|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8** | |
| **по дисциплине** | |
| «Системное программное обеспечение»  по теме: «Потоки» | |
|  | |
| Выполнил студент группы ИКБО-16-18 | Пономарев А.Р. |
| Принял преподаватель | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

Москва 2021

**Цель работы**

Получение практических навыков по использованию Win32 API для исследования потоков.

**Задание**

1. Учитывая особенности методов синхронизации, сформулировать три задачи, демонстрирующие возможности ОС по синхронизации потоков следующими способами:
   1. Критические секции;
   2. Мьютексы
   3. События.

**Ход работы**

Итоговая программа представлена в листинге 1. Результат работы представлен на рисунках 1-3.

Листинг 1

#include <windows.h>

#include <stdio.h>

#include <iostream>

CRITICAL\_SECTION cs;

HANDLE hMutex;

HANDLE hEvent1, hEvent2;

int a[5];

HANDLE hThr;

unsigned long uThrID;

using namespace std;

void ThreadCrit(void\* pParams)

{

int i, num = 0;

while (1)

{

EnterCriticalSection(&cs);

for (i = 0; i < 5; i++) a[i] = num;

num++;

LeaveCriticalSection(&cs);

}

}

void ThreadMutex(void\* pParams)

{

int i, num = 0;

while (1)

{

WaitForSingleObject(hMutex, INFINITE);

for (i = 0; i < 5; i++) a[i] = num;

num++;

ReleaseMutex(hMutex);

}

}

void ThreadEvent(void\* pParams)

{

int i, num = 0;

while (1)

{

WaitForSingleObject(hEvent2, INFINITE);

for (i = 0; i < 5; i++) a[i] = num;

num++;

SetEvent(hEvent1);

}

}

int main(void)

{

setlocale(LC\_ALL, "Russian");

cout << "Выберите вариант:\n1-Критические секции\n2-Мьютексы\n3-События\n";

int v;

cin >> v;

switch (v)

{

case 1:

InitializeCriticalSection(&cs);

hThr = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ThreadCrit, NULL, 0, &uThrID);

while (1)

{

EnterCriticalSection(&cs);

printf("%d %d %d %d %d\n", a[0], a[1], a[2], a[3], a[4]);

LeaveCriticalSection(&cs);

}

break;

case 2:

hMutex = CreateMutex(NULL, FALSE, NULL);

hThr = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ThreadMutex, NULL, 0, &uThrID);

while (1)

{

WaitForSingleObject(hMutex, INFINITE);

printf("%d %d %d %d %d\n", a[0], a[1], a[2], a[3], a[4]);

ReleaseMutex(hMutex);

}

break;

case 3:

hEvent1 = CreateEvent(NULL, FALSE, TRUE, NULL);

hEvent2 = CreateEvent(NULL, FALSE, FALSE, NULL);

hThr = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ThreadEvent , NULL, 0, &uThrID);

while (1)

{

WaitForSingleObject(hEvent1, INFINITE);

printf("%d %d %d %d %d\n", a[0], a[1], a[2], a[3], a[4]);

SetEvent(hEvent2);

}

break;

default:

cout << "Неправильный вариант";

break;

}

return 0;

}

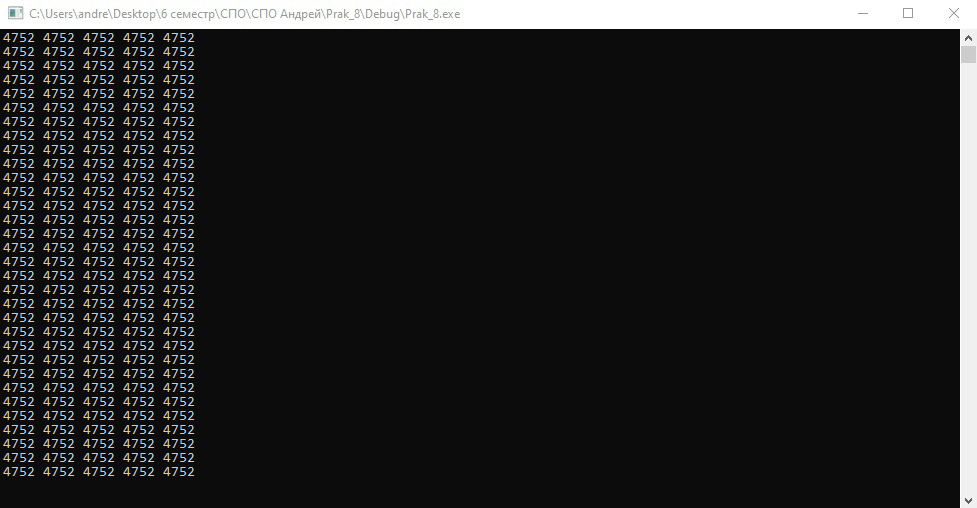


Рис. 1 – Критические секции.

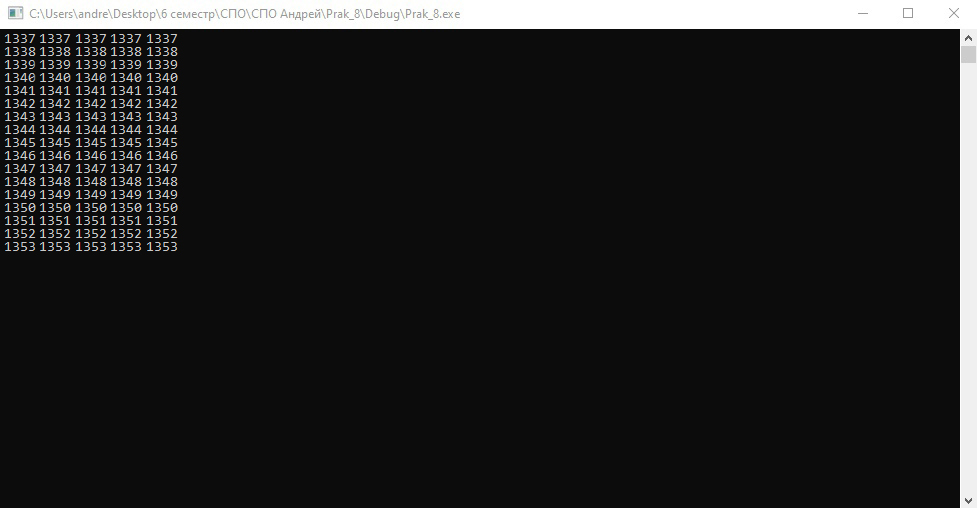


Рис. 2 – Мьютексы.

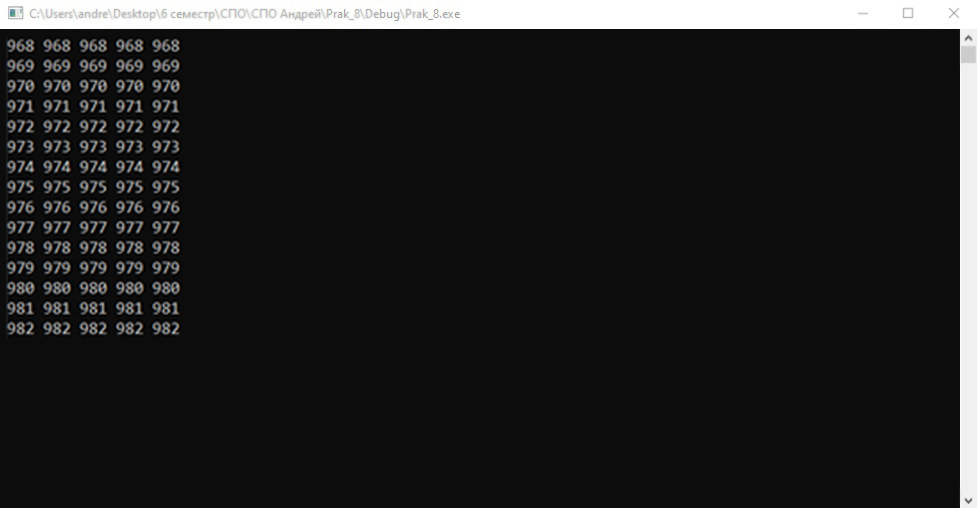


Рис. 3 – События.

**Вывод**

В ходе практической работы мы изучили работу с потоками с помощью Win32 API.

**Список использованных источников**

1. Пошаговое руководство. Создание классического оконного приложение [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160
2. Администрирование в Windows 10. [Электронный ресурс] : [https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10 /](https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10%20/) Ричард Ворд 2020 — 100 с.
3. «Тайм-менеджмент для системных администраторов». / Лимончелли Т.А. М.: Символ-плюс 2007 — 247 с. ISBN:978-5-93286-090-8