



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## Отчёт по лабораторной работе №6 по курсу «Операционные системы»

Тема Реализация монитора Хоара «Читатели-писатели» под ОС Windows

Студент Якуба Д.В.

Группа ИУ7-53Б

Оценка (баллы)

Преподаватель Рязанова Н.Ю.

Москва, 2020 г

## Задание.

В лабораторной работе необходимо разработать многопоточное приложение, используя API ОС Windows такие как, потоки, события (event) и мьютексы (mutex). Потоки разделяют единственную глобальную переменную. Приложение реализует монитор Хоара «Читатели-писатели».

Листинг кода 1

```
#include <stdio.h>
#include <stdbool.h>
#include <windows.h>

#define WRITERS_SLEEP_IN_MILLI 1000
#define READERS_SLEEP_IN_MILLI 1000

#define WRITERS_AMOUNT 3
#define READERS_AMOUNT 5

#define AMOUNT_OF_ACTIONS 4

#define SUCCESS 0

#define MUTEX_ERR 1
#define EVENT_ERR 2
#define WRITER_ERR 3
#define READER_ERR 4

HANDLE mutex, canRead, canWrite;

HANDLE writersThreads[WRITERS_AMOUNT];
HANDLE readersThreads[READERS_AMOUNT];

bool writing = false; // Активный писатель

LONG writersInQueue = 0; // Ждущие писатели
LONG readersInQueue = 0; // Ждущие читатели
LONG readingMembers = 0; // Активные читатели

int sharedMemory = 0;

void startWrite()
{
    InterlockedIncrement(&writersInQueue);

    if (writing || readingMembers > 0)
        WaitForSingleObject(canWrite, INFINITE);

    InterlockedDecrement(&writersInQueue);
    writing = true;
}

void stopWrite()
{
    writing = false;

    if (readersInQueue > 0)
        SetEvent(canRead);
    else
        SetEvent(canWrite);
}

DWORD WINAPI writer(LPVOID lpParams)
{
    for (int i = 0; i < AMOUNT_OF_ACTIONS; i++)
```

```

    {
        startWrite();

        sharedMemory++;
        printf("<---Writer[ID = %d]: write value %d\n", lpParams, sharedMemory);

        stopWrite();
        Sleep(WRITERS_SLEEP_IN_MILLI);
    }

    return SUCCESS;
}

void startRead()
{
    WaitForSingleObject(mutex, INFINITE);

    InterlockedIncrement(&readersInQueue);

    if (writing || writersInQueue > 0)
        WaitForSingleObject(canRead, INFINITE);

    InterlockedIncrement(&readingMembers);
    SetEvent(canRead);

    ReleaseMutex(mutex);
}

void stopRead()
{
    InterlockedDecrement(&readingMembers);

    if (readingMembers == 0)
        SetEvent(canWrite);
}

DWORD WINAPI reader(LPVOID lpParams)
{
    while (sharedMemory < WRITERS_AMOUNT * AMOUNT_OF_ACTIONS)
    {
        startRead();

        printf("->>Reader[ID = %d]: read value %d\n", lpParams, sharedMemory);

        stopRead();
        Sleep(READERS_SLEEP_IN_MILLI);
    }

    return SUCCESS;
}

int main()
{
    if ((mutex = CreateMutex(NULL, FALSE, NULL)) == NULL)
    {
        perror("Mutex error");
        exit(MUTEX_ERR);
    }

    if ((canRead = CreateEvent(NULL, FALSE, TRUE, NULL)) == NULL)
    {
        perror("CanRead event error");
        exit(EVENT_ERR);
    }

    if ((canWrite = CreateEvent(NULL, FALSE, TRUE, NULL)) == NULL)
    {
        perror("CanWrite event error");
    }
}

```

```

        exit(EVENT_ERR);
    }

    for (int i = 0; i < WRITERS_AMOUNT; i++)
        if ((writersThreads[i] = CreateThread(NULL, 0, writer, i, 0, NULL)) == NULL)
        {
            perror("Writer thread creation error");
            exit(WRITER_CRT_ERR);
        }

    for (int i = 0; i < READERS_AMOUNT; i++)
        if ((readersThreads[i] = CreateThread(NULL, 0, reader, i, 0, NULL)) == NULL)
        {
            perror("Reader thread creation error");
            exit(READER_CRT_ERR);
        }

    WaitForMultipleObjects(WRITERS_AMOUNT, writersThreads, TRUE, INFINITE);
    WaitForMultipleObjects(READERS_AMOUNT, readersThreads, TRUE, INFINITE);

    CloseHandle(mutex);
    CloseHandle(canRead);
    CloseHandle(canWrite);

    return SUCCESS;
}

```

Демонстрация работы программы предоставлена на следующих страницах.

```
<<---Writer[ID = 0]: write value 1
<<---Writer[ID = 1]: write value 2
->>Reader[ID = 0]: read value 2
->>Reader[ID = 3]: read value 2
->>Reader[ID = 1]: read value 2
->>Reader[ID = 2]: read value 2
->>Reader[ID = 4]: read value 2
<<---Writer[ID = 2]: write value 3
<<---Writer[ID = 0]: write value 4
->>Reader[ID = 3]: read value 4
->>Reader[ID = 0]: read value 4
->>Reader[ID = 1]: read value 4
->>Reader[ID = 2]: read value 4
->>Reader[ID = 4]: read value 4
<<---Writer[ID = 1]: write value 5
<<---Writer[ID = 0]: write value 6
->>Reader[ID = 1]: read value 6
->>Reader[ID = 3]: read value 6
->>Reader[ID = 0]: read value 6
->>Reader[ID = 2]: read value 6
->>Reader[ID = 4]: read value 6
<<---Writer[ID = 2]: write value 7
->>Reader[ID = 1]: read value 7
<<---Writer[ID = 1]: write value 8
->>Reader[ID = 3]: read value 8
->>Reader[ID = 0]: read value 8
->>Reader[ID = 2]: read value 8
->>Reader[ID = 4]: read value 8
<<---Writer[ID = 0]: write value 9
<<---Writer[ID = 1]: write value 10
->>Reader[ID = 1]: read value 10
->>Reader[ID = 0]: read value 10
->>Reader[ID = 3]: read value 10
->>Reader[ID = 2]: read value 10
->>Reader[ID = 4]: read value 10
<<---Writer[ID = 2]: write value 11
->>Reader[ID = 1]: read value 11
->>Reader[ID = 0]: read value 11
->>Reader[ID = 2]: read value 11
->>Reader[ID = 3]: read value 11
->>Reader[ID = 4]: read value 11
<<---Writer[ID = 2]: write value 12
-
```

Рисунок 1, Демонстрация работы программы (задержка писателей 1 секунда, задержка читателей 1 секунда)

```
<<---Writer[ID = 0]: write value 1
<<---Writer[ID = 1]: write value 2
->>Reader[ID = 0]: read value 2
->>Reader[ID = 1]: read value 2
->>Reader[ID = 3]: read value 2
->>Reader[ID = 4]: read value 2
->>Reader[ID = 2]: read value 2
<<---Writer[ID = 2]: write value 3
->>Reader[ID = 3]: read value 3
->>Reader[ID = 1]: read value 3
->>Reader[ID = 0]: read value 3
->>Reader[ID = 4]: read value 3
->>Reader[ID = 2]: read value 3
->>Reader[ID = 3]: read value 3
->>Reader[ID = 1]: read value 3
->>Reader[ID = 0]: read value 3
->>Reader[ID = 2]: read value 3
->>Reader[ID = 4]: read value 3
<<---Writer[ID = 1]: write value 4
<<---Writer[ID = 0]: write value 5
->>Reader[ID = 3]: read value 5
->>Reader[ID = 0]: read value 5
->>Reader[ID = 1]: read value 5
->>Reader[ID = 2]: read value 5
->>Reader[ID = 4]: read value 5
<<---Writer[ID = 2]: write value 6
->>Reader[ID = 3]: read value 6
->>Reader[ID = 0]: read value 6
->>Reader[ID = 1]: read value 6
->>Reader[ID = 2]: read value 6
->>Reader[ID = 4]: read value 6
<<---Writer[ID = 0]: write value 7
<<---Writer[ID = 1]: write value 8
->>Reader[ID = 3]: read value 8
->>Reader[ID = 1]: read value 8
->>Reader[ID = 0]: read value 8
->>Reader[ID = 4]: read value 8
->>Reader[ID = 2]: read value 8
<<---Writer[ID = 2]: write value 9
->>Reader[ID = 3]: read value 9
->>Reader[ID = 4]: read value 9
->>Reader[ID = 1]: read value 9
->>Reader[ID = 0]: read value 9
->>Reader[ID = 2]: read value 9
->>Reader[ID = 3]: read value 9
->>Reader[ID = 4]: read value 9
->>Reader[ID = 0]: read value 9
->>Reader[ID = 1]: read value 9
->>Reader[ID = 2]: read value 9
<<---Writer[ID = 1]: write value 10
<<---Writer[ID = 0]: write value 11
<<---Writer[ID = 2]: write value 12
```

Рисунок 2, Демонстрация работы программы (задержка писателей 2.5 секунды, задержка читателей 1 секунда)

```
<<---Writer[ID = 0]: write value 1
<<---Writer[ID = 1]: write value 2
->>Reader[ID = 0]: read value 2
->>Reader[ID = 1]: read value 2
->>Reader[ID = 2]: read value 2
->>Reader[ID = 3]: read value 2
->>Reader[ID = 4]: read value 2
<<---Writer[ID = 2]: write value 3
<<---Writer[ID = 0]: write value 4
<<---Writer[ID = 2]: write value 5
<<---Writer[ID = 0]: write value 6
->>Reader[ID = 0]: read value 6
->>Reader[ID = 2]: read value 6
->>Reader[ID = 4]: read value 6
->>Reader[ID = 1]: read value 6
->>Reader[ID = 3]: read value 6
<<---Writer[ID = 1]: write value 7
<<---Writer[ID = 0]: write value 8
<<---Writer[ID = 1]: write value 9
<<---Writer[ID = 1]: write value 10
->>Reader[ID = 0]: read value 10
->>Reader[ID = 4]: read value 10
->>Reader[ID = 1]: read value 10
->>Reader[ID = 2]: read value 10
->>Reader[ID = 3]: read value 10
<<---Writer[ID = 2]: write value 11
<<---Writer[ID = 2]: write value 12
```

Рисунок 3, Демонстрация работы программы (задержка писателей 1 секунда, задержка читателей 2.5 секунды)