Именованные каналы

Аникин Константин Сергеевич

# Цель работы

Приобретение практических навыков работы с именованными каналами.

# Задание

* Ознакомиться с теоретическим материалом.
* Написать три скрипта из задания.

# Теоретическое введение

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. [[1]](https://www.opennet.ru/docs/RUS/linux_parallel/node17.html)

В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общеюниксные (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты).

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью каналов довольно трудно организовать обмен асинхронными сообщениями между процессами.

Краткий справочник команд Linux [[2]](https://hpc.icc.ru/documentation/cmnds.pdf)

# Выполнение лабораторной работы

## Скрипт 1

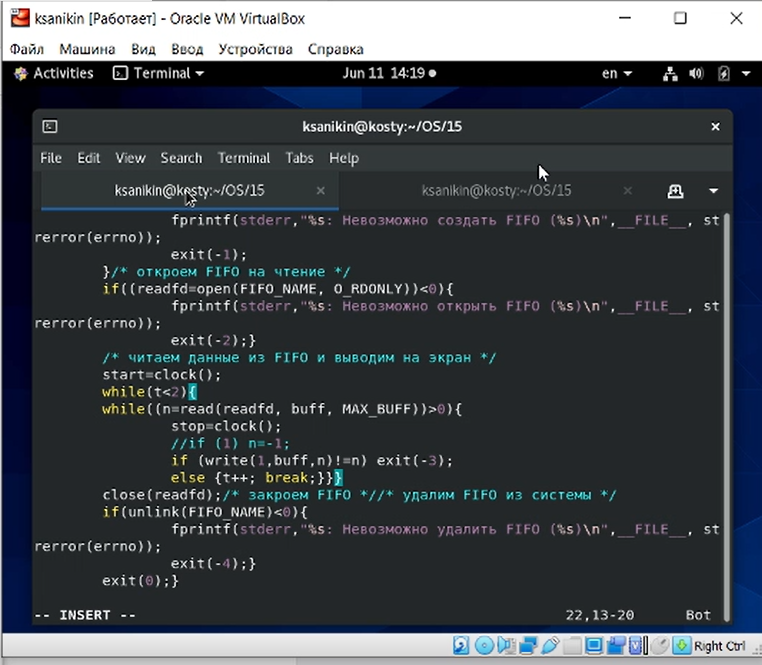
Полностью скрипты приводить не буду, покажу лишь места, в которые внесены изменения.

Непосредственно server.c. (рис. 1)

Приём нескольих клиентов обеспечивается через переменную t.

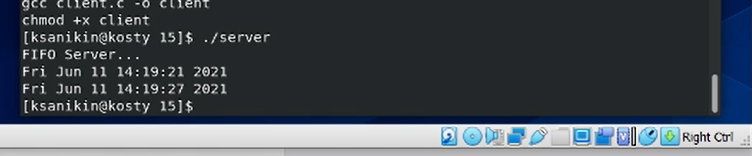
Когда мы получили первое сообщение, мы увеличиваем t на 1 и выходим из цикла read.

И так до тех пор, пока мы не получили t сообщений.



*Рис. 1: Скрипт 1*

Результаты работы скрипта 1. Второй раз клиент был запущен через 6 секунд. (рис. 2)



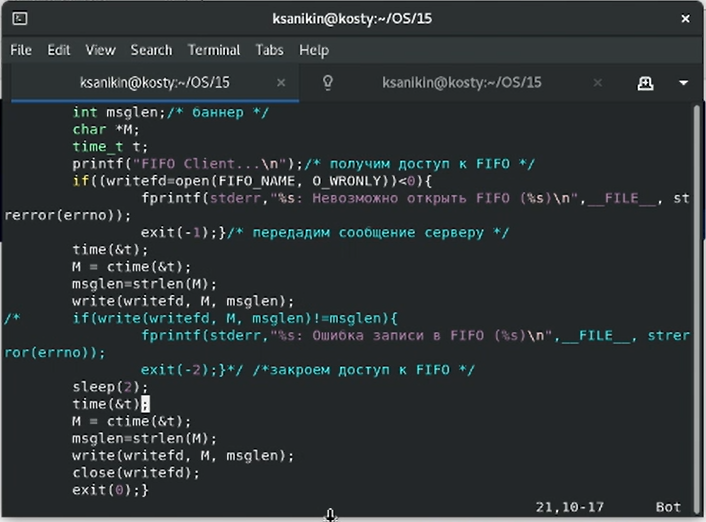
*Рис. 2: Результаты работы скрипта 1*

## Скрипт 2

Непосредственно client.c. (рис. 3)

Получаем текущее время и записываем его в файл.

Потом спим 2 секунды и повторяем операцию.



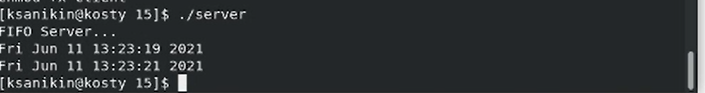
*Рис. 3: Скрипт 2*

Клиент закрывается не сразу(рис. 4)



*Рис. 4: Работаюший клиент*

Результаты работы скрипта 2. Между записями времени прошло 2 секунды. (рис. 5)



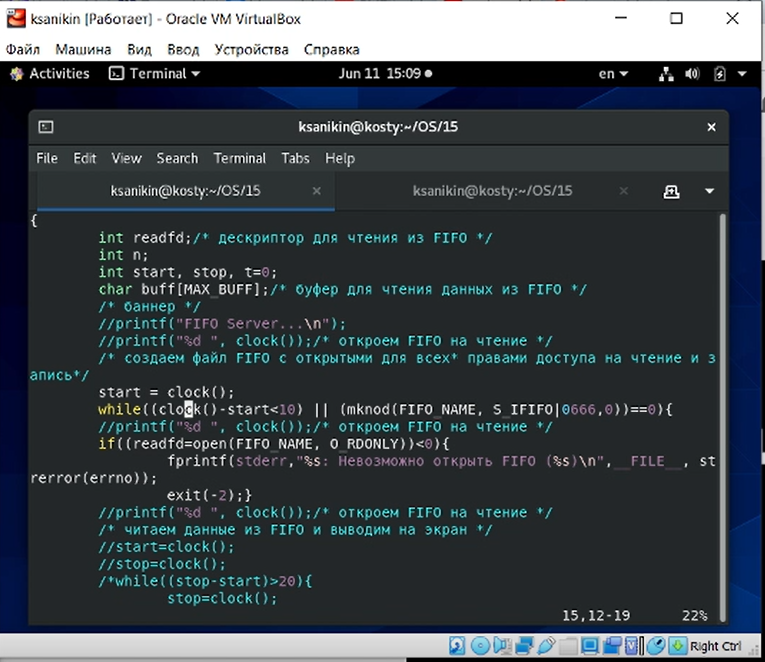
*Рис. 5: Результаты работы скрипта 2*

## Скрипт 3

Скрипт не получился, однако были продвижения. (рис. 6)

Я понял, что мне нужно закрыть программу либо после 10 секунд, либо после

подключения клиента. Но как это сделать - я так и не понял.

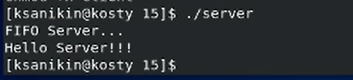


*Рис. 6: Скрипт 3*

На рисунке этого не видно, я взял другой момент, но сервер может

вывести только баннер, но закроется он всё равно лишь после подключения клиента.

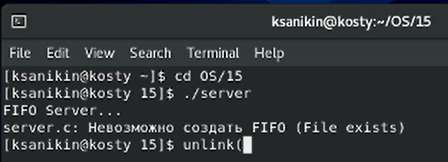
Выведется ли Hello или нет зависит от скорости подключения. (рис. 7)



*Рис. 7: Результаты работы скрипта 3*

Если закрыть сервер нештатно, то временный файл не будет удален.

Новый сервер не сможет создать канал, так как он уже будет существовать. (рис. 8)



*Рис. 8: Неразлинкованный fifo*

# Выводы

Скрипты созданы и почти все работают. Работа выполнена на 80%.

## Библиографический список

[1]: [FIFO - именованные каналы](https://www.opennet.ru/docs/RUS/linux_parallel/node17.html)

[2]: [Краткий справочник команд Linux](https://hpc.icc.ru/documentation/cmnds.pdf)