

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., к.т.н,
должность, уч. степень, звание

подпись, дата

О. А. Кононов
инициалы, фамилия

Отчет о лабораторной работе №3

«Обмен сообщениями»

по курсу: «Системы реального времени»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

1742

подпись, дата

Д. В. Коробков
инициалы, фамилия

Санкт-Петербург 2020

1. Цель

Научится делать программы с клиент серверным взаимодействием на C в QNX

2. Листинг программы

2.1. текст файла «server.c»

```
#include <stdio.h>
#include <pthread.h>
#include <inttypes.h>
#include <errno.h>
#include <sys/neutrino.h>
#include <string.h>

void server(void){
    int rcvid;
    int chid;
    char message[512];

    printf("Server start working \n");

    chid=ChannelCreate(0);
    printf("Chanel id: %d \n", chid);
    printf("Pid: %d \n", getpid());
    while(1)
    {
        rcvid=MsgReceive(chid,message,sizeof(message), NULL);
        printf("Polychili soobshenie, rcvid %X \n", rcvid );
        printf("Soobshenie takoe : \"%s\". \n", message );

        strcpy(message,"Eto otvet");

        MsgReply(rcvid, EOK, message, sizeof(message));
        printf("\"%s\". \n", message );
    }
    ChannelDestroy(chid);
}

int main(void){
    printf("Prog server \n");
    server();
    sleep(5);
    return(1);
}
```

Рисунок 1. Текст файла

2.2. текст файла «client.c»

```
#include <stdio.h>
#include <pthread.h>
#include <inttypes.h>
#include <errno.h>
#include <sys/neutrino.h>
#include <string.h>

int main(void){
char smsg[20];
char rmsg[200];
int coid;
long serv_pid;
printf("Prog client, Vvedite PID servera \n");
scanf("%ld", &serv_pid);
printf("Vveli %ld \n", serv_pid);
coid=ConnectAttach(0, serv_pid, 1, 0, 0);
printf("Connect res %d, vvedite soobshenie: \n", coid);
scanf("%s", &smsg);
printf("Vveli %s \n", smsg);
if(MsgSend(coid, smsg, strlen(smsg)+1, rmsg, sizeof(rmsg))==-1)
{
    printf("Error MsgSend \n");
}
ConnectDetach(coid);
return(1);
}
```

Рисунок 2. Текст файла

3. Последовательность выполнения лабораторной

После компиляции программ сервера и клиента у нас будет 2 исполняемых файла. Назовём их server и client соответственно. Программу server необходимо запустить в фоновом режиме # server &. Она начнёт работать: создаст канал, выведет номер канала и идентификатор процесса сервера, будет ждать сообщения от клиента. Потом необходимо запустить клиента. Он попросит ввести идентификатор процесса сервера, для установления соединения с ним, и само сообщение (20 символов). Далее можно наблюдать, как сервер получит сообщение, выведет его и пошлёт ответ. На этом клиент закончит свою работу, но его можно запустить ещё раз и послать другое сообщение. Остановить работу сервера можно функцией kill < идентификатор процесса сервера >.

4. Результат

```
# `pwd`/server &
[1] 434211
# Prog server
Server start working
Chanel id: 1
Pid: 434211

# `pwd`/client
Prog client, Uvedite PID servera
434211
Uveli 434211
Connect res 3, vedite soobshenie:
privet
Uveli privet
Polychili soobshenie, rcvid 2
Soobshenie takoe : "privet".
"Eto otvet".
# `pwd`/client
Prog client, Uvedite PID servera
434211
Uveli 434211
Connect res 3, vedite soobshenie:
i_eshe_privet
Uveli i_eshe_privet
Polychili soobshenie, rcvid 2
Soobshenie takoe : "i_eshe_privet".
"Eto otvet".
# kill 434211
# _
```

Рисунок 3. Запуск программы

5. Вывод

Научился делать клиент-серверную программу, которая обменивается сообщениями.