отчёта по лабораторной работе 15

Ндри Ив Алла Ролан

Содержание

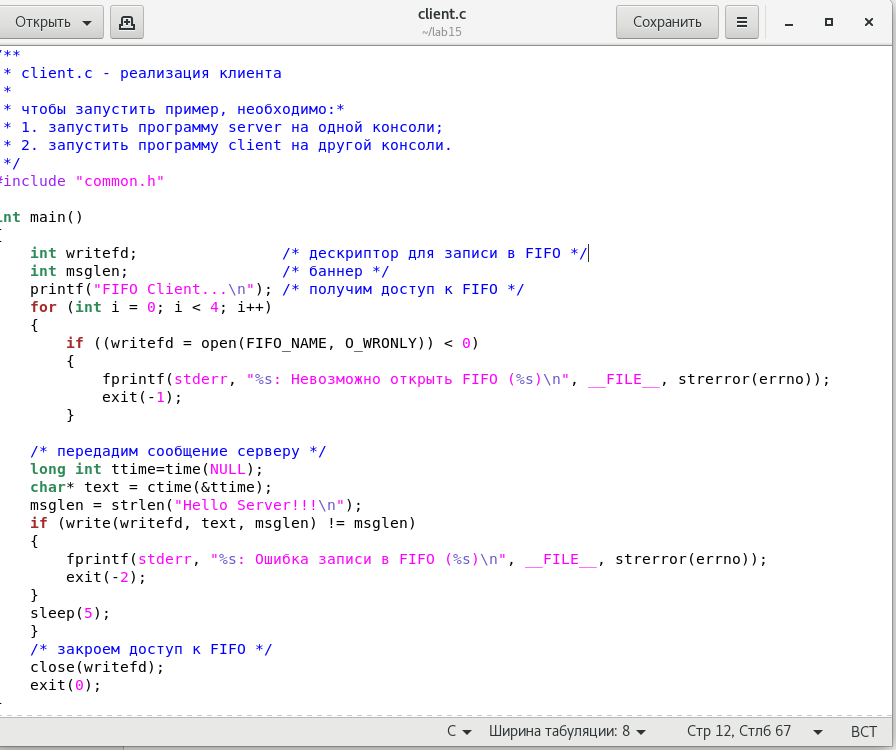
## Цель работы

Приобретение практических навыков работы с именованными каналами.

## Выполнение работы.

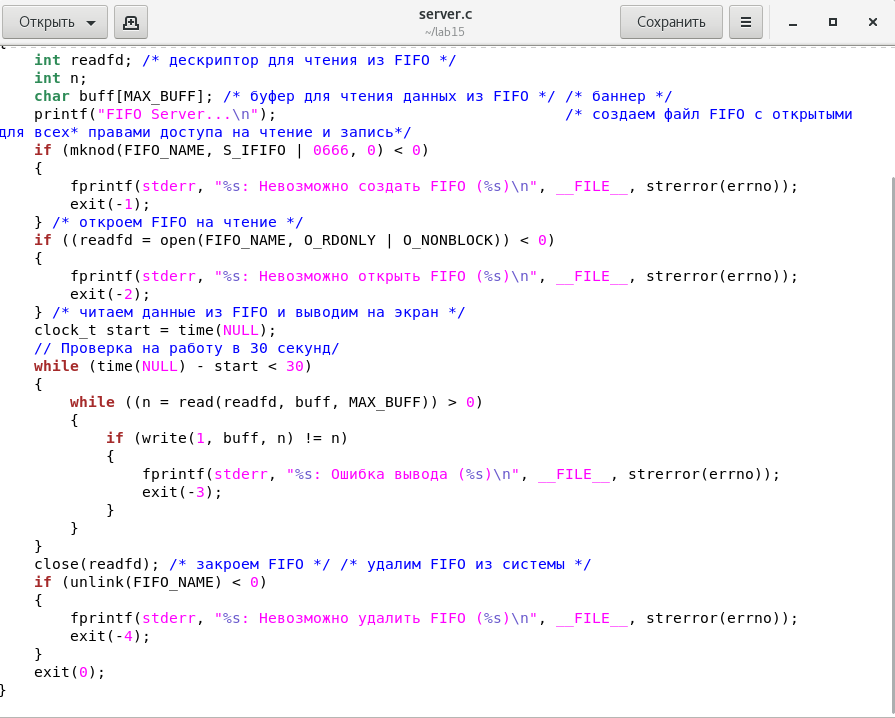
Ход работы: 1. Мы изучили и написали приведенные в тексте задания к лабораторной работе программы common.h, server.c и client.c

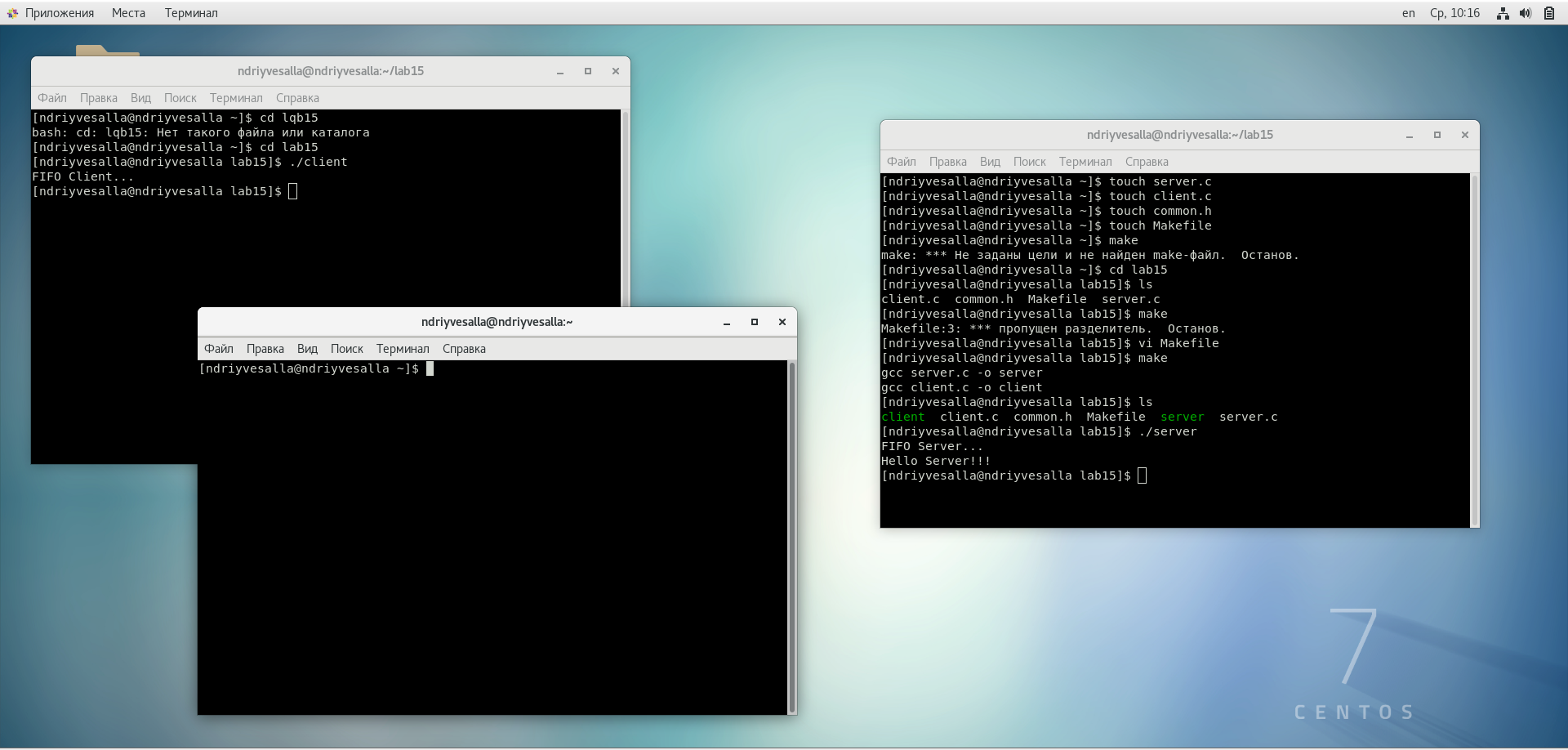
(рис. **¿fig:001?**)

 { # fig:001 width=70%} (рис. **¿fig:002?**)

 { # fig:002 width=70%}

(рис. **¿fig:003?**)

 { # fig:003 width=70%}

 { # fig:004 width=70%}

Как видно сервер откликается через 5 секунд от каждого нового запуска клиента, а через 30 секунд программа завершается с выводом общего количества времени и интервал от последнего отклика клиента до завершения сервера

Если сервер завершит работу не закрыв канал, то при следующем запуске будет выведено следующее (errno)

## Вывод:

Написали FIFO канал на C (server/client)

## Ответы на контрольные вопросы:

1. В чем ключевое отличие именованных каналов от неименованных? Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла).
2. Возможно ли создание неименованного канала из командной строки? Неименованные каналы создаются вызовом pipe() и предоставляют возможность только однонаправленной (односторонней) передачи данных:

#include <unistd.h> int fd[2]; pipe(fd);

/\* возвращает 0 в случае успешного завершения, -1 - в случае ошибки;\*/

Функция возвращает два файловых дескриптора: fd[0] и fd[l], причемпервый открыт для чтения, а второй – для записи.

Возможно создание неименованного канала из командной строки только с созданием временного именнованного канала.

1. Возможно ли создание именованного канала из командной строки?

Да, с помощью mkfifo или, использованной с давних времен, mknod

1. Опишите функцию языка С, создающую неименованный канал. #include <unistd.h>

int fd[2]; pipe(fd);

/\* возвращает 0 в случае успешного завершения, -1 - в случае ошибки;\*/

Функция возвращает два файловых дескриптора: fd[0] и fd[l], причемпервый открыт для чтения, а второй – для записи.

1. Опишите функцию языка С, создающую именованный канал. #include <sys/types.h>

#include <sys/stat.h>

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

1. Что будет в случае прочтения из fifo меньшего числа байтов, чем находится в канале? Большего числа байтов?

При чтении меньшего числа байтов, чем находится в канале или FIFO,

возвращается требуемое число байтов, остаток сохраняется для последующих чтений.

При чтении большего числа байтов, чем находится в канале или FIFO, возвращается доступное число байтов. Процесс, читающий из канала, должен соответствующим образом обработать ситуацию, когда прочитано меньше, чем заказано.

1. Аналогично, что будет в случае записи в fifo меньшего числа байтов, чем позво-ляет буфер? Большего числа байтов?

Запись числа байтов, меньшего емкости канала или FIFO, гарантированно атомарно. Это означает, что в случае, когда несколько процессов одновременно записывают в канал, порции данных от этих процессов не перемешиваются.

При записи большего числа байтов, чем это позволяет канал или FIFO, вызов write(2) блокируется до освобождения требуемого места. При этом атомарность операции не гарантируется. Если процесс пытается записать данные в канал, не открытый ни одним процессом на чтение, процессу генерируется сигнал SIGPIPE, а вызов write(2) возвращает 0 с установкой ошибки (errno=ЕР1РЕ) (если процесс не установил обработки сигнала SIGPIPE, производится обработка по умолчанию – процесс завершается).

1. Могут ли два и более процессов читать или записывать в канал?

Да, если введено достаточное количество байтов для buff, при этом порции процессов не перемешиваются, иначе, при большем вводе байтов, чем указано, возможна ошибка.

1. Опишите функцию write (тип возвращаемого значения, аргументы и логику ра-боты). Что означает 1 (единица) в вызове этой функции в программе server.c (строка 42)?

Функция записывает length байтов из буфера buffer в файл, определенный дескриптором файла fd. Эта операция чисто ‘двоичная’ и без буферизации. При единице возвращает действительное число байтов.

Функция write возвращает число действительно записанных в файл байтов или -1 при ошибке, устанавливая при этом errno.

1. Опишите функцию strerror

Используется, чтоб получить pointer к ошибке в строке буффера и возврата errnum, выводя сообщения с указанием errno. Чаще используется strerror\_s (secured — защищенный), нежели strerror из-за безопасности и утечки данных.