# Reverse

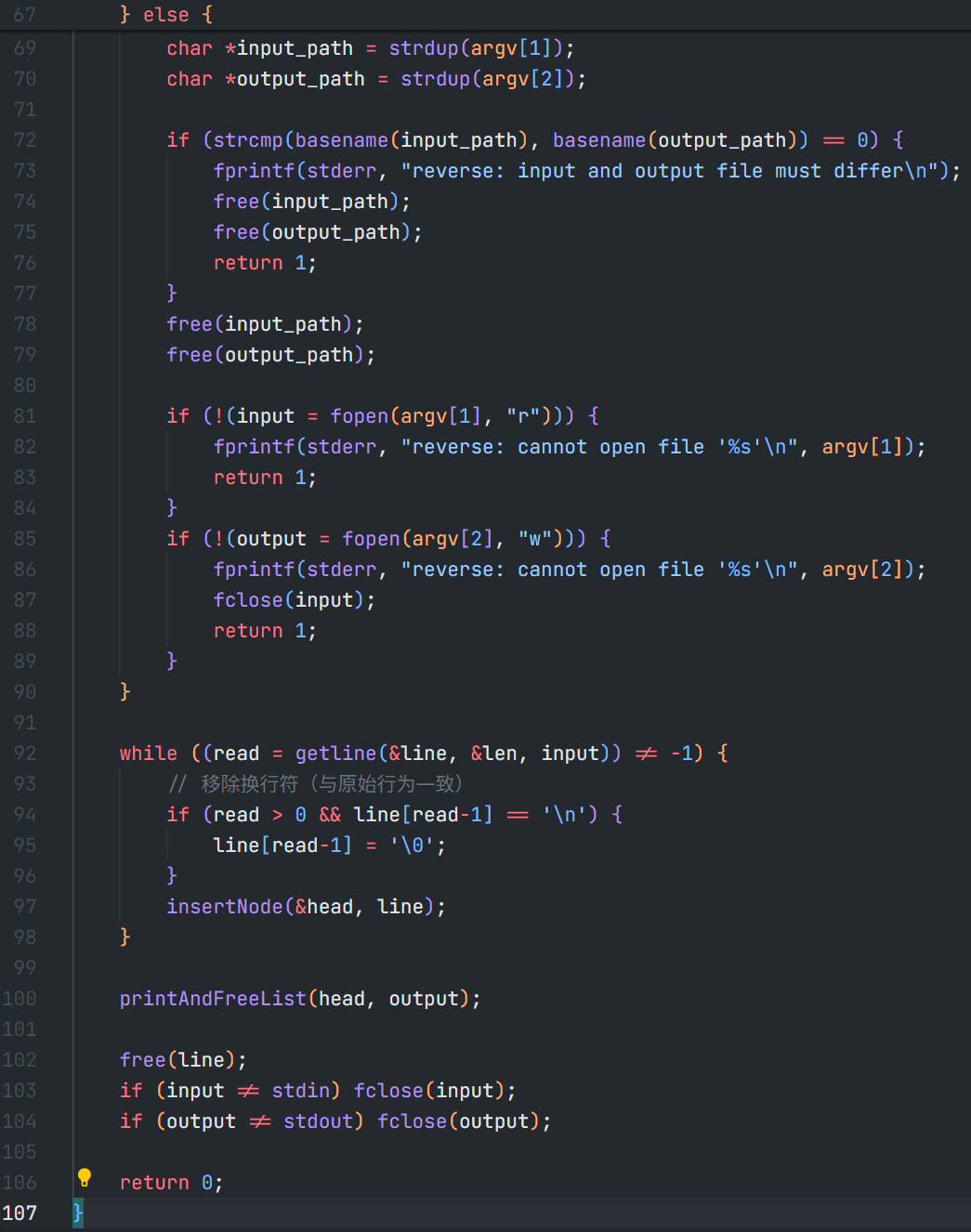
这个实验的主要任务是编写一个能够反转文本的程序。 使用链表的数据结构来存储和读取字符串。完成正向输入和反向输出。

定义存储文本的结构体，以及基本的结点创建，与反向输出所需的从头节点插入结点以达到LIFO的效果



主函数包括参数数量不同的处理和逆序输出



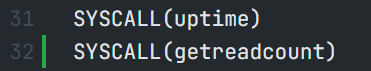


# Syscall

先对xv6以及qemu进行配置

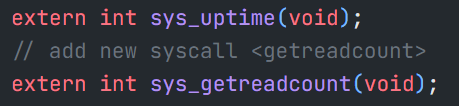
1. Usy.S

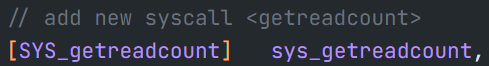
这是一个汇编语言文件，其中包含用户空间系统调用的实现。它定义了一组汇编宏，用于将系统调用请求从用户空间程序传递到内核。我们先将我们所写的getreadcount添加进去



1. Syscall.c

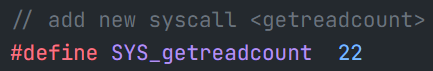
此文件包含 user-space 系统调用的实现。它定义了一个名为 syscall（） 的函数，该函数负责将系统调用号和参数传递给内核并接收内核返回的结果。将我们的添加





1. Syscall.h

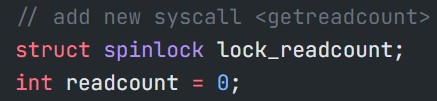
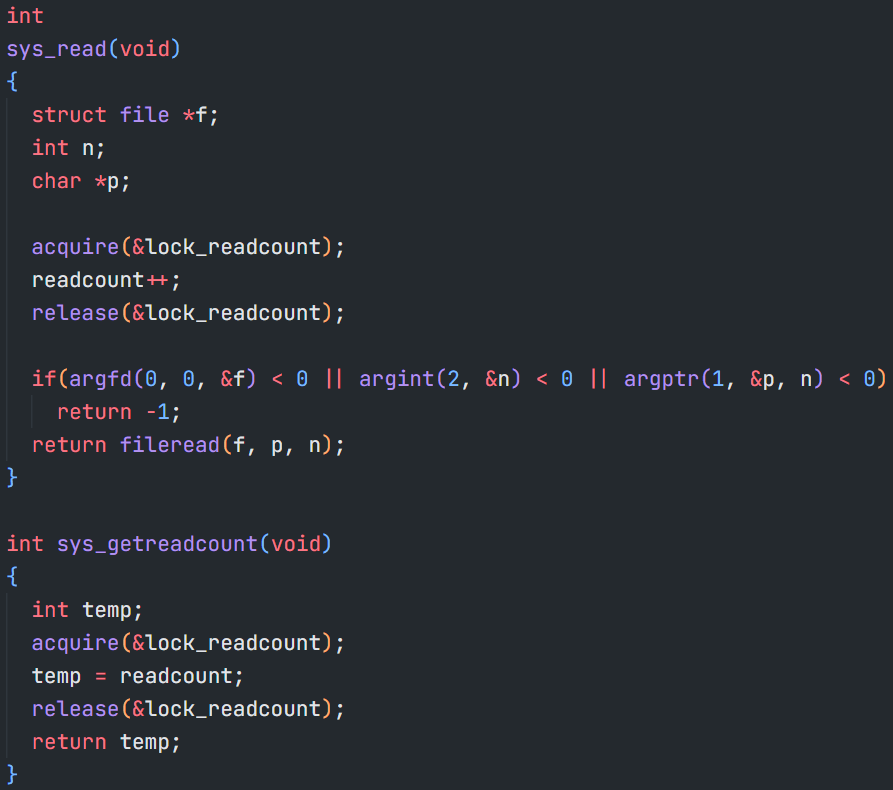
此头文件定义 system 调用的编号。它包含一系列 #define 指令，每个指令对应于一个唯一的系统调用号。用户空间程序可以使用这些数字来请求不同的系统调用。



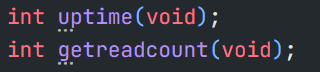
1. Sysfile.c

此文件包含与文件系统相关的系统调用的实现。它定义了一组用于用户空间和文件系统之间交互的函数，例如打开文件、读取文件、写入文件和关闭文件。

为get操作加上锁防止其在读的时候值改变

1. Sysfile.c



两个实验的运行测试

