

Лабораторная работа № 14

Андрианова Марина Георгиевна
RUDN University, Moscow, Russian Federation
NEC–2022, 03 June

Цель работы

Приобретение практических навыков работы с именованными каналами.

Создание файлов

1). Создала необходимые файлы с помощью команды “touch common.h server.c client.c Makefile” и открыла редактор Emacs в фоновом режиме для их редактирования(рис.1).

```
[mgandrianova@fedora ~]$ touch common.h server.c client.c Makefile
[mgandrianova@fedora ~]$ emacs &
[1] 64400
```

Рис.1: Создание файлов

Изменение кода `common.h`

2). Изменила коды программ, представленных в тексте лабораторной работы. В файл `common.h` добавляем стандартные заголовочные файлы `unistd.h` и `time.h`, необходимые для работы кодов других файлов. `Common.h` предназначен для заголовочных файлов, чтобы в остальных программах их не прописывать каждый раз (рис.2).

```

#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__ */

```

U:--- **common.h** Bot L22 (C/*l Abbrev)
Wrote /home/mgandrianova/common.h

Рис.2: Изменение файла common.h

Изменение кода server.c

3). В файл server.c добавила цикл while для контроля времени работы сервера. Разница между текущим временем time(NULL) и временем начала работы clock_t start=time(NULL)(инициализация до цикла) не должна превышать 30 секунд(рис.3).

```

        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 */
    while (time(NULL)-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);
            }
        }
    }

```

Рис.3: Изменение файла server.c

Изменение кода client.c

4). В файл client.c добавила цикл, который отвечает за количество сообщений о текущем времени, которое получается в результате выполнения команд(/* текущее время */) и команду sleep(5) для приостановления работы клиента на 5 секунд(рис.4,рис.5).


```

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

#include "common.h"

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

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

    /* цикл, отвечающий за отправку сообщения о текущем времени */
    int i=0;
    for(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);
        }
    }
}

```

Рис.4: Изменение файла client.c

```

        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(text);
    if(write(writefd, text, msglen) != msglen)
    {
        fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
        exit(-2);
    }

    /* приостановка работы времени клиента на 5 секунде */
    sleep(5);
}

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

```

Рис.5: Изменение файла client.c

Makefile

5). Makefile(для сборки) не изменяем(рис.6).

```
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.6: Makefile

Компиляция файлов

6). После написания кодов ввела команду “make all” для компиляции необходимых файлов(рис.7).

```
[mgandrianova@fedora ~]$ make all  
gcc server.c -o server  
gcc client.c -o client
```

Рис.7: Компиляция файлов

Проверка написанного кода

7). Далее проверила работу написанного кода. Открыла 3 консоли(терминала) и запустила: в первом терминале - `“./server”`, в остальных двух - `“./client”`. В результате каждый терминал-клиент вывел по 4 сообщения. Спустя 30 секунд работа сервера была прекращена. Программа работает корректно(рис.8,рис.9).

```
[mgandrianova@fedora ~]$ ./server
FIFO Server...
Fri Jun  3 19:50:36 2022
Fri Jun  3 19:50:41 2022
Fri Jun  3 19:50:43 2022
Fri Jun  3 19:50:46 2022
Fri Jun  3 19:50:47 2022
Fri Jun  3 19:50:48 2022
Fri Jun  3 19:50:51 2022
Fri Jun  3 19:50:52 2022
Fri Jun  3 19:50:53 2022
Fri Jun  3 19:50:57 2022
Fri Jun  3 19:50:58 2022
Fri Jun  3 19:51:02 2022
```

Рис.8: Проверка работы кода

```
[mgandrianova@fedora ~]$ ./client
FIFO Client...
```

Рис.9: Проверка работы
кода

Выводы

Я приобрела практические навыки работы с
именованными каналами.