


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.

```
Generated matrix:
1.000000 1.000000 4.000000
3.000000 3.000000 4.000000

Average values in rows:
Row 0: 2.000000
Row 1: 3.333333

Average values in columns:
Column 0: 2.000000
Column 1: 2.000000
Column 2: 4.000000
```

Программа работает правильно

2. Проанализируйте программу и введите в нее изменения, которые по Вашему мнению повысят ее 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 / numb_cols;
            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 / numb_rows;
            average_vals[j] = sum.get_value() / numb_rows;
        }
        break;
    }
    default:
    {
        throw("Incorrect value for parameter 'proc_type' in function FindAverageValues() call!");
    }
    }
}
```

3. 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. Запускайте анализы на разных уровнях (*Narrowest*, *Medium*, *Widest*).

Detect Leaks

Target Analysis Type Collection Log Summary

ID	Type	Sources	Modules	Object Size	State
P1	Missing allocation	[Unknown]	ucrtbase.dll		New
P2	Missing allocation	[Unknown]	ucrtbase.dll		New
P3	Mismatched allocation/deallocation	[Unknown]	ucrtbase.dll; vcruntime140.dll		New
P4	Mismatched allocation/deallocation	xthread	IPS.exe		New
P5	Mismatched allocation/deallocation	xthread	IPS.exe		New
P6	Memory leak	task_for_lecture5.cpp	IPS.exe	8	New
P7	Memory leak	task_for_lecture5.cpp	IPS.exe	48	New
P8	Memory leak	task_for_lecture5.cpp	IPS.exe	16	New
P9	Memory leak	task_for_lecture5.cpp	IPS.exe	24	New
P10	Memory not deallocated	exe_common.inl; stdio.h	IPS.exe	8250	New

Filters Severity Sort

Error 9 item(s)
Warning 1 item(s)

Type
Memory leak 4 item(s)
Memory not deallocated 1 item(s)
Mismatched allocation/deallocation 3 item(s)
Missing allocation 2 item(s)

Source
[Unknown] 3 item(s)
exe_common.inl 1 item(s)
stdio.h 1 item(s)
task_for_lecture5.cpp 4 item(s)

Code Locations: Memory leak

Description Source Function Module Object Size Variable

Allocation site task_for_lecture5.cpp:150 main IPS.exe 24 block allocated at task_for_lecture5.cpp:150

```
148 double* average_vals_in_rows = new double[numb_rows];
149
150 double* average_vals_in_cols = new double[numb_cols];
151
152 InitMatrix( matrix, numb_rows, numb_cols );
```

Timeline
mainCRTStartup (17720)

Detect Memory Problems

Target Analysis Type Collection Log Summary

ID	Type	Sources	Modules	Object Size	State
P1	Memory leak	task_for_lecture5.cpp	IPS.exe	8	New
P2	Memory leak	task_for_lecture5.cpp	IPS.exe	48	New
P3	Memory leak	task_for_lecture5.cpp	IPS.exe	16	New
P4	Memory leak	task_for_lecture5.cpp	IPS.exe	24	New
P5	Memory leak	xthread	IPS.exe	32	New
P6	Memory not deallocated	exe_common.inl; stdio.h; task_for_lecture5.cpp	IPS.exe	8250	New

Code Locations: Memory leak

Description Source Function Module Object Size Offset Variable

Allocation site task_for_lecture5.cpp:143 main IPS.exe 8 block allocated at task_for_lecture5.cpp:143

```
141 const size_t numb_cols = 3;
142
143 double** matrix = new double*[numb_rows];
144 for ( size_t i = 0; i < numb_rows; ++i )
145 {
```

IPS.exe!main - task_for_lecture5.cpp:143
IPS.exe!_scrt_common_main_seh - exe_common
kernel32.dll!BaseThreadInitThunk
ntdll.dll!RtlGetAppContainerNamedObjectPath
ntdll.dll!RtlGetAppContainerNamedObjectPath

Locate Memory Problems


Target
Analysis Type
Collection Log
Summary

Problems

ID	Type	Sources	Modules	Object Size	State
P1	Memory leak	task_for_lecture5.cpp	IPS.exe	8	New
P2	Memory leak	task_for_lecture5.cpp	IPS.exe	48	New
P3	Memory leak	task_for_lecture5.cpp	IPS.exe	16	New
P4	Memory leak	task_for_lecture5.cpp	IPS.exe	24	New
P5	Memory not deallocated	exe_common.inl; stdio.h; task_for_lecture5.cpp	IPS.exe	8250	New

1 of 1
All
Code Locations: Memory leak

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	task_for_lecture5.cpp:150	main	IPS.exe	24		block allocated at task_for_lecture5.cpp:150
148						IPS.exe!main - task_for_lecture5.cpp:150
149	double* average_vals_in_rows = new double[numb_rows]					IPS.exe!_sclr_common_main_seh - exe_common
150	double* average_vals_in_cols = new double[numb_cols]					kernel32.dll!BaseThreadInitThunk
151						ntdll.dll!RtlGetAppContainerNamedObjectPath
152	InitMatrix(matrix, numb_rows, numb_cols);					ntdll.dll!RtlGetAppContainerNamedObjectPath

 **Detect Deadlocks**

◀ Target

Analysis Type


Collection Log

Summary

Problems

No Problems Detected

Intel Inspector detected no problems at this analysis scope. If this result is unexpected, try rerunning the target using an analysis type with a wider scope. Press F1 for more information.

 **Detect Deadlocks and Data Races**

◀ Target

Analysis Type

Collection Log

Summary

Problems

No Problems Detected

Intel Inspector detected no problems at this analysis scope. If this result is unexpected, try rerunning the target using an analysis type with a wider scope. Press F1 for more information.

 **Locate Deadlocks and Data Races**

◀ Target

Analysis Type

Collection Log

Summary

Problems

No Problems Detected

Intel Inspector detected no problems at this analysis scope. If this result is unexpected, try rerunning the target using an analysis type with a wider scope. Press F1 for more information.

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

```
for (size_t i = 0; i < numb_rows; i++)
{
    delete[] matrix[i];
}
delete[] matrix;
delete[] average_vals_in_rows;
delete[] average_vals_in_cols;
```

Код добавлен в конец блока try функции main().

The screenshot shows the 'Detect Leaks' application window. The top bar includes tabs for 'Target', 'Analysis Type', 'Collection Log', and 'Summary'. Below this is a 'Problems' table with columns for ID, Type, Sources, Modules, Object Size, and State. The table lists six problems (P1-P6), with P6 being a 'Memory not deallocated' issue of size 4154, which is 'Not fixed'.

Below the table, the 'Code Locations: Mismatched allocation/deallocation' section is expanded, showing details for problem P4. It includes a 'Description' table with columns for Source, Function, Module, Object Size, and Variable. The 'Mismatched deallocation site' is shown at xthread:127, thread IPS.exe, block. The code snippet shows a function that destroys an object, with the deallocation site highlighted in yellow. The 'Allocation site' is shown at xthread:92, thread IPS.exe, block, with the code snippet showing the allocation of the object, also highlighted in yellow.

Detect Memory Problems

Target
Analysis Type
Collection Log
Summary

Problems

ID	Type	Sources	Modules	Object Size	State
P1	Memory not deallocated	exe_common.inl; stdio.h	IPS.exe	4154	Not fixed

1 of 2 All

Code Locations: Memory not deallocated

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	exe_common.inl:167	pre_c_initialization	IPS.exe	58		block
Source file not found. Suggestion: Specify locations in a Project Properties dialog box search tab.				ucrtbase.dll!configure_narrow_argv IPS.exe!pre_c_initialization - exe_common ucrtbase.dll!initterm_e IPS.exe!_scrt_common_main_seh - exe_commo kernel32.dll!BaseThreadInitThunk		

Locate Memory Problems

Target
Analysis Type
Collection Log
Summary

Problems

ID	Type	Sources	Modules	Object Size	State
P1	Memory not deallocated	exe_common.inl; stdio.h	IPS.exe	4154	Not fixed

1 of 2 All

Code Locations: Memory not deallocated

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	exe_common.inl:167	pre_c_initialization	IPS.exe	58		block
Source file not found. Suggestion: Specify locations in a Project Properties dialog box search tab.				ucrtbase.dll!configure_narrow_argv IPS.exe!pre_c_initialization - exe_common ucrtbase.dll!initterm_e IPS.exe!_scrt_common_main_seh - exe_commo kernel32.dll!BaseThreadInitThunk		

Как можно видеть, ошибки, связанные с утечкой памяти в коде файла task_for_lecture5.cpp, исправлены