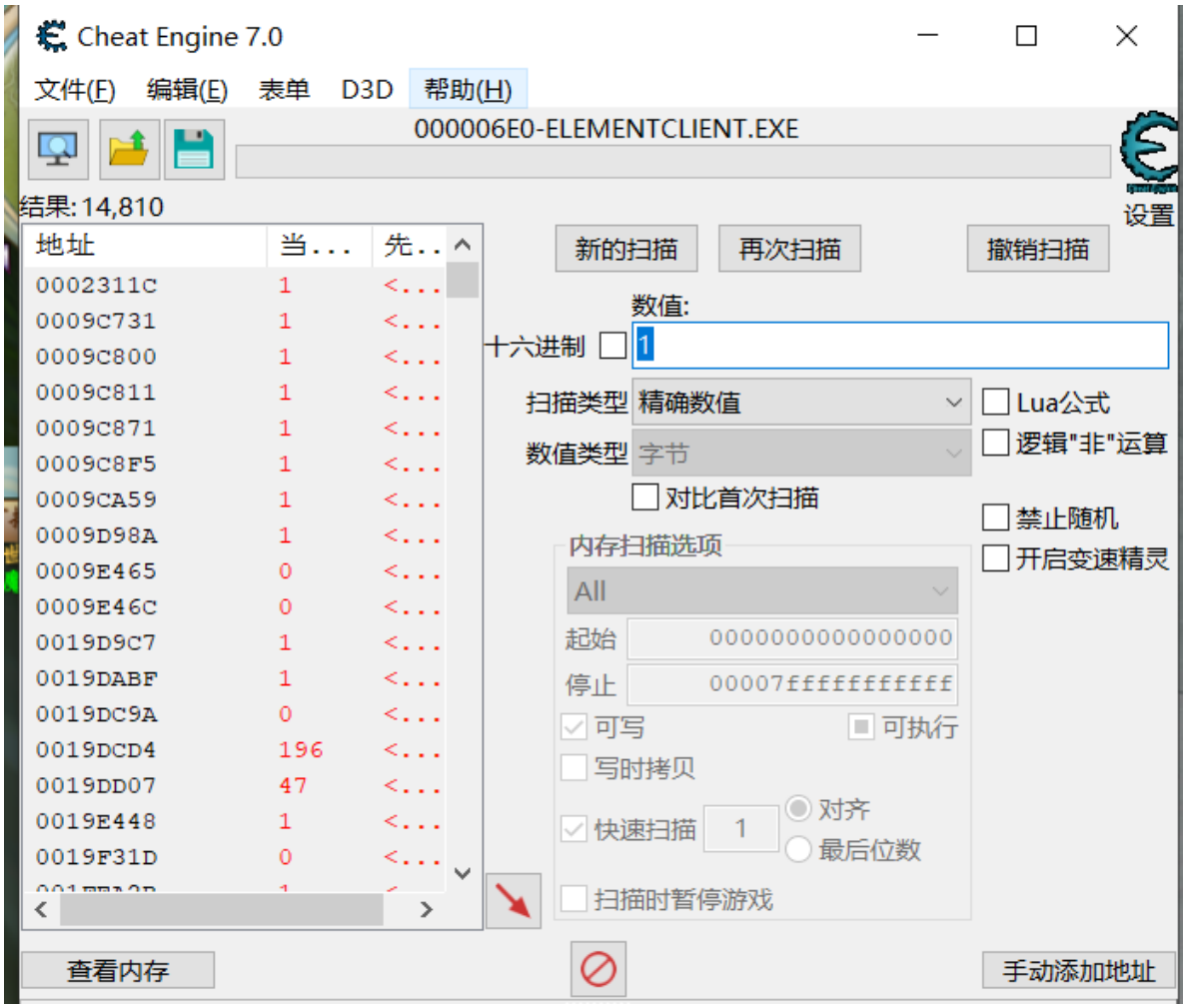
那我们就可以先用CE搜索到背包打开和关闭的状态，之后再对这个状态下一个写入断点，通过栈回溯的方式就能找到这个万能按键call。

CE搜索背包状态

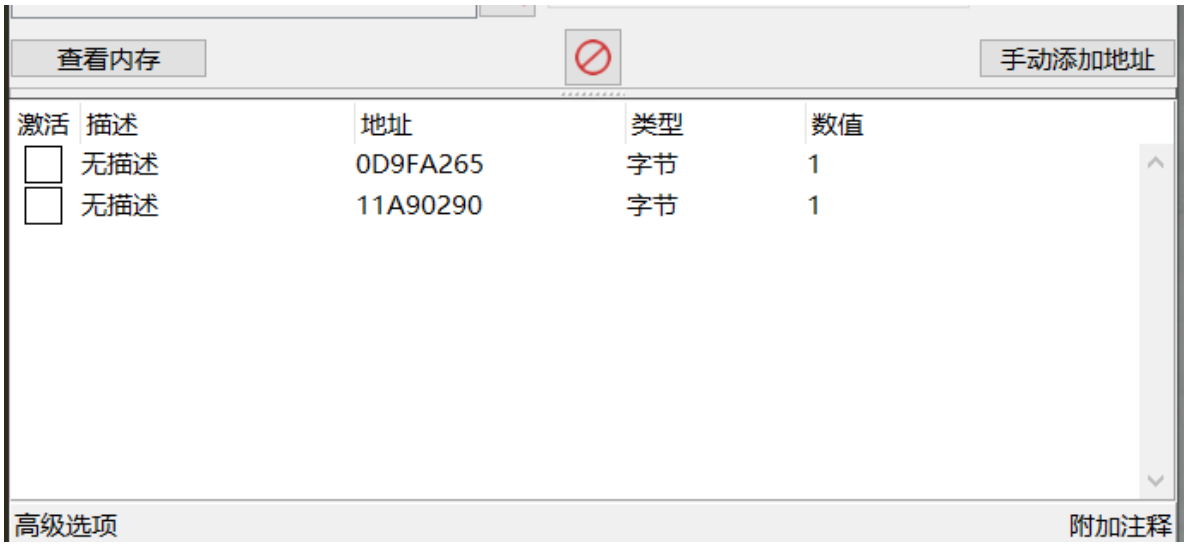
背包打开和关闭的状态是一个标志位，可能是0和1，也可能不是。这里只能进行尝试，如果0和1没找到就需要利用未知的初始值和变化的值来找到这个标志位。



在背包打开状态下扫描1，**注意这里需要用1字节进行扫描，4字节是扫不出来的**



关闭状态下扫描0



重复这个过程(没啥技术含量)，直到筛选出唯一一个背包状态标志位，我这里过滤剩下两个。

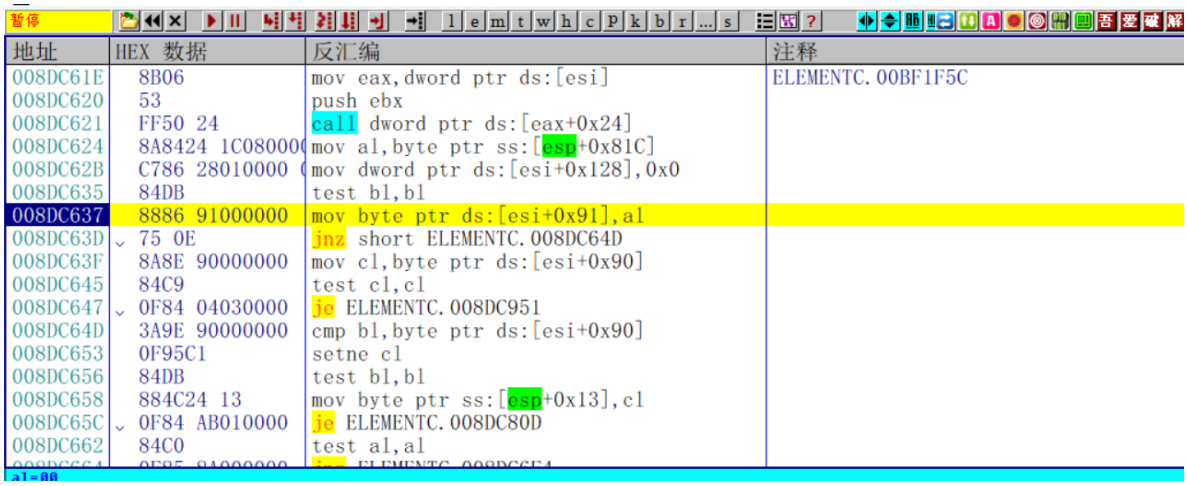
通过内存关系找万能按键call

接着我们在这个地址下**一个字节的硬件写入断点**，这里有两个要注意的点：

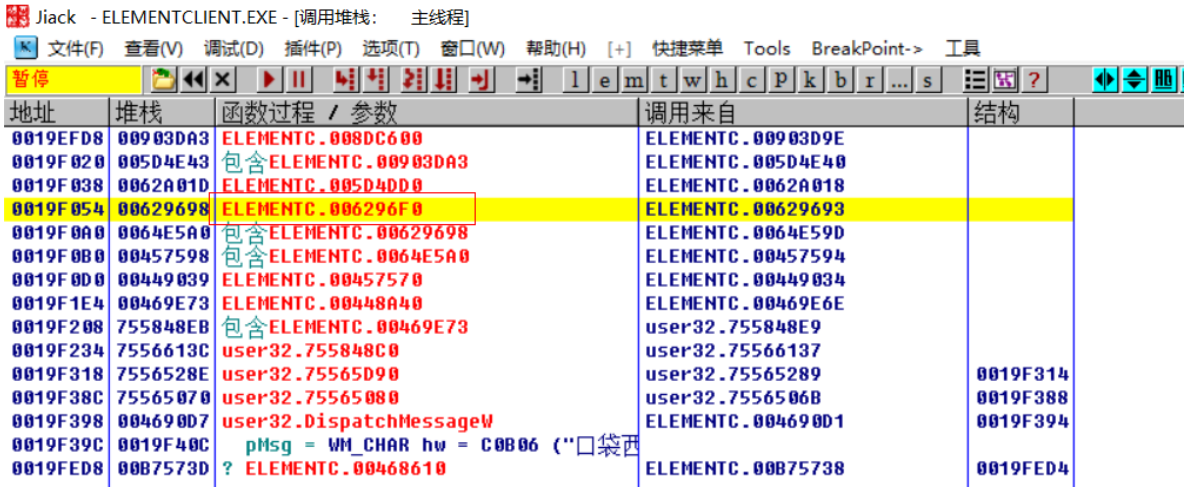
第一个点是吾爱的OD硬件断点有点问题，建议换个OD；

第二个点如果下了硬件断点之后这个地方断的很频繁，而且数值的变化范围超出了0和1说明标志位找错了。

然后点击B键，让游戏断下



删除硬件断点，打开调用堆栈



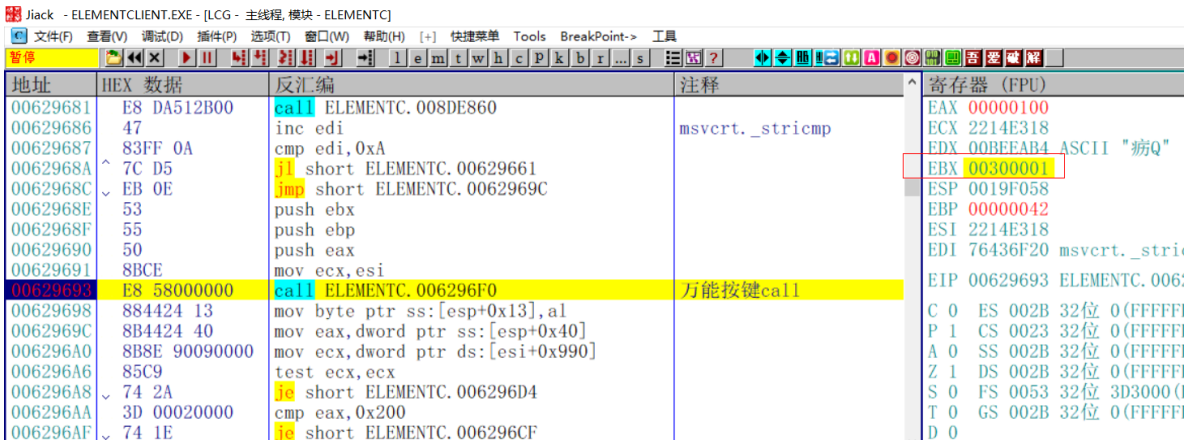
挨个下断点分析参数排查，6296F0这个call就是我们要的万能按键call

万能按键call参数分析

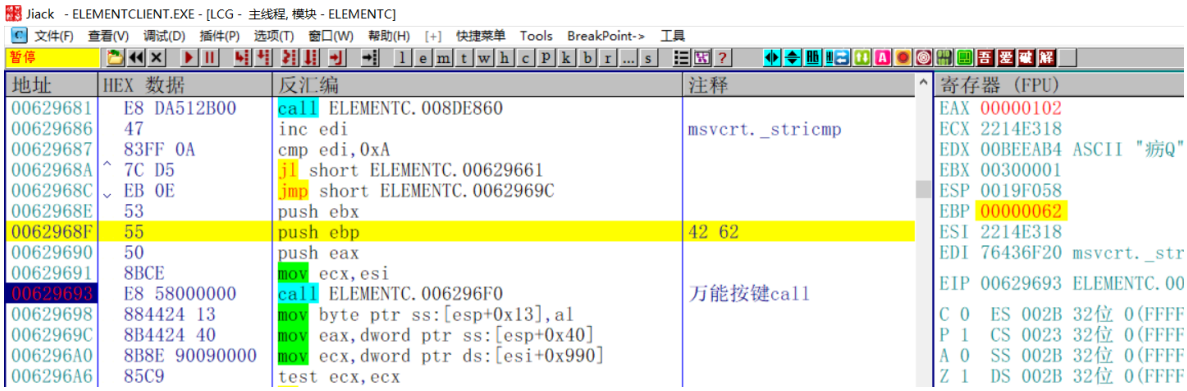
这个call和我们之前找的call有一定的区别。正常找的call一般是调用一次，而这个地方会断下来两次。

原因在于这个call是用来响应处理按下键盘事件的，按下键盘事件实际上分为两个动作，一个是按键按下，另一个是按键抬起，对应Windows的键盘事件就是WM_KeyDown和WM_KeyUp。

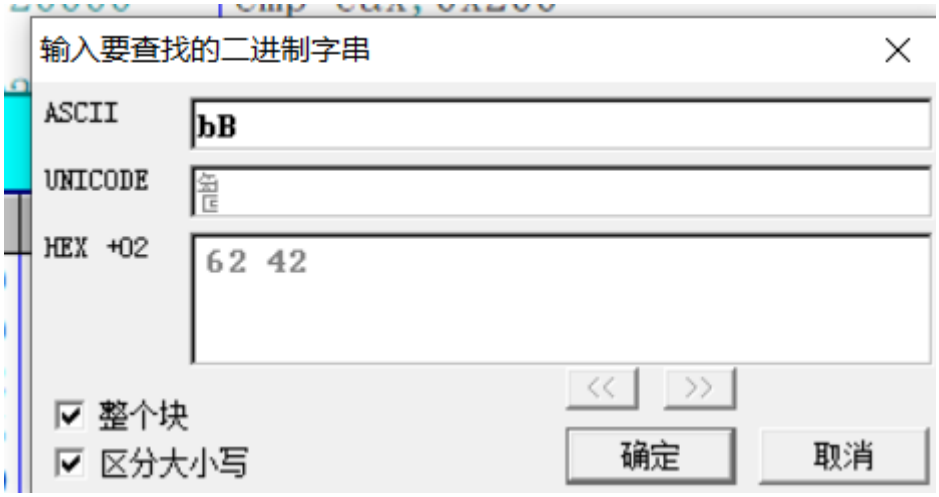
接下来分析这个call的参数



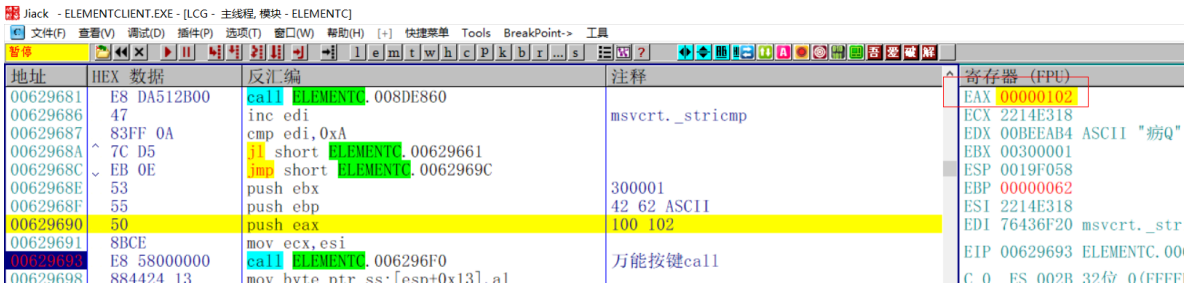
ebx是300001，无论是按下还是抬起都是一样的，具体含义暂时未知



ebp在第一次断下的时候是0x42，第二次断下的时候是62



正好是对应按键B的ASCII值，大写的B表示按键被按下，小写的b表示键盘抬起



再看eax，按下的时候eax等于100，，抬起的时候eax等于102

这个参数对应的是当前的键盘状态是按下还是抬起，区别在于把KeyUp改成了102。

而ecx是一个数据结构，需要往上追ecx的来源，直到把基址追出来。这里就不一步一步追数据，本次的重点是找call，各位有兴趣可以自己去追一下ecx数据的基址。

总结

在api断点不方便逆向的时候，可以通过内存的访问关系下断，然后利用栈回溯来找到我们需要的call。

另外这个万能按键call，如果是在实际写代码应用的时候，需要写两个call，一次是按下，一次是抬起，在按下和抬起的中间还应该加一个延迟，尽量去模拟真实的键盘事件。

相关工具：

https://github.com/TonyChen56/GameReverseNote