Отчёт по лабораторной работе №14

Именованные каналы

Леденев Егор Олегович

Содержание

# 1 Цель работы

Приобретение практических навыков работы с именованными каналами.

# 2 Выполнение лабораторной работы

Пишем и редактируем программы на С, так чтобы на одном сервере можно было запускать сначала один клиент. Далее напишем и отредактируем программы, так чтобы на одном сервере можно было запускать больше серверов, чем один, интервал между клиентами будет 5 секунд, сервер завершится через 30 секунд. Мы имеем 4 файла (программы) это заголовочный файл (common.h) клиент (client.c), сервер (server.c), и Makefile.

////////////////////////////////////  
// common.h  
#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\_\_ \*/

////////////////////////////////////  
// client.c  
#include "common.h"  
#define MESSAGE "Hello Server!!! \n"  
int main ()  
{  
int writefd;   
int msglen;  
printf("FIFO Client...\n");  
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);  
}  
close (writefd);  
exit(0);  
}

////////////////////////////////////  
// server.c  
#include "common.h"  
void display() {  
 printf("/n Server timeout...%u seconds passed!\n Total elapsesed time is 30 seconds!\n",clock());  
}  
int main()  
{  
clock\_t start, now;  
start = time(NULL);  
int readfd;  
int n;  
char buff[MAX\_BUFF];  
printf("Hell server...\n\n");  
if(mknod(FIFO\_NAME, S\_IFIFO | 0666, 0) < 0)  
{  
fprintf(stderr, "%s: Unable to create FiFO(%s)\n", \_\_FILE\_\_, strerror(errno));  
exit(-1);  
}  
if((readfd = open(FIFO\_NAME, O\_RDONLY)) < 0)  
{  
fprintf(stderr, "%s: Unable to open FIFO (%s)\n", \_\_FILE\_\_, strerror(errno));  
exit(-2);  
}  
for(;;)  
{  
while((n = read(readfd, buff, MAX\_BUFF)) > 0)  
{  
if(write(1, buff, n) != n)  
{  
fprintf(stderr, "%s: Input error (%s)\n", \_\_FILE\_\_, strerror(errno));  
}  
sleep(5);  
}  
now=time(NULL);  
if (now-start>30)  
{  
display();  
return 0;  
}  
}  
close(readfd);   
if(unlink(FIFO\_NAME) < 0)  
{  
fprintf(stderr, "%s: unable to delete FIFO (%s)\n", \_\_FILE\_\_, strerror(errno));   
exit(-4);  
}  
exit(0);  
}

////////////////////////////////////  
// makefile  
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

Запуск пректа

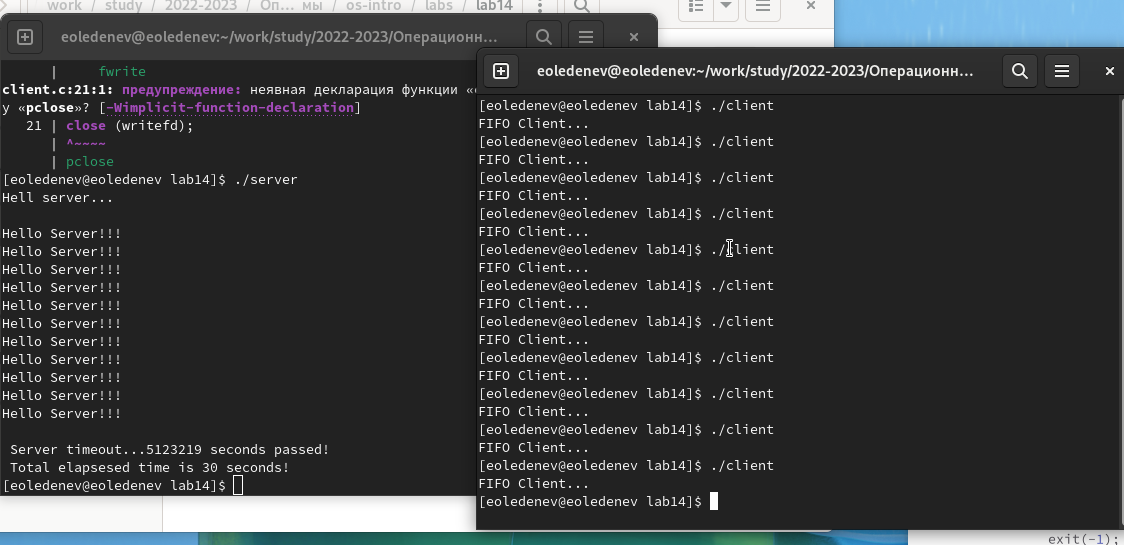


Figure 1: Запуск клиента и сервера

# 3 Вывод

В данной работе мы приобрели практические навыки работы с именованными каналами по типу клиент-сервер.