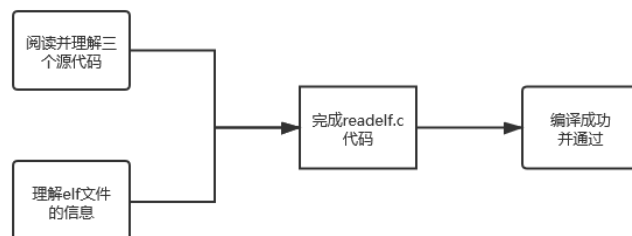


一 . 思考题:

1. 因为生成的内核文件的编码方式是大尾端编码, 我们的 readelf 程序只能解析小尾端编码。之后的工具 readelf 可以解析大尾端编码格式。
2. 内核入口在 start.S 中, main 函数在 init 目录下的 main.c 中, 通过无条件跳转指令跳到 main 函数, 通过 makefile 编译链接实现跨文件调用函数。

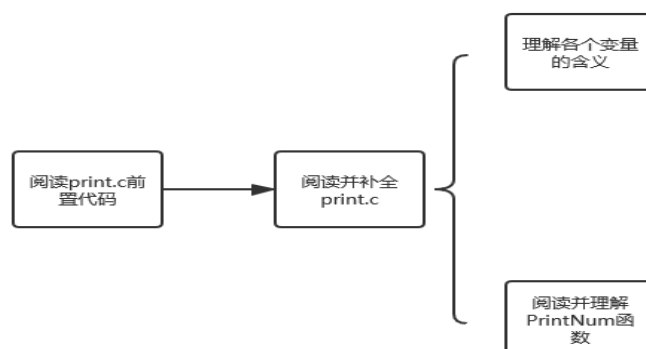
二 . 实验难点

1.



练习 1.2 是我花费时间最多的练习, 阅读三个源代码较为简单, elf 文件的格式最初理解的不是很透彻, 我感觉最难的就是完成 readelf.c 代码, 主要是对各个 elf 文件各个变量的类型不是很清晰, 还有就是对指针操作有一点生疏, 错误频频, 浪费了很多时间。

2.



练习 1.5 不是很难, 主要在于理解 PrintNum 函数, 这是这次练习的核心内容, 只有理解这个函数才能清楚各个变量的含义, 才能在主函数中正确使用。

三 . 体会和感想

这次的 lab1 课下内容总体上说难度不大, 个人原因包括指针运用生疏等导致完成的时间比较长, 从周一下午 2 点左右到晚上 12 点左右, 其中练习 1.2 花去了近一半的时间, 在练习 1.5 中通过课程群和同学讨论成功修复 bug 完成所有实验。

通过这次的实验我知道了操作系统的启动过程, 包括可执行文件的格式、内核的入口、栈的入口等, 也知道了 C 语言中最常用的 printf 函数的实现方式, 对操作系统有了一点浅显的理解。

四 . Lab1-extra

这次的 lab1-extra 难度不是很大, 花费了 2 个多小时, 主要原因是大尾端存储的数据转换成小尾端存储的方法不清楚, 花了一些时间去查资料。readelf