

Лабораторная работа №15

Выполнил Гамаюнов Никита, 1032201719, НПМбд-01-20

Прагматика выполнения работы

Цель работы

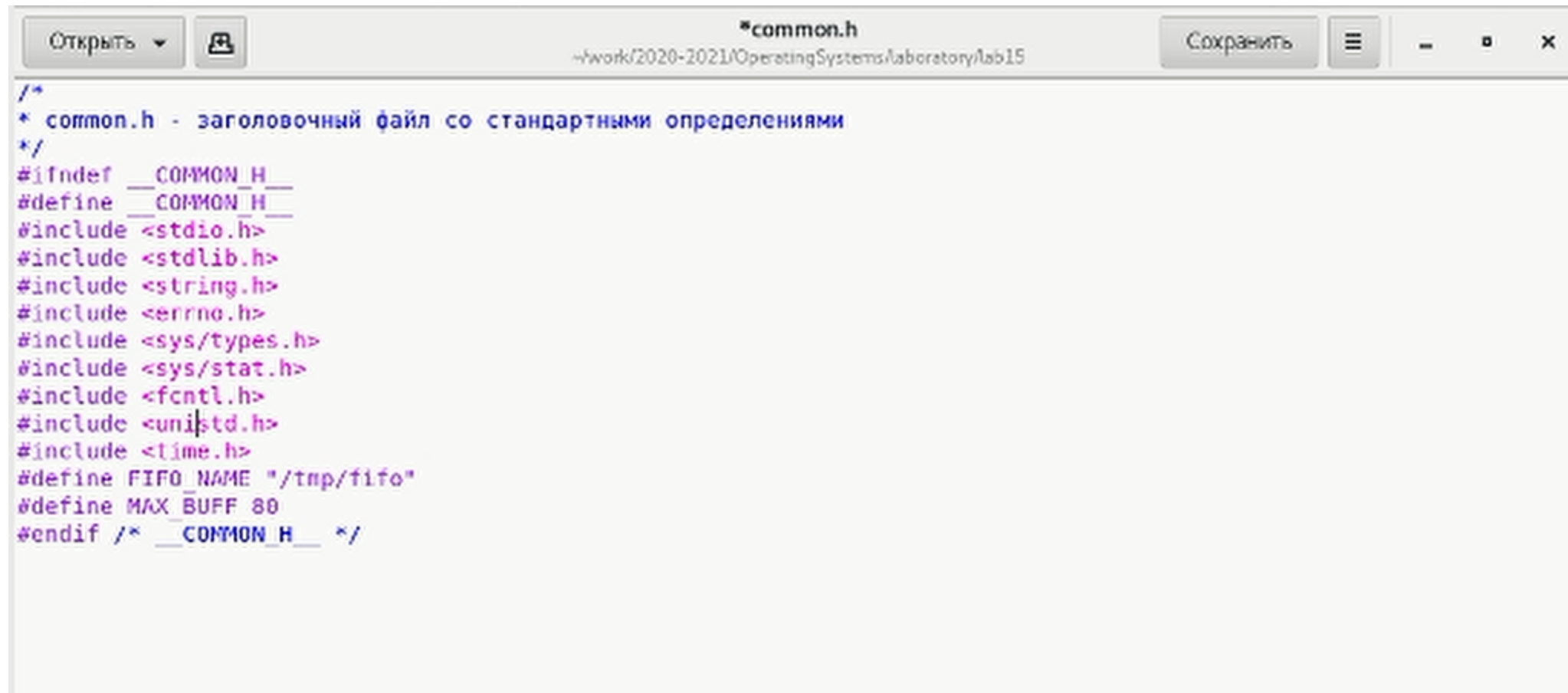
Приобретение практических навыков работы с именованными каналами.

Задания

1. Изучить пример из методических материалов.
2. На его основе написать новые программы, доработав существующие так, что:
 - i. Работает не 1 клиент, а несколько (например, два);
 - ii. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Использовать функцию `sleep()`;
 - iii. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Использовать функцию `clock()`.

Процесс выполнения работы

1. Для корректной работы программ нужно подключить ещё несколько библиотек, - в файл `common.h` добавил `unistd`, чтобы можно было пользоваться `sleep()` и `time.h` для функции `clock()`

A screenshot of a code editor window titled "common.h" with a path "~/work/2020-2021/OperatingSystems/laboratory/lab15". The editor contains C header file code for common.h. The code includes standard headers like <stdio.h>, <stdlib.h>, <string.h>, <errno.h>, <sys/types.h>, <sys/stat.h>, <fcntl.h>, <unistd.h>, and <time.h>. It also defines FIFO_NAME, MAX_BUFF, and uses __COMMON_H__ for guard macros.

```
/*
 * 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__ */
```

2. В файл client.c добавил остановку с помощью sleep и зациклил всю работу с FIFO так, чтобы цикл повторился 6 раз

```
/*
 * 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"
#define MESSAGE "Hello Server!!!\n"
int
main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;

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

    /* 6 раз отправляем сообщение серверу (30/5=6) */
    int i;
    for(i=0; i<6; 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);
        }
    }
}
```

```
/*Получим время*/
long int ttime=time(NULL);
char* MESSAGE1=ctime(&ttime);

/* передадим сообщение серверу */
msglen = strlen(MESSAGE1);
if(write(writefd, MESSAGE1, msglen) != msglen)
{
    fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
        __FILE__, strerror(errno));
    exit(-2);
}

sleep(5);

}

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

3. Реализовал ограничение работы сервера в файле server.c:

```
#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 tmr=time(NULL);

while (time(NULL)-tmr < 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);
        }
    }
    close(readfd); /* закроем FIFO */

    /* удалим FIFO из системы */
    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-4);
    }

    exit(0);
}
```

4. Проверил работу скриптов:

```
[neganayunov@neganayunov lab15]$ ./server
FIFO Server...
Sun Jun  6 17:18:42 2021
Sun Jun  6 17:18:43 2021
Sun Jun  6 17:18:47 2021
Sun Jun  6 17:18:48 2021
Sun Jun  6 17:18:52 2021
Sun Jun  6 17:18:53 2021
Sun Jun  6 17:18:57 2021
Sun Jun  6 17:18:58 2021
Sun Jun  6 17:19:02 2021
Sun Jun  6 17:19:03 2021
Sun Jun  6 17:19:07 2021
Sun Jun  6 17:19:08 2021
[neganayunov@neganayunov lab15]$
```

Выводы

В ходе выполнения лабораторной работы я приобрёл практические навыки работы с именованными каналами.