**操作系统实验报告**

**2024年秋季学期**

|  |  |
| --- | --- |
| **哈尔滨工业大学（威海）计算机科学与技术学院** | |
| **专 业** | 软件工程 |
| **姓 名** | 杨涛 |
| **班 级** | 2211106 |
| **学 号** | 2022211939 |

**哈尔滨工业大学（威海）计算机科学与技术学院**

**操作系统实验报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 杨涛 | | 专 业 | 软件工程 | | 指导老师 | | 朴学峰 |
| 班 级 | | 2211106 | | | 学 号 | | 2022211939 | |
| 实验时间 | | 2024.11.14 | | | | | | |
| 实验名称 | | 静态链接、动态链接、动态加载 | | | | | | |

* 1. 预习报告(对实验主要内容的认识)

本实验旨在加深对静态链接、动态链接和动态加载机制的理解。通过编写和执行多种链接方式的C程序，我们能够验证不同链接模式下内存使用、库加载时机的差异。在静态链接中，所有代码在编译阶段已嵌入执行文件；而在动态链接中，库文件在运行时被载入内存，节省了执行文件大小；而动态加载则是在运行时按需加载库文件，灵活性更高。

* 1. 实验内容（思路，过程，代码解析、运行结果）

1. **创建并编辑源码文件**

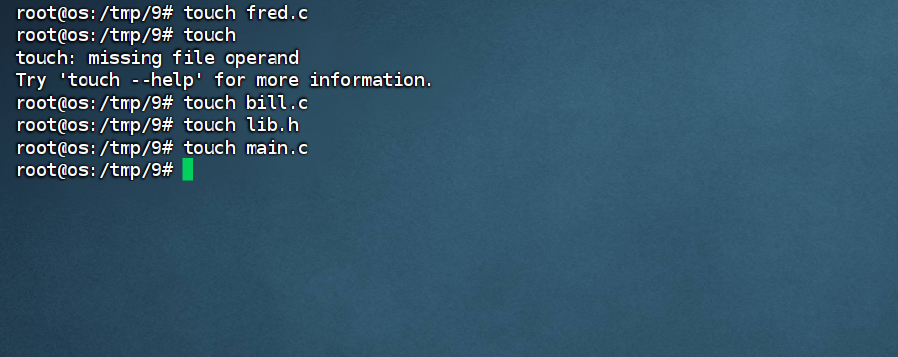


图 1 创建并编辑源码文件

1. **生成静态库文件**

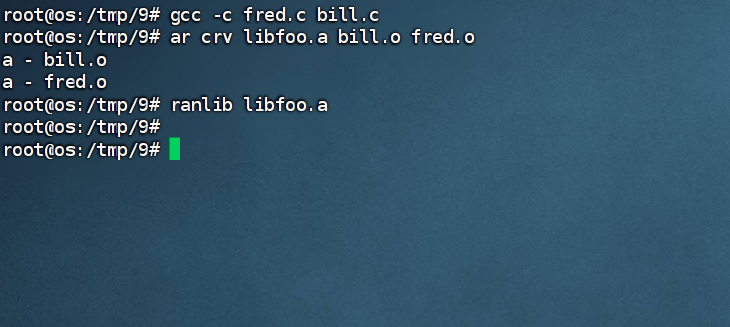


图 2 生成静态库文件

1. **静态链接并生成可执行文件**

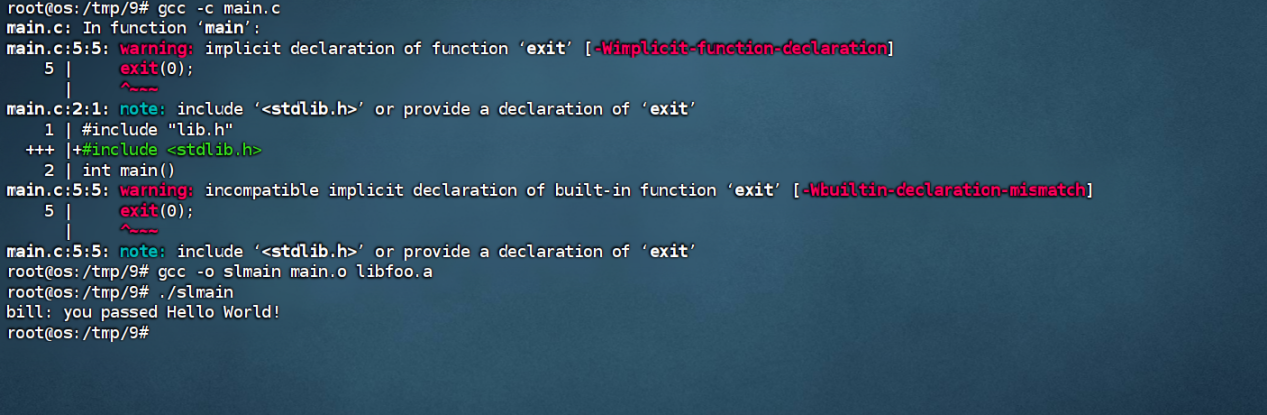


图 3 生成可执行文件

1. **运行后检查可执行文件大小**

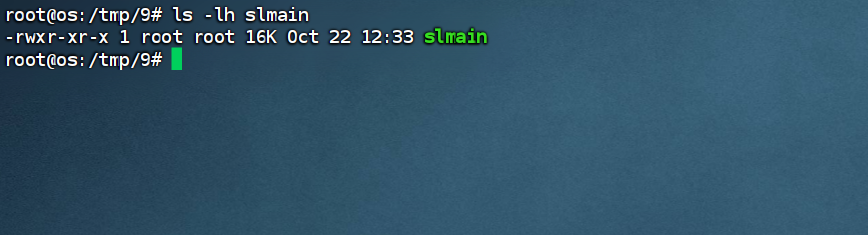


图 4 检查可执行文件大小

1. **编译并链接动态库生成可执行文件**

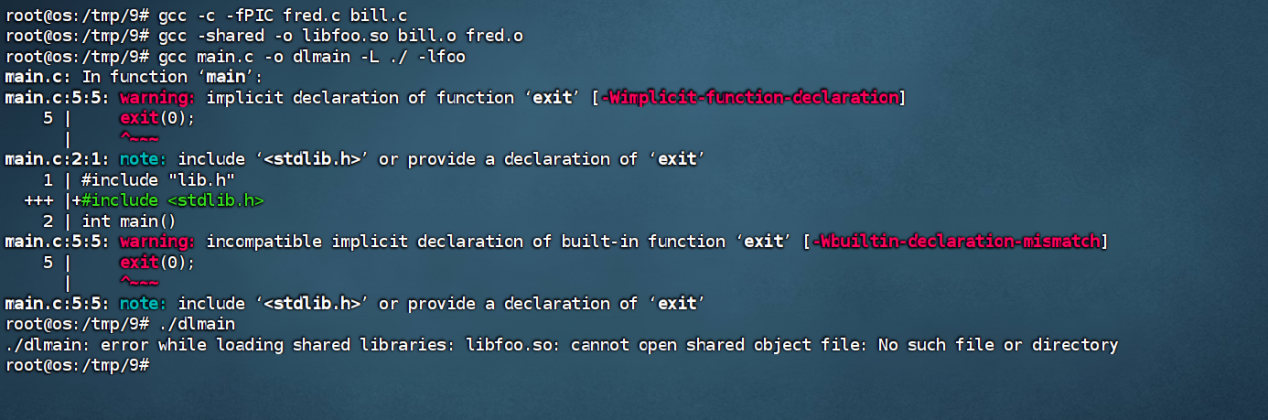


图 5 生成可执行文件

1. **再次检查生成的可执行文件大小**



图 6 检查生成的可执行文件大小

1. **编译并运行动态加载**

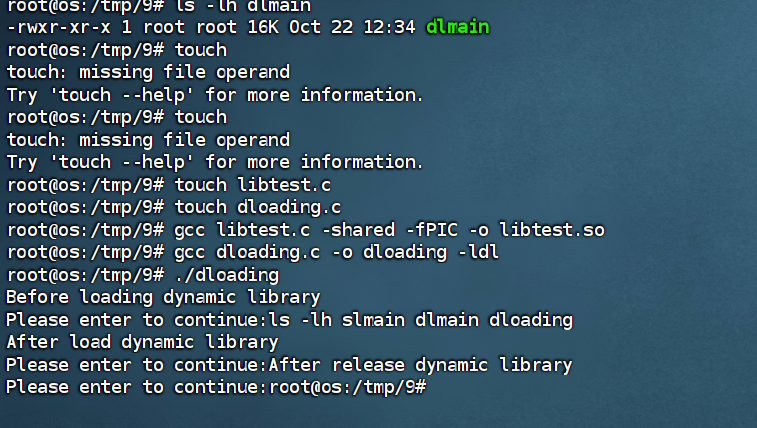


图 7 编译并运行动态加载

* 1. 通过实验所掌握的知识点总结

通过实验，我掌握了静态库、动态库和动态加载的基本原理及其实际应用。了解了如何通过gcc编译器生成静态库和动态库，以及使用dlopen等函数动态加载库文件。还学会了如何使用ldd工具查看动态依赖，以及ar和ranlib工具管理静态库。通过对比不同链接方式生成的可执行文件的大小与内存使用，进一步理解了三种方式在不同场景下的优劣。