

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)
Институт интеллектуальных кибернетических систем
Кафедра Кибернетики

**Лабораторная работа №3 по курсу
«Разработка ПО ОС UNIX»**

Выполнил студент группы Б17-511:

Чудновец Иван

Проверил:

Ктитров С.В.

Москва, 2020

Задание

Разработать программу, реализующую обмен сообщениями между программами-клиентами (чат). Клиентов может быть несколько, при этом клиент должен знать, от кого получено сообщение. Программа должна собираться из нескольких файлов с использованием make.

Код программы

settings.sh:

```
#!/bin/bash
```

```
make
```

```
rm -f *.o
```

Makefile:

```
CC=gcc
```

```
CFLAGS=-Wall
```

```
CXX=g++
```

```
CXXFLAGS=-Wall
```

```
PROJECT=queue
```

```
all: $(PROJECT)
```

```
svr4_queue.o: svr4_queue.c svr4_queue.h
```

```
$(CC) $(CFLAGS) -c svr4_queue.c -o $@
```

```
main.o: main.cpp svr4_queue.h
```

```
$(CXX) $(CXXFLAGS) -c main.cpp -o $@
```

```
$(PROJECT): main.o svr4_queue.o
```

```
$(CXX) $(CXXFLAGS) main.o svr4_queue.o -o $@
```

svr4_queue.h:

```
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
```

```
const int MSGSIZE = 256;
```

```
typedef struct {
    long type;
    char text[1];
} mq_message;
```

```
int create(key_t key);
int delete_(int msqid);
int write(int msqid, long type, const char* text);
mq_message* read(int msqid, long type);
```

svr4_queue.c:

```
#include "svr4_queue.h"
```

```
int create(key_t key){
    int msqid;
    msqid = msgget(key, IPC_CREAT | 0666);

    if(msqid == -1){
        printf("create error: %s\n", strerror(errno));
    }
}
```

```
        return msqid;
    }
}
```

```
int delete_(int msqid){
    if(msgctl(msqid, IPC_RMID, NULL) == -1){
        printf("delete error: %s\n", strerror(errno));
        return 1;
    }
    return 0;
}
```

```
int write(int msqid, long type, const char* text){
    mq_message msg;
    msg.type = type;
    strcpy(msg.text, text);

    if(msgsnd(msqid, (void *)&msg, sizeof(msg), IPC_NOWAIT) == -1){
        printf("write error: %s\n", strerror(errno));
        return 1;
    }
    return 0;
}
```

```
mq_message* read(int msqid, long type){

    mq_message* msg = (mq_message*)malloc(sizeof(mq_message));
    if(msgrcv(msqid, (void *)msg, MSGSIZE, type, IPC_NOWAIT | MSG_NOERROR) == -1){
        printf("read error: %s\n", strerror(errno));
        return NULL;
    }
    return msg;
}
```

```
}
```

main.cpp:

```
#include <string>
```

```
#include <iostream>
```

```
using namespace std;
```

```
extern "C"{
```

```
    #include "svr4_queue.h"
```

```
}
```

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

```
    string cmd = "";
```

```
    int msqid = -1;
```

```
    int key;
```

```
    mq_message* msg = NULL;
```

```
    while(cmd != "exit"){
```

```
        cin >> cmd;
```

```
        if (cmd == "c"){
```

```
            cin >> key;
```

```
            msqid = create(static_cast<key_t>(key));
```

```
            if(msqid == -1) continue;
```

```
            cout << "Создана очередь с id: " << msqid << endl;
```

```
        }
```

```
        else if(cmd == "d"){
```

```
            if(delete_(msqid) == 0){
```

```
                cout << "Удалена очередь." << endl;
```

```
                msqid = -1;
```

```

        }
    }
    else if(cmd == "w"){
        cin >> key;
        getline(cin, cmd);
        cmd = string(cmd.begin() + 1, cmd.end());

        if(write(msqid, key, cmd.c_str()) == 0){
            cout << "Отправлено сообщение: " << cmd << endl;
        }
    }
    else if(cmd == "r"){
        cin >> key;
        msg = read(msqid, key);

        if(msg){
            cout << "Принято сообщение от: " << msg->type << endl;
            cout << "Текст сообщения: " << msg->text << endl;

            delete msg;
        }
    }
}

return 0;
}

```

Выполнение:

Терминал №0:

Код:

./settings.sh

Вывод:

```
g++ -Wall -c main.cpp -o main.o
```

```
gcc -Wall -c svr4_queue.c -o svr4_queue.o
```

```
g++ -Wall main.o svr4_queue.o -o queue
```

Терминал №1:

Код:

```
./queue
```

```
c
```

```
1
```

Вывод:

Создана очередь с id: 9

Код:

```
w
```

```
1 Hi, all!
```

Вывод:

Отправлено сообщение: Hi, all!

Терминал №2:

Код:

```
./queue
```

```
c
```

```
1
```

Вывод:

Создана очередь с id: 9

Код:

```
r
```

```
1
```

Вывод:

Принято сообщение от: 1

Текст сообщения: Hi, all!

Код:

w

2 Hello!

Вывод:

Отправлено сообщение: Hello!

Терминал №1:

Код:

r

2

Вывод:

Принято сообщение от: 2

Текст сообщения: Hello!