Отчёт по лабораторной работе

Лабораторная работа № 14

Мурзаев Замир Зейнадинович

Содержание

1 Цель работы **5**

2 Задание **6**

3 Выполнение лабораторной работы **7**

4 Выводы **10**

5 Ответы на вопросы **11**

Список литературы **13**

Список иллюстраций

3.1 Команда touch 7
3.2 файл server.c 7
3.3 файл client.c 8
3.4 файл makefile 9
3.5 файл common.h 9

Список таблиц

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, внеся 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 (напри- мер, 30 сек). Используйте функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

3 Выполнение лабораторной работы

Создаем файлы необходимые (рис. 3.1).

```
zzmurzaev@dk2n25 ~/work/study/2022-2023/Операционные системы/study_2022-2023_os-intro/labs/lab14 $ touch
 common.h server.c client.c Makefile
```

Рис. 3.1: Команда touch

Релизация сервера, создаем (рис. 3.2).

```
/*
* 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);
}
/* начало отсчёта времени */
 clock_t start = time(NULL);

/* цикл работает пока с момента начала отсчёта времени прошло меньше 30 секунд */
```

Рис. 3.2: файл server.c

Изменяем еще один файл (рис. 3.3).

```c
#include "common.h"

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

int
main()
{
int writefd; /* дескриптор для записи в FIFO */
int msglen;

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

/* цикл отвечающий за отправку сообщения о текущем времени */
for(int i=0; i<4; i++)
{
/* получим доступ к FIFO */
if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
{
fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
__FILE__, strerror(errno));
exit(-1);
break;
}

/* текущее время */
long int ttime=time(NULL);
char* text=ctime(&ttime);

/* передадим сообщение серверу */
msglen = strlen(MESSAGE);
if(write(writefd, MESSAGE, msglen) != msglen)
{
fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
__FILE__, strerror(errno));
exit(-2);
}
/* приостановка работы клиента на 5 секунд */
 sleep(5);
}

/* закроем доступ к FIFO */
```

Рис. 3.3: файл client.c

Создаем файл, с помощью которого будем создавать исполняемые файлы (рис. 3.4).

```
all: server client

server: server.c common.h
	gcc server.c -o server

client: client.c common.h
	gcc client.c -o client

clean:
	-rm server client *.o
```

Рис. 3.4: файл makefile

Создаем заголовочный файл (рис. 3.5).

```
/*
* 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>
#include <unistd.h>
#include <time.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80

#endif /* __COMMON_H__ */
```

Рис. 3.5: файл common.h

4 Выводы

Приобретены практические навыки работы с именованными каналами.

5 Ответы на вопросы

В чем ключевое отличие именованных каналов от неименованных?

Ответ: именованные каналы имеют идентификатор.

Возможно ли создание неименованного канала из командной строки?

Ответ: sí.

Возможно ли создание именованного канала из командной строки?

Ответ: sí.

Опишите функцию языка C, создающую неименованный канал.

Ответ: int pipe (int filedes[2]).

Опишите функцию языка C, создающую именованный канал.

Ответ: int mkfifo(const char *pathname, mode_t mode).

Что будет в случае прочтения из fifo меньшего числа байтов, чем находится в канал

Ответ: в первом случае возвращается требуемое число байтов, остаток сохраняется для последующих чтений. При чтении числа байт, большего чем находится в канале, возвращается доступное число байт.

Что будет в случае записи в fifo меньшего числа байтов, чем позволяет буфер? Боль

Ответ: свойство - анализ кода; для анализа необходимо скомпилировать программу.

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

Ответ: ну у меня получилось. Значит, вроде да.

Опишите функцию write (тип возвращаемого значения, аргументы и логику работы). Чт

Ответ: write(int fildes, const void *buf, size_t nbyte, off_t offset); 1 (единица) - значит 1 (единица).

Опишите функцию strerror.

Ответ: char* strerror(int errnum).

Список литературы