**Отчет по лабораторной работе № 3**

Программирование для многопоточных приложений.

OpenMP

Студенты:

Потапова Алина,

Запорожская Елисавета

Группа:

3530202/90002

**Код программы**

#include <omp.h>

#include <iostream>

#include <windows.h>

#include <stdio.h>

#include <conio.h>

**using** **namespace** std;

**void** func()

{

**for** (**int** i = 0; i < 500000; i++)

{

rand();

}

}

//MASTER

**int** main()

{

omp\_set\_num\_threads(2);

#pragma omp parallel for

**for** (**int** i = 0; i < 100; i++)

{

cout << i;

func();

}

#pragma omp parallel

{

#pragma omp sections nowait

{

#pragma omp parallel for ordered

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

{

#pragma omp ordered

{

cout << i;

func();

}

}

#pragma omp section

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

{

cout << '/';

func();

}

}

#pragma omp master

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

{

cout << 3;

func();

}

}

**return** 0;

}

//SINGLE

**int** main()

{

omp\_set\_num\_threads(2);

#pragma omp parallel for

**for** (**int** i = 0; i < 100; i++)

{

cout << 0;

func();

}

#pragma omp parallel

{

#pragma omp sections nowait

{

#pragma omp single nowait

{

printf("\nPrinting: \n");

}

#pragma omp section

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

{

cout << i;

func();

}

#pragma omp section

**for** (**int** i = 0; i < 20; i++)

{

cout << 2;

func();

}

}

#pragma omp barrier

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

{

cout << 3;

func();

}

}

**return** 0;

}

//CRITICAL

**int** main()

{

omp\_set\_num\_threads(2);

#pragma omp parallel for

**for** (**int** i = 0; i < 100; i++)

{

cout << i;

func();

}

#pragma omp parallel

{

#pragma omp sections nowait

{

#pragma omp critical

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

{

cout << 1;

func();

}

#pragma omp critical

**for** (**int** i = 0; i < 20; i++)

{

cout << 2;

func();

}

}

#pragma omp barrier

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

{

cout << 3;

func();

}

}

**return** 0;

}

//ПРИМЕР

**int** main()

{

omp\_set\_num\_threads(2);

#pragma omp parallel for

**for** (**int** i = 0; i < 100; i++)

{

cout << i;

func();

}

#pragma omp parallel

{

#pragma omp sections nowait

{

#pragma omp section

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

{

cout << 1;

func();

}

#pragma omp section

**for** (**int** i = 0; i < 20; i++)

{

cout << 2;

func();

}

}

#pragma omp barrier

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

{

cout << 3;

func();

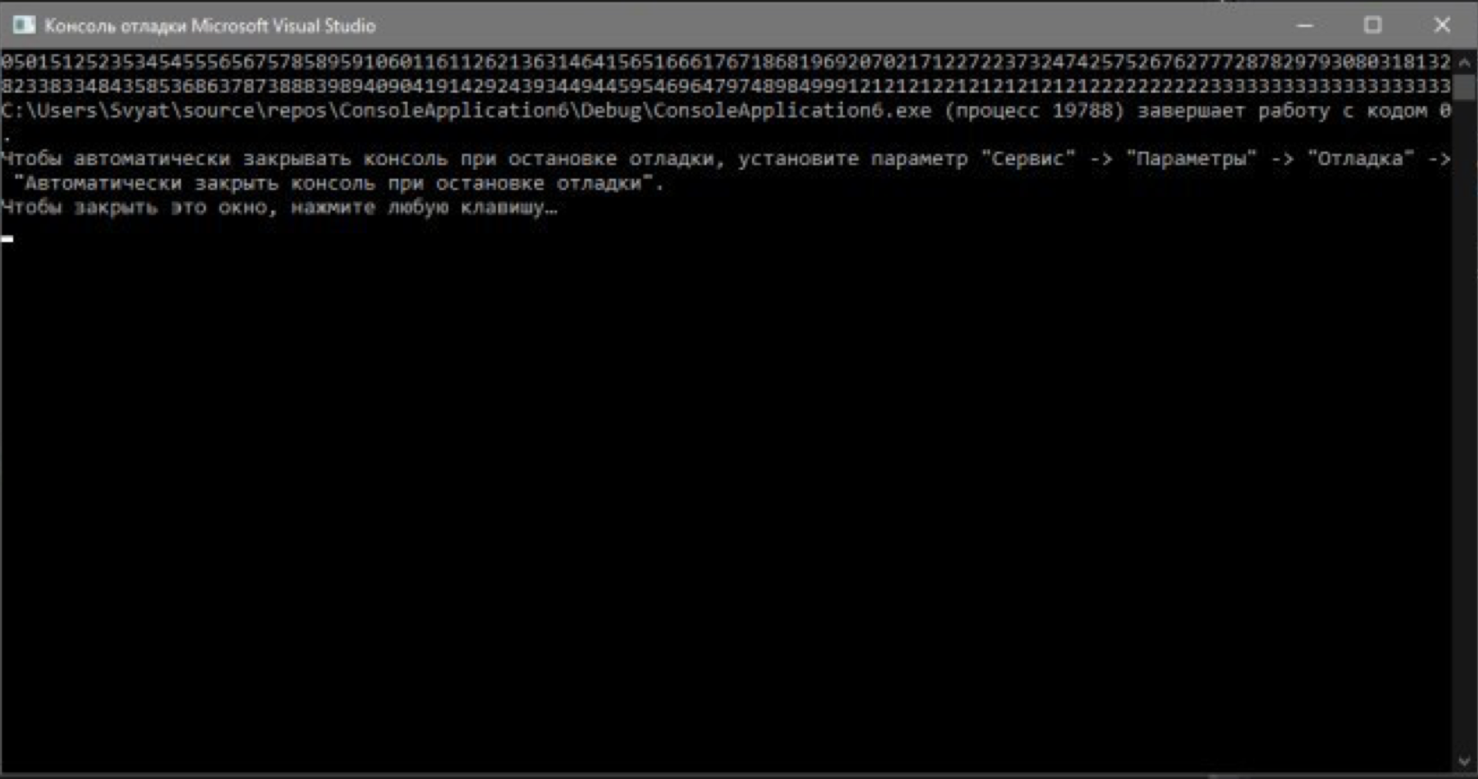
}

}

**return** 0;

}

**Пример выполнения программы**

****