Отчет по лабораторной работе 15

Именованные каналы

Шалыгин Георгий Эдуардович, НФИбд-02-20

Содержание

# Цель работы

Приобретение практических навыков работы с именованными каналами

Объект исследования: система UNIX.

Предмет исследования: работа с именованными каналами в UNIX.

# Техническое обеспечение:

* Характеристики техники: AMD Ryzen 5 3500U 2.1 GHz, 8 GB оперативной памяти, 50 GB свободного места на жёстком диске;
* ОС Windows 10 Home
* Git 2.31.1
* Google Chrome 91.0.4472.19
* VirtualBox 2.0
* CentOS 7

# Условные обозначения и термины:

**Именованный канал** или **именованный конвейер** (англ. *named pipe*) — один из методов [межпроцессного взаимодействия](https://ru.wikipedia.org/wiki/Межпроцессное_взаимодействие), расширение понятия [конвейера](https://ru.wikipedia.org/wiki/Конвейер_(UNIX)) в [Unix](https://ru.wikipedia.org/wiki/Unix) и подобных [ОС](https://ru.wikipedia.org/wiki/Операционная_система).[1]

**Сокет домена Unix** или **IPC-сокет** (сокет межпроцессного взаимодействия) — конечная точка обмена данными, подобная [Интернет-сокету](https://ru.wikipedia.org/wiki/Сокет_(программный_интерфейс)), но не использующая сетевого протокола для взаимодействия (обмена данными)

**Командный язык** - это язык, на котором пользователь взаимодействует с системой в интерактивном режиме.

**Командный интерпретатор**, интерпретатор командной строки - компьютерная программа, часть операционной системы, обеспечивающая базовые возможности управления компьютером посредством интерактивного ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов.[3]

Подробнее в [2] и [3].

# Теоретическое введение:

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому.

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.

Файлы именованных каналов создаются функцией mkfifo(3).

Первый параметр — имя файла, идентифицирующего канал, второй параметр — маска прав доступа к файлу.

После создания файла канала процессы, участвующие в обмене данными, должны открыть этот файл либо для записи, либо для чтения. При закрытии файла сам канал продолжает существовать. Для того чтобы закрыть сам канал, нужно удалить его файл, например с помощью вызова unlink(2).

Рассмотрим работу именованного канала на примере системы клиент–сервер. Сервер создаёт канал, читает из него текст, посылаемый клиентом, и выводит его на терминал.

Вызов функции mkfifo() создаёт файл канала (с именем, заданным макросом FIFO\_NAME):

mkfifo(FIFO\_NAME, 0600);

В качестве маски доступа используется восьмеричное значение 0600, разрешающее процессу с аналогичными реквизитами пользователя чтение и запись. Можно также установить права доступа 0666. Открываем созданный файл для чтения:

f = fopen(FIFO\_NAME, O\_RDONLY);

Ждём сообщение от клиента. Сообщение читаем с помощью функции read() и печатаем на экран. После этого удаляется файл FIFO\_NAME и сервер прекращает работу. Клиент открывает FIFO для записи как обычный файл:

f = fopen(FIFO\_NAME, O\_WRONLY);

Посылаем сообщение серверу с помощью функции write().

Для создания файла FIFO можно использовать более общую функцию mknod(2), предназначенную для создания специальных файлов различных типов (FIFO, сокеты, файлы устройств и обычные файлы для хранения данных).

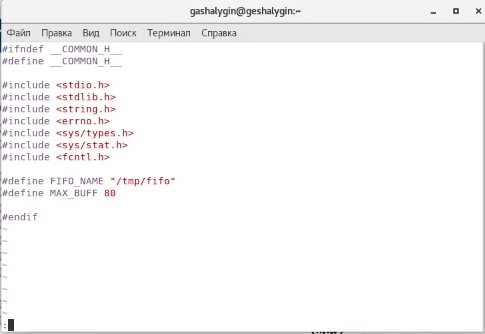
int mknod(const char \*pathname, mode\_t mode, dev\_t dev);

Тогда, вместо mkfifo(FIFO\_NAME, 0600); пишем mknod(FIFO\_NAME, S\_IFIFO | 0600, 0);

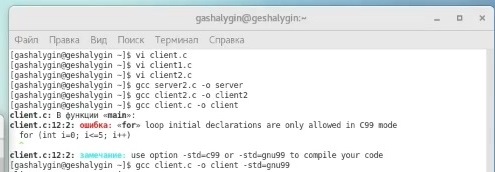
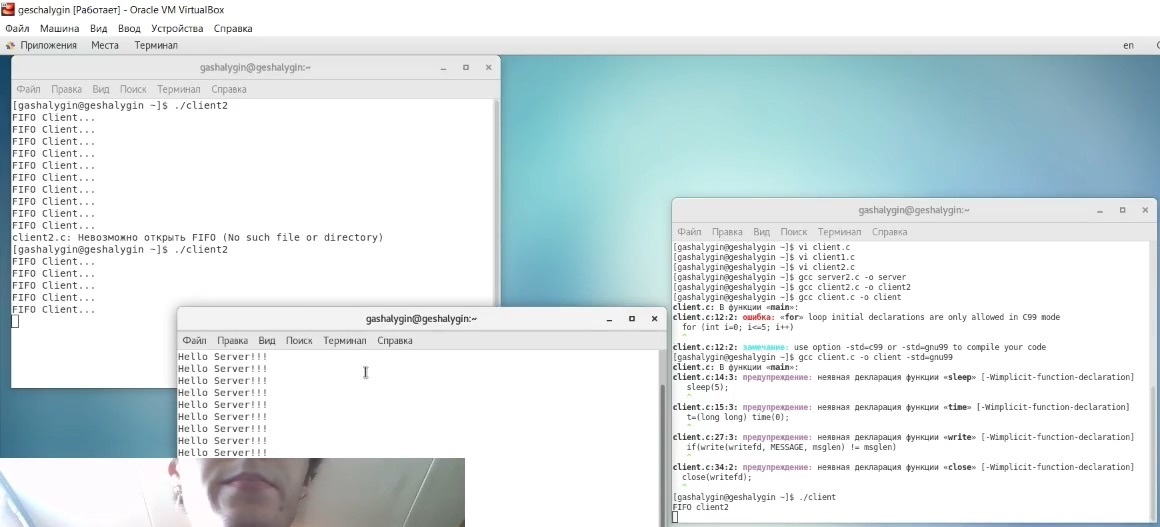
Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью каналов довольно трудно организовать обмен асинхронными сообщениями между процессами.

# Выполнение лабораторной работы

1. Создадим файлы: common.h, server.c, client.c, client2.c для работы с именованными каналами со следующими критериями:
   * Работает не 1 клиент, а несколько (например, два).
   * Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию sleep() для приостановки работы клиента.
   * Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера.

* В случае, если сервер завершит работу, не закрыв канал, в дальнейшем его не удастся создать, т.к. не удалятся временные файлы.
* (рис. 1, 2, 3, 4 ).
* 
* Figure 1: Текст commoh.h
* 
* Figure 2: Текст client2.c
* 
* Figure 3: Текст client.c
* 
* Figure 4: Текст client.c

1. Выполним компиляцию программы посредством gcc. (рис. 5).

* Запустим программы в разных терминалах и увидим взаимодействие между ними на рис. 6.
* 
* Figure 5: Компиляция
* 
* Figure 6: Запуск

# Выводы

В процессе работы над лабораторной работы были изучены основы программирования в оболочке ОС UNIX, получен опыт работы с именованными каналами.

# Библиография

1. https://parallel.uran.ru/book/export/html/464
2. https://www.opennet.ru/docs/RUS/linux\_parallel/node17.html
3. Д.С. Кулябов, А.В. Королькова / Администрирование локальных систем. Лабораторные работы. — М.: Российский университет дружбы народов, 2017. — 119 с.