
Front matter

lang: ru-RU title: Именованные каналы subtitle: ДР по ОС №15 author: Аникин Константин Сергеевич group: НПИбд-01-20

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes: - \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph. - \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph. - \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binoppenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph - \displaywidowpenalty=50 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplaypenalty=10000 # penalty for breaking before a display - \postdisplaypenalty=0 # penalty for breaking after a display - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in standard LaTeX) - \raggedbottom # or \flushbottom - \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

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. [\[1\]](#)

В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общенюиксные (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 [\[2\]](#)

Выполнение лабораторной работы

Скрипт 1

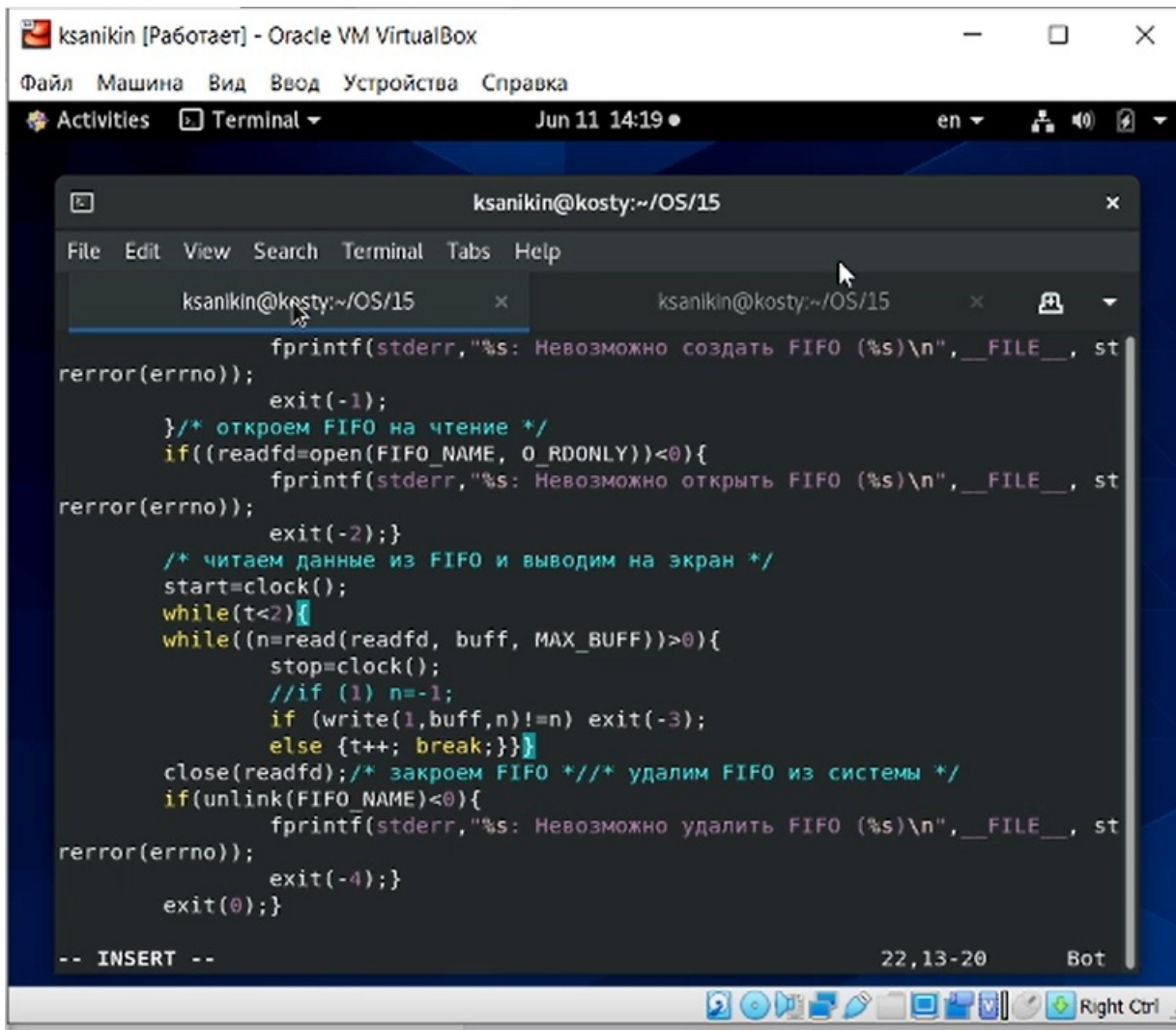
Полностью скрипты приводить не буду, покажу лишь места, в которые внесены изменения.

Непосредственно server.c. (рис. 1)

Приём нескольких 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.



```
ksanikin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Activities  Terminal  Jun 11 14:19  en  [user icon] [volume icon] [network icon]

ksanikin@kosty:~/OS/15
File Edit View Search Terminal Tabs Help

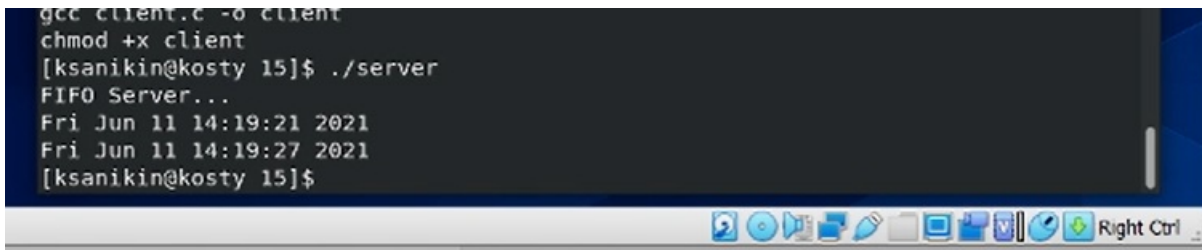
ksanikin@kosty:~/OS/15  x  ksanikin@kosty:~/OS/15  x  [user icon] [dropdown]

fprintf(stderr,"%s: Невозможно создать FIFO (%s)\n",__FILE__, st
rerror(errno));
exit(-1);
}/* откроем FIFO на чтение */
if((readfd=open(FIFO_NAME, O_RDONLY))<0){
fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",__FILE__, st
rerror(errno));
exit(-2);}
/* читаем данные из FIFO и выводим на экран */
start=clock();
while(t<2){
while((n=read(readfd, buff, MAX_BUFF))>0){
stop=clock();
//if (1) n=-1;
if (write(1,buff,n)!=n) exit(-3);
else {t++; break;}}
close(readfd);/* закроем FIFO *//* удалим FIFO из системы */
if(unlink(FIFO_NAME)<0){
fprintf(stderr,"%s: Невозможно удалить FIFO (%s)\n",__FILE__, st
rerror(errno));
exit(-4);}
exit(0);}

-- INSERT -- 22,13-20 Bot
```

Рис. 1: Скрипт 1

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



```
gcc client.c -o client
chmod +x client
[ksanikin@kosty 15]$ ./server
FIFO Server...
Fri Jun 11 14:19:21 2021
Fri Jun 11 14:19:27 2021
[ksanikin@kosty 15]$
```

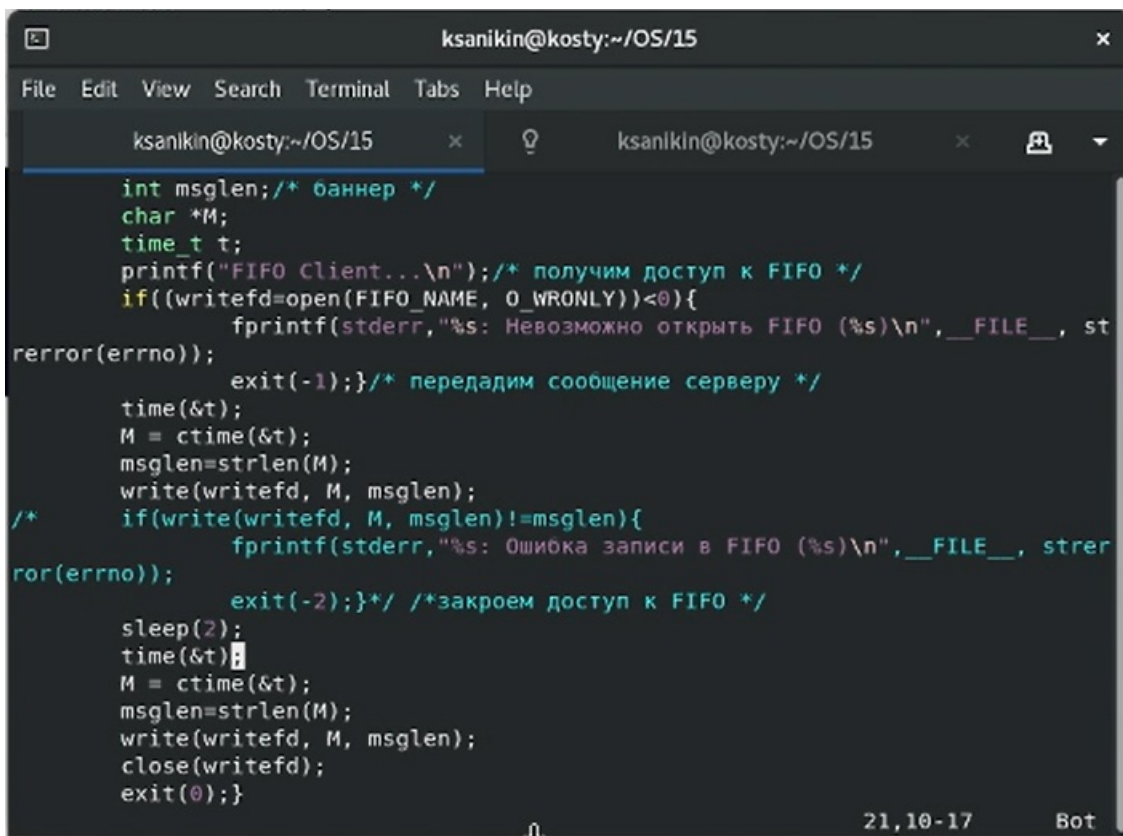
Рис. 2: Результаты работы скрипта 1

Скрипт 2

Непосредственно client.c. (рис. 3)

Получаем текущее время и записываем его в файл.

Потом спим 2 секунды и повторяем операцию.



```
ksanikin@kosty:~/OS/15
File Edit View Search Terminal Tabs Help

ksanikin@kosty:~/OS/15 x ksanikin@kosty:~/OS/15 x Bot

int msglen; /* баннер */
char *M;
time_t t;
printf("FIFO Client...\n"); /* получим доступ к FIFO */
if((writefd=open(FIFO_NAME, O_WRONLY))<0){
    fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",__FILE__, st
error(errno));
    exit(-1); /* передадим сообщение серверу */
    time(&t);
    M = ctime(&t);
    msglen=strlen(M);
    write(writefd, M, msglen);
} /*
if(write(writefd, M, msglen)!=msglen){
    fprintf(stderr,"%s: Ошибка записи в FIFO (%s)\n",__FILE__, strer
ror(errno));
    exit(-2); /* закроем доступ к FIFO */
    sleep(2);
    time(&t);
    M = ctime(&t);
    msglen=strlen(M);
    write(writefd, M, msglen);
    close(writefd);
    exit(0);
}
```

Рис. 3: Скрипт 2

Клиент закрывается не сразу(рис. 4)



```
[ksanikin@kosty 15]$ ./client
FIFO Client...
█
```

Рис. 4: Работающий клиент

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



```
[ksanikin@kosty 15]$ ./server
FIFO Server...
Fri Jun 11 13:23:19 2021
Fri Jun 11 13:23:21 2021
[ksanikin@kosty 15]$ █
```

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.

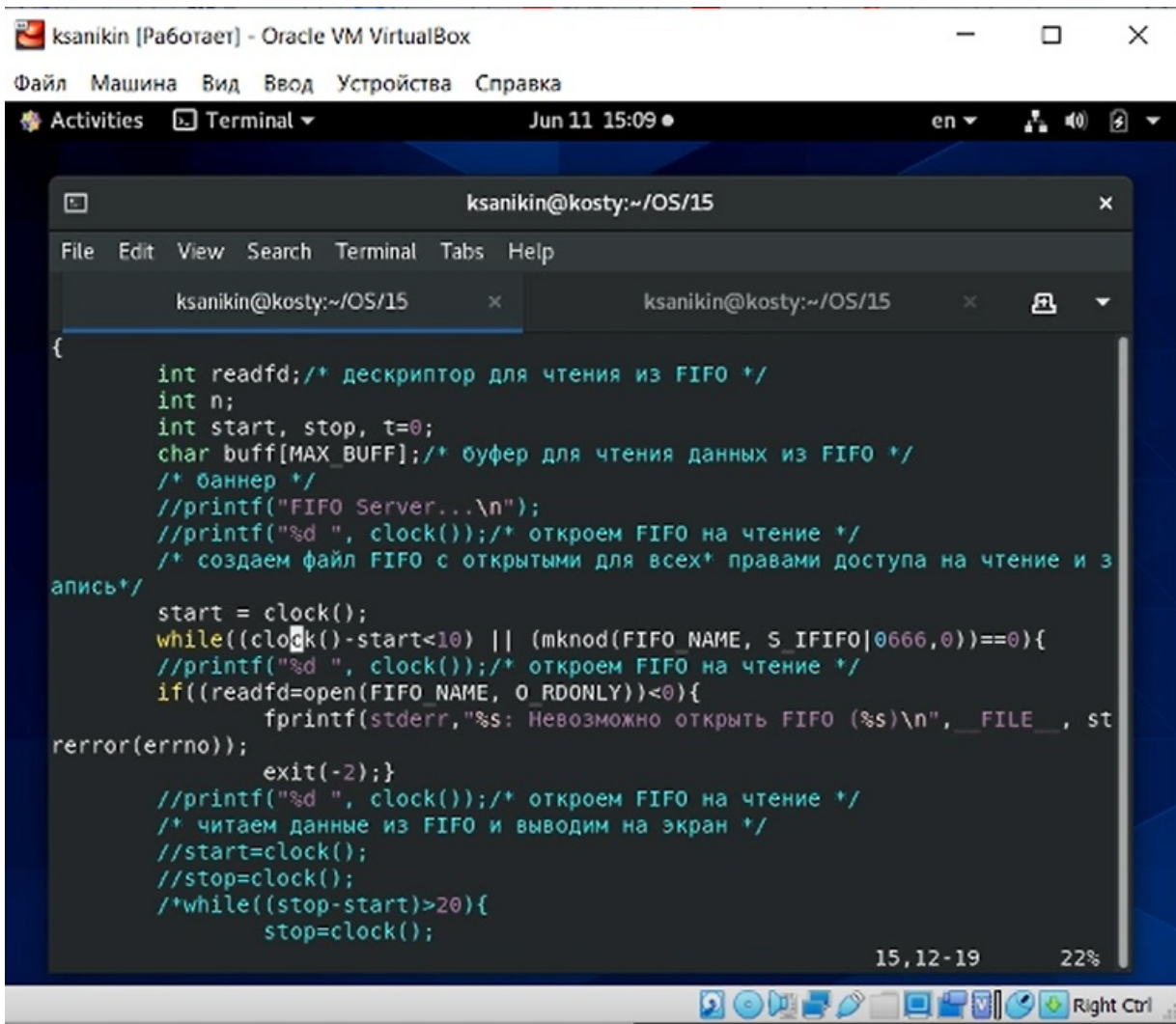


Рис. 6: Скрипт 3

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)

```

[ksanikin@kosty 15]$ ./server
FIFO Server...
Hello Server!!!
[ksanikin@kosty 15]$

```

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)

```

ksanikin@kosty:~/OS/15
File Edit View Search Terminal Help
[ksanikin@kosty ~]$ cd OS/15
[ksanikin@kosty 15]$ ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
[ksanikin@kosty 15]$ unlink(

```

Рис. 8: Неразлинкованный fifo

Выводы

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

Библиографический список

[1]: [FIFO - именованные каналы](#)

[2]: [Краткий справочник команд Linux](#)