南开大学

**CTF Challenge**

**（汇编语言与逆向技术实验 10）**

****

**姓名：申宗尚**

**学号：2213924**

**专业：信息安全**

1. **实验目的**

1.熟悉静态反汇编工具IDA Freeware；

2.掌握对二进制代码内部逻辑关系的分析；

3.掌握对二进制代码的修改和保存。

**二．实验环境**

1. ida
2. game.exe

**三．实验原理**

1. CTF

CTF是一种流行的信息安全竞赛形式，可意译为“夺旗赛”。其大致流程是，参赛团队之间通过进行攻防对抗、程序分析等形式，率先从主办方给出的比赛环境中得到一串具有一定格式的字符串或其他内容，并将其提交给主办方，从而夺得分数。

CTF竞赛模式具体分为以下三类：

一、解题模式（Jeopardy）

在解题模式CTF赛制中，参赛队伍可以通过互联网或者现场网络参与，这种模式的CTF竞赛与ACM编程竞赛、信息学奥赛比较类似，以解决网络安全技术挑战题目的分值和时间来排名，通常用于在线选拔赛。题目主要包含**逆向分析**、漏洞挖掘与利用、Web渗透、密码、取证、隐写、安全编程等类别。

二、攻防模式（Attack-Defense）

在攻防模式CTF赛制中，参赛队伍在网络空间互相进行攻击和防守，挖掘网络服务漏洞并攻击对手服务来得分，修补自身服务漏洞进行防御来避免丢分。

三、混合模式（Mix）

结合了解题模式与攻防模式的CTF[赛制](https://baike.baidu.com/item/%E8%B5%9B%E5%88%B6)，比如参赛队伍通过解题可以获取一些初始分数，然后通过攻防对抗进行得分增减的零和游戏，最终以得分高低分出胜负。

1. 解题

Flag隐藏在game.exe的二进制代码中。通过对game.exe的修改，使game.exe能够顺利的执行，完成对Flag的解密。

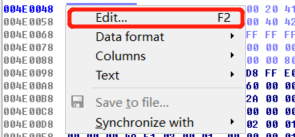


1. 技巧A：利用IDA Pro修改静态资源

* 第一步，在反汇编代码中（IDA View）找到静态资源。

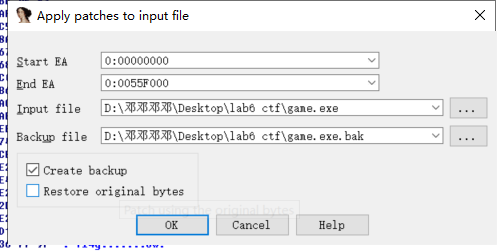


* 第二步，在十六进制视图中（Hex View）找到指定区域，右键选择Edit对资源进行修改。修改完毕后，右键选择Apply changes应用修改。





* 第三步，点击Edit->Patch program->Apply patches to input file，建议选中创建备份的选项，完成修改。

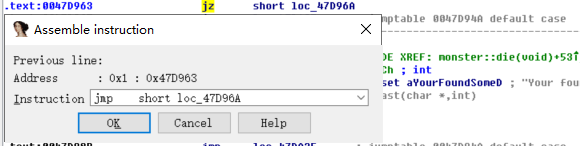


1. 技巧B：利用IDA Pro修改汇编指令

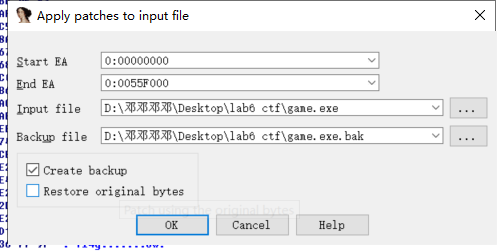
* 第一步，在反汇编代码中（IDA View）找到需要修改的汇编指令。



* 第二步，点击Edit->Patch program->Assemble，输入新的汇编指令。

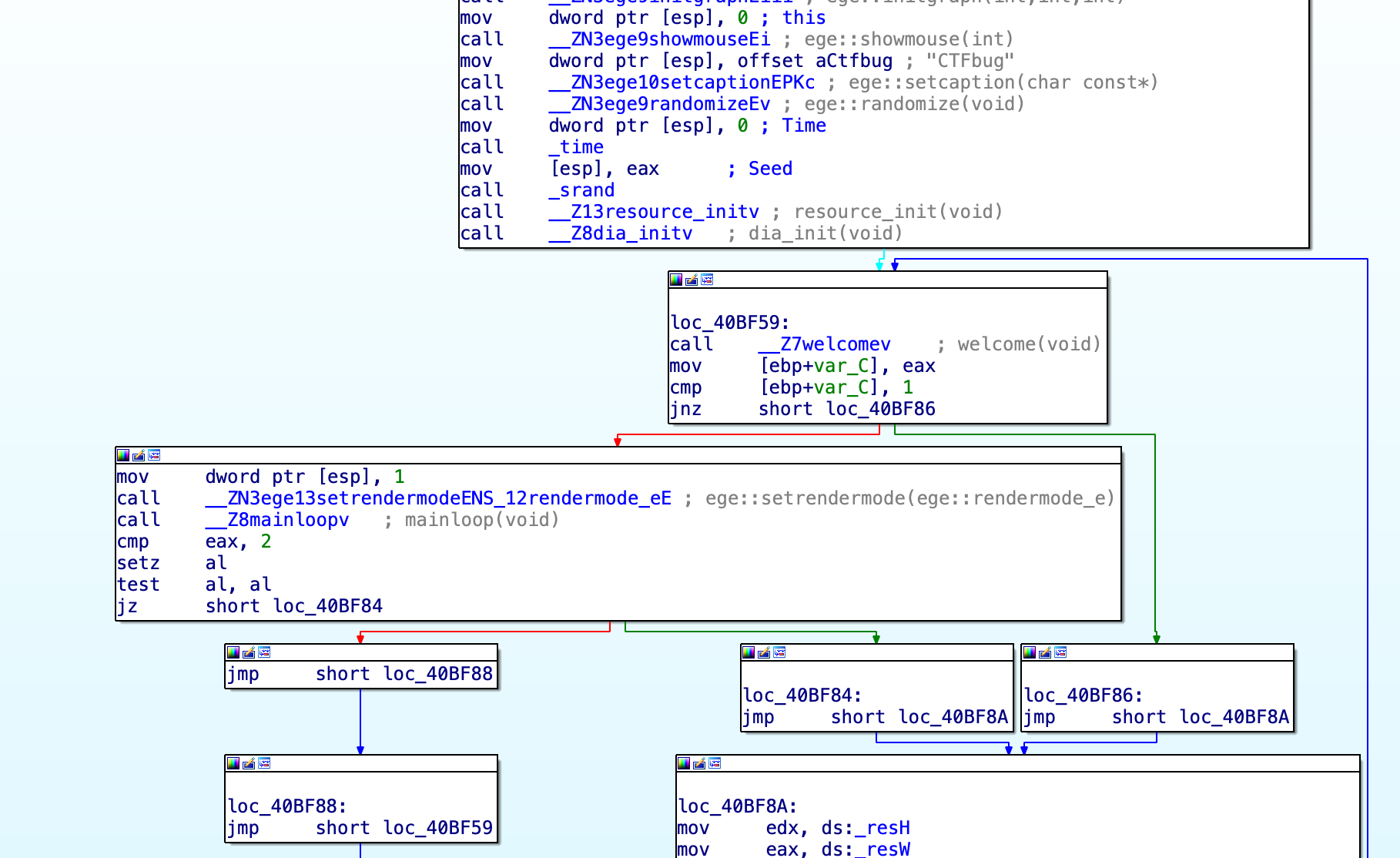


* 第三步，点击Edit->Patch program->Apply patches to input file，建议选中创建备份的选项，完成修改。

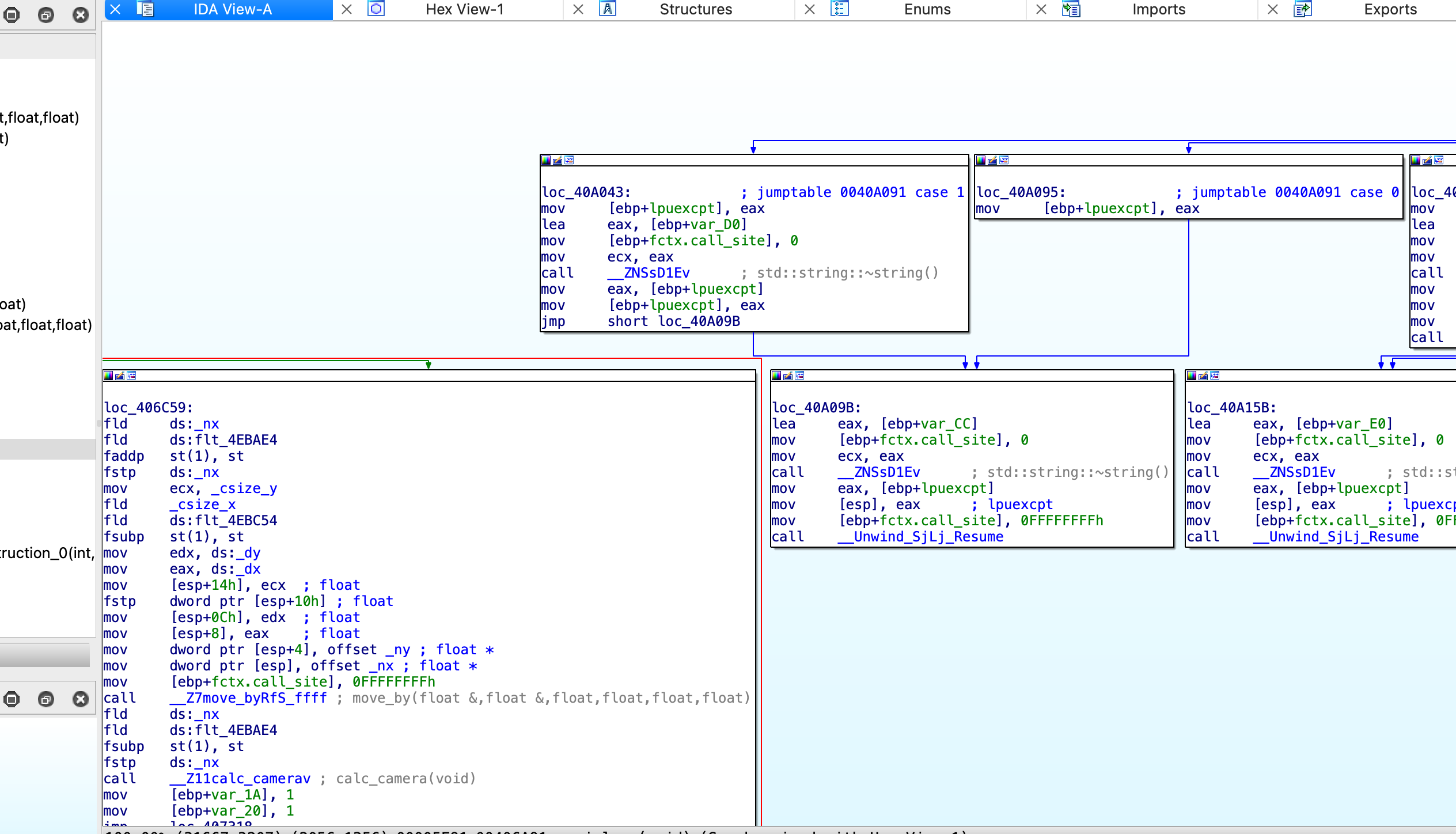


**四.实验内容**

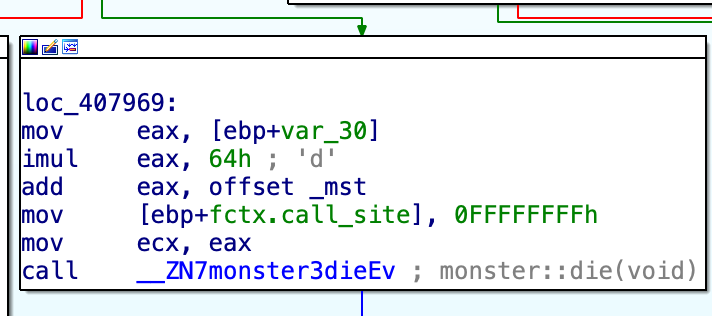
如图，将game.exe导入ida64 Freeware中，可以得到二进制代码的反汇编代码

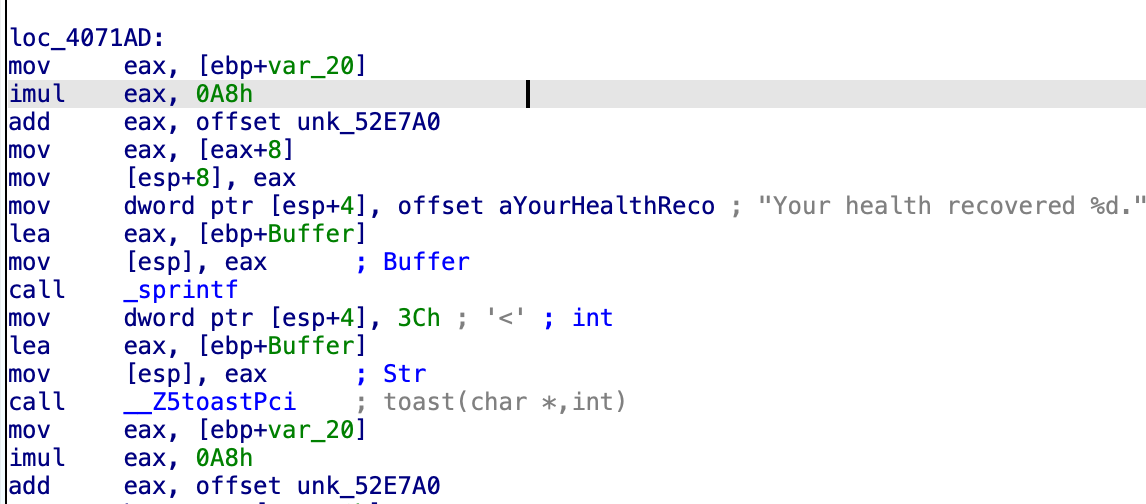
如上图为程序的\_main部分，对其分析可知，该程序一开始使用call指令调用了许多初始化函数，如bgm的配置，初始模式的设置，图形加载等，然后call了\_\_Z8mainloopv部分，由注释和具体分析可知，这部分mainloop函数为游戏的主体，故主要对其进行分析。



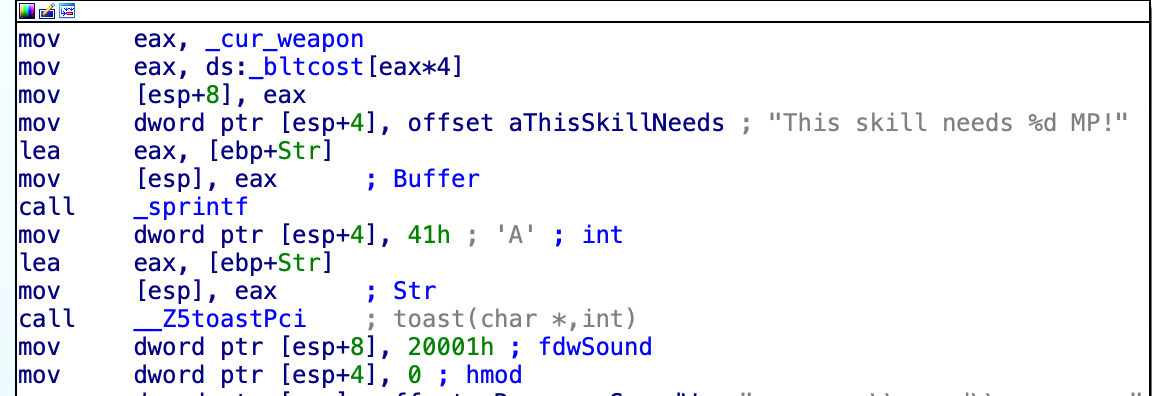
在mainloop函数中，定义了多个小函数，来执行对应的功能与操作。



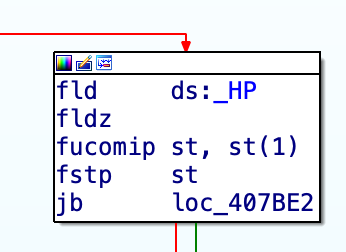
（怪物对象的死亡）



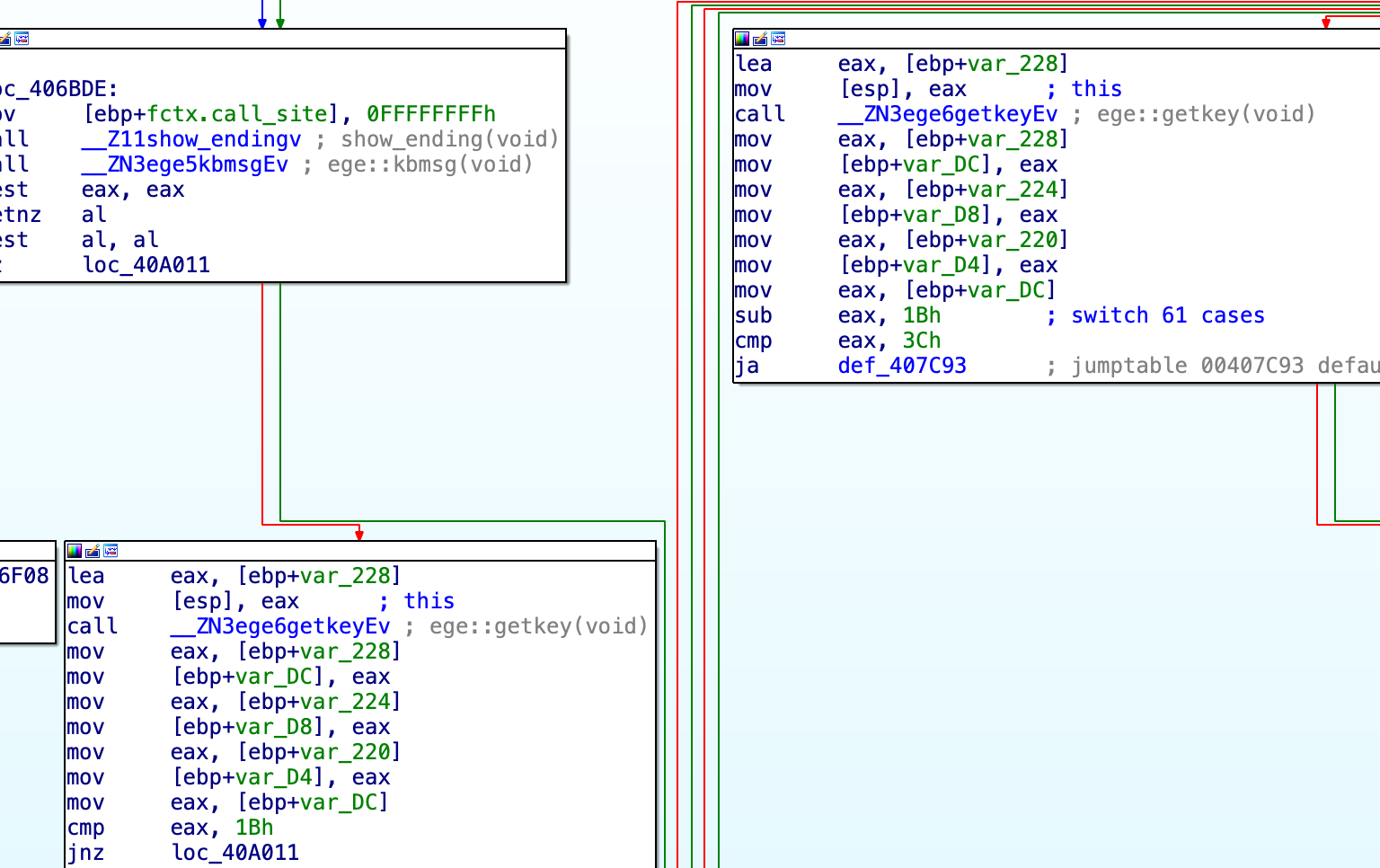
（血量恢复系统）



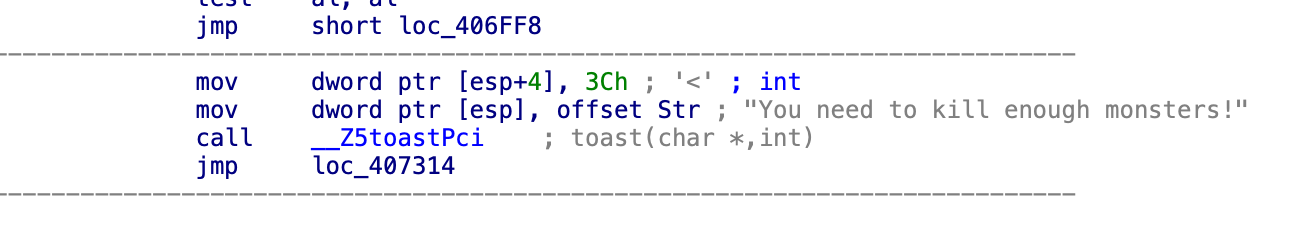
（技能施法系统）



在这部分中，通过将\_HP与0值进行比较，如果HP小于0，则跳转至主人公死亡函数loc\_407BE2，游戏失败结束。



在这部分定义了getkey函数，获取游戏结局时的Flag，通过对其跳转本身上一个模块的寻找，可以找到其触发条件：



可以知道，游戏胜利获得Flag的条件是击杀足够的怪物数。

从而，对以上过程的分析，我们可以知道

1. 开始游戏后，主人公会因为游戏难度过高，HP小于0而死亡，从而导致游戏结束
2. 最终需要击败一定数量的怪物数，从而获得游戏胜利，夺旗

对于（1），我们可以将上面对于\_HP变量的比较段代码进行改写，目前是与0值进行比较，但我们可以将其改为与一个极小的负数进行比较好（如-10000），从而让血量不会更小，也不会通过jb指令跳转至主人公死亡函数。

修改代码如下：

fld ds:\_HP ; 将 \_HP 的值加载到 FPU 栈顶

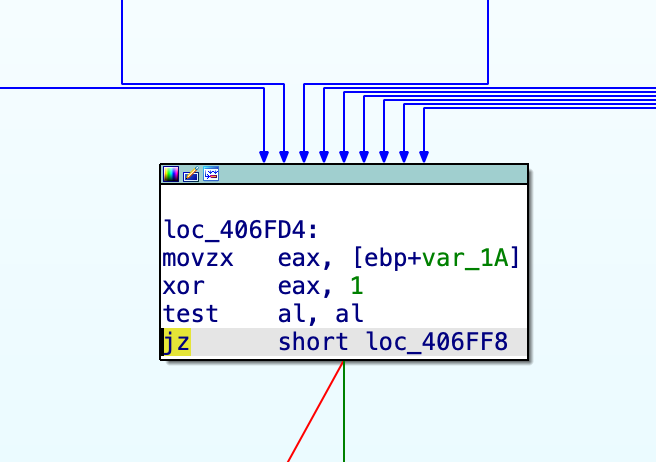
fld qword ptr [ebp-10000.0] ; 将值 -10000.0 加载到 FPU 栈顶

fucomip st, st(1) ; 对比 \_HP 和 -10000.0，结果存储在 EFLAGS 寄存器中

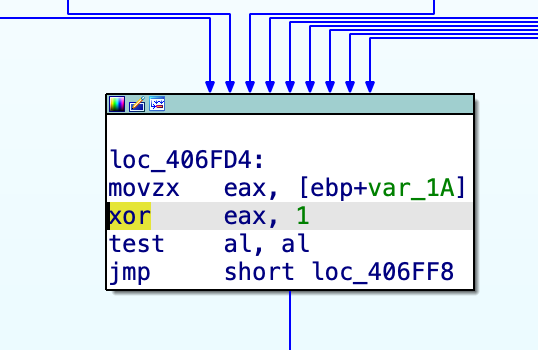
fstp st ; 弹出 FPU 栈顶元素

jb loc\_407BE2 ; 如果\_HP 小于 -10000.0，跳转到 loc\_407BE2 处

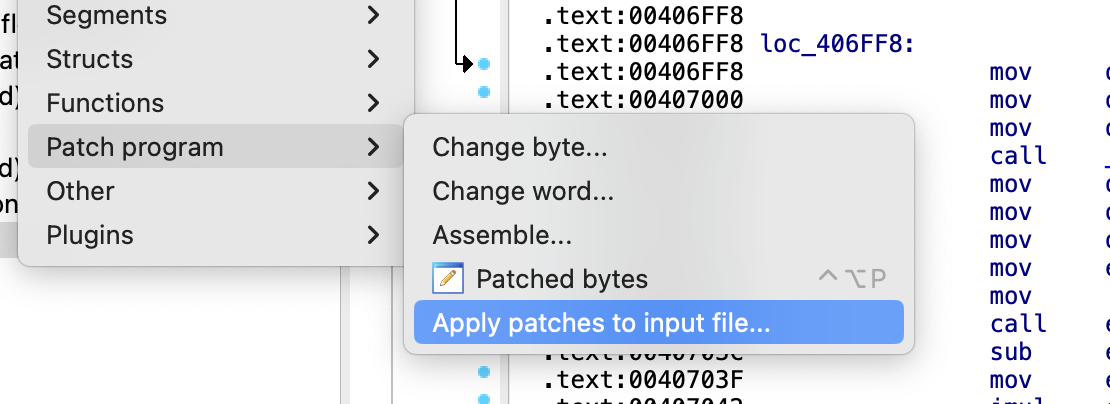
对于（2），对其判断代码块的上一个代码块进行分析，可以得知：



其通过jz指令，条件跳转至成功夺旗函数，所以，我们可以将条件跳转jz改为无条件跳转指令jmp，使击败足够怪物数则获胜变为无条件获胜，从而顺利夺旗。



最后，应用我们所有的修改，生成新的.exe文件





如图 启动游戏后，主角即使被攻击也不会死亡

穿过四个地图，最终可以通关游戏，获得flag{a2fdkd80xo}