**操作系统课程设计实验报告**

实验题目： Linux 进程管理

姓 名：

学 号：

组 号：

专 业：

班 级：

老师姓名：

日 期： 2024 年 06 月 04 日

目 录

[一 题目介绍 1](#_Toc9066)

[二 实验思路 2](#_Toc22209)

[三 遇到问题及解决方法 2](#_Toc10340)

[四 核心代码及实验结果展示 3](#_Toc6201)

[五 个人实验改进与总结 6](#_Toc25457)

[5.1 个人实验改进 6](#_Toc5843)

[5.2 个人实验总结 6](#_Toc19967)

[六 参考文献 6](#_Toc17199)

# 一 题目介绍

**1、实验介绍**

本实验通过编写模拟Shell，管道通信程序，消息通信程序，共享内存机制，了解Linux进程的创建，进程通信的方法。本实验为分组实验，小组协作完成，每人完成一个程序，但需要教会同组其他同学所有实验内容。

**2、任务描述**

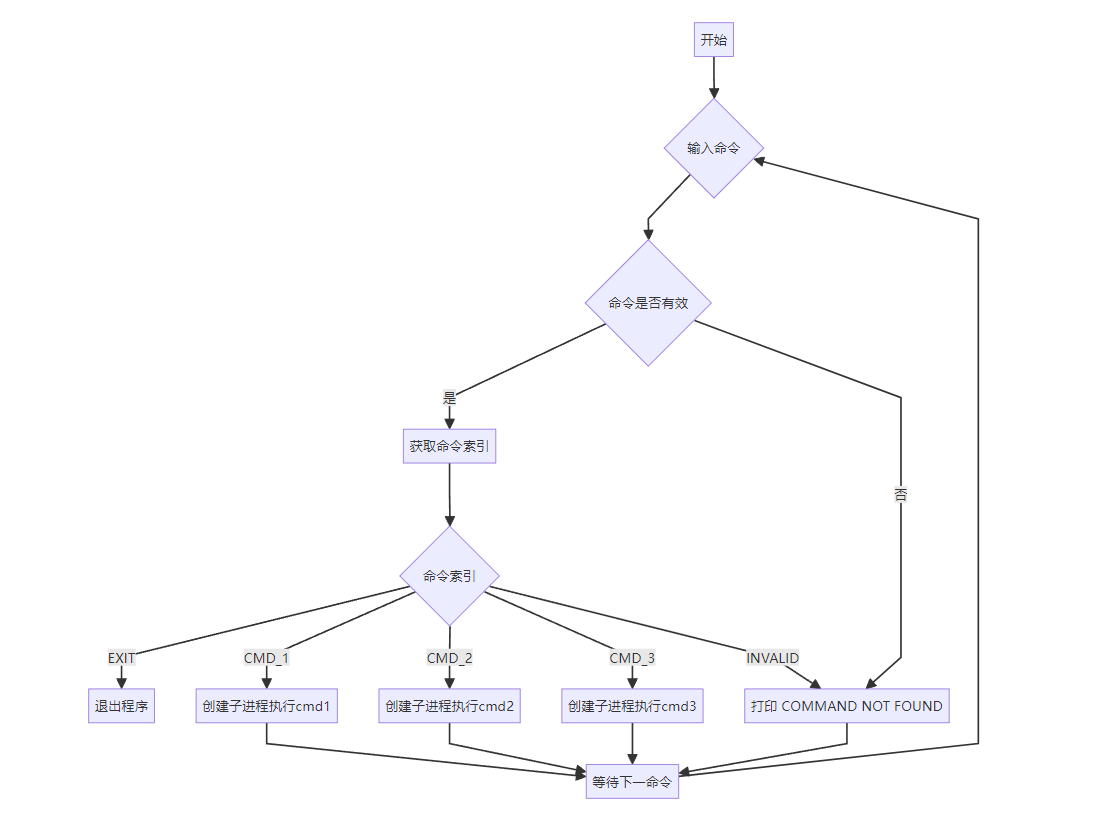
* 在Linux系统下，实现一个模拟的 shell
* 在Linux系统下，实现一个管道通信程序
* 利用 Linux 的消息队列通信机制实现两个线程间的通信
* 利用 Linux 的共享内存通信机制实现两个进程间的通信

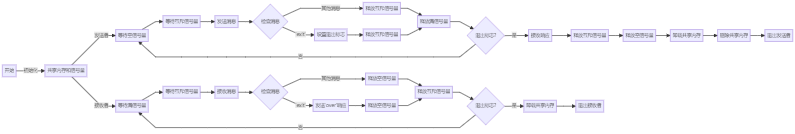
**3、实验目的**

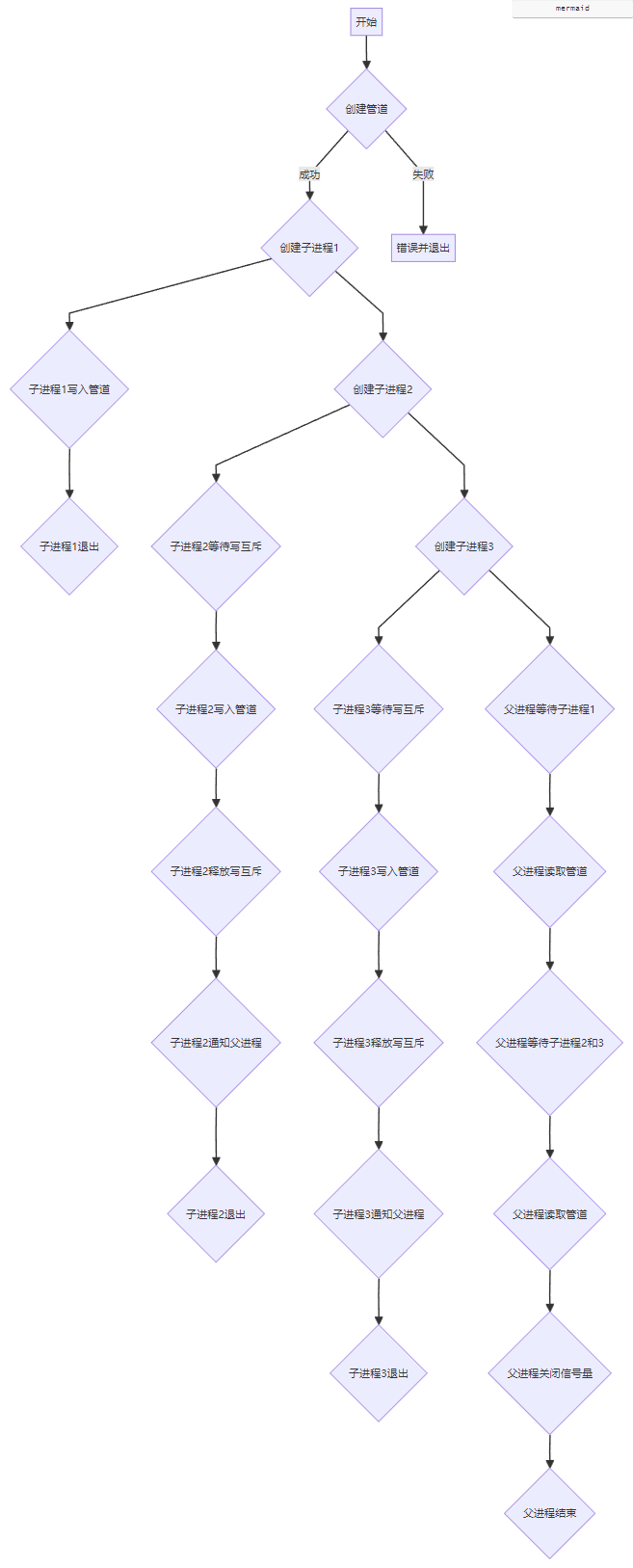
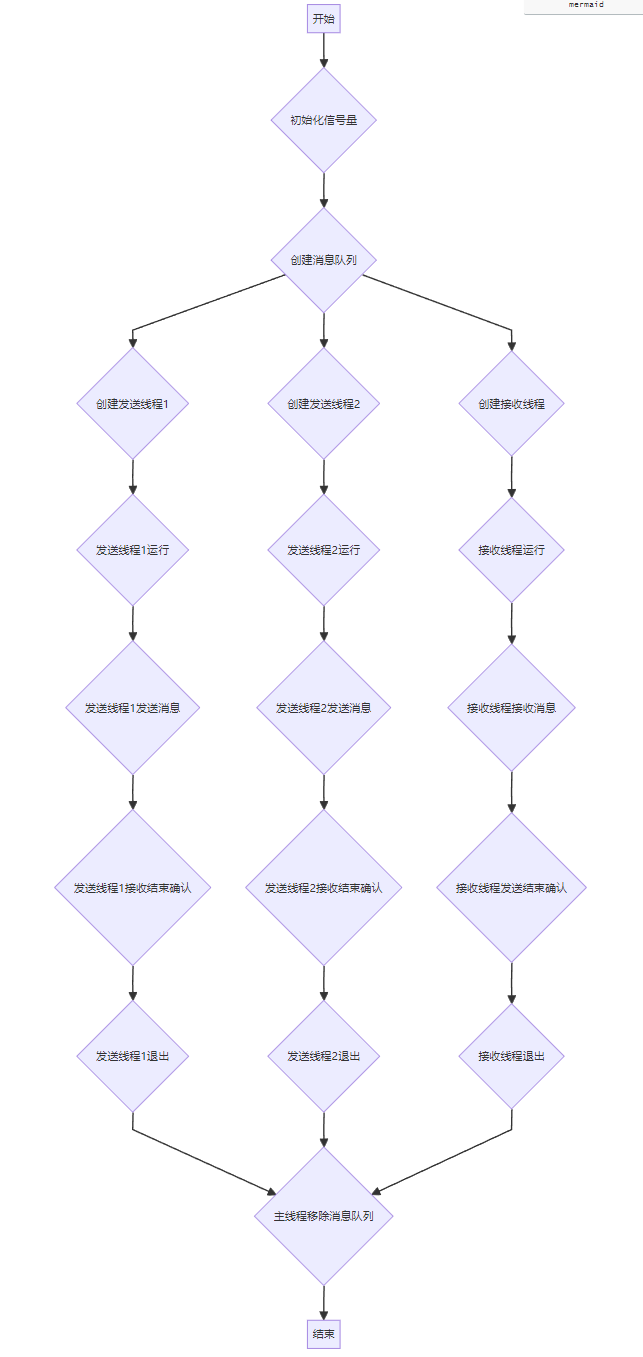
* 通过对 Linux 进程控制的相关系统调用的编程应用，进一步加深对进程概念的理解，明确进程和程序的联系和区别，理解进程并发执行的具体含义。
* 通过 Linux 管道通信机制、消息队列通信机制、共享内存通信机制的应用，加深 对不同类型的进程通信方式的理解。
* 通过对 Linux 的 Posix 信号量及 IPC 信号量的应用，加深对信号量同步机制的理解。
* 请根据自身情况，进一步阅读分析相关系统调用的内核源码实现。

# 二 实验思路

实验思路使用流程图进行表示，四个任务使用四张图表示，流程图使用 mermaid 编写。因为实验流程本身的原因，流程图的截取略有困难，所以没按照顺序来。







# 三 遇到问题及解决方法

1. 编译时错误

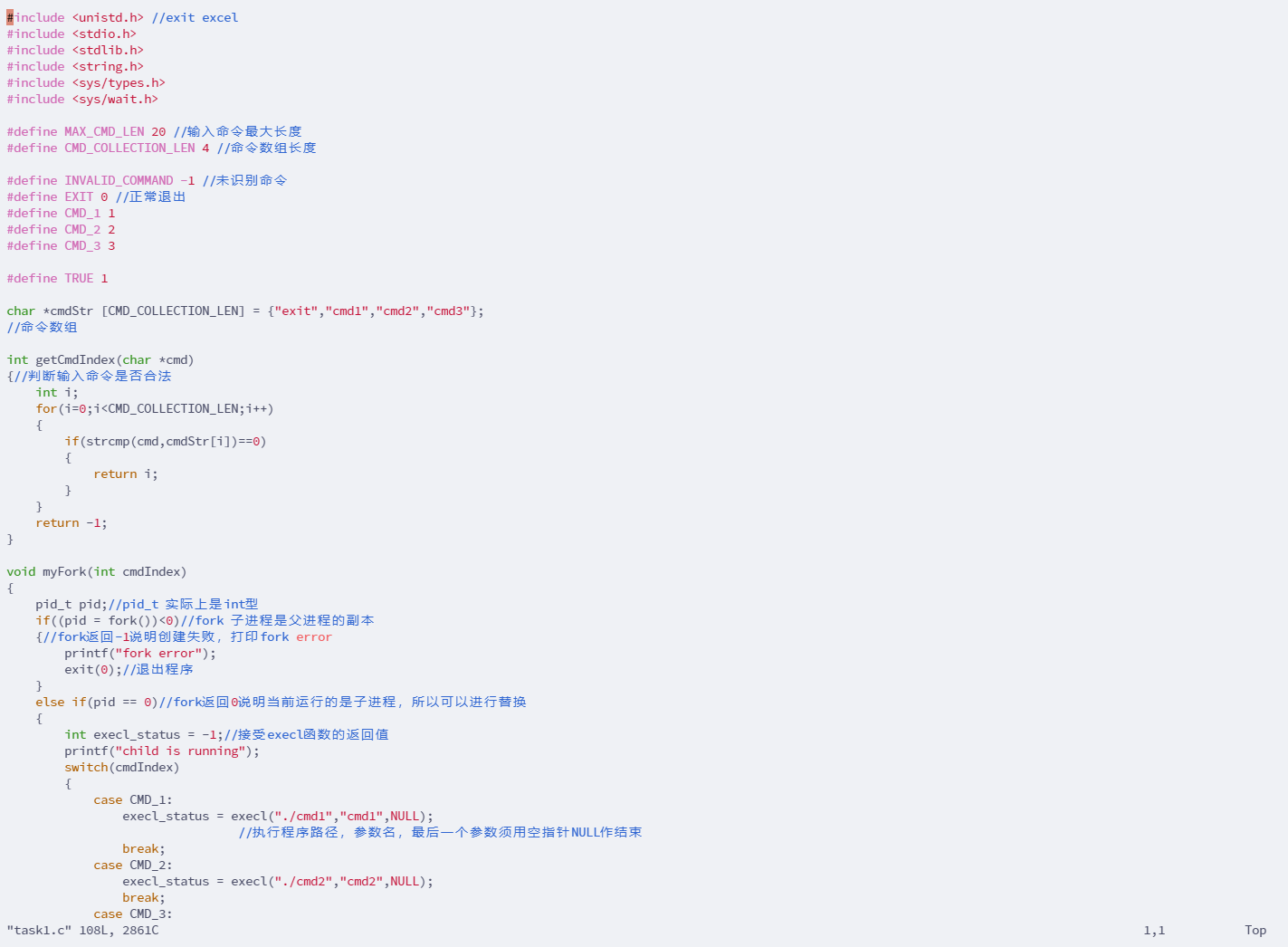
在编译程序时没有链接POSIX线程库。sem\_open、sem\_wait、sem\_post、sem\_close 和 sem\_unlink 这些函数都是 POSIX 信号量相关的函数，需要链接 -lpthread 库才能使用。 使用以下命令来编译代码：

gcc -o main task2.c -lpthread

这个命令会告诉 gcc 编译器在链接阶段加入 -lpthread 选项，以便找到 POSIX 线程库中的信号量相关函数的实现。

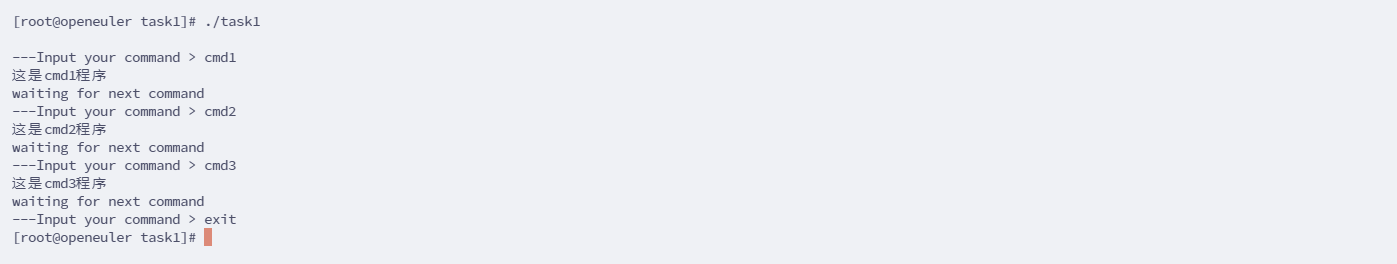
# 四 核心代码及实验结果展示

1. 实现一个模拟的 shell，并在 shell 中自定义三条指令，分别实现执行当前目录下 cmd1、cmd2、cmd3 可执行文件（实际上可根据需求请求系统调用自定义更多可使用指令），以下为核心代码展示：

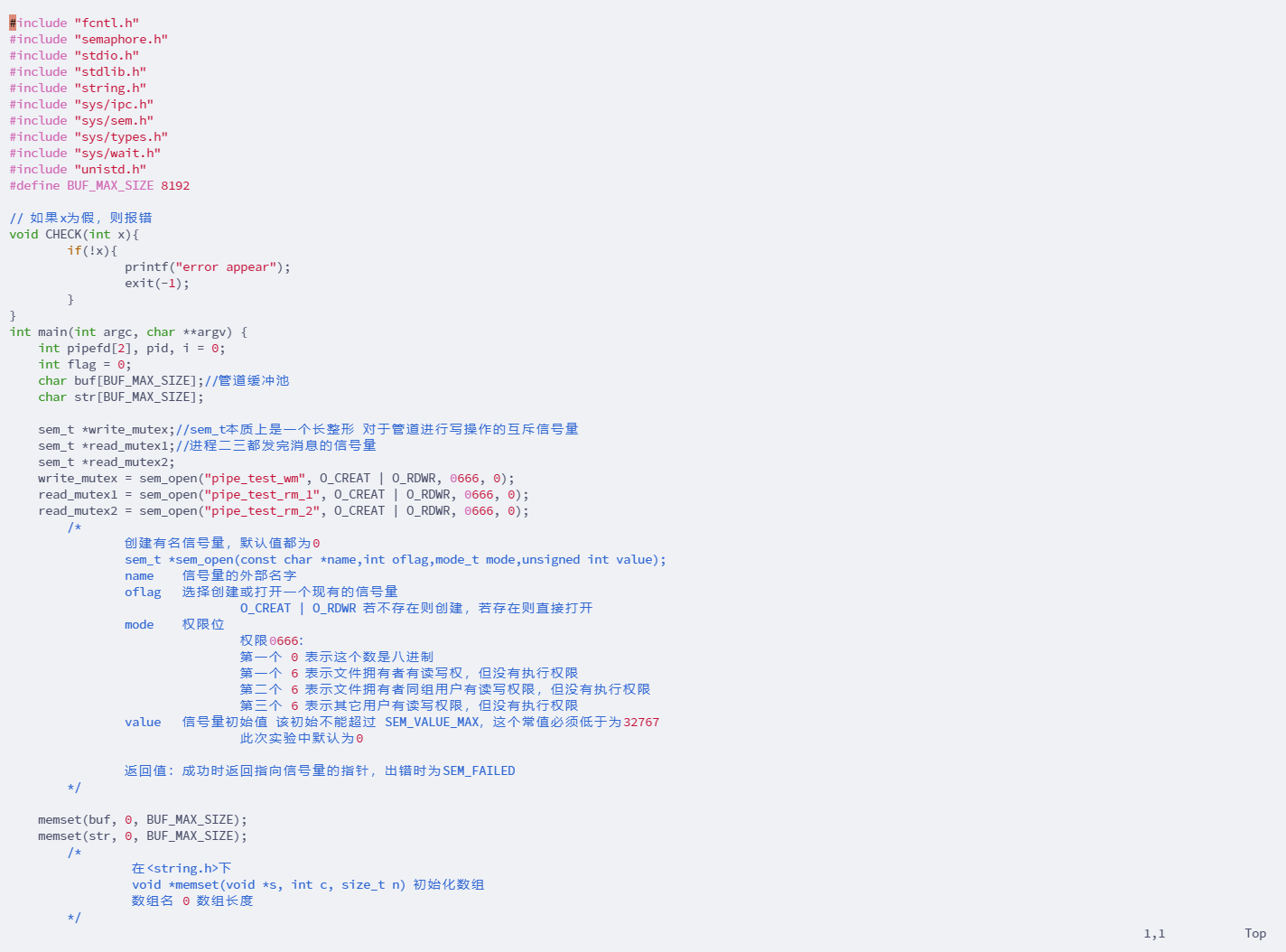


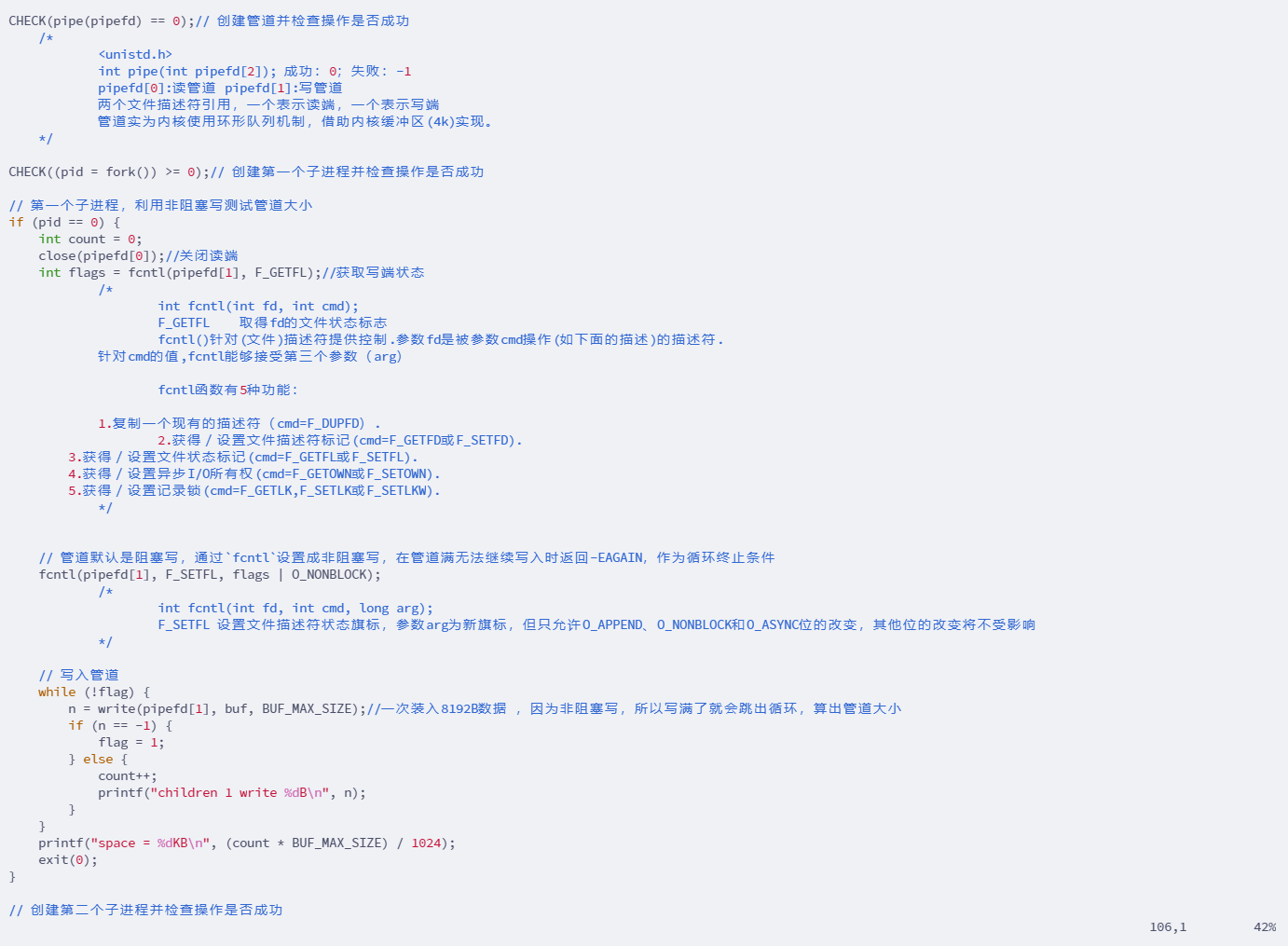


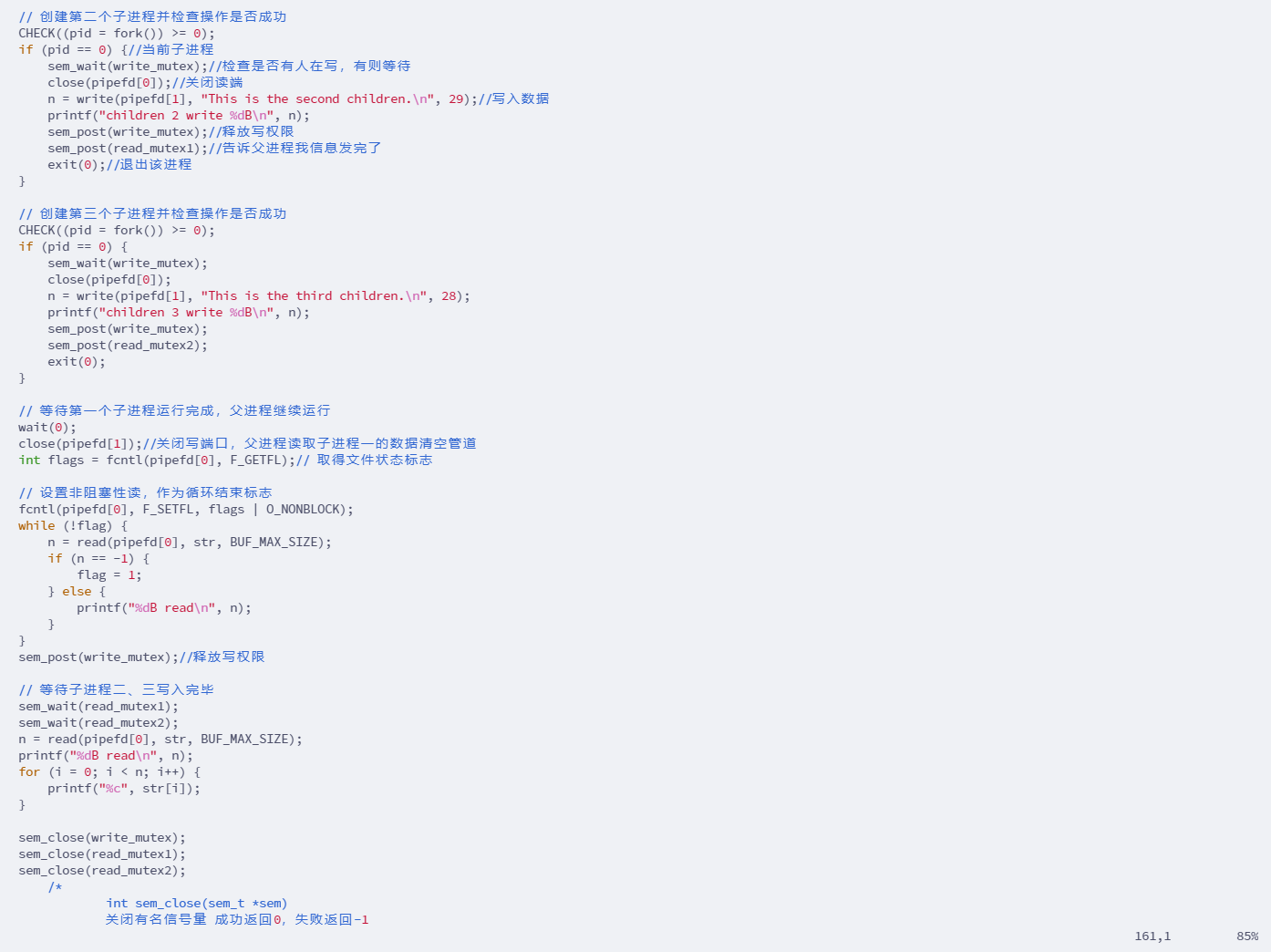
以下展示 shell 启动后执行三条预定义指令，以及退出 shell 的效果：



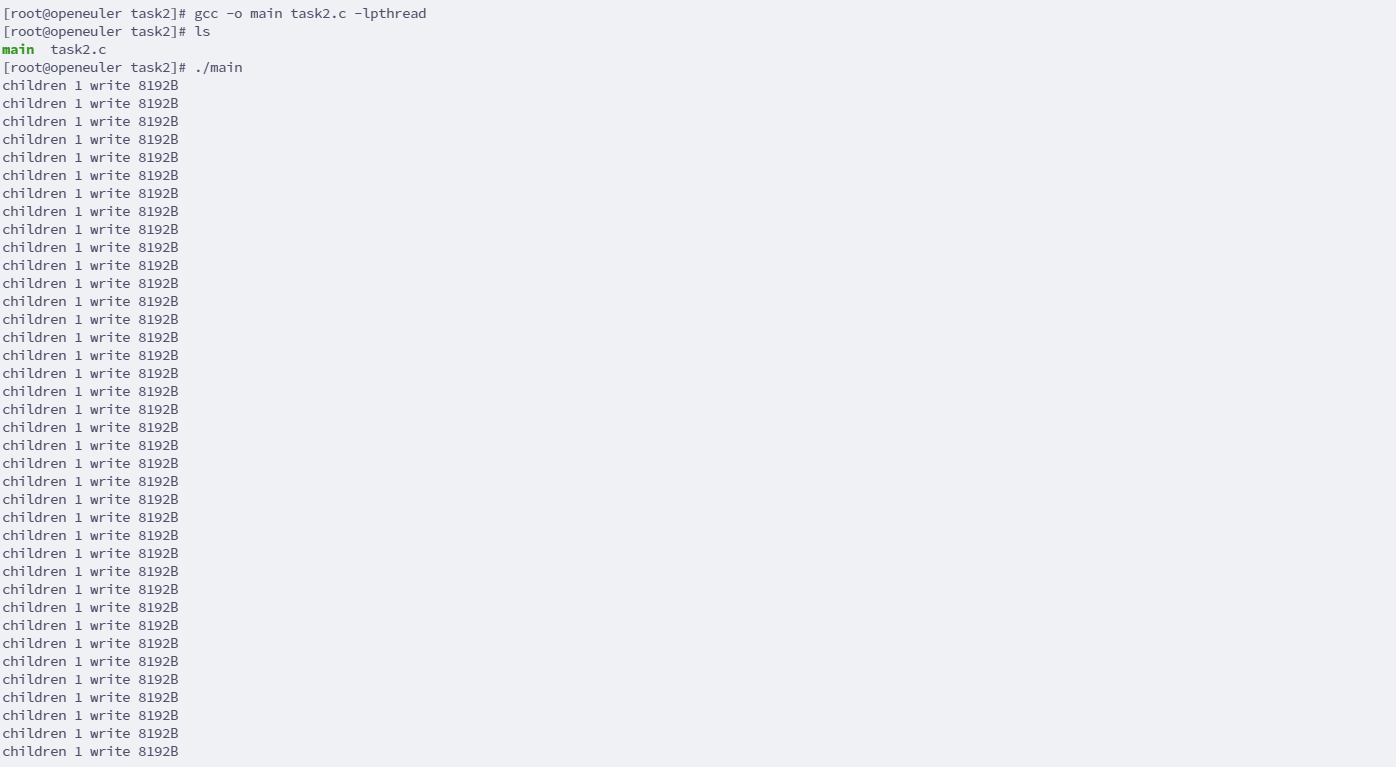
1. 实现管道通信





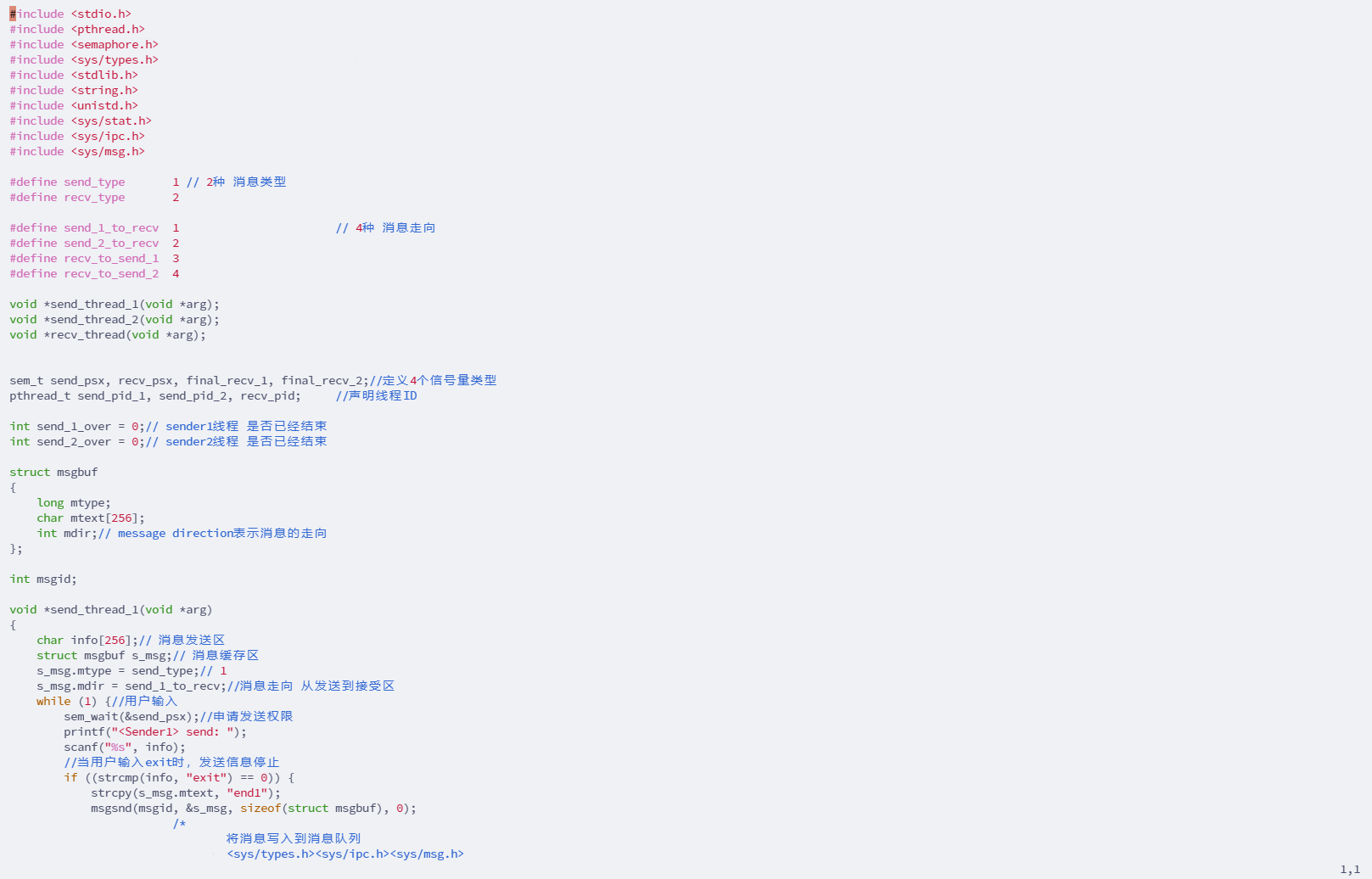


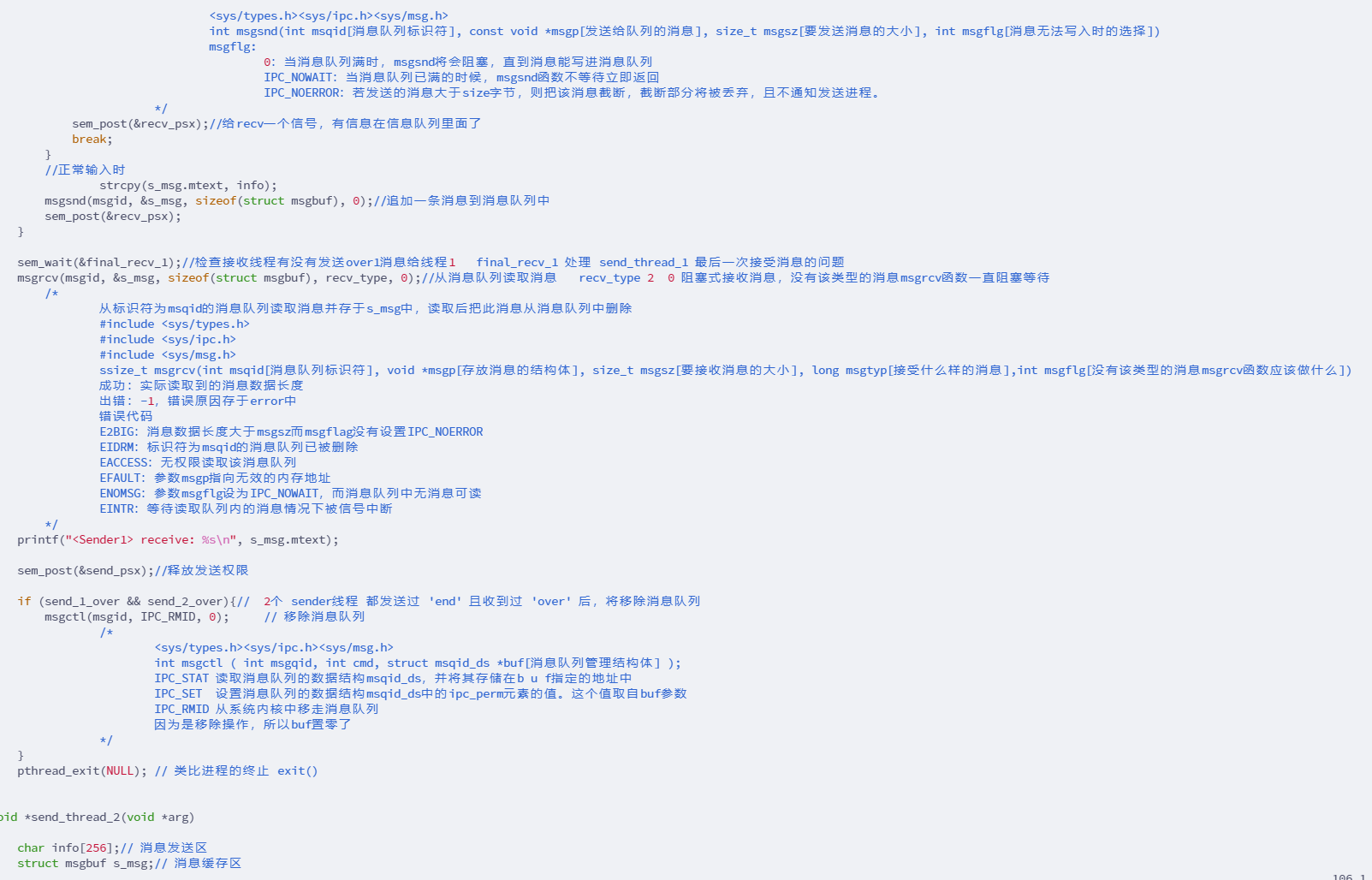
编译时需要链接POSIX线程库，代码中调用的一些函数是POSIX信号量相关的函数，需要链接到-lpthread库才能使用，以下是编译并运行后的代码，因为输出太长所以没有全部截取，截图只截取了部分输出。

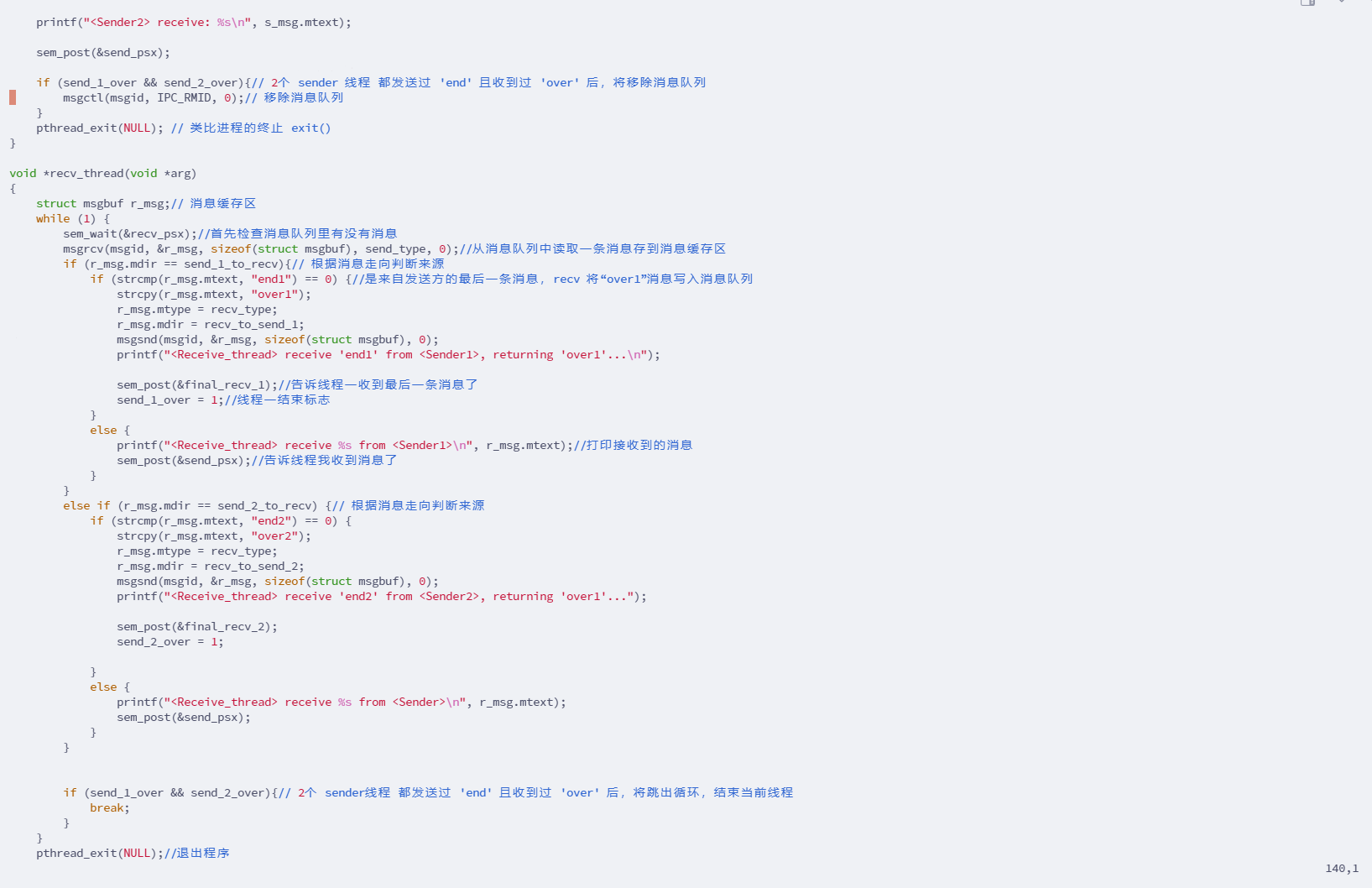


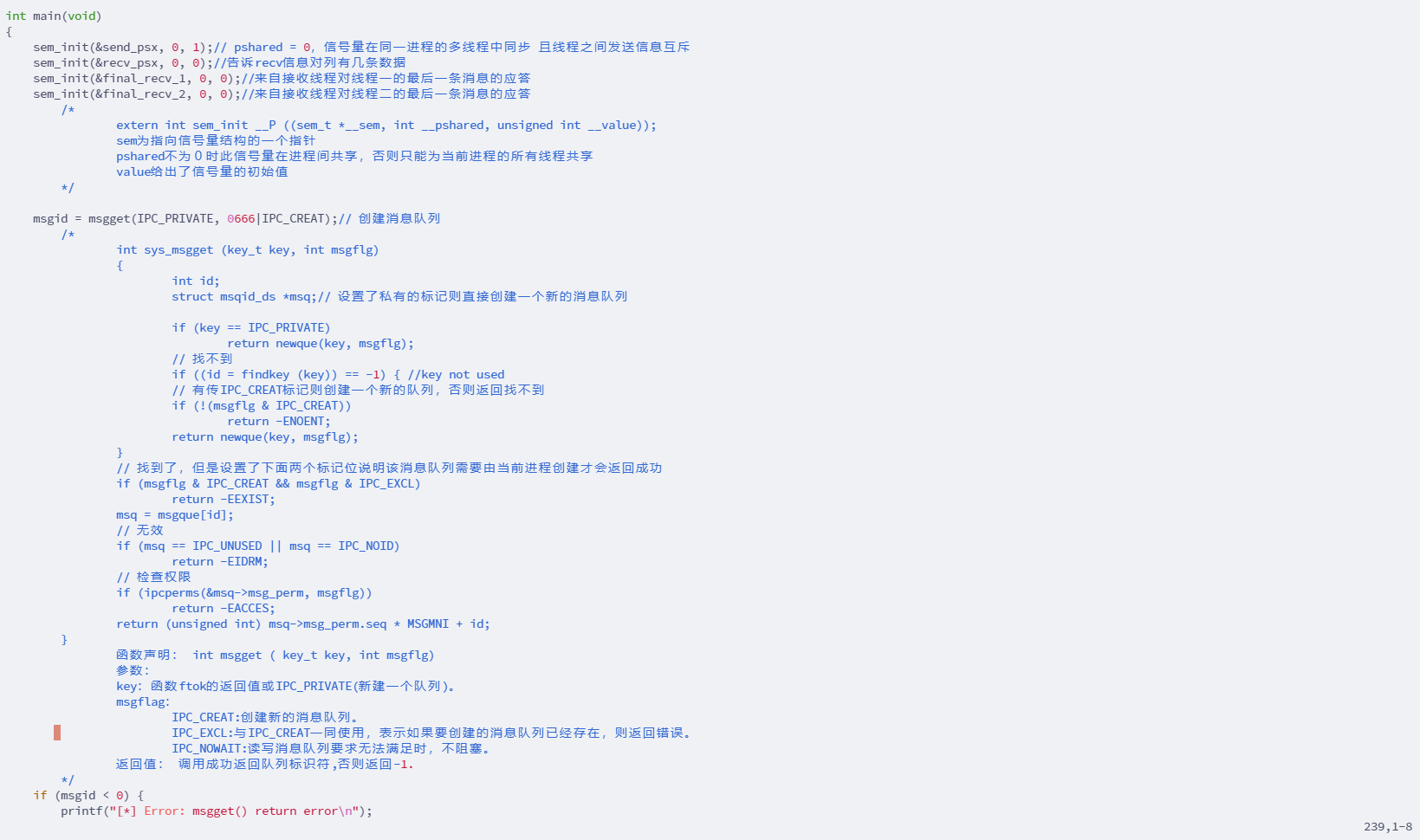
输出内容为子进程的写入和读取操作，以及父进程的管道读取操作，还包含一些错误检查和打印信息。

1. 利用Linux的消息队列通信机制实现了两个线程之间的通信

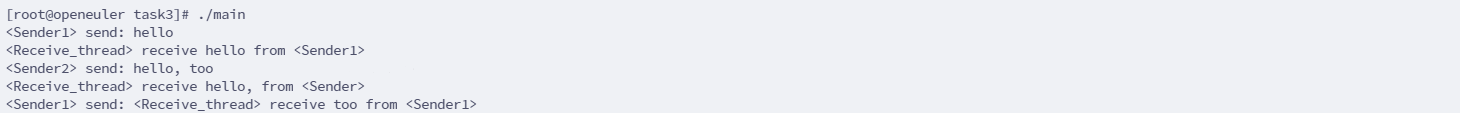






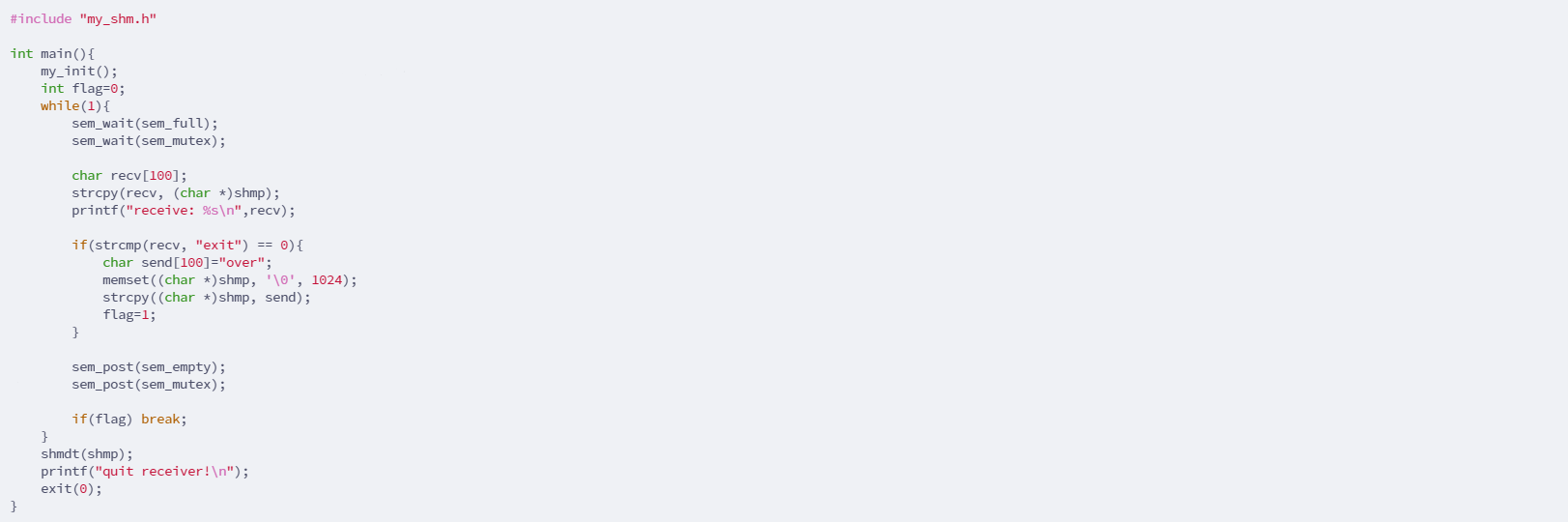


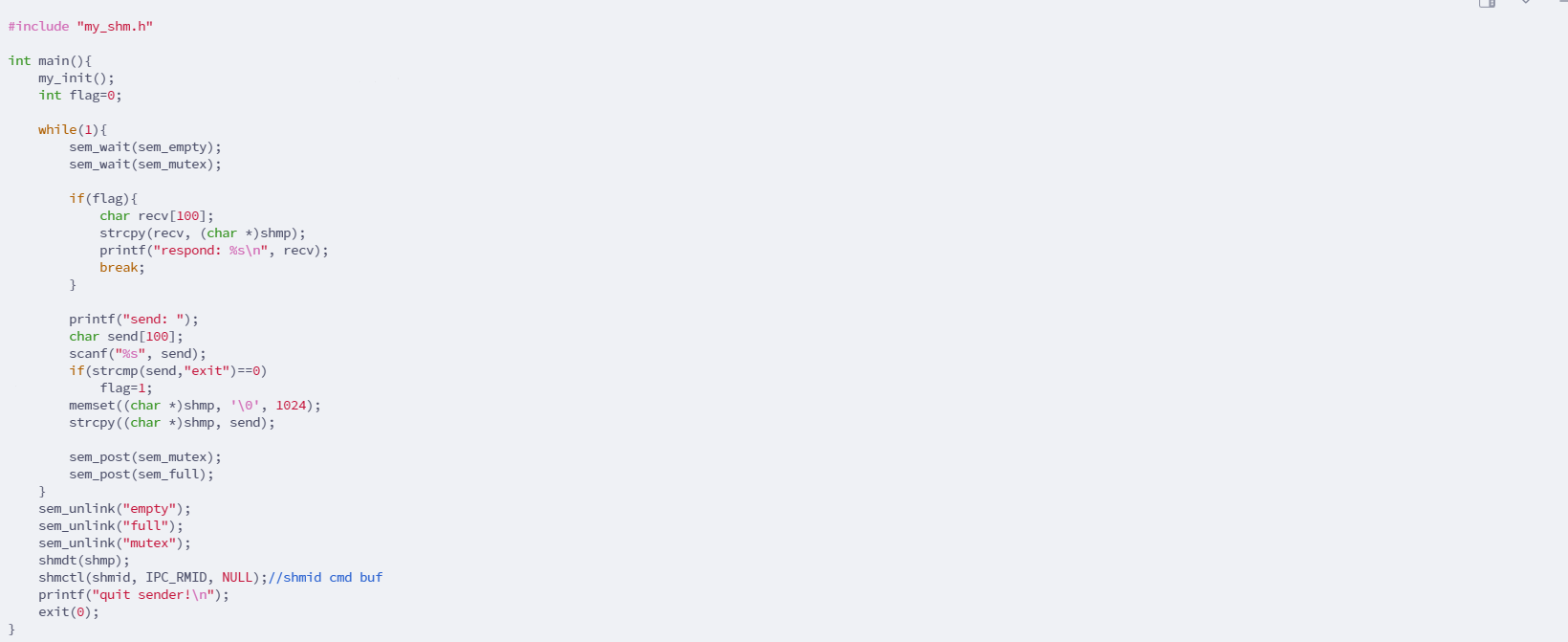
编译时需要链接POSIX线程库，代码中调用的一些函数是POSIX信号量相关的函数，需要链接到-lpthread库才能使用，以下是编译并运行的效果，在两个线程之间发送信息，实验成功：

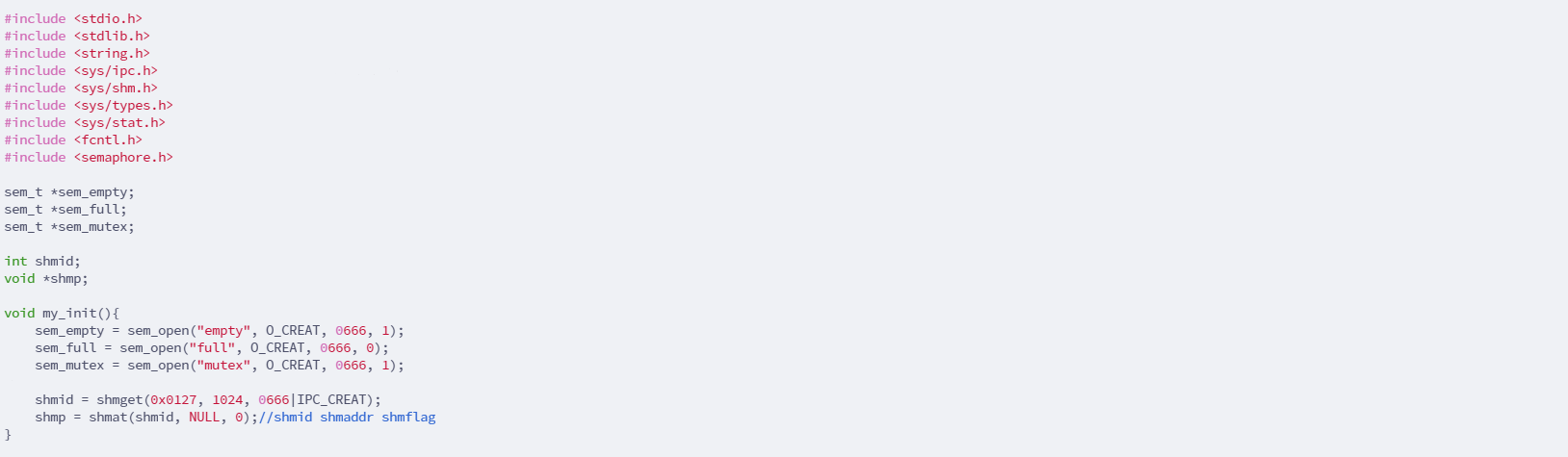


1. 利用Linux的共享内存通信机制实现了两个进程之间的通信

编写了三个文件，分别是 reciever.c sender.c 和 my\_shm.h，前两者编译后实现课题要求，以下为核心代码截图：







编译时需要链接POSIX线程库，代码中调用的一些函数是POSIX信号量相关的函数，需要链接到-lpthread库才能使用，以下是编译并运行的效果，在两个线程之间发送信息，实验成功：



# 五 个人实验改进与总结

## 5.1 个人实验改进

在本次实验中，我主要负责了模拟Shell的编写和消息队列通信机制的实现。通过这次实验，我发现了自己在系统编程方面的不足，也学到了很多宝贵的知识。首先，我意识到了代码的健壮性对于系统级编程的重要性。在实现消息队列通信时，我遇到了线程同步的问题，这让我认识到了同步机制的重要性。因此，我计划深入学习POSIX信号量和IPC信号量，以便更好地控制线程间的同步。

此外，我还计划学习更多关于Linux内核的知识，特别是系统调用的内核源码实现，这将帮助我更深入地理解操作系统的工作原理。我还意识到了团队合作的重要性。在未来的项目中，我会更加注重与队友的沟通和协作，确保每个成员都能理解和掌握所有实验内容。

5.2 个人实验总结

通过本次实验，我不仅提高了自己的编程技能，也加深了对Linux进程控制和进程通信机制的理解。我学会了如何创建和管理Linux进程，如何使用管道、消息队列和共享内存进行进程间通信。我还学习了如何使用信号量进行线程同步，这对于并发编程来说是非常重要的。

实验过程中，我遇到了一些挑战，比如依赖错误和线程同步问题，但我通过查阅文档和与队友讨论找到了解决方案。这些经历不仅增强了我的问题解决能力，也让我更加自信地面对未来的编程挑战。总的来说，这次实验是一次宝贵的学习经历，我期待将这些新知识应用到未来的项目中。

# 六 参考文献

1. [Devv\_](https://devv.ai)