Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

## Кафедра ТС и ВС

**Лабораторная работа № 9**

**По дисциплине: Операционные системы**

**Выполнил**: Зарубин Максим Евгеньевич

**Группа**: ИА-831

## Вариант: 5

**Проверила**: Моренкова Ольга Ильинична

Новосибирск, 2020 г

**Взаимодействие процессов. Каналы. FIFO.**

**Задание на лабораторную работу.**

1. Написать две программы, использующие именованные каналы для обмена информацией. Назовем их условно «клиент» и «сервер»

2. Клиент и сервер должны иметь общий заголовочный файл, определяющий требуемые соглашения (имя FIFO, длины строк и т.д.).

3. Сервер создаёт именованный канал согласно соглашению и ждёт поступления от клиента строки формата: «FILENAME~имя\_файла». Информация, поступившая до указанной строки, игнорируется. После поступления указанной строки сервер открывает файл, имя которого было передано, и выполняет обработку указанного файла согласно варианту задания (имя файла и задание взять из занятия 6). Далее, всё, что нашел сервер должно быть записано в новый файл. Процедура взаимодействия заканчивается отправкой клиенту имени созданного нового файла.

4. Клиент выводит приглашение пользователю для ввода имени файла, после чего формирует строку указанного формата и передаёт её серверу. После получения итогового сообщения от сервера клиент выводит информацию из файла на экран.

**Результаты работы программы.**

Рис.1. Открытые окна для работы с клиентом и сервером.

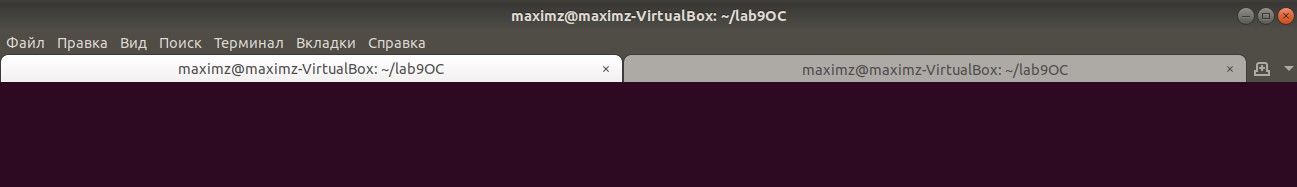


Рис.2. Работа сервера.

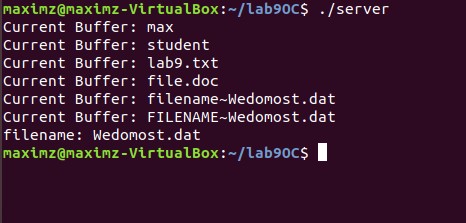
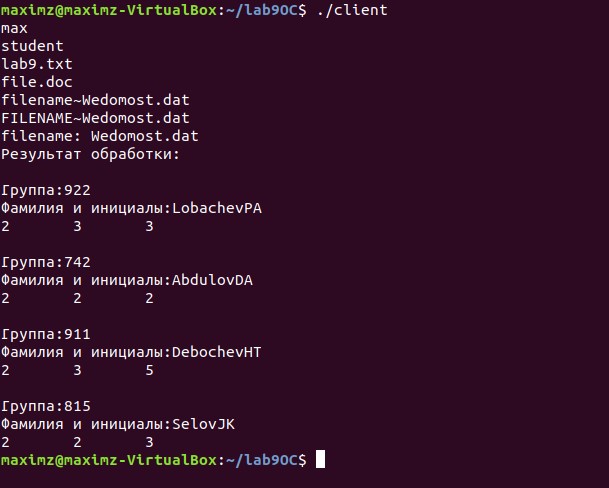


Рис.3. Работа клиента.



**Листинг программы.**

**Lab9.h:**

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <termios.h>

#define FIFO "fifo.l"

#define MAXBUFF 80

#define FILENAME "FILENAME~"

#define FNBEGIN 9

typedef struct stud

{ int group;

char name[10];

int A[3]; } stud;

stud t;

int fd;

void Processfile(char \*filename, char \*newfile);

void Printnewfile(char \*filename);

**client.c:**

#include "./lab9.h"

void Printnewfile(char \*filename){

int j=0;

FILE \*f;

printf("Результат обработки:\n");

f=fopen(filename,"rb");

fread(&t, sizeof(t), 1, f);

while (!feof(f)){

printf("\nГруппа:%3d \nФамилия и инициалы:%s\n",t.group, t.name);

for(int i=0;i<3;i++)

printf("%d\t",t.A[i]);

j++;

printf("\n");

fread(&t, sizeof(t), 1, f);

}

fclose(f);

if (j==0) printf("Таких студентов нет\n");

}

int main(int argc, char \*argv[]){

char f[MAXBUFF], newname[MAXBUFF];

int n;

char \*buff = (char \*)calloc(MAXBUFF, sizeof(char));

if((fd = open(FIFO, O\_WRONLY)) < 0){

printf("Error in open FIFO\n");

exit(1);

}

while(1){

scanf("%s", f);

if(write(fd, f, sizeof(f)) != sizeof(f)){

printf("Error write\n");

exit(1);

}

if(strstr(f,FILENAME) != NULL){

for(short int i = 0; i<strlen(f)-FNBEGIN;i++){

newname[i] = f[i+FNBEGIN];

}

printf("filename: %s\n",newname);

break;

}

}

close(fd);

sleep(5);

if((fd = open(FIFO, O\_RDONLY)) < 0){

printf("Error in open FIFO\n");

exit(1);

}

while ((n = read(fd, buff, MAXBUFF)) > 0){

Printnewfile(buff);

}

exit(0);

}

**Server.c:**

#include "./lab9.h"

char \*newfile = "RESULT.dat";

void Processfile(char \*filename, char \*newfile){

FILE \*f,\*f2;

int k,l=0;

f=fopen(filename,"rb");

f2=fopen(newfile,"wb");

fread(&t, sizeof(t), 1, f);

while (!feof(f)){

k=0;

for(int i=0;i<3;i++)

if(t.A[i]==2)

k++;

if(k>=1)

{

fwrite(&t, sizeof(t), 1, f2);

l++;

}

fread(&t, sizeof(t), 1, f);

}

fclose(f);

fclose(f2);

}

int main (void){

int n;

char \*f = (char \*)calloc(MAXBUFF , sizeof(char));

char \*buff = (char \*)calloc(MAXBUFF , sizeof(char));

if (mknod(FIFO, S\_IFIFO | 0666, 0) < 0){

printf("Невозможно создать FIFO\n");

exit(1);

}

if ((fd = open(FIFO, O\_RDONLY)) < 0){

printf("Невозможно открыть FIFO\n");

exit(1);

}

while ((n = read(fd, buff, MAXBUFF)) > 0){

printf("Current Buffer: %s\n",buff);

if(strstr(buff,FILENAME) != NULL){

for(short int i = 0; i<strlen(buff)-FNBEGIN;i++){

f[i] = buff[i+FNBEGIN];

}

printf("filename: %s\n",f);

break;

}

}

Processfile(f, newfile);

free(buff);

close(fd);

if ((fd = open(FIFO, O\_WRONLY)) < 0){

printf("Невозможно открыть FIFO\n");

exit(1);

}

if(write(fd, newfile, strlen(newfile)) != strlen(newfile)){

printf("Error write\n");

exit(1);

}

close(fd);

if (unlink(FIFO) < 0){

printf("Невозможно удалить FIFO\n");

exit(1);

}

exit(0);

}