

Лабораторная работа №14

Дисциплина: Операционные системы

Кондратьев Арсений Вячеславович

26.09.2022

Содержание

1	Цель работы	3
2	Теоретическое введение	4
3	Выполнение лабораторной работы	5
4	Выводы	9
5	Контрольные вопросы	10

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Теоретическое введение

Вызов функции `mkfifo()` создаёт файл канала

Файлы именованных каналов создаются функцией `mkfifo`

Посылаем сообщение серверу с помощью функции `write()`

3 Выполнение лабораторной работы

1. Чтобы сервер работал не бесконечно, а прекращал работу через некоторое время, я добавил в цикл `while` в файле сервера доп условие, чтобы разница между текущим и стартовым временем была меньше 30(рис.3.1)

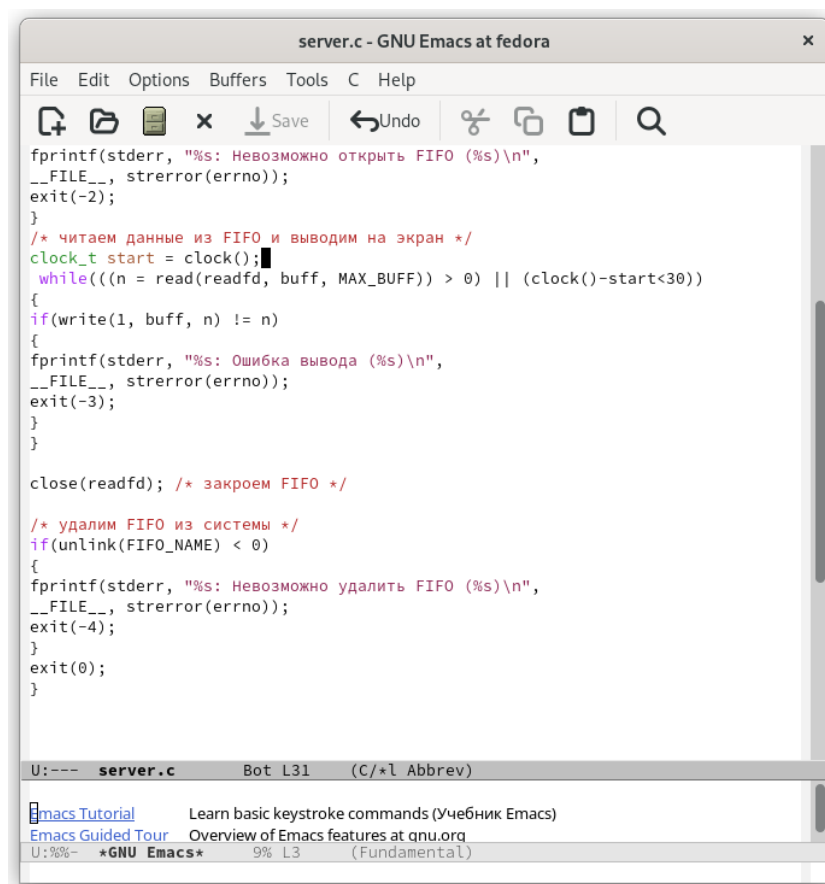


Figure 3.1: Рис. 1

2. Написал цикл `for`, чтобы клиент передавал серверу текущее время, и с помощью команды `sleep()` установил интервал 5 сек. И с помощью функции

time(NULL) передавал в переменную текущее время, а потом передавал сообщение на сервер.(рис.3.2)

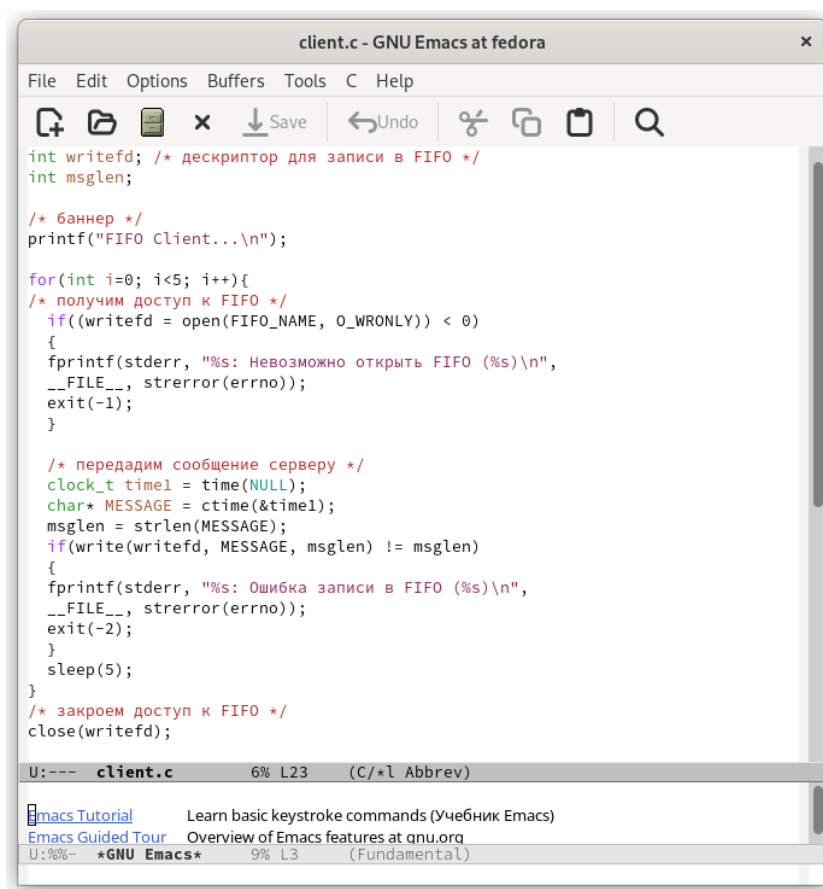


Figure 3.2: Рис. 2

Мной было замечено, что функция clock неверно вычисляет время(рис.3.3), но она корректно считает секунды, поэтому я ее использовал для завершения работы сервера

```
avkondratev@fedora:~/lab14
client.c:26:6: предупреждение: неявная декларация функции «write»; имелось в вид
y «fwrite»? [-Wimplicit-function-declaration]
 26 |     if(write(writefd, MESSAGE, msglen) != msglen)
    |         ^~~~~
    |         fwrite
client.c:32:3: предупреждение: неявная декларация функции «sleep» [-Wimplicit-fu
nction-declaration]
 32 |     sleep(5);
    |     ^~~~~
    |     pclose
client.c:35:1: предупреждение: неявная декларация функции «close»; имелось в вид
y «pclose»? [-Wimplicit-function-declaration]
 35 |     close(writefd);
    |     ^~~~~
    |     pclose
[avkondratev@fedora lab14]$ ./server
bash: ./server: Нет такого файла или каталога
[avkondratev@fedora lab14]$ ./server
FIFO Server...
Thu Jan 1 03:09:45 1970
Thu Jan 1 03:10:50 1970
Thu Jan 1 03:11:51 1970
Thu Jan 1 03:13:10 1970
Thu Jan 1 03:14:24 1970
[avkondratev@fedora lab14]$
```

Figure 3.3: Рис. 3: Неправильное время

Результат: клиент выводит на сервер текущее время с интервалом 5 сек, а сервер завершает работу после 30 сек(рис.3.4)

```
avkondratev@fedora:~/lab14
gcc client.c -o client
client.c: В функции «main»:
client.c:26:6: предупреждение: неявная декларация функции «write»; имелось в вид
y «fwrite»? [-Wimplicit-function-declaration]
 26 |     if(write(writefd, MESSAGE, msglen) != msglen)
    |         ^~~~~
    |         fwrite
client.c:32:3: предупреждение: неявная декларация функции «sleep» [-Wimplicit-fu
nction-declaration]
 32 |     sleep(5);
    |     ^~~~~
    |     pclose
client.c:35:1: предупреждение: неявная декларация функции «close»; имелось в вид
y «pclose»? [-Wimplicit-function-declaration]
 35 |     close(writefd);
    |     ^~~~~
    |     pclose
[avkondratev@fedora lab14]$ ./server
FIFO Server...
Mon Sep 26 18:41:27 2022
Mon Sep 26 18:41:32 2022
Mon Sep 26 18:41:37 2022
Mon Sep 26 18:41:42 2022
Mon Sep 26 18:41:47 2022
[avkondratev@fedora lab14]$
```

Figure 3.4: Рис. 4

3. Проверил работу 2-х клиентов. Можно заметить, как каждый отправляет на

сервер текущее время с собственным интервалом 5 сек(рис.3.5)

Figure 3.5: Рис. 5

4 Выводы

Я приобрел практические навыки работы с именованными каналами.

5 Контрольные вопросы

1. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл
2. Можно, с помощью объединения нескольких процессов символом |
3. Можно, с помощью команд `mkfifo` или `mknod`
4. `int pipe(int fds[2]);`
Единственный параметр-массив включает два файловых дескриптора — `fd[0]` для чтения и `fd[1]` для записи
5. `mkfifo(FIFO_NAME, 0600);` или `mknod(FIFO_NAME, S_IFIFO | 0600, 0);`
6. При чтении числа байт, меньшего чем находится в канале, возвращается требуемое число байтов, остаток сохраняется для последующих чтений. При чтении числа байт, большего чем находится в канале, возвращается доступное число байт.
7. Данные не будут перемешиваться
8. Да
9. `Write` передает сообщение на сервер, записывает байты из буфера в файл.

Возвращает число записанных байтов

EACCES – файл открыт для чтения или закрыт для записи, EBADF – неверный handle-р файла, ENOSPC – на устройстве нет свободного места. Единица в вызове функции write в программе server.c означает идентификатор потока

10. Интерпритирует номер ошибки, передаваемый в функцию