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. КТбо2-8

Нестеренко П.А.

Проверил:

Ассистент каф. МОП ЭВМ

Альминене Т. А.

# 1 Цель работы

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

2 Задание

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

Процесс A в цикле просит пользователя ввести 5 чисел и затем передает их процессу B. Процесс B 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 С++. 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\* triger;

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);

triger = 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(triger);

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 (triger[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\* triger;

HANDLE hMapping, hReceivedEvent, hAnswerEvent, hStopEvent;

hMapping = OpenFileMapping(FILE\_MAP\_WRITE, FALSE, \_T("Lab5\_Mapping"));

arrayReceived = MapViewOfFile(hMapping, FILE\_MAP\_WRITE, 0, 0, 250);

triger = 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("\nПрододжить выполение?(1/0):");

scanf("%d", &choose);

triger[0] = choose;

SetEvent(hAnswerEvent);

} while (triger[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 отрабатывает 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;

1. Структура, содержащие в себе информацию о процессах, их хэндлы и переопределенные потоки входных и выходных данных (если это необходимо):

PROCESS\_INFORMATION procInfo;

1. Хэндлеры входных и выходных структур, через которые дочерние процессы будут взаимодействовать с родительским:

HANDLE hMapping, hDataSentEvent, hChooseEvent;

HANDLE hAnswerEvent;

1. 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.
2. Далее в цикле идёт приём 5 цифр в общую область памяти, затем с помощью команды SetEvent(hDataSentEvent); программа сообщает что данные готовы для использвоания.
3. 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».