Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

ОТЧЁТ  
ЛАБОРАТОРНАЯ РАБОТА №4  
по дисциплине «Операционные системы и системное программное обеспечение»

на тему:

«Управление процессами в ОС Windows»  
Вариант 6

Выполнили  
студенты группы КТбо2-6 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю. А. Беликова \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. И. Домбрина

Принял  
ассистент кафедры МОП ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Гуляев

1 Цель работы

Целью данной работы является изучение средств управления процессами в ОС Windows, практическое овладение способами управления и синхронизации процессов и нитей.

2 Вариант задания

В программе процесса описан буферный массив целых чисел длиной 10 элементов. Буфер первоначально пуст. Нить A с интервалом в 2 с добавляет в буфер случайное число в диапазоне от 1 до 100. Если буфер полон, нить ожидает свободного места. Нить B ожидает, чтобы в буфере накопилось не менее трех чисел, удаляет из буфера три последних числа и подсчитывает их сумму. Затем нить B делает паузу случайной длительности от 4 до 8 с. Обе нити выводят на экран сообщения о своих действиях. Работа прекращается после того, как нить B подсчитает 20 сумм.

**3 Ход работы**

Пример выполнялся при помощи языка программирования С.

Для создания объекта-события используется функция CreateEvent. При помощи API-функции CreateThread создаем две нити и вызываем функции RandomNumber (для первой нити, отвечает за добавление случайного числа в буфер) и DeleteNumber (для второй нити, отвечает за удаление чисел). Для синхронизации этих нитей используем системную функцию ожидания WaitForMultipleObjects.

Листинг программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "windows.h"

#include <stdlib.h>

#include <stdio.h>

#include <locale.h>

#include <tchar.h>

#include <time.h>

HANDLE Waiting;

int Buffer[10];

int CurrentSize = 0, Count = 0;

DWORD WINAPI RandomNumber() {

srand(time(NULL));

Waiting = OpenEvent(EVENT\_ALL\_ACCESS, FALSE, \_T("Waiting"));

printf("%s\n", "Нить №1. Начало.");

while (TRUE) {

endwaiting:

if (Count == 20) {

printf("%s\n", "Нить №1. Конец.");

return 0;

}

if (CurrentSize < 10 && Count != 20) {

Buffer[CurrentSize] = ((rand()) % 100) + 1;

CurrentSize++;

}

else {

WaitForSingleObject(Waiting, INFINITE);

goto endwaiting;

}

printf("%s\n", "Нить №1. Добавлено число в буфер.");

Sleep(2000);

ResetEvent(Waiting);

}

}

DWORD WINAPI DeleteNumber() {

srand(time(NULL));

printf("%s\n", "Нить №2. Начало.");

while (TRUE) {

int Sum = 0;

int Wait = (((rand()) % 5) + 4) \* 1000;

if (CurrentSize > 2) {

Sum = Buffer[CurrentSize] + Buffer[CurrentSize - 1] + Buffer[CurrentSize - 2];

Buffer[CurrentSize] = 0;

Buffer[CurrentSize - 1] = 0;

Buffer[CurrentSize - 2] = 0;

CurrentSize -= 3;

Count++;

SetEvent(Waiting);

printf("%s%d.\n", "Нить №2. Удалены 3 последних числа. Их сумма равна = ", Sum);

if (Count == 20) {

printf("%s\n", "Нить №2. Конец.");

return 0;

}

Sleep(Wait);

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

system("cls");

HANDLE Threads[2];

Waiting = CreateEvent(NULL, TRUE, FALSE, \_T("Waiting"));

Threads[0] = CreateThread(NULL, NULL, RandomNumber, NULL, 0, NULL);

Threads[1] = CreateThread(NULL, NULL, DeleteNumber, NULL, 0, NULL);

WaitForMultipleObjects(2, Threads, TRUE, INFINITE);

CloseHandle(Waiting);

CloseHandle(Threads[0]);

CloseHandle(Threads[1]);

system("pause");

return 0;

}

**4 Пример**

Пример выполнения работы программы. (Рисунок 1)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1-Результат выполнения

**5 Вывод**

Во время выполнения лабораторной работы мы ознакомились со средствами управления процессами в ОС Windows и приобрели практические навыки по управлению и синхронизации процессов и нитей.