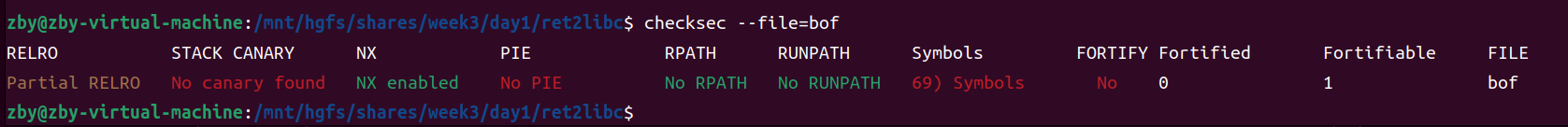
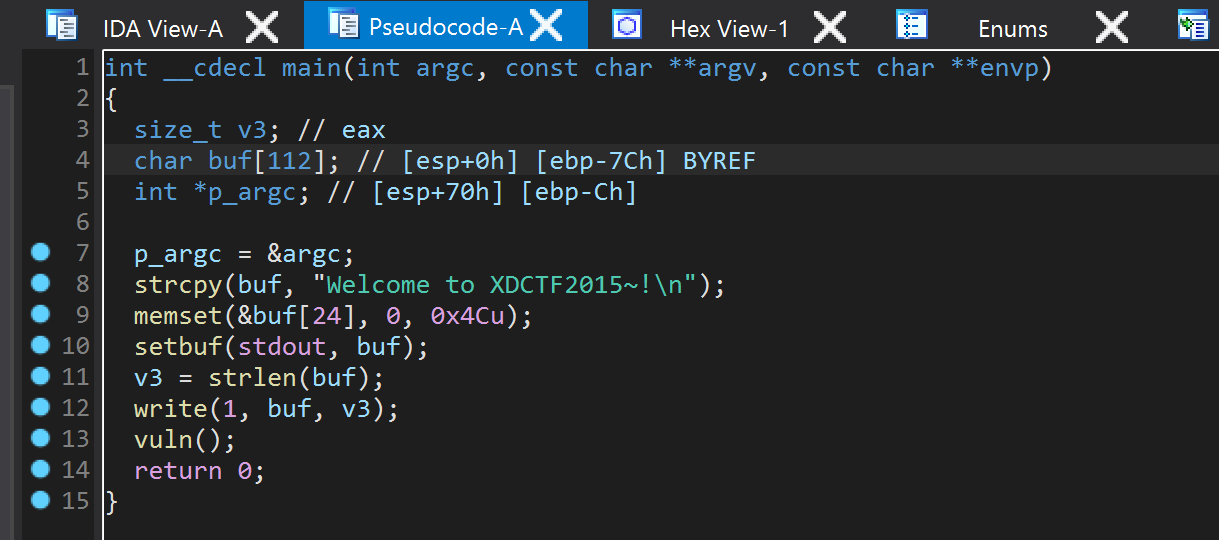
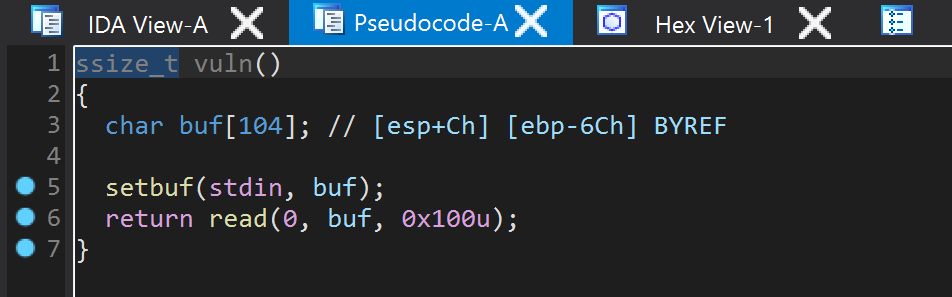
首先查看保护机制，只开启了NX也就是说可以使用栈溢出

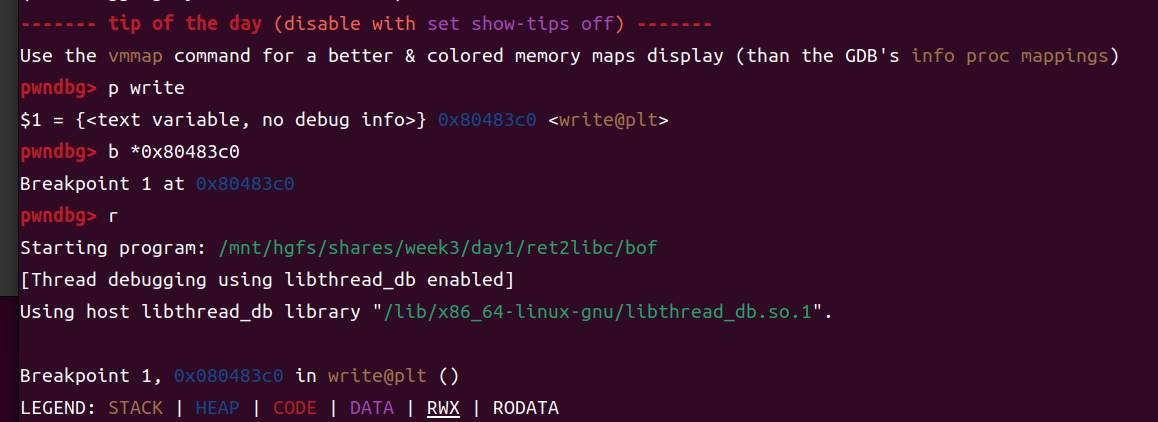


将代码在IDA32位中进行反汇编得到的结果如下

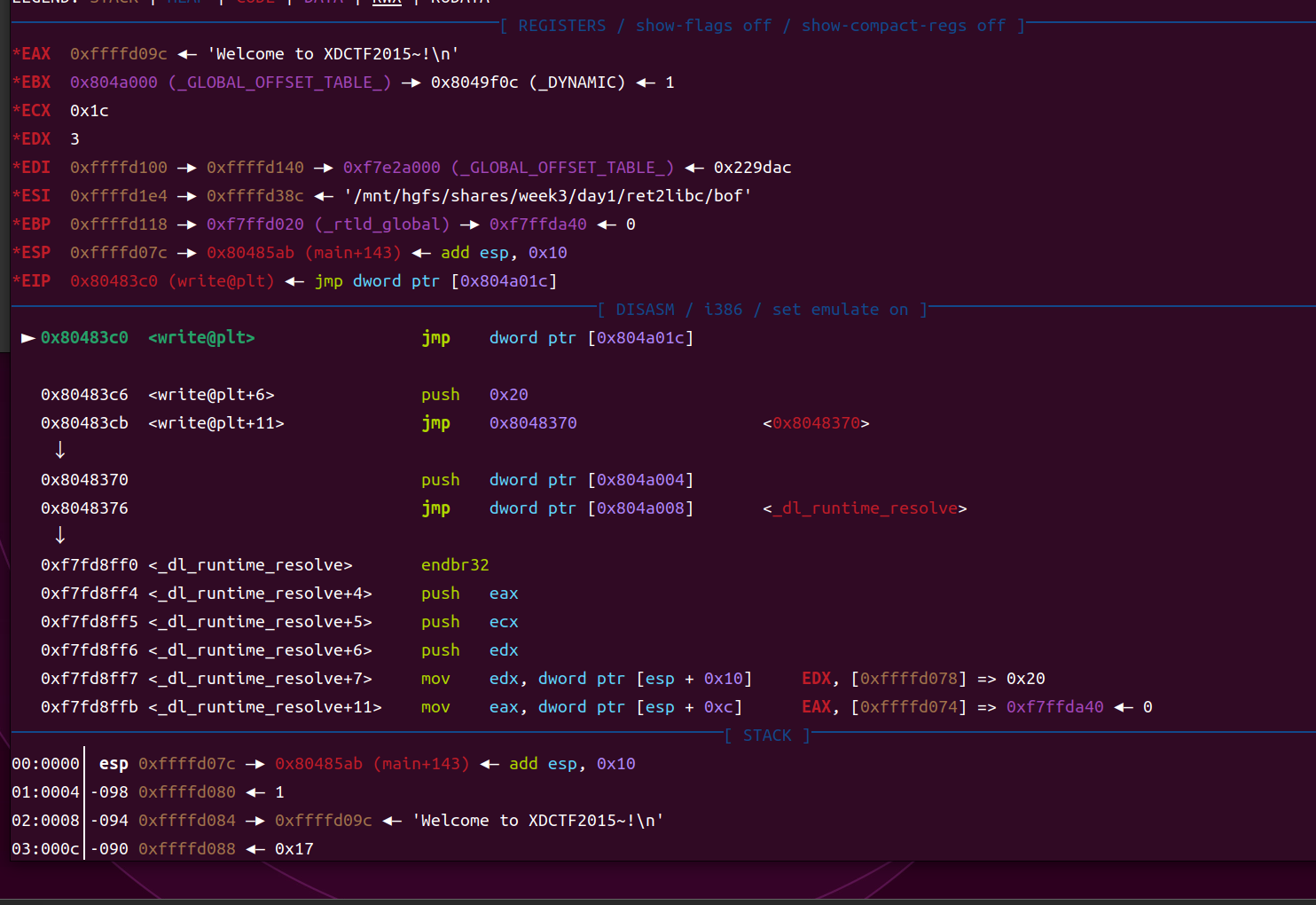


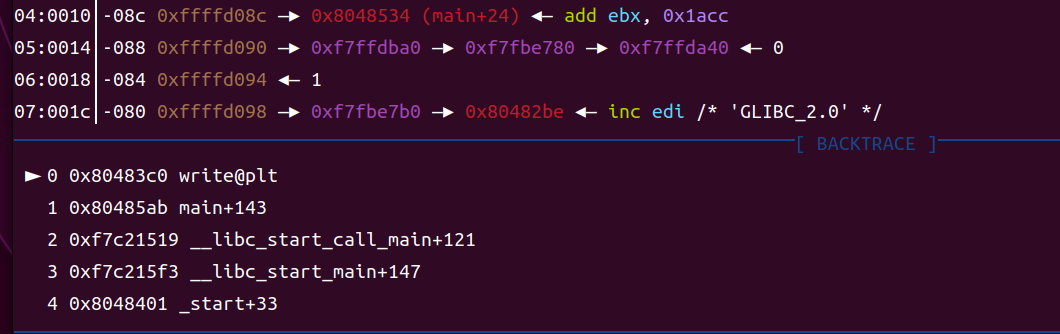


在write函数下断点然后运行



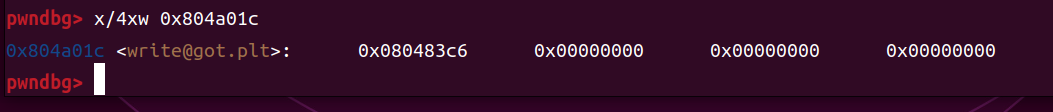
运行后的结果如下



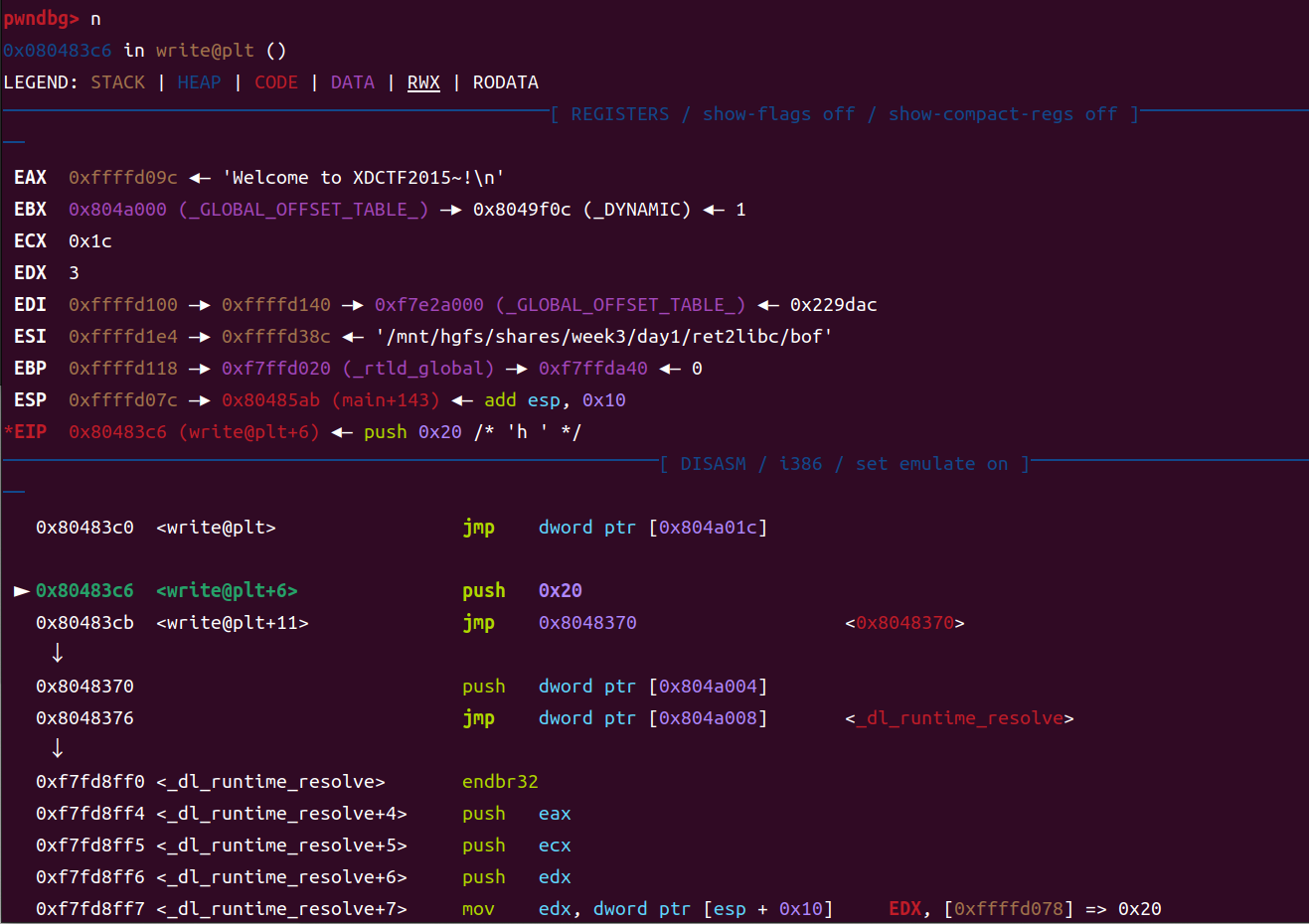


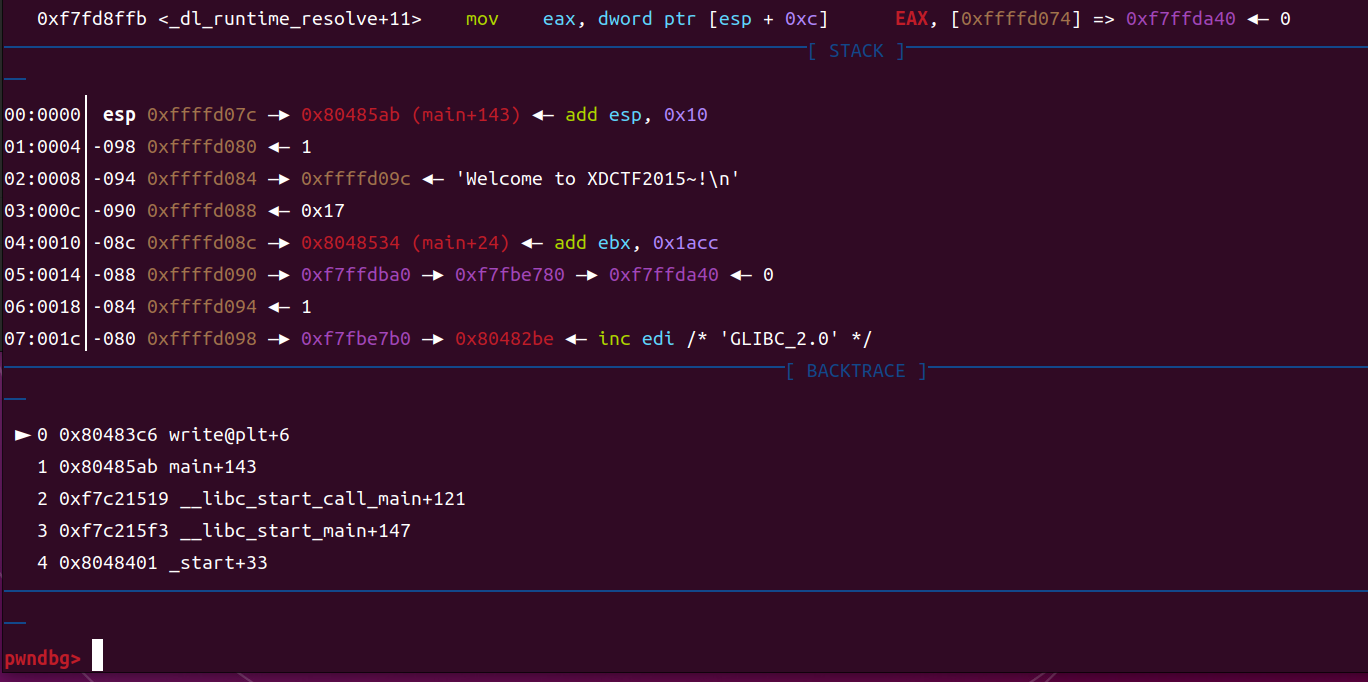
eip将要跳转的值为0x804a01c

查看内存中0x804a01c位置的值为0x080483c6即为write@plt的地址+6

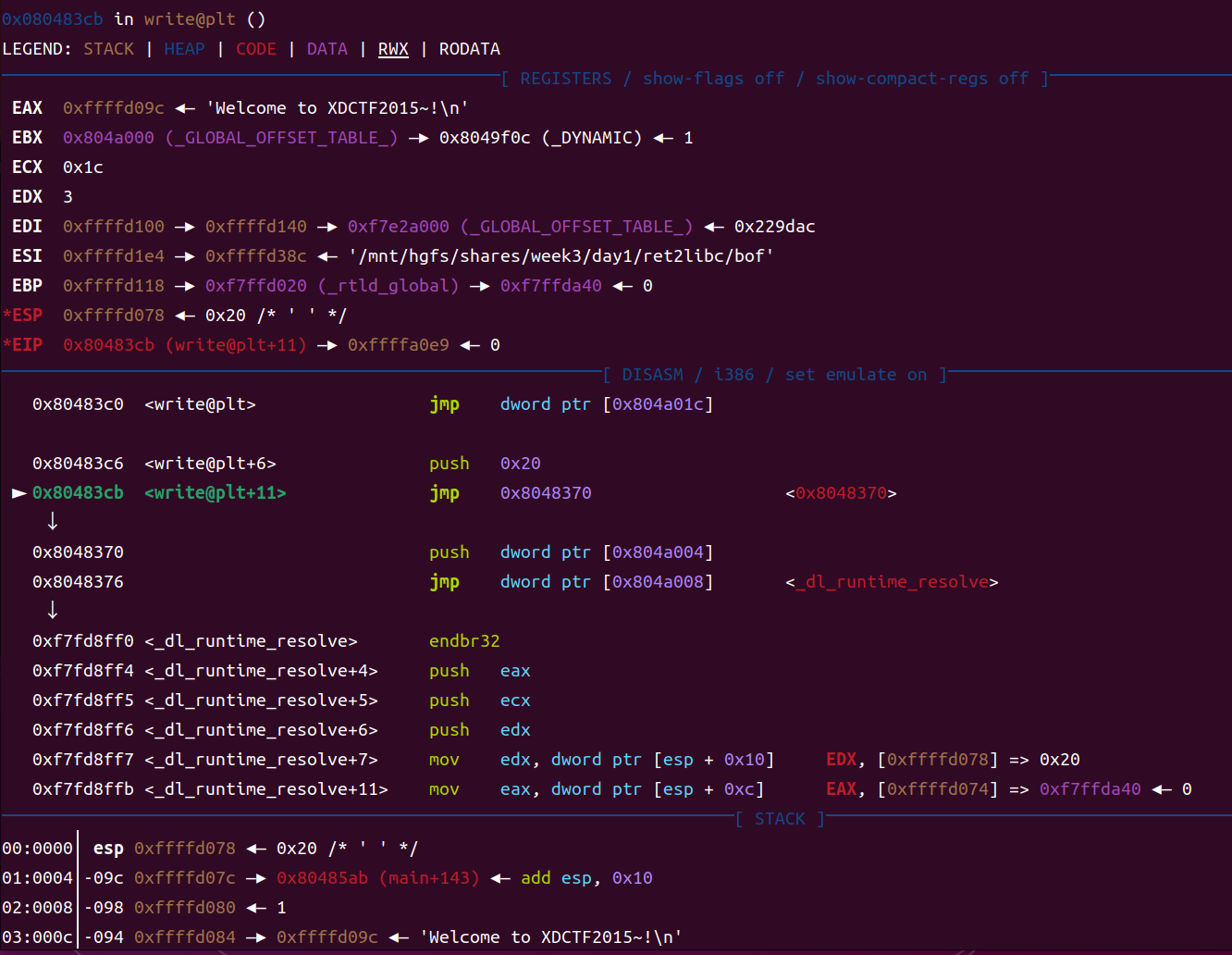


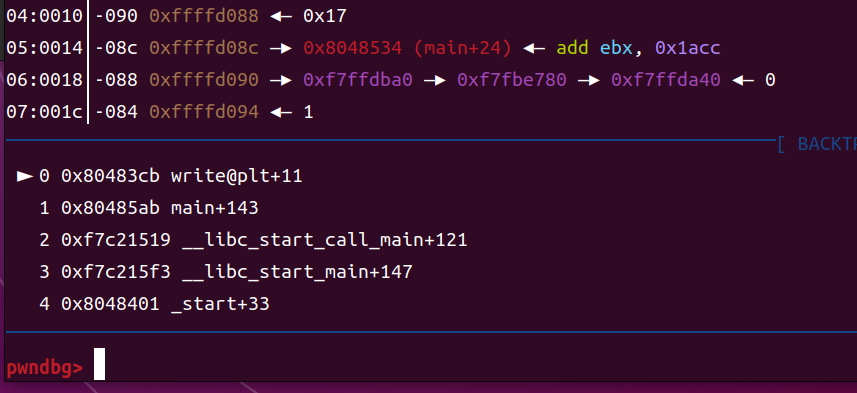
单步运行之后发现该指令是将0x20入栈，0x20是导入函数标识，是ELF\_Rel在。rel.plt中的偏移



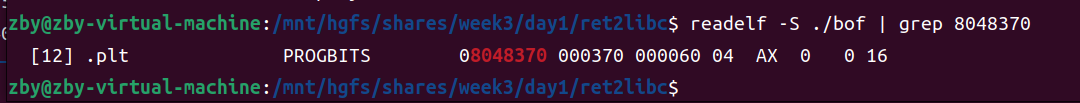


再次单步调试，是一个跳转指令，跳转到0x8048370，

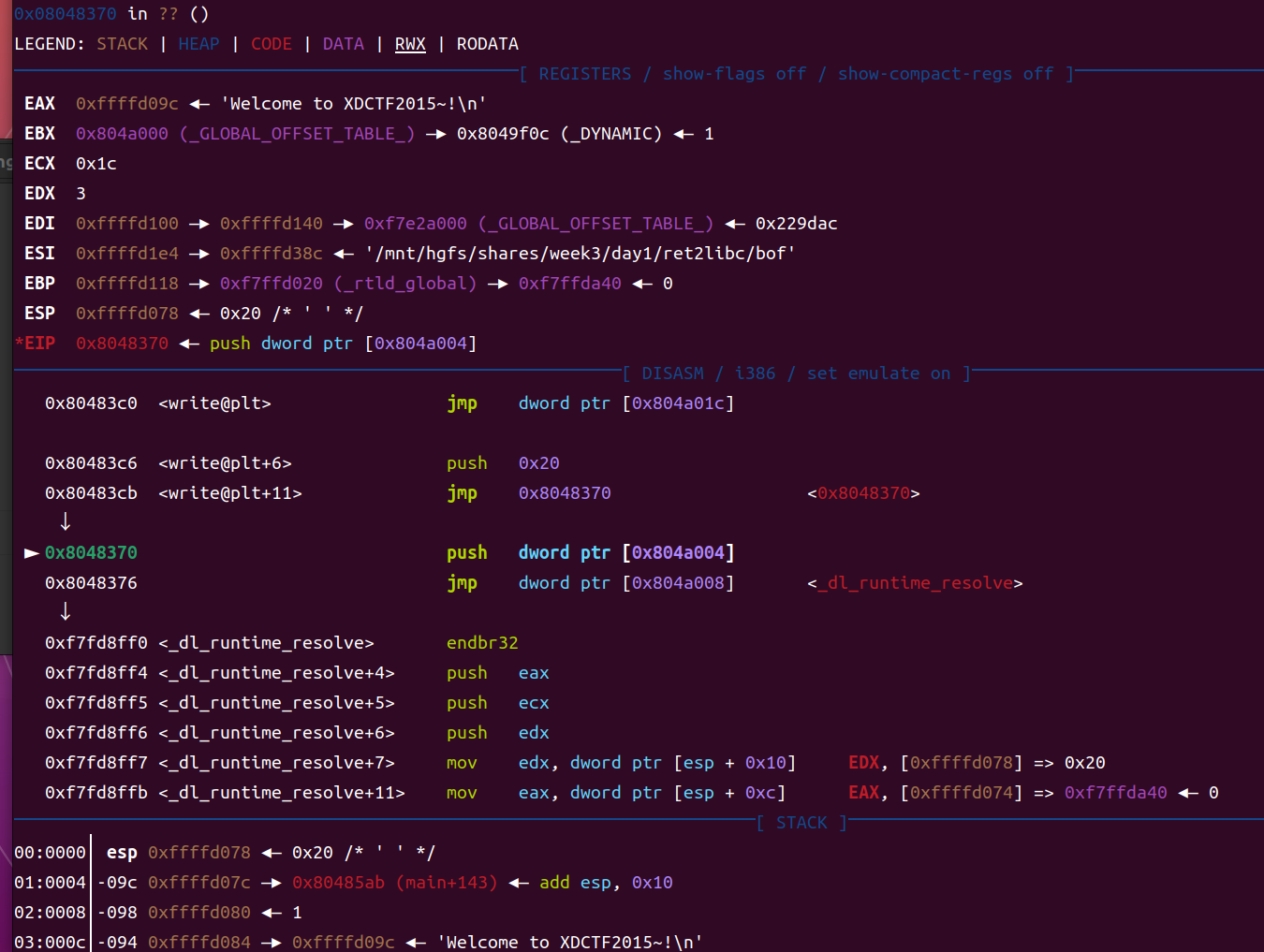


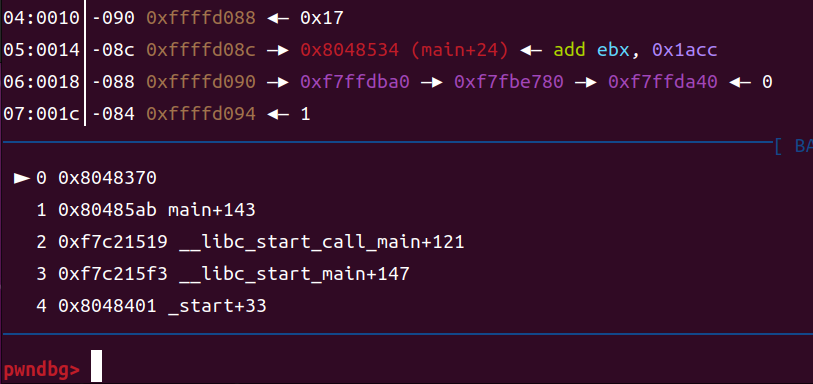


使用readelf工具查找该地址发现地址代表的是.plt段的开头PLT[0]

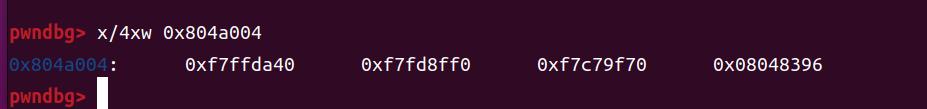


再次单步运行进入到PLT[0]处的代码

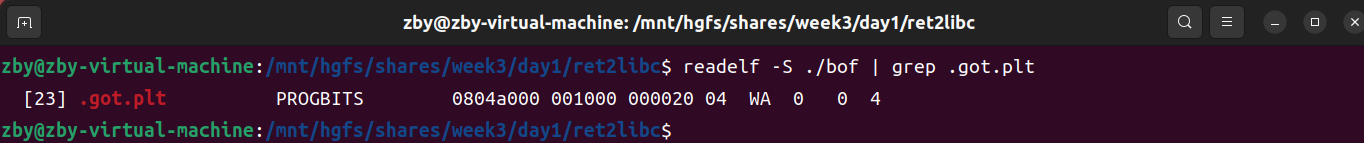




查看eip寄存器中段偏移的值0x804a004位置的值



接下来查看.got.plt段如下，可以得出GOT[1]是0x804a004，GOT[2]是0x804a008



所以我们直接反汇编GOT[2]位置指向的地址如下所示，我们可以发现PLT[0]就是调用如下函数

