

Лабораторная работа № 14

Именованные каналы

Хамдамова Айжана

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	14
	Список литературы	15

Список иллюстраций

figno:1.	8
figno:2.	9
figno:3.	10
figno:4.	11
figno:5.	12
figno:6.	12
figno:7.	12
figno:8.	13

Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами

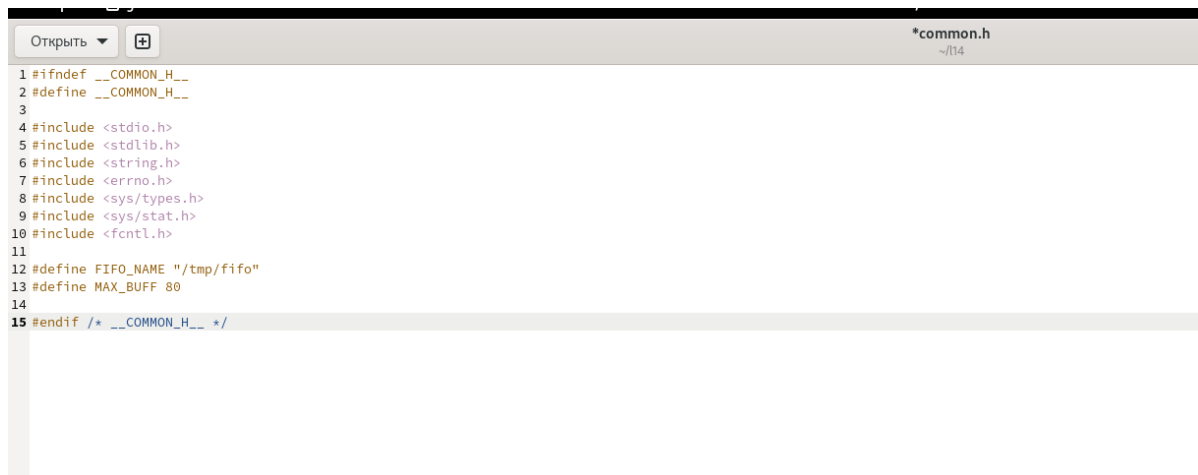
2 Теоретическое введение

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: общедоступные (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

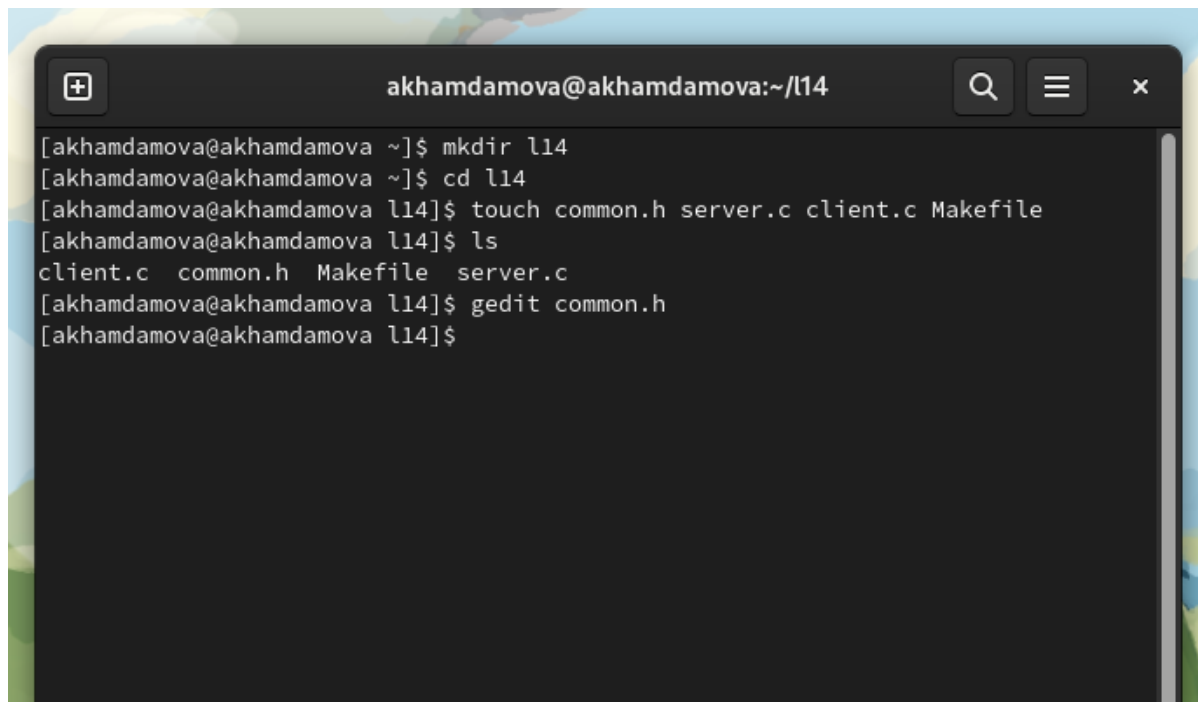
макросом FIFO_NAME)

3 Выполнение лабораторной работы

Изучите приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два).




```
Открыть + *common.h ~/114
1 #ifndef __COMMON_H__
2 #define __COMMON_H__
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <errno.h>
8 #include <sys/types.h>
9 #include <sys/stat.h>
10 #include <fcntl.h>
11
12 #define FIFO_NAME "/tmp/fifo"
13 #define MAX_BUFF 80
14
15 #endif /* __COMMON_H__ */
```

A terminal window with a dark background and a light blue title bar. The title bar contains a window control icon on the left, the text "akhamdamova@akhamdamova:~/l14" in the center, and search, menu, and close icons on the right. The terminal shows a sequence of commands and their outputs:

```
[akhamdamova@akhamdamova ~]$ mkdir l14
[akhamdamova@akhamdamova ~]$ cd l14
[akhamdamova@akhamdamova l14]$ touch common.h server.c client.c Makefile
[akhamdamova@akhamdamova l14]$ ls
client.c  common.h  Makefile  server.c
[akhamdamova@akhamdamova l14]$ gedit common.h
[akhamdamova@akhamdamova l14]$
```

```
Открыть  *server.c
~/114

1 #include "common.h"
2 int
3 main()
4 {
5     int readfd;
6     int n;
7     char buff[MAX_BUFF];
8     printf("FIFO Server...\n");
9
10    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
11    {
12        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
13            __FILE__, strerror(errno));
14        exit(-1);
15    }
16
17    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
18    {
19        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
20            __FILE__, strerror(errno));
21        exit(-2);
22    }
23    clock_t now=time(NULL), start=time(NULL);
24    while(now-start<30)
25    {
26        while((n = read(readfd, buff, MAX_BUFF)) > 0)
27        {
28            if(write(1, buff, n) != n)
29            {
30                fprintf(stderr, "%s: Ошибка вывода (%s)\n",
31                    __FILE__, strerror(errno));
32            }
33            now=time(NULL);
34        }
35        printf("server timeout, %li - seconds passed\n", (now-start));
36        close(readfd);
37
38        if(unlink(FIFO_NAME) < 0)
39        {
40            fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
41                __FILE__, strerror(errno));
42            exit(-4);
43        }
44        exit(0);
45    }
46 }
47
```

2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.

Открыть ▾

+

*Makefile
~/114

```
1 all: server client
2
3 server: server.c common.h
4     gcc server.c -o server
5
6 client: client.c common.h
7     gcc client.c -o client
8
9 clean:
10     -rm server client *.o
```

```

akhamdamova@akhamdamova:~/l14
[akhamdamova@akhamdamova ~]$ mkdir l14
[akhamdamova@akhamdamova ~]$ cd l14
[akhamdamova@akhamdamova l14]$ touch common.h server.c client.c Makefile
[akhamdamova@akhamdamova l14]$ ls
client.c common.h Makefile server.c
[akhamdamova@akhamdamova l14]$ gedit common.h
[akhamdamova@akhamdamova l14]$ gedit server.c
[akhamdamova@akhamdamova l14]$ gedit client.c
[akhamdamova@akhamdamova l14]$ gedit Makefile
[akhamdamova@akhamdamova l14]$ touch client2.c
[akhamdamova@akhamdamova l14]$ gedit client2.c
[akhamdamova@akhamdamova l14]$ make all
gcc server.c -o server
server.c: 8 функции «main»:
server.c:23:15: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
 23 |     clock_t now=time(NULL), start=time(NULL);
    |               ^~~~~
server.c:2:1: замечание: «time» is defined in header «<time.h>»; did you forget to «#include <time.h>»?
 1 | #include "common.h"
+++ |++include <time.h>
 2 | int
server.c:26:18: предупреждение: неявная декларация функции «read»; имелось в виду «fread»? [-Wimplicit-function-declaration]
 26 |     while((n = read(readfd, buff, MAX_BUFF) > 0)
    |                  ^~~~~
server.c:28:14: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
 28 |         if(write(1, buff, n) != n)
    |            ^~~~~
server.c:37:3: предупреждение: неявная декларация функции «close»; имелось в виду «pclose»? [-Wimplicit-function-declaration]
 37 |     close(readfd);
    |     ^~~~~
server.c:39:6: предупреждение: неявная декларация функции «unlink» [-Wimplicit-function-declaration]
 39 |     if(unlink(FIFO_NAME) < 0)
    |        ^~~~~
gcc client.c -o client
client.c: 8 функции «main»:
client.c:13:5: предупреждение: неявная декларация функции «sleep» [-Wimplicit-function-declaration]
 13 |     sleep(3);
    |     ^~~~~
client.c:14:7: предупреждение: неявная декларация функции «time» [-Wimplicit-function-declaration]
 14 |     t=time(NULL);
    |       ^~~~~
client.c:2:1: замечание: «time» is defined in header «<time.h>»; did you forget to «#include <time.h>»?
 1 | #include "common.h"
+++ |++include <time.h>
 2 |
client.c:35:8: предупреждение: неявная декларация функции «write»; имелось в виду «fwrite»? [-Wimplicit-function-declaration]
 35 |     write(1, buff, n);
    |     ^~~~~

```

```

[akhamdamova@akhamdamova l14]$ ls
client client2.c client.c common.h Makefile server server.c
[akhamdamova@akhamdamova l14]$

```

3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера.

```

[akhamdamova@akhamdamova l14]$ ./server
FIFO Server...
^C
[akhamdamova@akhamdamova l14]$ ./server
bash: ./server: Нет такого файла или каталога
[akhamdamova@akhamdamova l14]$ ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
[akhamdamova@akhamdamova l14]$

```

```
bash: ./server: Нет такого файла или каталога
[akhamdamova@akhamdamova l14]$ ./server
FIFO Server...
server.c: Невозможно создать FIFO (File exists)
[akhamdamova@akhamdamova l14]$ ls
client  client2.c  client.c  common.h  Makefile  server  server.c
[akhamdamova@akhamdamova l14]$ ./client
FIFO Client...
```

4 Выводы

Я приобрела практические навыки работы с именованными каналами

Список литературы