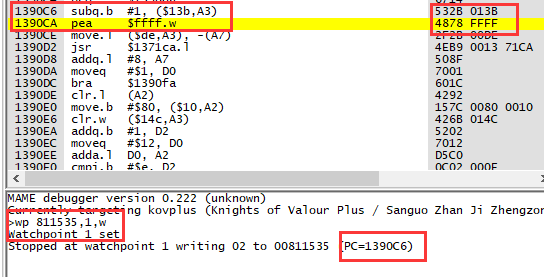
\*注：pea/jsr/bsr等指令会改写堆栈寄存器A7，故最好不要使用包含此类语句的指令于跳转

Hook跳转的原理是是将原代码跳转到另一个空地址块，

后面可以任意写多少代码，执行完后跳转回去

例子是使用技能不消耗气，反而每次加2个气



对1p的气地址811535写入断点，并使用1个气

会断在1390CA，而上方的subq才是导致1p气减1的语句

因为hook跳转需要至少6个字节，

故将这两句的字节码532B 013B 4878 FFFF 记录下，后面称为原代码

Hook跳转有两种方式

4EB9 XXXX XXXX // jsr 地址

....

4E75 // rts

和 ==============

4EF9 XXXX XXXX // jmp 地址

....

4EF9 XXXX XXXX // jmp 地址

一般用前者，jsr跳转时就已经记录下一条语句的地址，当执行到rts会返回到记录的下一条语句地址继续执行，但这里用后者，因为pea会改变jsr记录的地址

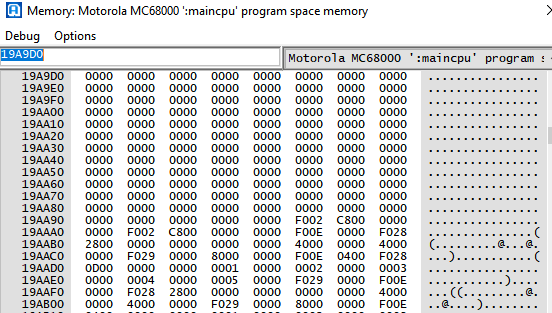
后面的XX XX XX XX代表要跳转到的地址，一般而言需要一块比较大的空代码块，即全都是00 00 00 00 .... 或是自己找一块无用代码区域

这里使用find命令查找空代码块，100000是起始地址，后面的fffff代表查找范围为100000-1fffff，q为数据类型，q.0即00 00 00 00 00 00 00 00

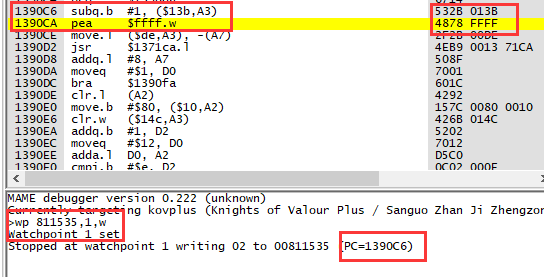
find 100000,fffff,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0,q.0

会搜索到1个地址





不算很大，但举例够用



回到之前的导致气减少代码 (共用代码，同时处理1p-4p)，

可以确定代码为

4EB9 0019 A9D0

后面的0019 A9D0即前面搜索出来的空地址，前面00为补位

即 maincpu.rq@1390C6=4EF90019A9D04E71



后面的4E71为了nop掉无用字节，因为只需3个字节即可跳转



继续执行会跳转到19A9D0

此时就可以在这里写代码

比如把之前的原代码

532B 013B 改成 522B 013B 就是把气减1变成加1

对齐地址，再多写一条，并把原代码4878 FFFF写上，

(一般而言原代码必须放到语句块里执行，不然会死机之类)

最后跳转回去 (下一条语句，即1390CE)，就完成了

maincpu.rd@19A9D0=522B013B

maincpu.rd@19A9D4=522B013B

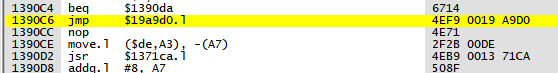
maincpu.rd@19A9D8=4878FFFF

maincpu.rq@19A9DC=4EF9001390CE4E71

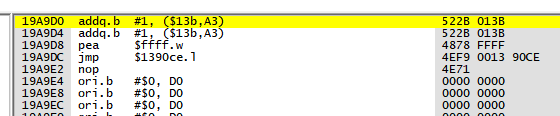
现在回游戏放技能，每次气都会加2而不会减少

代码图：

气减少代码跳转到空代码块



空代码块写自己的代码和原代码



空代码块执行完后跳转到原代码下一句

