

快乐假期

第二天，0x3704 又问我要指令，我有点生气：“你就不能记住吗？”

0x3704 说：“我可不能记住，万一你被重新装载了，指令变了怎么办？”

我告诉他指令是“`iload_1`”，他刚把数据扔到桶里，古怪的事情发生了：身手敏捷的 0x3704 突然好像凝固了一样，不动了。

只听到 String 欢呼：“遇到断点了，码农开始调试了，我们放假了！”

“调试？什么调试？”

“就是码农会单步、手工地执行这些指令。他们慢死了，可能 1 秒才能执行一步。由于我们的时间比他们的快得多，他们的 1 秒简直就是我们的十几天。走，出去玩去！”

“出去玩？能上哪儿玩？”我觉得这里无聊透顶了。

“找我们 new 出来的对象玩去！”

我想到了之前联系过我的对象 `Account@659e0bfd`，想着去看看也不错。

这个叫“堆”的大楼更加拥挤，全是人，String 的对象当然最多，String 类左右逢源，不停地打招呼，从我这个类创建出来的 Account 对象几乎找不到。

一队全副武装的士兵不停地巡逻，时不时地把对象拉出来，塞到车里去。

“这是在干吗？”我问 String。

“这些人叫清理者，专门清理没用的对象。你看，车里那不是 `Account@659e0bfd` 吗？”

“啊？昨天他还和我联系了，他怎么会没用了呢？”

“他很有可能只是一个方法的局部变量，方法结束后，就没人引用了，白白地占用空间。你看这幢楼太拥挤了，如果不清理，很快就会住满，系统崩溃，Out Of Memory 了。”

“那这幢楼就不能盖得更高一点吗？”我心里有点可怜这些被回收的对象们。

“楼有多高，是由码农决定的，他们在启动虚拟机的时候会指定参数。”

“那清理者怎么知道谁有用、谁没用？”

“可达性分析呗。这些清理者非常厉害，手里掌握着一些叫作 GC Roots 的对象，从这些节点出发四处搜索被 GC Roots 直接引用的对象，然后再找这些对象所引用的对象，这么一层一层找下去，就形成了一条以 GC Roots 为起点的引用链条。如果你很不幸，不在这条链条上（比如图 2-1 中的 Object C、Object E、Object F、Object G），那就很有可能被清理掉了！”