作业：Shellcode 文件注入

题目：

实现一个程序，完成针对给定 PE 文件的 shellcode 注入与清除。

功能需求：

1. 能够校验给定文件是否合法 PE 文件，并打印输出
2. 可支持 32 位或 64 位 PE 文件
3. 能够检查 PE 文件是否开启 ASLR，并打印输出，如果开启可以提示选择关闭
4. 能够根据 shellcode 注入需求从 PE 头提取和计算相应信息，包含：节数目、文件特征信息、程序入口点相对地址、程序入口点文件偏移、各节名称、相对地址、内存大小、文件偏移、文件大小信息，并打印输出
5. 注入的 shellcode 默认为运行计算器，也可以指定原始 shellcode 二进制文件作为输入
6. 支持选择新增节或者代码空洞方式注入 shellcode
7. 新增节方式自动修改相应 PE 字段，将 Shellcode 放置到新增节中
8. 代码空洞方式支持根据给定的 Shellcode 查找 PE 文件中合适的代码空洞，如果不存在合适空洞，打印提示，如果存在合适的空洞，提示用户进行选择
9. 自动修改程序入口地址执行注入的 shellcode，shellcode 执行完成后需要返回到原程序执行入口执行
10. 支持移除使用本工具注入的 shellcode
11. 可根据指定，输出对应名称的注入或者移除 shellcode 后 PE 文件
12. 可运行在 32 位或 64 位 windows 操作系统，使用 python 或 C/C++实现

概要设计（供参考）：

1. 接口设计

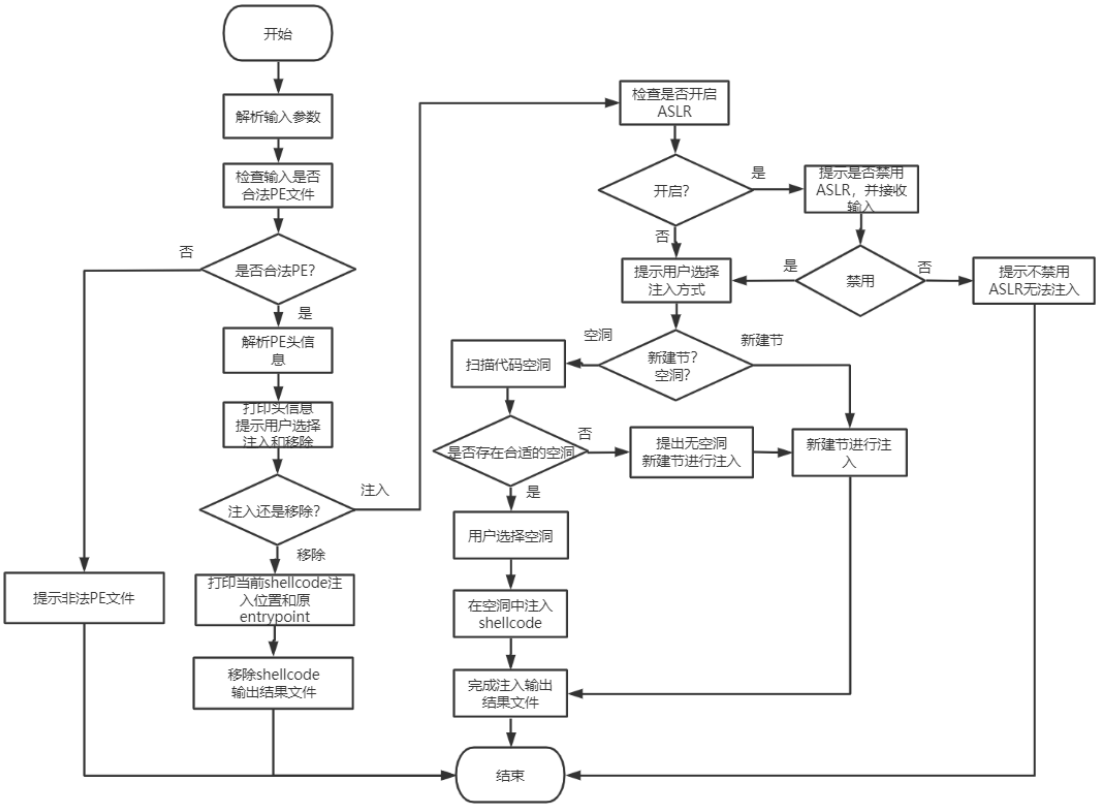
   程序为可执行文件，接收如下命令行参数：

   a) 必选，需要注入或者移除 shellcode 的文件路径
   b) 必选，输出的文件路径
   c) 可选，执行 Shellcode 注入还是移除动作，默认 shellcode 注入
   d) 可选（仅当选择动作为 shellcode 注入），需要注入的 shellcode 二进制文件，默认注入执行计算器

   程序执行过程中，终端输出提示信息，并接受用户选择输入；

   程序执行完成输出已注入或者移除 shellcode 后的 PE 文件。

2. 整体执行流程



3. 关键功能点
   a) PE 头解析
      按照 PE 头结构定义进行解析，可以参照现有结构体定义代码
   b) ASLR 与 shellcode 注入
      理解 ASLR 对 shellcode 注入的影响，以及关闭方式
   c) 节的新建
      新建节需要修改的 PE 头字段
   d) 空洞的查找
      可参考 Cave Miner 等已有代码，需要考虑空洞所在节的执行权限
   e) Shellcode 的地址计算
      根据 RVA、Image Base、File Offset 等信息计算 shellcode 注入位置和执行地址
   f) 恢复源程序执行
      需要在注入 shellcode 时计算并保存源程序入口地址，shellcode 执行完成后返回原程序正常执行流程
   g) Shellcode 的生成
      可使用 msfvenom 生成 shellcode，注意-f、-e、-b 参数的含义
   h) Shellcode 的移除
      注入时需要考虑移除功能需要，保留相关注入存根信息，方便 shellcode 移除

测试用例:

参考功能需求 1-12

参考材料:

1. https://en.wikipedia.org/wiki/Portable_Executable
2. https://docs.microsoft.com/en-us/windows/win32/debug/pe-format
3. https://www.offensive-security.com/metasploit-unleashed/msfvenom/
4. https://en.wikipedia.org/wiki/Address_space_layout_randomization
5. https://github.com/Antonin-Deniau/cave_miner