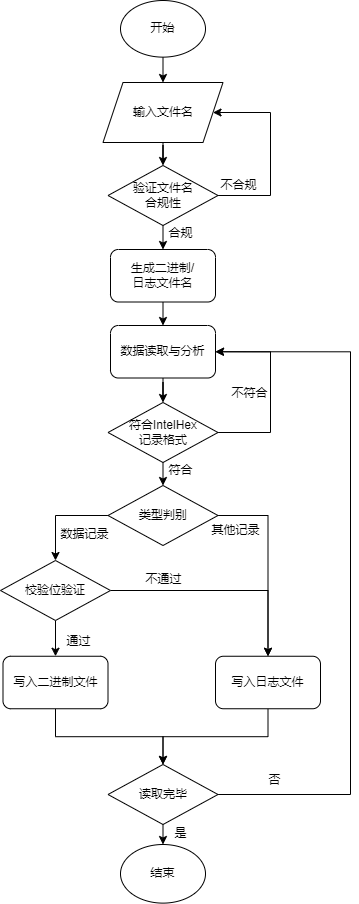


1. **用户输入阶段：**  
   用户通过终端输入待处理文件名。
2. **文件名合规性验证：**调用xx函数，对用户输入的文件名进行合规性检查。  
   a) 确保输入不为空。   
   b) 验证文件类型是否匹配。  
   c) 验证所选文件是否真实存在于文件系统中。
3. **合规性反馈：**  
   如果用户输入的文件名不符合规定，程序将提示用户重新输入，直至输入合规。
4. **文件名生成：**调用xx函数，根据输入的文件名自动生成相应的二进制文件名和日志文件名。
5. **数据分割处理：**  
   调用xx函数，逐行读取原始文件中的数据，并执行奇偶位分割。   
   a) 二进制文件无法打开或写入，程序捕获异常，释放内存, 回收空间。   
   b) 分割成功，数据将分别保存到指定的二进制文件中。
6. **数据对比分析：**  
   调用xx函数，分别读取两个二进制文件中的数据，并逐位进行对比  
   a) 文件读取失败，程序捕获异常，释放内存, 回收空间。   
   b) 文件读取成功，程序将逐位对比数据，并输出对比结果。
7. **程序结束：**数据对比分析完成后，程序执行必要的清理工作，释放内存, 回收空间。



1. **用户输入阶段：**  
   用户通过终端输入待处理文件名。
2. **文件名合规性验证：**调用xx函数，对用户输入的文件名进行合规性检查。  
   a) 确保输入不为空。   
   b) 验证文件类型是否匹配。  
   c) 验证所选文件是否真实存在于文件系统中。
3. **合规性反馈：**  
   如果用户输入的文件名不符合规定，程序将提示用户重新输入，直至输入合规。
4. **文件名生成：**调用xx函数，根据输入的文件名自动生成相应的二进制文件名和日志文件名。
5. **数据读取与分析：**  
   调用xx函数，从文件中逐行读取数据，并进行分析
   1. 校验输入数据是否符合标准的 IntelHex 数据格式；如果不是，程序将跳过当前行，继续读取下一行数据。
   2. 对数据类型进行判别，区分数据记录和其他记录。
6. **数据记录：**

进行校验位计算和验证

* 1. 数据合规，将该记录写入二进制文件
  2. 数据不合规，将该记录写入日志文件

1. **其他记录：**

对于其他类型数据，直接记录到日志文件中

1. **判断是否读取完毕：**
   1. 未完毕，继续读取数据并进行分析
   2. 完毕，结束程序并安全退出

性能优化

1. 使用fread逐行读取数据，避免将数据全部保存到内存中
2. 对用户输入进行验证，即使用户输入错误，也不会直接退出程序，而是可以让用户重新输入结果。
3. 使用多线程，并行执行数据读取和分析操作
4. 多线程执行数据分析操作时，将需要保存的数据提取出来，暂存到一个缓冲区中。后续，从缓冲区一次性写入，从而确保写入顺序和读取顺序保持一致。

时间提升 xxx

存在问题：

由于需要一次性写入数据，数据暂存在缓冲区，内存负荷较大。

1. 对用户输入进行验证，即使用户输入错误，也不会直接退出程序，而是可以让用户重新输入结果。同时允许用户对指定目录下文件进行查看，避免用户因遗忘目录名称而输错文件名
2. 将生成结果直接保存到当前目录下
3. 对文件内容进行校验，提升程序的健壮性。针对输入为空、输入非IntelHex数据、输入多个结束标记、以及结束标记后仍有数据/空行等情况进行了考虑，确保程序能正确处理这些问题。