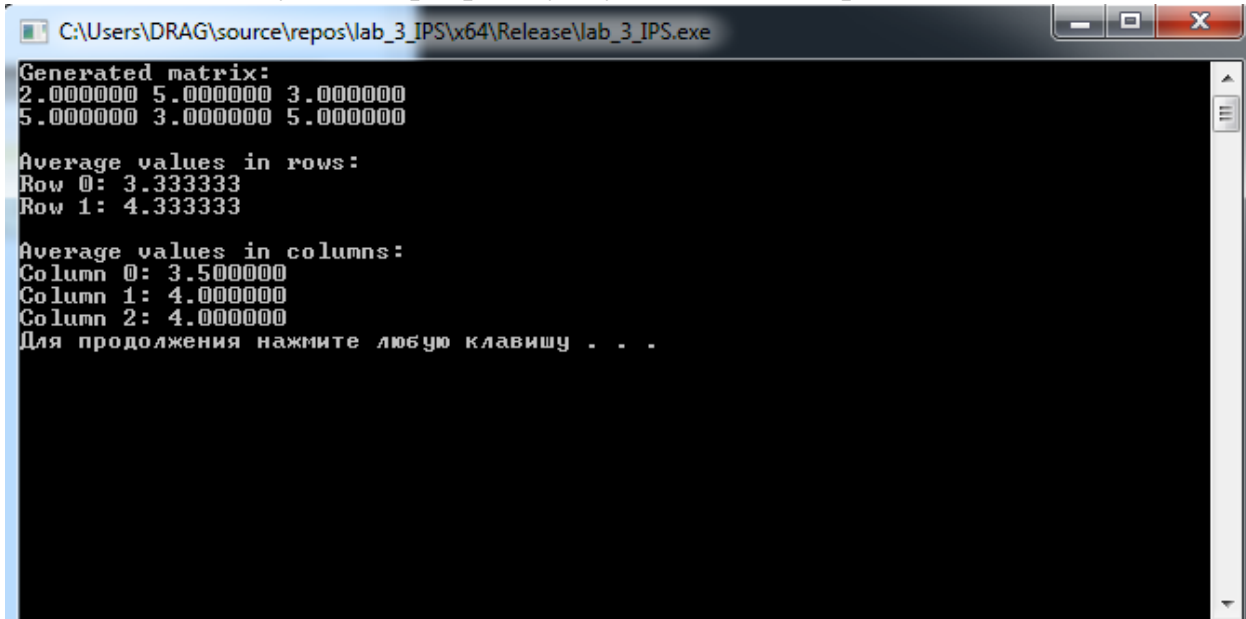


## Лабораторная работа №3

Выполнил: Прохоров А.В.

Группа: ПМ-21М

1. Разберите программу представленную в файле task\_for\_lecture5.cpp. В программе создается 2 потока, каждый из которых вычисляет средние значения матрицы, один по строкам исходной матрицы matrix, а другой по столбцам. Запустите программу и убедитесь в ее работоспособности



```
C:\Users\DRAG\source\repos\lab_3_IPS\x64\Release\lab_3_IPS.exe
Generated matrix:
2.000000 5.000000 3.000000
5.000000 3.000000 5.000000

Average values in rows:
Row 0: 3.333333
Row 1: 4.333333

Average values in columns:
Column 0: 3.500000
Column 1: 4.000000
Column 2: 4.000000
Для продолжения нажмите любую клавишу . . .
```

Программа выдает верные результаты.

2. Проанализируйте программу и введите в нее изменения, которые по Вашему мнению повысят её производительность.

```

void FindAverageValues(eprocess_type proc_type, double** matrix, const size_t numb_rows, const size_t numb_cols, double* average_vals)
{
    switch (proc_type)
    {
        case eprocess_type::by_rows:
        {
            cilk_for(size_t i = 0; i < numb_rows; ++i)
            {
                //double sum( 0.0 );
                cilk::reducer_opadd<double> sum(0.0);
                cilk_for(size_t j = 0; j < numb_cols; ++j)
                {
                    sum += matrix[i][j];
                }
                average_vals[i] = sum.get_value() / numb_cols;
            }
            break;
        }
        case eprocess_type::by_cols:
        {
            cilk_for(size_t j = 0; j < numb_cols; ++j)
            {
                //double sum( 0.0 );
                cilk::reducer_opadd<double> sum(0.0);
                cilk_for(size_t i = 0; i < numb_rows; ++i)
                {
                    sum += matrix[i][j];
                }
                average_vals[j] = sum.get_value() / numb_rows;
            }
            break;
        }
        default:
            return;
    }
}

```

- Определите с помощью Intel Parallel Inspector наличие в программе таких ошибок как: взаимная блокировка, гонка данных, утечка памяти. Сделайте скрины результатов анализа Parallel Inspector (вкладки Summary, Bottom-up) для всех упомянутых ошибок, где отображаются обнаруженные ошибки, либо отражается их отсутствие.

Detect Memory Problems

Target

Analysis Type

Collection Log

Summary

Problems

ID	Type	Sources	Modules	Object Size	State
#P1	Memory leak	main.cpp	lab_3_ips.exe	16	Not fixed
#P2	Memory leak	main.cpp	lab_3_ips.exe	48	Not fixed
#P3	Memory leak	main.cpp	lab_3_ips.exe	16	Not fixed
#P4	Memory leak	main.cpp	lab_3_ips.exe	24	Not fixed

1 of 1

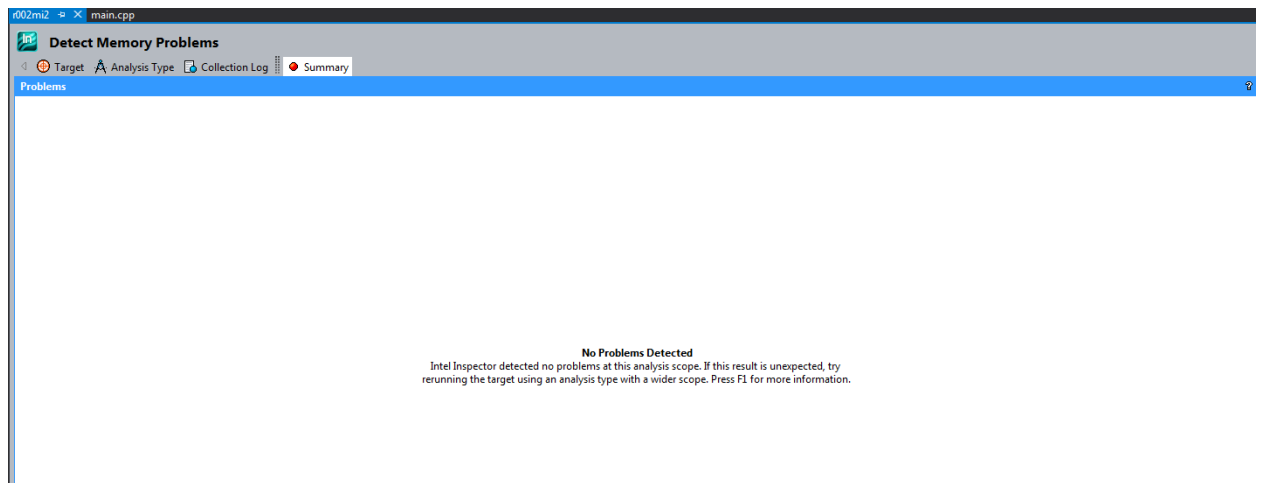
All

Code Locations: Memory leak

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	main.cpp:140	main	lab_3_ips.exe	16	block allocated at main.cpp:140	
138	const size_t numb_cols = 3;					lab_3_ips.exe!main - main.cpp:140
139						
140	double** matrix = new double*[numb_rows];					lab_3_ips.exe!_scrt_common_main_seh - exe_common.inl:288
141	for (size_t i = 0; i < numb_rows; ++i)					kernel32.dll!BaseThreadInitThunk
142	{					ntdll.dll!RtlUserThreadStart

Видим утечку из-за не освобождения выделенной памяти

- Измените код программы таким образом, чтобы Inspector при проверке не находил в программе ошибок, перечисленных в п.3. Сделайте скрины результатов запуска Parallel Inspector.



Как видно, ошибка была устранена.