

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №6**

По дисциплине «ОСиСП»

за 4-й семестр

Тема: «Средства межпроцессного взаимодействия»

Выполнил:  
студент 2 курса  
группы ПО-3 (1)  
Гаврилкович Е.В.  
Проверил:  
Давидюк Ю.И.

## Ход работы:

### Вариант 5.

Написать программу, которая порождает дочерний процесс, и общается с ним через средства взаимодействия согласно варианту, передавая и получая информацию согласно варианту. Передачу и получение информации каждым из процессов сопровождать выводом на экран информации типа "процесс такой-то передал/получил такую-то информацию".

Сообщение вводит пользователь через терминал. Дочерние процессы начинают операции после получения сигнала SIGUSR1 от родительского процесса.

После отработки дочерний процесс должен возвращать результат родительскому процессу

5	Очереди сообщений	Родитель передает потомку три стороны треугольника, потомок возвращает его площадь
---	-------------------	--

Текст программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <mqueue.h>
#include "math.h"
#include <sys/types.h>
#include <sys/msg.h>
#include <sys/ipc.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <ctype.h>
#define MQNAME "/message"
/// Обработчик сигналов
void handler(int idsignal)
{

if (idsignal==0) {
printf("Parent send MS\n");
idsignal++;
```

```

    }
else {
    printf("Child received MS\n");
    idsignal--;
    }
}

int main()
{

    pid_t pid;
    printf("Parent proces\n");
    struct mq_attr mqAttr;
    int rc;
    char buffer[2048];
    char MQMSG [100];
    char str [23][3];
    int n;
    signal(SIGUSR1, handler);
    printf("Enter the sides of the triangle: ");
    gets(MQMSG);
    //Устанавливаем атрибуты очереди
    mqAttr.mq_maxmsg = 10;
    mqAttr.mq_msgsize = 1024;
    //Открываем очередь
    mqd_t sndHndl;
    sndHndl = mq_open (MQNAME, O_RDWR|O_CREAT, S_IWUSR|S_IRUSR,
    &mqAttr);
    if (sndHndl < 0) {
        printf ("\t\tError %d (%s) mq_open for send.\n",
            errno, strerror (errno));
        exit (-1);
    }
    printf ("\tOpened mqd_t of %d.\n", sndHndl);
    //посылаем сообщение в очередь
    rc = mq_send (sndHndl, MQMSG, sizeof (MQMSG), 1);
    if (rc < 0) {
        printf ("\t\tError %d (%s) mq_send.\n",
            errno, strerror (errno));
    }
}

```

```

        exit (-1);
    }
    printf ("\tParent send MS %d.\n", sndHndl);
    //Порождение 2 процесса
    if ((pid = fork()) == -1)
        printf("Erro!\n");
    else if (pid == 0)
    {
        printf("Child procces\n");
        mqd_t rcHndl;
        //открываем очередь
        rcHndl = mq_open (MQNAME, O_RDWR);
        if (rcHndl < 0) {
            printf ("\tError %d (%s) mq_open for receive.\n",
                    errno, strerror (errno));
            exit (-1);
        }
        printf ("\tOpened MS %d.\n", rcHndl);
        printf ("\tChild received MS %d.\n", rcHndl);
        //принимаем сообщение из очереди
        rc = mq_receive (rcHndl, buffer, sizeof (buffer), NULL);
        if (rc < 0) {
            printf ("\tError %d (%s) mq_receive.\n",
                    errno, strerror (errno));
            exit (-1);
        }
        char *pfir = strtok(buffer, " ");
        n = 0;
        while (pfir != NULL){
            strcpy(str[n], pfir);
            pfir = strtok(NULL, " ");
            n++;
        }
        //Нахождение площади
        double p=0;
        double res;
        double a = atof(str[0]);
        double b = atof(str[1]);
        double c = atof(str[2]);
    }

```

```

p=(a+b+c)/2;
res=pow((p*(p-a)*(p-b)*(p-c)),0.5);
printf("\nArea of a triangle: %f\n", res);
//удаляем очередь
if (mq_unlink (MQNAME) < 0) {
    printf ("Warning %d (%s) mq_unlink.\n",
            errno, strerror (errno));
}
printf("End Child process \n");
sleep(1);
exit(0);
}
sleep(2);
printf("End Parent procces \n");
exit(0);
return 0;
}

```

```

kate@kate-Lenovo-ideapad-100-15IBD:~/4 сем/ОСиСП/Lab6$ ./LAB_6
Parent procces
Enter the sides of the triangle: 3 4 5
    Opened mqd t of 3.
    Parent send MS 3.
Child procces
    Opened MS 4.
    Child received MS 4.

Area of a triangle: 6.000000
End Child process
End Parent procces

```

**Вывод:** были изучены Средства межпроцессного взаимодействия в Linux.