

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности
Кафедра математического обеспечения и применения ЭВМ

Отчет
По лабораторной работе №5
на тему: **«Обмен данными между процессами в Windows»**

по курсу
**«Операционные системы и системное программное
обеспечение»**
Вариант №8

Выполнил:
студент гр. КТ6о2-8
Нестеренко П.А.

Проверил:
Ассистент каф. МОП ЭВМ
Альминене Т. А.

1 Цель работы

Изучение механизмов межпроцессного обмена данными в Windows и приобретения навыков использования различных средств обмена данными при разработке программ.

2 Задание

Необходимо разработать программу, реализующую следующий алгоритм, используя Windows API:

Процесс А в цикле просит пользователя ввести 5 чисел и затем передает их процессу В. Процесс В отображает на экране эти числа в обратном порядке и спрашивает пользователя, продолжить работу или завершить процессы. Для передачи данных используется отображение страничного файла.

3 Ход работы

Для написания программы был выбран язык программирования C++. По ходу выполнения лабораторной работы был составлен следующий код:

Код программы «родительского» процесса:

```
#define _CRT_SECURE_NO_WARNINGS
#include "stdafx.h"
#include <windows.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <locale.h>

int main(void)
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
```

```

int errCode;
HANDLE hMapping, hDataSentEvent, hChooseEvent;
HANDLE hAnswerEvent;
int* Array;
int* trigger;
BOOL procBRuns;
STARTUPINFO startInfo = { sizeof(startInfo) };
PROCESS_INFORMATION procInfo;
SECURITY_ATTRIBUTES pipeAttributes = { sizeof(SECURITY_ATTRIBUTES), NULL, TRUE };

hMapping = CreateFileMapping(INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE, 0, 4096,
_T("Lab5_Mapping"));
Array = MapViewOfFile(hMapping, FILE_MAP_WRITE, 0, 0, 250);
trigger = MapViewOfFile(hMapping, FILE_MAP_WRITE, 0, 0, 2);
hDataSentEvent = CreateEvent(NULL, FALSE, FALSE, _T("Lab5"));
hAnswerEvent = CreateEvent(NULL, FALSE, FALSE, _T("Lab5"));
hChooseEvent = CreateEvent(NULL, FALSE, FALSE, _T("Lab5"));

procBRuns = CreateProcess(_T("../Debug\\Lab5_B.exe"), NULL, NULL, NULL, TRUE,
CREATE_NEW_CONSOLE, NULL, NULL, &startInfo, &procInfo);

if (!procBRuns) {
    errCode = 1;
    printf("%s", "Не удалось запустить процесс B\n");
    getchar();
    return 1;
}

WaitForSingleObject(hAnswerEvent, INFINITE);
do {
    printf(trigger);
    printf("%s", "Введите 5 чисел: ");
    for (int i = 0; i < 5; i++)
        scanf("%d", &Array[i]);

    SetEvent(hDataSentEvent);
    WaitForSingleObject(hAnswerEvent, INFINITE);
    printf("=====\n\n");
} while (trigger[0] == 1);

```

```

    UnmapViewOfFile(Array);
    CloseHandle(hMapping);

    return 0;
}

```

Код программы дочернего процесса:

```

#define _CRT_SECURE_NO_WARNINGS
#include "stdafx.h"
#include "windows.h"
#include <stdlib.h>
#include <locale.h>

int main(void)
{
    int* arrayReceived;
    int* trigger;
    HANDLE hMapping, hReceivedEvent, hAnswerEvent, hStopEvent;

    hMapping = OpenFileMapping(FILE_MAP_WRITE, FALSE, _T("Lab5_Mapping"));
    arrayReceived = MapViewOfFile(hMapping, FILE_MAP_WRITE, 0, 0, 250);
    trigger = MapViewOfFile(hMapping, FILE_MAP_WRITE, 0, 0, 2);
    hReceivedEvent = OpenEvent(EVENT_ALL_ACCESS, FALSE, _T("Lab5"));
    hAnswerEvent = OpenEvent(EVENT_ALL_ACCESS, FALSE, _T("Lab5"));

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    puts("");
    fflush(stdout);

    SetEvent(hAnswerEvent);

    do {
        int choose = 1;
        WaitForSingleObject(hReceivedEvent, INFINITE);
        printf("Получено от A:");
        for(int i = 4; i >= 0; i--)
            printf("%d ", arrayReceived[i]);
        printf("\nПродолжить выполнение?(1/0):");
    } while (1);
}

```

```

        scanf("%d", &choose);
        trigger[0] = choose;
        SetEvent(hAnswerEvent);
    } while (trigger[0] == 1);

    SetEvent(hAnswerEvent);
    UnmapViewOfFile(arrayReceived);

    CloseHandle(hMapping);

    return 0;
}

```

Исходный код был написан и протестирован в среде разработки Visual Studio 2019. Программа решает поставленную задачу и отработывает без ошибок.

4 Пояснения

Решение поставленного задания состоит из двух частей, которые располагаются в разных .cpp файлах, и соответственно компилируются в две различные программы.

В «родительском» исходном коде расположена логика создания «дочернего» процесса. Для процессов потомков создаются все необходимые для операционной системы структуры:

- 1) Структуры с информацией, необходимой для запуска дочерних процессов:

```
STARTUPINFO startInfo = { sizeof(startInfo) }; STARTUPINFO startupInfoB;
```

- 2) Структура, содержащая в себе информацию о процессах, их хэнды и переопределенные потоки входных и выходных данных (если это необходимо):

```
PROCESS_INFORMATION procInfo;
```

- 3) Хэндлеры входных и выходных структур, через которые дочерние процессы будут взаимодействовать с родительским:

```
HANDLE hMapping, hDataSentEvent, hChooseEvent;  
HANDLE hAnswerEvent;
```

- 4) С помощью команды `hMapping = CreateFileMapping(INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE, 0, 4096, _T("Lab5_Mapping"))`; Было создано отображение страничного файла через который родительский и дочерний процесс будут общаться.
- 5) Далее в цикле идёт приём 5 цифр в общую область памяти, затем с помощью команды `SetEvent(hDataSentEvent)`; программа сообщает что данные готовы для использования.
- 6) Дочерний процесс ждёт “ивентов” и так же в цикле в обратном порядке выводит все полученные данные из массива

5 Выводы

В ходе выполнения лабораторной работы я закрепил навыки работы со средствами управления потоками и процессами, а так познакомился с одним из возможных способов передачи сообщений (отображение страничного файла) между процессами в операционной системе Windows с помощью «Windows API».