

Лабораторная работа № 14

Операционные системы

Оразгелдиев Язгелди

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	10

Список иллюстраций

3.1	Код	7
3.2	Код	8
3.3	Код	8
3.4	Код	9
3.5	Код	9
3.6	Прописывание команды	9

Список таблиц

1 Цель работы

Приобретение практических навыков работы с именованными каналами.

2 Задание

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.

3 Выполнение лабораторной работы

1. Прописал все коды и команды из методички к лабораторной работе

```
#include "common.h"
#define MESSAGE "Hello Server!!!"
int
main()
{
    int msg, len, i;
    long int t;
    for(i=0; i<20; i++)
    {
        sleep(3);
        t=time(NULL);
        printf("FIFO Client...\n");

        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr,"%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        len = strlen(MESSAGE);

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

Рис. 3.1: Код

```

#include "common.h"
#define MESSAGE "Hello Server!!!"
int
main()
{
    int writefd, msglen, count;
    long long int t;
    char message[10];
    for(count=0; count<-5; ++count)
    {
        sleep(5);
        t=(long long int) time(0);
        sprintf(message, "%lli", t);
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        msglen = strlen(MESSAGE);

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

```

Рис. 3.2: Код

```

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

```

Рис. 3.3: Код


```
#include "common.h"
int
main()
{
    int readfd;
    int n;
    char buff[MAX_BUFF];
    printf("FIFO Server...");
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)",
            __FILE__, strerror(errno));
        exit(-1);
    }
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)",
            __FILE__, strerror(errno));
        exit(-2);
    }
    clock_t now=time(NULL), start=time(NULL);
```

Рис. 3.4: Код

```
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
```

--- makefile All 11 (GNUmakefile)

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

[Emacs Tutorial](#) Learn basic keystroke commands (Учебник Emacs)
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org

Рис. 3.5: Код

2. Проверил работу сервера с помощью сервера

```
[orazgeldiyevy@ Lab14]$ make
gcc server.c -o server
```

Рис. 3.6: Прописывание команды

4 Выводы

Приобрел практические навыки работы с именованными каналами.