**ПРОЦЕССЫ И ПОТОКИ. ПРОДОЛЖЕНИЕ**

1. ЗАПУСКАЕМ ПОТОК

Поток выполняет последовательный счет, но через пять секунд обрывается основным процессом.

#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include "string.h"

DWORD WINAPI myThread(LPVOID lpParameter)

{

unsigned int counter=0;

while(counter<20)

{

Sleep(1000);

counter++;

printf("\n Counter= %d",counter);

}

return 0;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

unsigned int myCounter = 0;

DWORD myThreadID;

HANDLE myHandle = CreateThread(0, 0, myThread, &myCounter, 0, &myThreadID);

Sleep(5000);

printf("\n Main Process sagt");

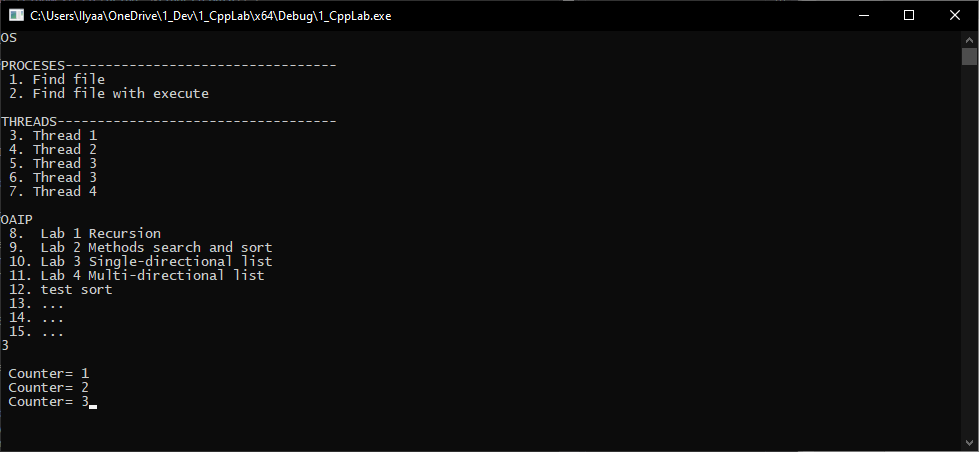
TerminateThread(myHandle,0);

getchar();

return 0;

}

Приведенный пример запускает поток myThread. Здесь важны третий и четвертый параметры. Третий параметр – имя функции, выполняемой потоком. Четвертый параметр – ссылка на переменную, передающую параметры. Например, эта переменная может представлять структуру, в которую помещены настоящие аргументы. Приведем пример передачи целочисленного параметра z;



#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include "string.h"

DWORD WINAPI myThread(void\* lpParameter)

{

int\* counterp= new int();

counterp=(int\*) lpParameter;

int counter=\*counterp;

while(counter<20)

{

Sleep(1000);

counter++;

printf("\n Counter= %d",counter);

}

return 0;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int z=5;

unsigned int myCounter = 0;

DWORD myThreadID;

HANDLE myHandle = CreateThread(0, 0, myThread, (void\*)&z, 0, &myThreadID);

Sleep(5000);

printf("\n Main Process sagt");

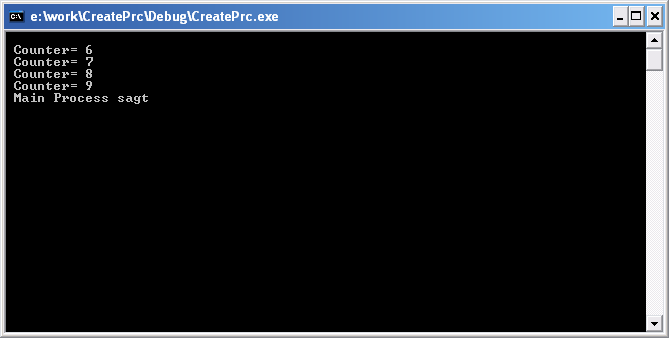
TerminateThread(myHandle,0);

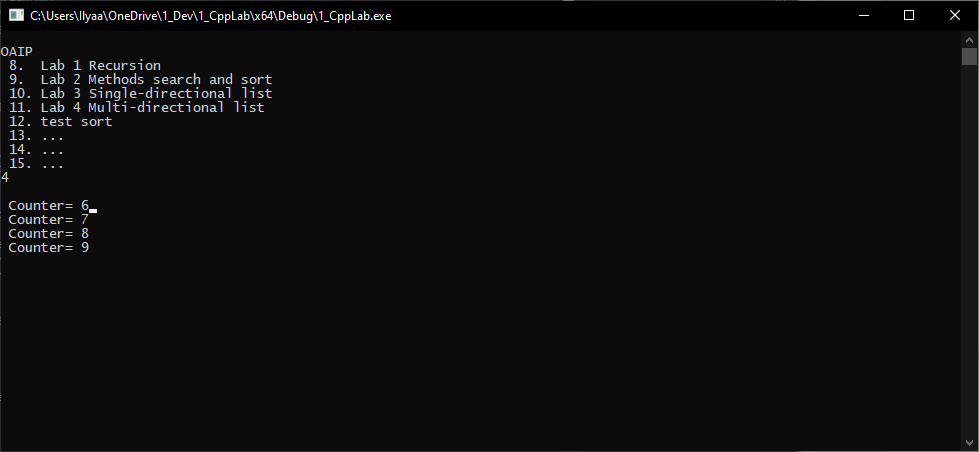
getchar();

return 0;

}

Данная программа выполняет простой отсчет, начиная с 6.





**ЗАДАНИЕ**. Запустите 2 потока, которые выполняют вывод чисел на консоль вперемежку.

**РЕШЕНИЕ:**

int counter = NULL;

int myRandom(time\_t\* mtime) {

srand(time(mtime));

return 1 + rand() % 10;

}

DWORD WINAPI myThreadRand2(void\* lpParameter)

{

HANDLE\* h = new HANDLE();

h = (HANDLE\*)lpParameter;

HANDLE hh = \*h;

while (counter < 10)

{

Sleep(300);

counter++;

printf("%d", myRandom(0));

}

return 0;

}

DWORD WINAPI myThreadRand1(void\* lpParameter)

{

int\* counterp = new int();

counterp = (int\*)lpParameter;

int counter = \*counterp;

while (counter < 10)

{

Sleep(300);

counter++;

printf("%d", myRandom(0));

}

return 0;

}

int mainThread5(int argc, \_TCHAR\* argv[]){

DWORD myThreadID, myThreadID2;

HANDLE myHandle = CreateThread(0, 0, myThreadRand1, (void\*)&counter, 0, &myThreadID);

Sleep(1000);

HANDLE myHandle2 = CreateThread(0, 0, myThreadRand2, (void\*)&myHandle, 0, &myThreadID2);

Sleep(1000);

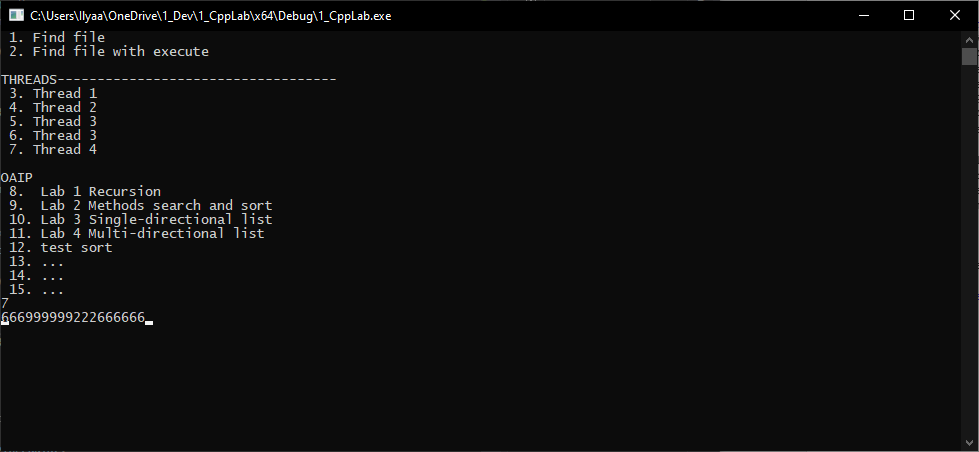
TerminateThread(myHandle, 0);

TerminateThread(myHandle2, 0);

getchar();

return 0;

}



РАБОТА НЕСКОЛЬКИХ ПОТОКОВ

Следующая программа запускает два потока. Первый поток начинает отсчет. Затем его приостанавливает второй поток. После приостановки первый поток вновь возобновляет работу. Приостановку делает функция

SuspendThread. Возобновление работы делает функция ResumeThread . В качестве аргумента в обеих функциях передается заголовок потока.

#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include "string.h"

DWORD WINAPI myThread2(void\* lpParameter)

{

HANDLE\* h= new HANDLE();

h=(HANDLE\*) lpParameter;

HANDLE hh=\*h;

printf("\n Thread 2 tries to suspend thread1");

SuspendThread(hh);

printf("\n Thread1 is suspended for 3 seconds");

Sleep(3000);

ResumeThread(hh);

printf("\n Thread is resumed");

return 0;

}

DWORD WINAPI myThread(void\* lpParameter)

{

int\* counterp= new int();

counterp=(int\*) lpParameter;

int counter=\*counterp;

while(counter<30)

{

Sleep(500);

counter++;

printf("\n Counter= %d",counter);

}

return 0;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int z=5;

unsigned int myCounter = 0;

DWORD myThreadID;

HANDLE myHandle = CreateThread(0, 0, myThread, (void\*)&z, 0, &myThreadID);

Sleep(5000);

DWORD myThreadID2;

HANDLE myHandle2 = CreateThread(0, 0, myThread2, (void\*)&myHandle, 0, &myThreadID2);

Sleep(15000);

printf("\n Main Process sagt");

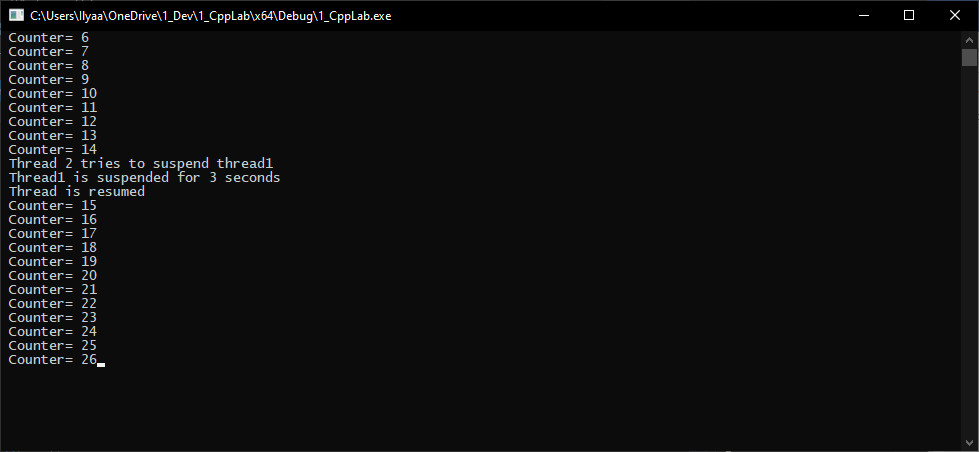
TerminateThread(myHandle2,0);

TerminateThread(myHandle,0);

getchar();

return 0;

}



**ЗАДАНИЕ**.

Написать приложение с двумя потоками. Один поток – писатель. Второй – читатель. Писатель записывает в ячейку (глобальную переменную) числа, производя отсчет через заданное время (Sleep(300)). После того как писатель запишет число, он запускает читателя, который считывает число и выдает его на консоль.

**РЕШЕНИЕ:**

int z = 0;

DWORD WINAPI myThreadReader(void\* lpParameter)

{

cout <<"Читатель считывает число, сейчас в ячейке находится число = " << z << "\n";

return 0;

}

DWORD WINAPI myThreadWriter(void\* lpParameter){

int i = 0;

cout << "Писатель записывает число ";

srand(time(0));

z = 1 + rand() % 10;

while (i < 3)

{

Sleep(50);

i++;

printf(".");

}

cout << "\n\nЧисло успешно записано\n\n";

return 0;

}

int mainThread4(int argc, \_TCHAR\* argv[])

{

DWORD myThreadID;

DWORD myThreadID2;

HANDLE myHandleReader = CreateThread(0, 0, myThreadReader, 0, CREATE\_SUSPENDED, &myThreadID2);

HANDLE myHandleWriter = CreateThread(0, 0, myThreadWriter, 0, 0, &myThreadID);

Sleep(300);

ResumeThread(myHandleReader);

Sleep(300);

TerminateThread(myHandleReader, 0);

TerminateThread(myHandleWriter, 0);

system("pause");

return 0;

}

