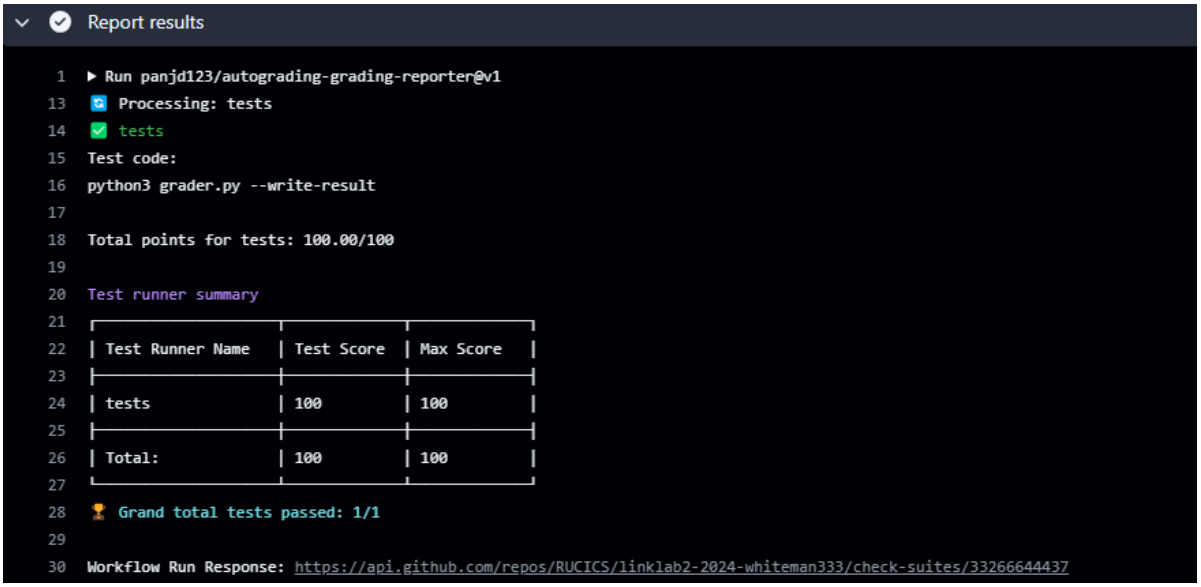
LinkLab 实验报告

姓名：卢虹宇
学号：2023202269



Part A: 思路简述

核心方法流程

一、扫描所有对象文件

1. **处理符号表中的静态变量**
   - 将静态变量名称前加上相应文件的名字作为前缀，以保证命名唯一性。
   - 修改其对应的重定位条目，确保重定位条目与改名后的变量一致。
2. **处理未定义符号**
   - 将未定义符号与全局符号表 `std::map<std::string, Symbol> all_symbols` 对照。
   - 如果到目前为止没有找到定义，则加入未定义符号表 `std::vector<Symbol> all_undefined_symbols`。
3. **按照节进行合并**
   - 对相应节创建或更新程序头。
   - 合并 `.bss` 段以外的 data 段。
   - 调整重定位条目的偏移量，并合并重定位表。
   - 更新符号偏移，将符号加入全局符号表，并根据新收集到的符号更新未定义符号表。
4. **检查未定义符号表**
   - 如果扫描结束后未定义符号表不为空，抛出错误。
5. **更新程序头的绝对地址**
   - 根据合并完成后各节的大小，设置节的绝对地址。
6. **确定程序入口点**
   - 设置程序入口点 `_start`。

二、扫描重定位条目

1. **计算需要填入的数据**
   - 根据重定位类型，计算目标值（地址或偏移量）。
2. **小端序写入**
   - 通过 `write_little_endian` 函数将值写入目标文件。

三、创建目标文件

Part B: 具体实现分析

符号解析实现

1. 处理不同类型的符号（全局/局部/弱符号）

- **全局符号和弱符号**：依照符号强弱规则处理，优先保留全局符号。
- **局部符号**：在符号名称前加上对应文件名作为前缀，同时修改与之相关的重定位条目，以避免不同文件的局部符号命名冲突。

2. 符号冲突解决方案

- 检查全局符号表是否存在同名符号：
  - 若新符号是全局符号（`GLOBAL`）：
    - 若原符号是弱符号（`WEAK`），则覆盖原符号。
    - 若原符号也是全局符号，则抛出错误，提示多重强符号定义。
  - 若新符号是弱符号：
    - 若全局符号表中已有全局符号，忽略该弱符号。
    - 若已有弱符号，则保留第一个弱符号。

3. 关键优化

- **静态变量处理优化**：通过修改变量名前缀，逻辑简单且避免了维护复杂的数据结构。
- **运行日志完善**：详细的调试输出帮助快速定位错误。

4. 边界情况与 Sanity Check

- 修改静态变量名称后，确保所有相关重定位条目都被更新。
- 避免只修改部分重定位条目导致隐蔽错误。

重定位处理

1. 支持的重定位类型

- **R_X86_64_32S**：检查是否可以通过 32 位符号扩展正确恢复 64 位地址。
- **R_X86_64_32**：直接截断并写入 32 位地址。
- **R_X86_64_64**：直接写入 64 位地址。
- **R_X86_64_PC32**：实现相对重定位，计算并写入偏移值。

2. 重定位计算方法

- **目标值计算公式**：

  目标值=符号节内偏移+符号对应节偏移+addend-重定位偏移-重定位条目对应段偏移

```
addr = sym.offset + section_addr + reloc.addend - reloc.offset -
header.vaddr;
```

3. 关键错误处理

- 检查地址是否超出目标字段的表示范围（例如 32 位符号扩展越界）。
- 其他问题主要来源于符号或段偏移计算错误。

段合并策略

1. 段组织和合并

- 独立处理子段（如 `.text.startup`、`.rodata.str1.1`）。
- 合并流程：
  1. 创建或更新程序头，更新 `size`。
  2. 合并数据（除 `.bss` 段外）。
  3. 调整重定位条目的偏移量并合并重定位表。
  4. 更新符号偏移，收集符号并更新未定义符号表。

2. 内存布局

- 由于可重定位文件中的相同类型的节相邻，所以在节合并后保持了该性质，相同权限需求的节被映射到同一块连续的内存区域中，以减少内存碎片。
- 根据节名前缀设置权限，将相同权限的节映射到连续的内存区域。

3. 对齐处理

- 在段合并完成后，运行地址按 4KB（0x1000）对齐，更新程序头。

Part C: 关键难点解决

1. 静态变量名冲突

- **难点**：静态变量仅对所在文件可见，但不同文件中可能有同名静态变量。
- **解决方案**：为静态变量名称添加文件名作为前缀，同时修改所有相关重定位条目。
- **效果**：避免静态变量命名冲突，逻辑简单清晰。

2. 收集符号时更新符号偏移

- **难点**：全局符号表是跨节的，但符号偏移是节内偏移，需要在节合并时正确更新符号偏移。
- **解决方案**：在合并节时顺便更新符号偏移，而非独立处理。
- **效果**：收集符号时逻辑更加清晰。

Part D: 实验反馈

1. 实验设计：实验难度适中
2. 实验文档：经过认真思考后还是容易理解，但是其中有一些很搞的细节还是应该着重强调，比如返回的object中重定位条目必须清空，要不然需要重定位的代码段就会被吞掉（感觉或许是个bug?）。还有就是debug方面感觉给出的工具没有很实用，基本上都是靠自己写log找出来的bug（毕竟等disasm派上用场我都到任务六了，而实际上前面几个任务还更难一些），不过readfle还是很好用的
3. 框架代码：框架代码写得相当清晰易懂了。

总体来说是一个很有趣的lab，在这个过程中我对链接器的工作原理又有了更加细致的理解，以后遇到链接带来的bug应该不至于束手无策。

参考资料

无