# 需求分析与复杂工程问题归纳

## 需求分析

首先我们对链接器部分的功能性需求进行分析。在本次项目中，链接器要求能够解析ARM架构下的可重定位文件，且生成的同样是ARM格式的可执行文件。链接器需要读取多个ELF格式的重定位文件，之后整合并输出为一个新的ELF可执行文件，具体过程可用数据流图反映如下图所示：

**图示

描述已自动生成**

**图1**

在可靠性方面，链接器需要做到尽最大可能的可靠，且在出现异常状况时给出正确恰当的错误信息。通过对链接过程的详细分析，不难发现将链接的过程进行细化，在读入可重定位文件之后，我们需要解析各个文件的文件头，之后根据文件头找到各个文件的section，并将section进行合并同时分配地址，同时我们还需要整合可重定位文件下的符号信息，并在分配地址之后进行符号的解析和重定位操作，完成了之后我们则整合所有的二进制信息，生成程序头并输出到文件系统之中，因此我们可以画出细化后的数据流图如下：

**图示

描述已自动生成**

**图2**

链接器要求在可重定位文件正确的情况下能够正确生成对应的可执行文件，而在重定位文件有误，比如遇到符号重复定义或者未定义、文件格式有误时链接器能够反映出正确的错误信息给用户，并提前终止。

在性能要求上，我们需要尽量减少内存的浪费，因此我们在实现链接器时还需要在执行过程中随时释放掉不需要的内存。

## 可行性分析

#### 经济可行性

在可行性分析方面，由于本次项目不会牵涉到任何经济相关的问题，可不考虑经济可行性以及财务可行性、投资可行性，我们默认相关可行性方面都是成立的。

#### 社会可行性

该项目中链接器不会牵涉到任何法律、政治体制等，同样我们认为该项目具有社会可行性。

#### 技术可行性

下面对链接器的技术可行性进行分析。

链接器相关的技术在当今已经非常成熟，且工程界已经有了许多公司的链接器和相关优化策略，同时还有许多链接器实现相关的参考资料可以学习，因此在技术上链接器的实现是一定可行的。

本次项目中链接器部分要实现的是ARM下的链接器，要求能够生成树莓派可以运行的可执行程序。经过分析，在读取可重定位文件方面可以直接通过编程语言已经封装好了的文件操作函数进行实现，对于ELF的文件格式解析则可以直接调用Linux下的头文件<elf.h>中的数据结构对读入的文件进行解析，同时还可通过已有的“elf-format”参考资料来了解具体ELF格式中不同部分的功能和关系，在进行最后的可执行文件组装和生成时，同样结合“elf-format”来正确生成最终的可执行文件，并直接通过编程语言的文件写操作来按ELF格式输出可执行文件的二进制内容即可实现。

因此在目前链接器技术已经成熟的情况下，链接器部分的技术实现是完全可行的。

# 测试环境构建与测试驱动开发

由于链接器作为一种较为特殊的软件，我们采用手动测试的方法来对其功能进行测试，观察其是否能够正确生成期望的可执行文件，并且在遇到错误时能够给出报错信息来提示用户进行改正，对此我们设计了测试链接器的用例如下表所示：

|  |  |  |
| --- | --- | --- |
| **用例编号** | **输入** | **期望结果** |
| **LK01** | 无 | 链接器报错，并显示正确使用方式。 |
| **LK02** | 含有未定义符号的重定位文件 | 链接器报错“undefined symbol xx” |
| **LK03** | 非ELF格式的文件 | 链接器报错“非法格式文件” |
| **LK04** | 不存在的文件 | 链接器报错“文件不存在” |
| **LK05** | 汉诺塔程序 “test.o”和相关文件“common.o”, 指定输出文件名“hanoi” | 链接器正确链接生成汉诺塔程序hanoi |
| **LK06** | 汉诺塔程序 “test.o”和相关文件“common.o”, 不指定输出文件名 | 链接器正确链接生成汉诺塔程序a.out |

之后我们手动进行测试，按照用例编号依次测试。

得到用例LK01测试结果如下图：

****

**图3**

用例LK02测试结果如下图：

**文本

描述已自动生成**

**图4**

用例LK03测试结果如下图所示：

**图形用户界面, 文本

中度可信度描述已自动生成**

**图5**

用例LK04测试结果如下图所示：

文本

中度可信度描述已自动生成

图6

用例LK05测试结果如下图所示：

图形用户界面

中度可信度描述已自动生成

图7-1

文本

描述已自动生成

图7-2

用例LK06测试结果如下图所示：



图8-1

图片包含 文本

描述已自动生成

图8-2

因此，通过上述测试可以发现链接器部分的功能基本正确。