Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине «Основы операционных систем»

## на тему «Порождение и завершение процессов и нитей»

Выполнили студенты группы 22ВВП1

Хоссейни Нежад С.А.С.М.

Захаров А. С.

Приняли:

Егоров В.Ю.

Федюнин Р.Н.

Пенза 2024

**Название**

Порождение и завершение процессов и нитей

**Цель работы**

Изучение методов и средств порождения процессов и нитей. Изучение способов синхронизации процессов и нитей через ожидание окончания их выполнения.

**Лабораторное задание**

Разработать программу, останавливающую и возобновляющую работу подчиненных нитей по очереди через равные интервалы времени.

**Описание программы**

В начале создаем нужное количество нитей в приостановочном состоянии с помощью параметра CREATE\_SUSPENDED в функции CreateThread.

Далее с помощью цикла for() мы проходим по всем нитям и возобновляем работу нити, ждем определенное время и приостанавливаем работу нити. Этот процесс будет продолжаться до тех пор, пока все нити не завершат работу, это выполнено с помощью помещения этого цикла в цикл do… while (WaitForMultipleObjects(THREAD\_COUNT, hThreads, TRUE, 50))

**Текст программы**

#include <windows.h>

#include <stdio.h>

#include <locale.h>

#define THREAD\_COUNT 3

DWORD WINAPI threadFunction()

{

Sleep(5000);

return 0;

}

int main()

{

setlocale(LC\_ALL, "Rus");

HANDLE hThreads[THREAD\_COUNT];

for (int i = 0; i < THREAD\_COUNT; i++)

{

hThreads[i] = CreateThread(NULL, 0, threadFunction, NULL, CREATE\_SUSPENDED, NULL);

if (hThreads[i] == NULL)

{

printf("Не удалось создать нить %d\n", i + 1);

return 1;

}

}

do

{

for (int i = 0; i < THREAD\_COUNT; i++)

{

ResumeThread(hThreads[i]);

printf("Нить номер %d начала / возобновила работу\n", i + 1);

Sleep(2000);

SuspendThread(hThreads[i]);

printf("Нить номер %d приостановлена\n\n", i + 1);

}

} while (WaitForMultipleObjects(THREAD\_COUNT, hThreads, TRUE, 50));

for (int i = 0; i < THREAD\_COUNT; i++)

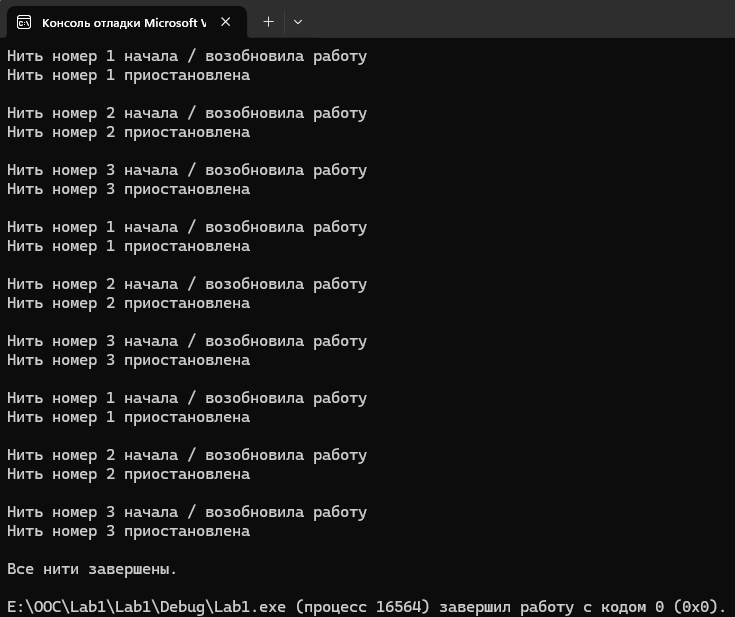
CloseHandle(hThreads[i]);

printf("Все нити завершены.\n");

return 0;

}

**Результат работы программы**

****

**Вывод**

Изучили методы и средства порождения процессов и нитей, способы синхронизации процессов и нитей через ожидание окончания их выполнения.