МГТУ им. Баумана

Лабораторная работа №6

По курсу: "Операционные системы"

«Реализация монитора Хоара «Читатели-писатели» под ОС Windows» (Hoare C.A.R.)

Работу выполнил: студент группы ИУ7-53Б
Наместник Анастасия

Преподаватель: Рязанова Н. Ю.

Москва, 2020

На листингах 1,...,5 представлено многопоточное приложение, демонстрирующее реализацию монитора Хоара «Читатели-писатели».

Листинг 1: Начальные установки

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

#define WHITE "\033[0m"
#define GREEN "\033[0;32m"

#define WRITERS 5
#define READERS 3
#define ITERATIONS_NUMBER 100
#define HANDLE_ERROR 1
#define THREAD_ERROR 2

HANDLE CanWrite;
HANDLE CanRead;
HANDLE MUTEX;
LONG SHARED_RESOURCE = 0;

bool active_writer = false;
LONG active_readers = 0;
LONG writers_queue = 0;  //quantity of writers waiting for
    CanWrite
LONG readers_queue = 0;  //quantity of readers waiting for
    CanRead

HANDLE writerThreads[WRITERS], readerThreads[READERS];
int writerID[WRITERS], readerID[READERS];
int value = 0;
```

Листинг 2: Код подпрограммы main()

```
int main(void)
{
    setbuf(stdout, NULL);
    srand(time(NULL));
```

```
5
6       if (!(MUTEX = CreateMutex(NULL, FALSE, NULL)))
7           return HANDLE_ERROR;
8
9       if (!(CanWrite = CreateEvent(NULL, FALSE, FALSE, NULL))
            )
10          return HANDLE_ERROR;
11      if (!(CanRead = CreateEvent(NULL, FALSE, FALSE, NULL)))
12          return HANDLE_ERROR;
13
14      if (Create_Threads() == THREAD_ERROR)
15          return THREAD_ERROR;
16
17      WaitForMultipleObjects(WRITERS, writerThreads, TRUE,
            INFINITE);
18      WaitForMultipleObjects(READERS, readerThreads, TRUE,
            INFINITE);
19
20      for (int i = 0; i < WRITERS; i++)
21          CloseHandle(writerThreads[i]);
22
23      for (int i = 0; i < READERS; i++)
24          CloseHandle(readerThreads[i]);
25
26      CloseHandle(CanWrite);
27      CloseHandle(CanRead);
28      CloseHandle(MUTEX);
29
30    return 0;
31  }
```

Листинг 3: Код подпрограммы создания потоков

```
1  int Create_Threads()
2  {
3      DWORD id = 0; //thread id
4
5      for (int i = 0; i < WRITERS; i++)
6      {
7          writerID[i] = i;
8          if (!(writerThreads[i] = CreateThread(NULL, 0, &
```

```
            Write, writerID + i, 0, &id)))
9               return THREAD_ERROR;
10      }
11
12      for (int i = 0; i < READERS; i++)
13      {
14          readerID[i] = i;
15          if (!(readerThreads[i] = CreateThread(NULL, 0, &
                Read, readerID + i, 0, &id)))
16              return THREAD_ERROR;
17      }
18      return 0;
19  }
```

Листинг 4: Код подпрограмм создания и работы писателя

```
1   void Start_Write()
2   {
3       InterlockedIncrement(&writers_queue);
4
5       if (active_readers > 0 || active_writer)
6           WaitForSingleObject(CanWrite, INFINITE);
7
8       InterlockedDecrement(&writers_queue);
9       active_writer = true;
10  }
11
12  void Stop_Write()
13  {
14
15      active_writer = false;
16
17      if (WaitForSingleObject(CanRead, 0) != WAIT_OBJECT_0)
18          SetEvent(CanRead);
19      else
20          SetEvent(CanWrite);
21  }
22
23  DWORD WINAPI Write(LPVOID Id)
24  {
25      int id = *(int *)Id;
```

```
26
27      for (int i = 0; i < ITERATIONS_NUMBER; i++)
28      {
29          int delay = rand() % 200;
30
31          Start_Write();
32          value++;
33          printf("%sWriter with id = %d wrote %d. Delay = %d\
                n", GREEN, id, value, delay);
34          Stop_Write();
35
36          Sleep(delay);
37      }
38  }
```

Листинг 5: Код подпрограмм создания и работы читателя

```
1   void Start_Read()
2   {
3
4       InterlockedIncrement(&readers_queue);
5
6       if (active_writer || WaitForSingleObject(CanWrite, 0)
           == WAIT_OBJECT_0)
7           WaitForSingleObject(CanRead, INFINITE);
8
9       WaitForSingleObject(MUTEX, INFINITE);
10
11      InterlockedDecrement(&readers_queue);
12      InterlockedIncrement(&active_readers);
13      SetEvent(CanRead);
14
15      ReleaseMutex(MUTEX);
16  }
17
18  void Stop_Read()
19  {
20      InterlockedDecrement(&active_readers);
21
22      if (active_readers == 0)
23          SetEvent(CanWrite);
```

```
24 }
25
26 DWORD WINAPI Read(LPVOID Id)
27 {
28     int id = *(int *)Id;
29
30     for (int i = 0; i < ITERATIONS_NUMBER; i++)
31     {
32         int delay = rand() % 200;
33
34         Start_Read();
35         printf("%sReader with id = %d read %d. Delay = %d\n
               ", WHITE, id, value, delay);
36         Stop_Read();
37
38         Sleep(delay);
39     }
40 }
```

На рисунке 1 приведен результат работы программы.

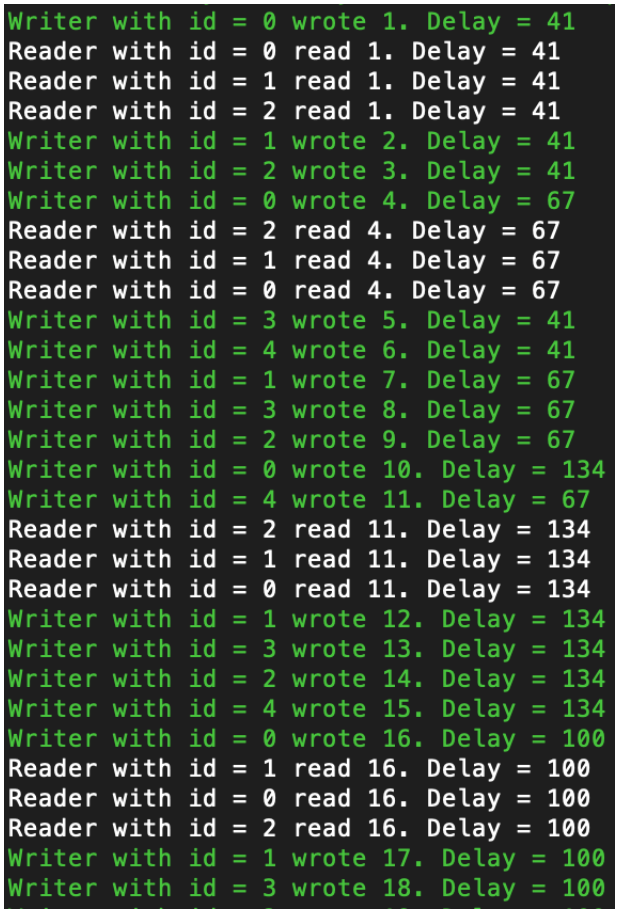

Рис 1: Результат работы программы