МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №8**

по дисциплине: «Системное программирование»

на тему: «Взаимодействие между процессами: семафоры, очереди сообщений»

Вариант №12

|  |  |
| --- | --- |
| Выполнил | Студент группы 21-ИТ-1  Чиникайло А.П. |
|  |  |
| Проверил | Преподаватель  Сергеев М. А. |

Полоцк 2023

**Ход работы**

1. Ознакомиться с теоретическим материалом.

2. Обеспечить синхронизацию процессов и передачу данных между ними на примере двух приложений «клиент» и «сервер», создав два процесса (два исполняемых файла) – процесс «клиент» (первый исполняемый файл) и процесс «сервер» (второй исполняемый файл). С помощью механизмов межпроцессного взаимодействия обеспечить передачу информации от «клиента» к «серверу» и наоборот. В качестве типа передаваемой информации можно использовать: данные, вводимые с клавиатуры; данные, считываемые из файла; данные, генерируемые случайным образом и т. п.

3.Обмен данными между процессами «клиент»-«сервер» осуществить с

использованием программных каналов (именованных либо неименованных),

использованием механизмов синхронизации процессов (например с помощью

семафоров), очередь сообщений;

Листинг 1 – client.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#define MAX\_TEXT\_SIZE 256

struct message {

long mtype;

char mtext[MAX\_TEXT\_SIZE];

};

int main() {

int Message\_ID;

key\_t key;

struct message Message;

key = ftok(".", 'M');

Message\_ID = msgget(key, 0);

printf("успешно запущен\n");

while (1) {

printf("Введите сообщение: ");

fgets(Message.mtext, sizeof(Message.mtext), stdin);

Message.mtext[strcspn(Message.mtext, "\n")] = '\0';

Message.mtype = 1;

msgsnd(Message\_ID, &Message, sizeof(Message.mtext), 0);

msgrcv(Message\_ID, &Message, sizeof(Message.mtext), 2, 0);

printf("Получен ответ от сервера: %s\n", Message.mtext);

}

return 0;}

Листинг 2 – server.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#define MAX\_TEXT\_SIZE 256

struct message {

long mtype;

char mtext[MAX\_TEXT\_SIZE];

};

int main() {

int Message\_ID;

key\_t key;

struct message Message;

key = ftok(".", 'M');

Message\_ID = msgget(key, IPC\_CREAT | 0666);

printf("Сервер запущен. Ожидание сообщений...\n");

while (1) {

msgrcv(Message\_ID, &Message, sizeof(Message.mtext), 1, 0);

printf("Получено сообщение от клиента: %s\n", Message.mtext);

printf("Введите сообщение: ");

fgets(Message.mtext, sizeof(Message.mtext), stdin);

Message.mtext[strcspn(Message.mtext, "\n")] = '\0';

Message.mtype = 2;

msgsnd(Message\_ID, &Message, sizeof(Message.mtext), 0);

}

return 0;

}

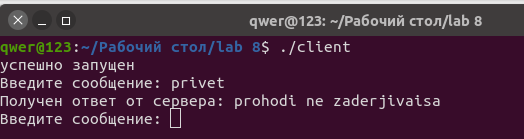


Рисунок 1 – Результат работы программы клиента

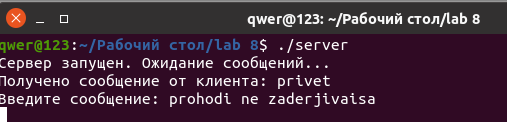


Рисунок 1 – Результат работы программы сервера