

Отчет по лабораторной работе №14

Именованные каналы

Максим Сергеевич Белов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	11

Список иллюстраций

4.1	common.h	8
4.2	server.c	9
4.3	server.c(2)	9
4.4	client.c	10
4.5	Проверка	10

List of Tables

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Задание

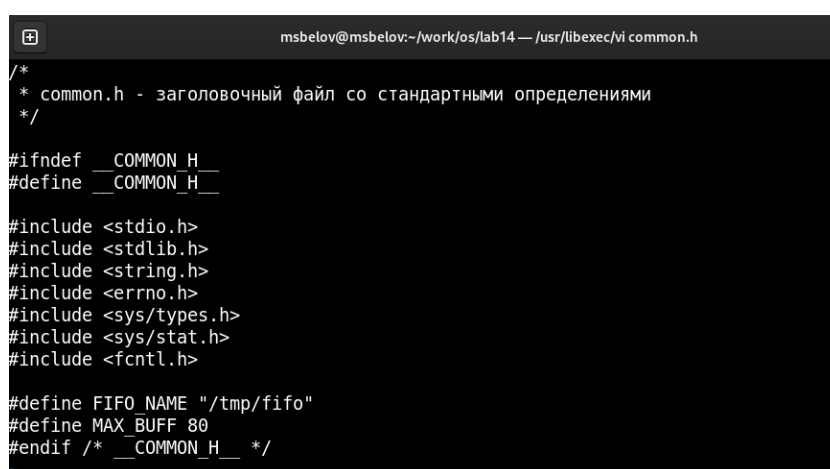
Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента. 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

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: общепонимание (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.

4 Выполнение лабораторной работы

1. Файл common.h (Рис. 4.1)



```
msbelov@msbelov:~/work/os/lab14 — /usr/libexec/vi common.h
/*
 * common.h - заголовочный файл со стандартными определениями
 */

#ifndef __COMMON_H__
#define __COMMON_H__

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80
#endif /* __COMMON_H__ */
```

Рис. 4.1: common.h

2. Файл server.c (изменен для прекращения работы через 30 секунд) (Рис. 4.2)


```
Обзор Терминал Сб, 4 июня msbelov@msbelov:~/work/os/lab

/*
server.c - реализация сервера

чтобы запустить пример, необходимо:
1. запустить программу server на одной консоли;
2. запустить программу client на другой консоли.
*/

#include "common.h"

int
main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    /* баннер */
    printf("FIFO Server...\n");

    /* создаем файл FIFO с открытыми для всех
    * правами доступа на чтение и запись
    */
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            _FILE_, strerror(errno));
        exit(-1);
    }

    /* откроем FIFO на чтение */
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            _FILE_, strerror(errno));
        exit(-2);
    }

    clock_t now=time(NULL), start=time(NULL);
    while(now-start<30)
    {
        /* читаем данные из FIFO и выводим на экран */
        while((n = read(readfd, buff, MAX_BUFF)) > 0)
        {
            if(write(1, buff, n) != n)
            {
                fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                    _FILE_, strerror(errno));
                exit(-3);
            }
        }
        now=time(NULL);
        printf("server timeout, %li second passed\n", (now-start));
        close(readfd);
        /* удалим FIFO из системы */
        if(unlink(FIFO_NAME) < 0)
        {
            fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
                _FILE_, strerror(errno));
            exit(-4);
        }
    }
    exit(0);
}
```

Рис. 4.2: server.c

(Рис. 4.3)

```
/* откроем FIFO на чтение */
if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
{
    fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
        _FILE_, strerror(errno));
    exit(-2);
}

clock_t now=time(NULL), start=time(NULL);
while(now-start<30)
{
    /* читаем данные из FIFO и выводим на экран */
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                _FILE_, strerror(errno));
            exit(-3);
        }
    }
    now=time(NULL);
    printf("server timeout, %li second passed\n", (now-start));
    close(readfd);
    /* удалим FIFO из системы */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
            _FILE_, strerror(errno));
        exit(-4);
    }
}
exit(0);
}
```

Рис. 4.3: server.c(2)

3. Файл client.c с функцией sleep() (Рис. 4.4)

```
Обзор Терминал C6, 4 июня 13:42 msbelov@msbelov:~/work/os/lab14 — /usr/libexec/vi client.c

* 1. запустить программу server на одной консоли.
* 2. запустить программу client на другой консоли.
*/

#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int
main()
{
    int msg; /* дескриптор для записи в FIFO */
    int len, i;
    long int t;
    for (i=0; i<20; i++)
    {
        sleep(3);
        t=time(NULL);
        /* выводим */
        printf("FIFO Client...\n");

        /* получим доступ к FIFO */
        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "fs: Невозможно открыть FIFO (%s)\n",
                FILE, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        len = strlen(MESSAGE);
        if(write(msg, MESSAGE, len) != len)
        {
            fprintf(stderr, "fs: Ошибка записи в FIFO (%s)\n",
                FILE, strerror(errno));
            exit(-2);
        }

        /* закроем доступ к FIFO */
        close(msg);
    }
    exit(0);
}
```

Рис. 4.4: client.c

4. Проверим работу. Напишем make all, затем запустим вторую консоль. В одной напишем ./server, а в другой ./client. (Рис. 4.5)

```
msbelov@msbelov lab14$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 30 second passed
msbelov@msbelov lab14$

msbelov@msbelov lab14$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
msbelov@msbelov lab14$
```

Рис. 4.5: Проверка

5 Выводы

В ходе работы я приобрел практические навыки работы с именованными каналами.