Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**11**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | **Using Counting semaphore, Create 10 numbers of Threads. Initialize the Semaphore value to 3.** |
| 2 | **Using Binary semaphore make second thread signaled and put first thread in waiting state.**   * + **First thread perform cube of given number at run time**   + **Second thread perform sorting of random integer list** |
|  |  |

Submitted On:

**12/01/2023**

**Task # 01: Using Counting semaphore, Create 10 numbers of Threads. Initialize the Semaphore value to 3.**

**Solution:**

#include<iostream>

#include<Windows.h>

using namespace std;

HANDLE hsemaphore;

DWORD threadId;

DWORD WINAPI MyFun(LPVOID lpParam) {

DWORD dwwaitresult;

BOOL bcondition = TRUE;

while (bcondition) {

dwwaitresult = WaitForSingleObject(hsemaphore, INFINITE);

switch (dwwaitresult)

{

case WAIT\_OBJECT\_0:

cout << "\nThread ID: " << GetCurrentThreadId() << endl;

bcondition = false;

Sleep(2000);

ReleaseSemaphore(hsemaphore, 1, 0);

break;

case WAIT\_TIMEOUT:

cout << "Wait is over" << endl;

break;

default:

break;

}

}

}

int main() {

hsemaphore = CreateSemaphore(NULL, 3, 3, NULL);

HANDLE hthread[10];

for (int i = 0; i < 10; i++) {

hthread[i] = CreateThread(NULL, 0, MyFun, NULL, 0, &threadId);

}

WaitForMultipleObjects(8, hthread, TRUE, INFINITE);

for (int j = 0; j < 10; j++) {

CloseHandle(hthread[10]);

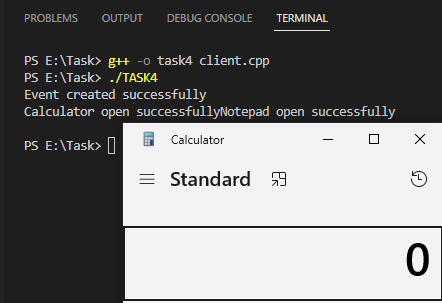
}

CloseHandle(hsemaphore);

return 0;

}

**Ouput:**



**Task # 02: Using Binary semaphore make second thread signaled and put first thread in waiting state.**

* + **First thread performs cube of given number at run time**
  + **Second thread perform sorting of random integer list**

**Solution:**

#include<iostream>

#include<Windows.h>

#include<math.h>

using namespace std;

HANDLE hTHreadA, hThreadB;

HANDLE hsemaphore;

DWORD threadid1, threadid2;

int a = 2, b = 1, c;

void bubbleSort(int arr[], int n)

{

int i, j;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - i - 1; j++)

if (arr[j] > arr[j + 1])

swap(arr[j], arr[j + 1]);

}

void ArraySorting() {

int arr[6] = { 4,2,7,1,9,8 };

cout << "\nArray before sorting: " << endl;

for (int i = 0; i < 6; i++) {

cout << arr[i] << " ";

}

cout << "\nAfter Sorting: " << endl;

bubbleSort(arr, 6);

for (int i = 0; i < 6; i++) {

cout << arr[i] << " ";

}

}

DWORD WINAPI MyFunA(void\* lpParam) {

ReleaseSemaphore(hsemaphore, 1, 0);

int a = (int)lpParam;

cout << "Cube of a number:" << a \* a \* a << endl;

cout << endl;

system("pause");

return 0;

}

DWORD WINAPI myFunB(LPVOID lpParam) {

WaitForSingleObject(hsemaphore, INFINITE);

ArraySorting();

// cout<<endl;

return 0;

}

int main() {

cout << "Enter Number: ";

int a;

system("pause");

cin >> a;

hTHreadA = CreateThread(NULL, 0, MyFunA, (void\*)a, 0, &threadid1);

hThreadB = CreateThread(NULL, 0, myFunB, NULL, 0, &threadid2);

WaitForSingleObject(hTHreadA, INFINITE);

WaitForSingleObject(hThreadB, INFINITE);

CloseHandle(hTHreadA);

CloseHandle(hThreadB);

system("pause");

return 0;

}

**Ouput:**