《漏洞利用及渗透测试基础》实验报告

姓名： 许健 学号：2013018 班级：信安班

**实验名称：**

程序插桩及Hook实验

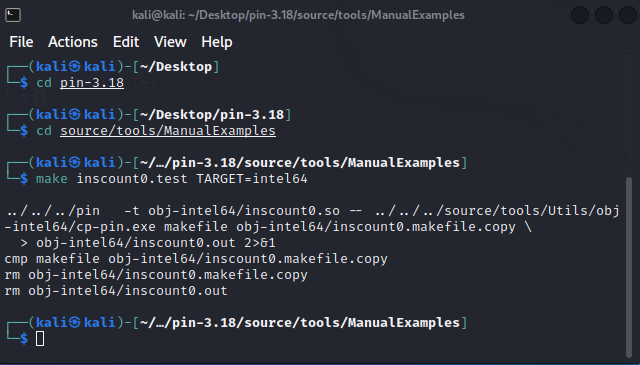
**实验要求：**

复现实验一，基于Windows MyPinTool或在Kali中复现malloctrace这个PinTool，理解Pin插桩工具的核心步骤和相关API，关注malloc和free函数的输入输出信息。

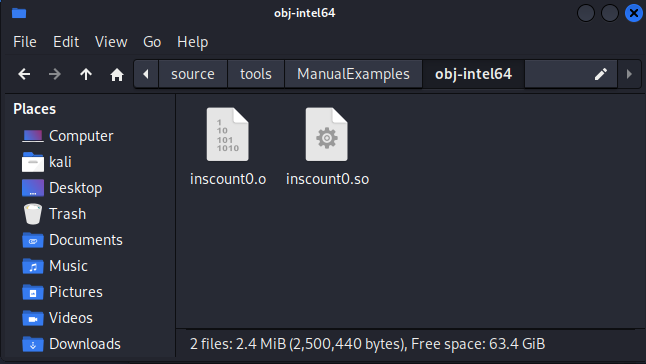
**实验过程：**

1. **Linux系统下复现实验一**
2. **编译inscount0.cpp**

进入source/tools/ManualExamples，对inscount0.cpp进行编译来产生其对应的动态链接库，所使用的命令为：make inscount0.test TARGET=intel64。



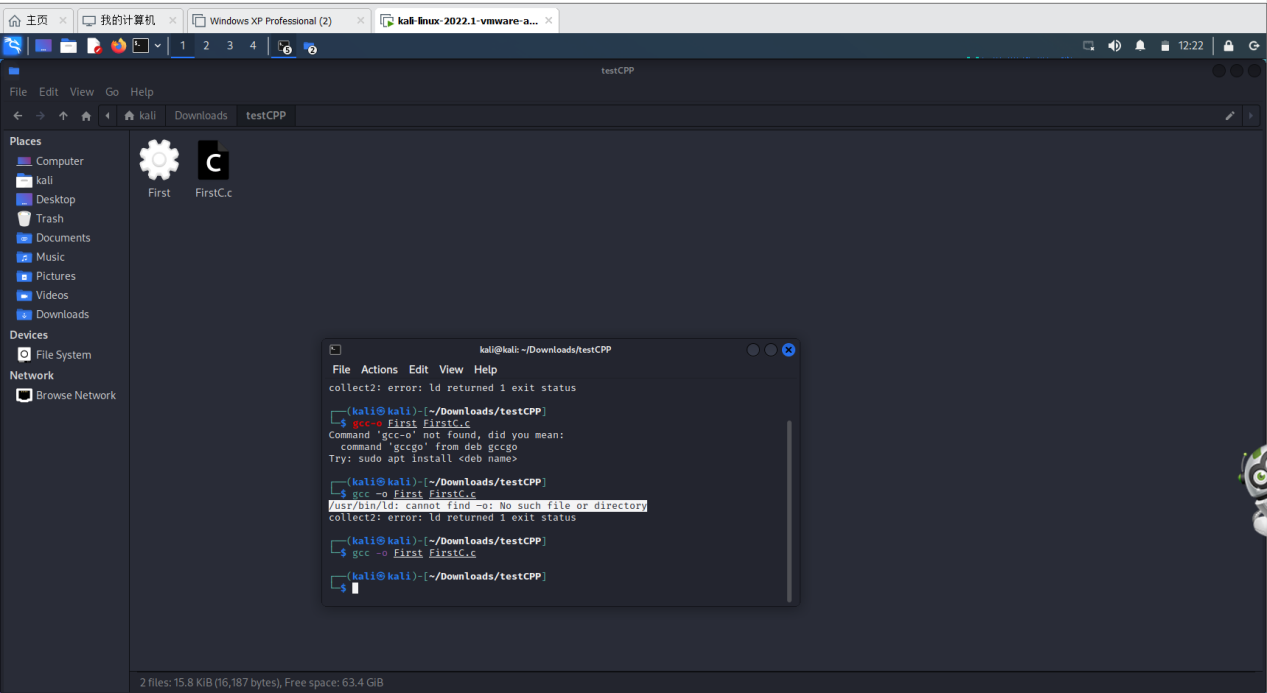
可以看到编译后的动态链接库，有inscount0.so



1. **编译c文件**

编写一个简单的控制台命令程序FirstC.c

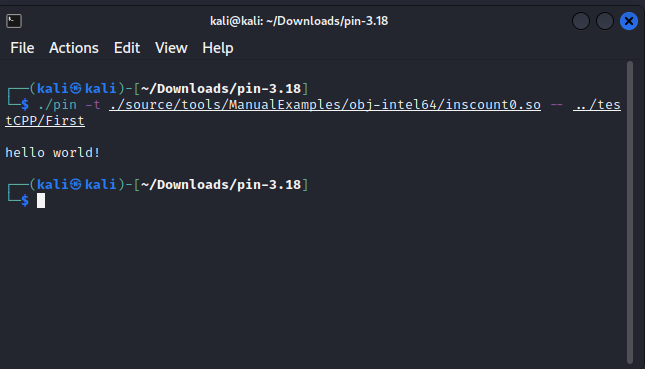
在Linux下编译c文件的命令为: gcc –o First FirstC.c

****

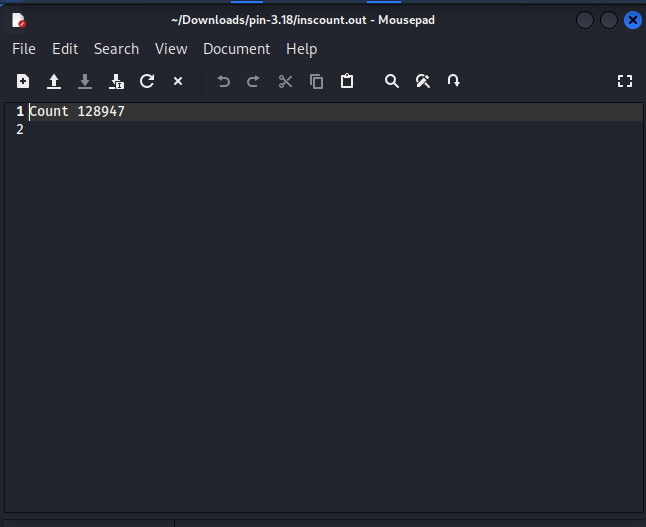
1. **程序插桩**

对First可执行程序进行程序插桩的Pin命令为：

./pin -t ./source/tools/ManualExamples/obj-intel64/inscount0.so -- ../testCPP/First

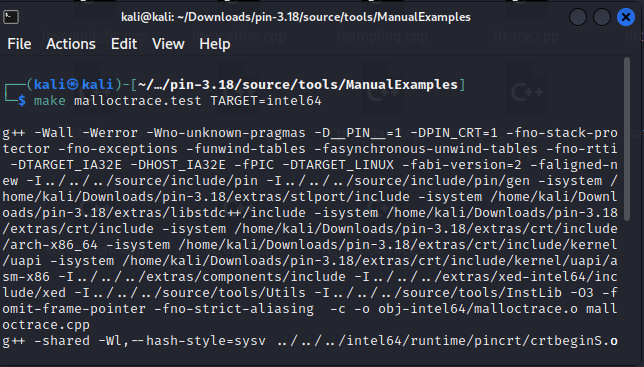


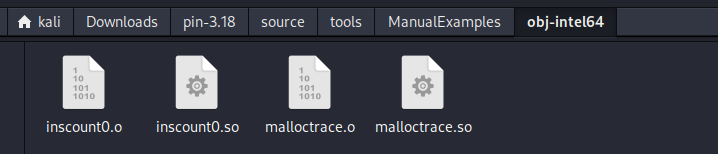
在pin-3.18路径下增加了一个输出文件inscount.out，即对指令数进行了插桩



1. **复现malloctrace**

对malloctrace.cpp进行编译来产生其对应的动态链接库，所使用的命令为：make malloctrace.test TARGET=intel64



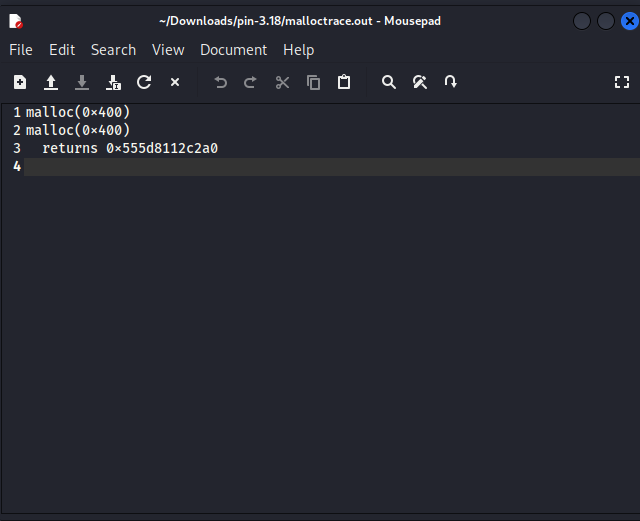


对First可执行程序进行程序插桩的Pin命令为：

./pin -t ./source/tools/ManualExamples/obj-intel64/malloctrace.so -- ../testCPP/First



查看生成的malloctrace.out具体内容



我们查看malloctrace.cpp的源码得知，这个 PinTool 可以实现函数粒度的插桩并实现对 malloc（内存分配） 与 free(释放) 的简单分析：记录 malloc 的参数及返回值，记录 free 的参数。

1. **Pintool核心步骤**
2. 调用函数PIN\_Init完成初始化
3. 注册插桩函数，在原始程序的每条指令被执行前，都会进入Instruction函数中
4. 注册退出回调函数，退出时调用该函数
5. 使用函数PIN\_StartProgram启动程序