

Национальный исследовательский университет
«Высшая школа экономики»

ОТЧЕТ ПО ЗАДАНИЮ №1
«Использование каналов»
Вариант 4

Выполнил:
студент 213 группы
Фам Ч. Х.

Преподаватель:
Легалов А. И.

Москва
2022

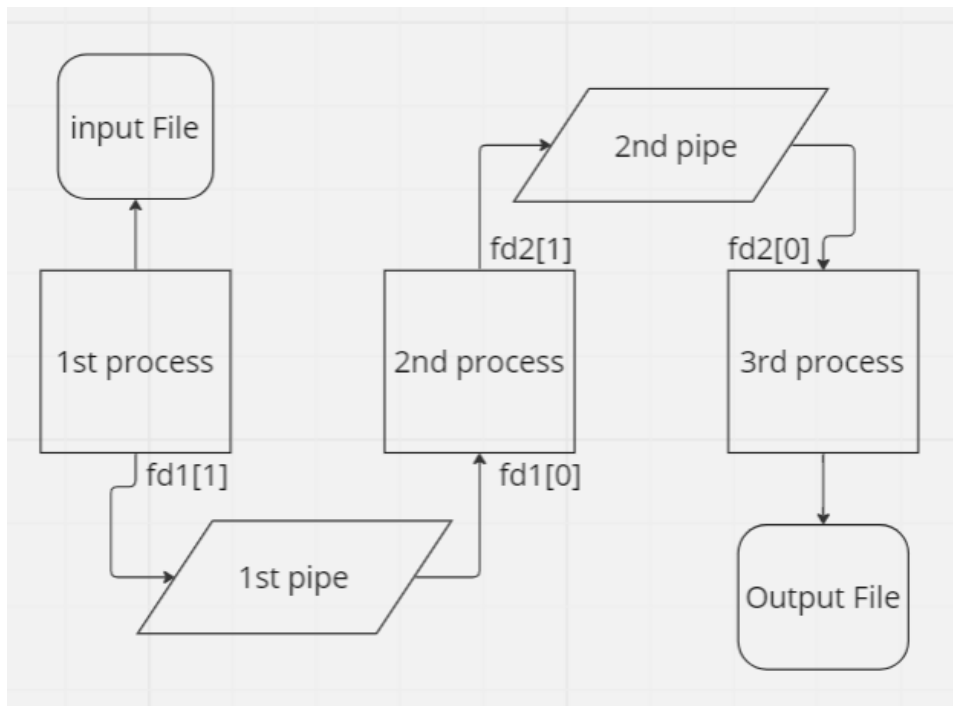
1. Условие задачи

Разработать программу, находящую в заданной ASCII-строке последнюю при перемещении слева направо последовательность N символов, каждый элемент которой определяется по условию «больше предшествующего» (N вводится как отдельный параметр).

2. Отчет на 4 балла

Разработка программы, осуществляющей взаимодействие между тремя дочерними процессами с использованием неименованных каналов

Процессы создаются через `fork()`, каналы создаются с помощью `pipe()`. Первый процесс считывает строку из файла в первый неименованный канал. Второй процесс берет данные из этого канала, обрабатывает их согласно условию и записывает результат во второй неименованный канал. Третий процесс открывает/создает файл для вывода и записывает в него из второго канала результат, полученный от второго процесса.



Пример запуска программы: `./four input.txt output.txt 2`

`./four` – имя программы

`input.txt` – входной файл

`output.txt` – файл с результатом

Размер буфера равняется ровно 5000 символов, весь ввод и вывод осуществлен через системные вызовы `read` и `write`

Тесты (Строка, параметр n, результат):

1. abc, 3, abc
2. abcdefg, 4, defg
3. zzzabczzzbcd, 3, bcd
4. gfedcba, 3,
5. aboba, 2, bo

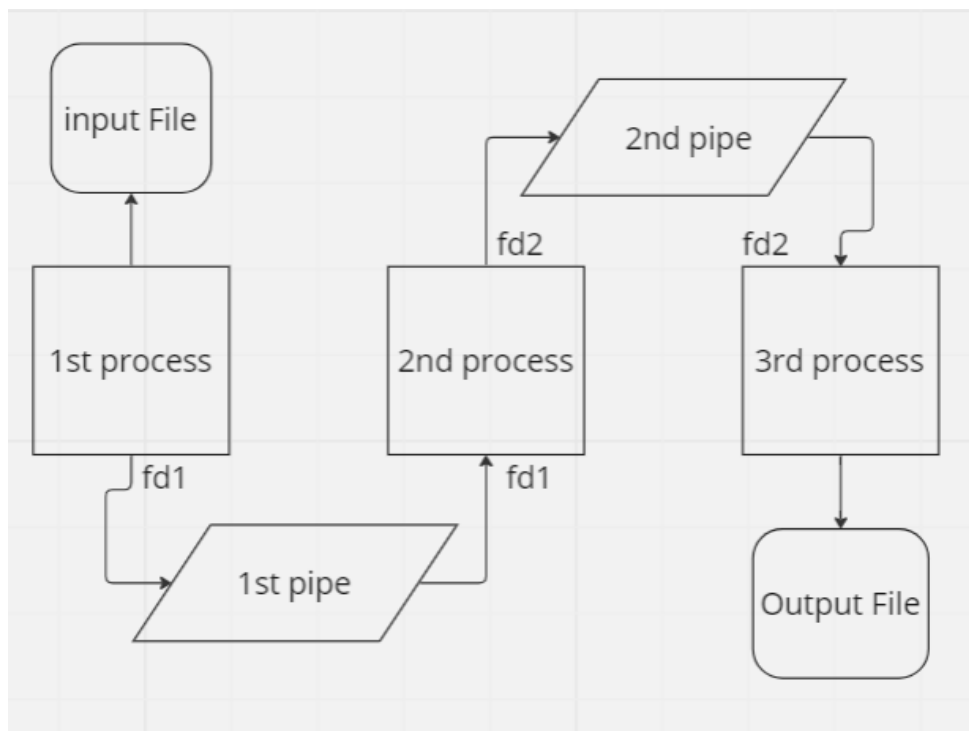
Эти тесты будут использоваться для каждой из программ. Результаты тестов в файле “tests_4”.

3. Отчет на 5 баллов

В дополнение к требованиям на предыдущую оценку необходимо разработать программу, в которой взаимодействие между тремя дочерними процессами осуществляется через именованные каналы

Имена каналов: первый - “FIFO_1”, второй - “FIFO_2”. Создаются каналы через mkfifo() с модификатором доступа 0666.

Первый процесс считывает строку из файла в первый неименованный канал. Второй процесс берет данные из этого канала, обрабатывает их согласно условию и записывает результат во второй неименованный канал. Третий процесс открывает/создает файл для вывода и записывает в него из второго канала результат, полученный от второго процесса. В отличие от неименованных каналов, именованные являются двусторонними, и теперь не придется закрывать одну из сторон.



Пример запуска программы: `./five input.txt output.txt 2`

`./five` – имя программы

`input.txt` – входной файл

`output.txt` – файл с результатом

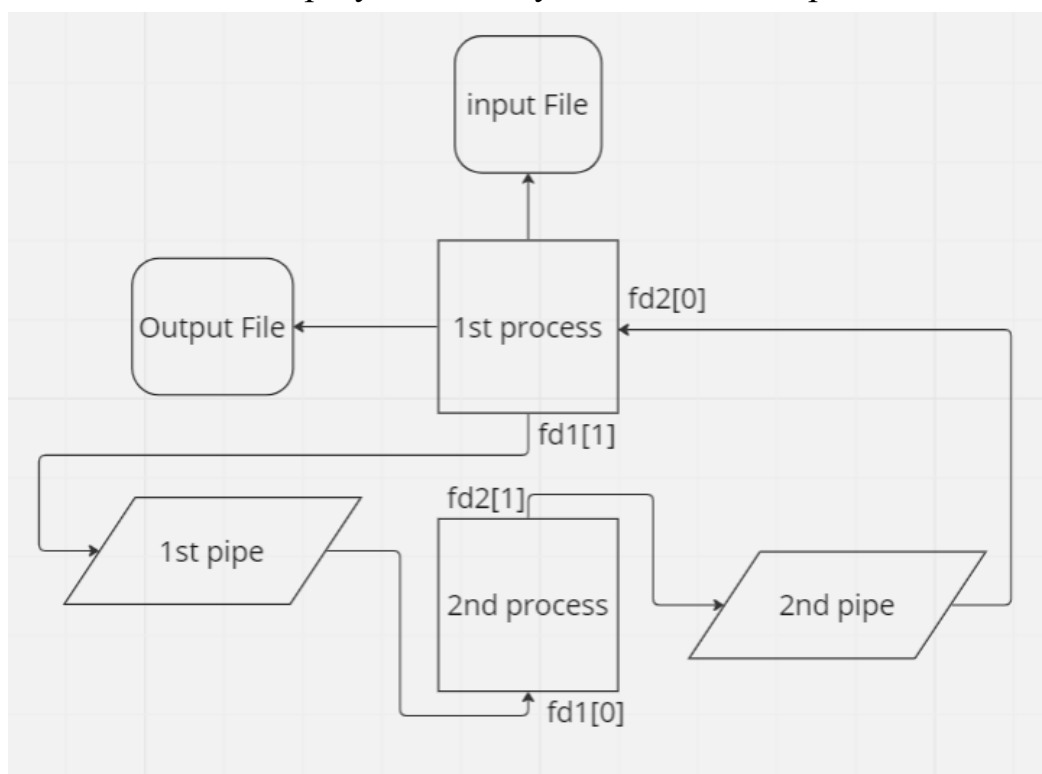
Размер буфера равняется ровно 5000 символов, весь ввод и вывод осуществлен через системные вызовы `read` и `write`

Тесты в папке “five”.

4. Отчет на 6 баллов

В дополнение к требованиям на предыдущую оценку разработать программу, которая осуществляет взаимодействие между двумя дочерними процессами с использованием неименованных каналов

Имена каналов `fd1[2]`, `fd2[2]`. Каналы создаются через `pipe()`. Первый процесс считывает данные из входного файла и передает их на обработку через 1ый неименованный канал во 2ой процесс. Второй процесс считывает из канала данные, обрабатывает их согласно условию и через второй канал отдает результат обратно первому процессу. Первый процесс создает результирующий файл и записывает в него результат, полученный от 2го процесса.



Пример запуска программы: `./six input.txt output.txt 2`

`./six` – имя программы

`input.txt` – входной файл

output.txt – файл с результатом

Размер буфера равняется ровно 5000 символов, весь ввод и вывод осуществлен через системные вызовы read и write

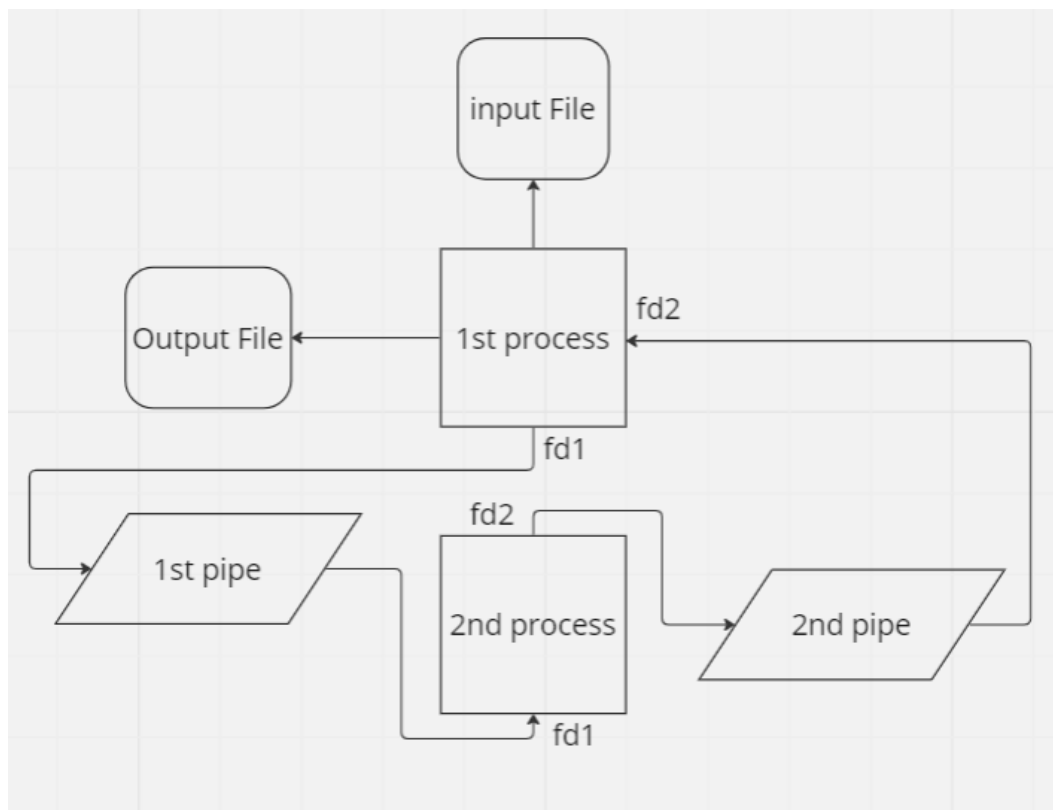
Тесты в папке “six”.

5. Отчет на 7 баллов

В дополнение к требованиям на предыдущую оценку разработать программу, которая осуществляет взаимодействие между двумя дочерними процессами с использованием именованных каналов

Имена каналов: первый - “FIFO_1”, второй - “FIFO_2”. Создаются каналы через mkfifo() с модификатором доступа 0666.

Первый процесс считывает данные из входного файла и передает их на обработку через 1ый неименованный канал во 2ой процесс. Второй процесс считывает из канала данные, обрабатывает их согласно условию и через второй канал отдает результат обратно первому процессу. Первый процесс создает результирующий файл и записывает в него результат, полученный от 2го процесса.



Пример запуска программы: ./seven input.txt output.txt 2

./seven – имя программы

input.txt – входной файл

output.txt – файл с результатом

Размер буфера равняется ровно 5000 символов, весь ввод и вывод осуществлен через системные вызовы read и write
Тесты в папке “seven”.