



**Нижегородский государственный университет
им. Н.И.Лобачевского**

Факультет Вычислительной математики и кибернетики

Образовательный комплекс

Технологии разработки параллельных программ

Intel Thread Profiler

Краткое описание

Часть 2: Пример использования

Корняков К.В., Шишков А.В.
Кафедра математического
обеспечения ЭВМ

Содержание

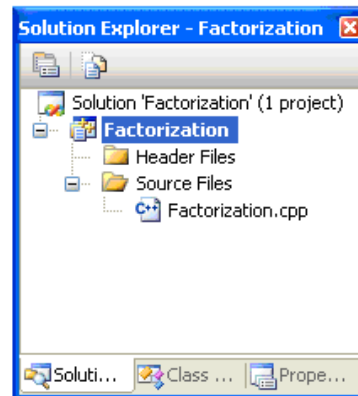
- ❑ Изучение профилируемого приложения
- ❑ Подготовка приложения для анализа
- ❑ Профилирование
- ❑ Анализ производительности приложения



Изучение профилируемого приложения

Задание 1

- ❑ Запустите приложение **Microsoft Visual Studio 2005**,
- ❑ В меню **File** выполните команду **Open→Project/Solution...**,
- ❑ В диалоговом окне **Open Project** выберите папку **C:\ITPLab\Factorization**,
- ❑ Дважды щелкните на файле **Factorization.sln** или, выбрав файл, выполните команду **Open**.
- ❑ После открытия проекта в окне **Solution Explorer** дважды щелкните на файле исходного кода **Factorization.cpp**.



Изучение профилируемого приложения

- Приложение **Factorization** ищет разложение множества чисел на простые множители.
- Алгоритм разложения основан на попытке деления нацело каждого числа на все числа, меньшие его.

$$12 = ? * ? * \dots * ?$$

$12 / 2 = 6$; // пробуем разделить на 2 еще раз

$6 / 2 = 3$; // пробуем разделить на 2 еще раз

$3 / 2 = 1,5$; // берем следующий делитель

$3 / 3 = 1$; // СТОП – получили единицу



Изучение профилируемого приложения

- В приложении объявлены следующие константы:

```
#define NUM_NUMBERS 100000  
#define NUM_THREADS 2
```

- Результаты разложения сохраняются в массив векторов

```
vector<int> divisors[NUM_NUMBERS+1];
```

Так, например,

```
divisors[5] = {5}
```

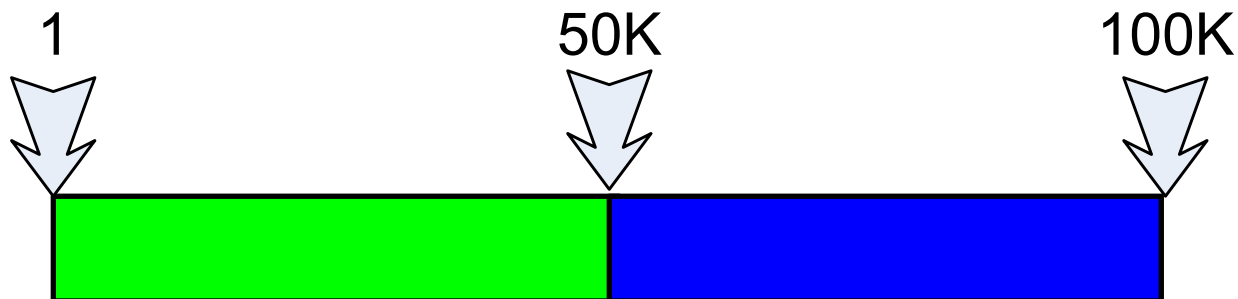
```
divisors[6] = {2, 3}
```

- Для разложения используется несколько потоков.



Изучение профилируемого приложения

- ❑ В данной работе нас будут интересовать только функции `main` и `factorization1`.
- ❑ Функция `main` содержит необходимые объявления, операции создания потоков и ожидания их завершения, а также вывод на экран, предназначенный для контроля правильности результатов.
- ❑ Функция `factorization1` представляет собой рабочую функцию потока. В ней реализуется простейшая стратегия распределения нагрузки между потоками.



Подготовка приложения для анализа

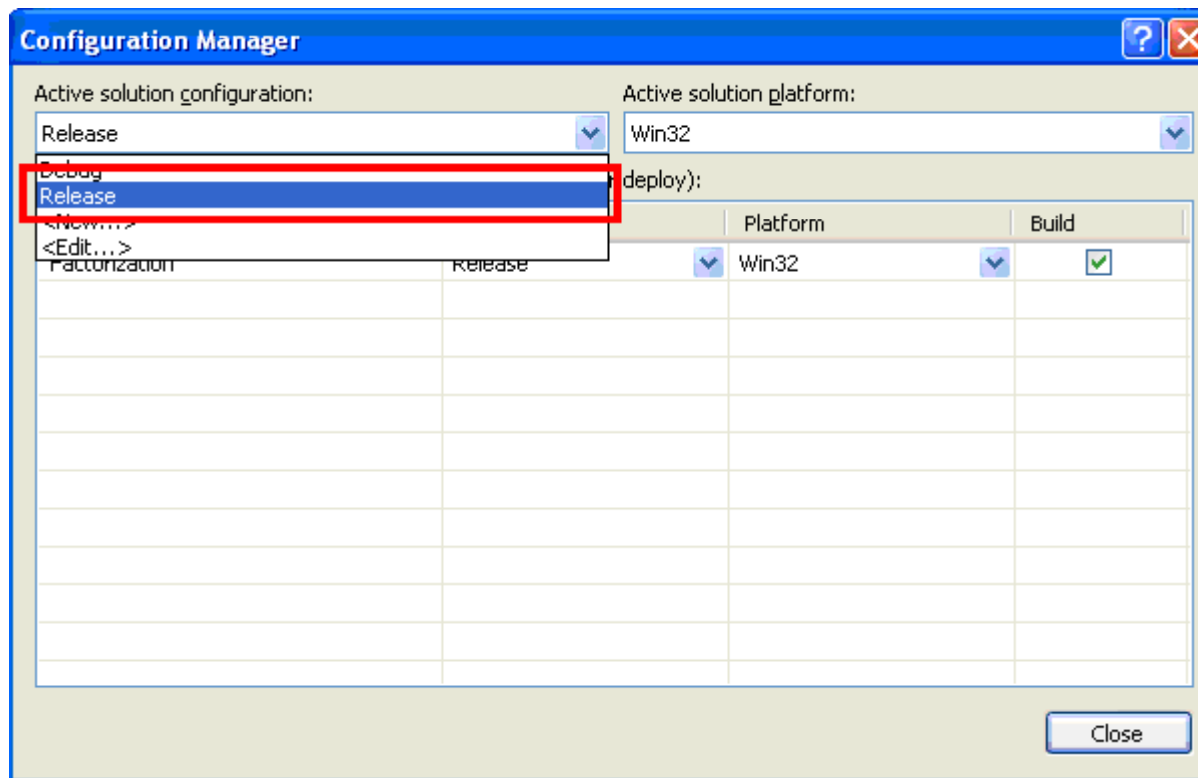
- ❑ Подготовка заключается в установке определенных опций компиляции и компоновки приложения. Большинство настроек установлены по умолчанию.
- ❑ Необходимые настройки
 - Компиляция в **Release** режиме
 - Формат отладочной информации **Program Database (/ZI)**
 - Включение в приложение отладочной информации
 - Использование потокобезопасных библиотек
 - Использование опции компоновщика **/fixed:no**
- ❑ После установки всех этих опций приложение готово к профилированию.



Подготовка приложения для анализа

Задание 2

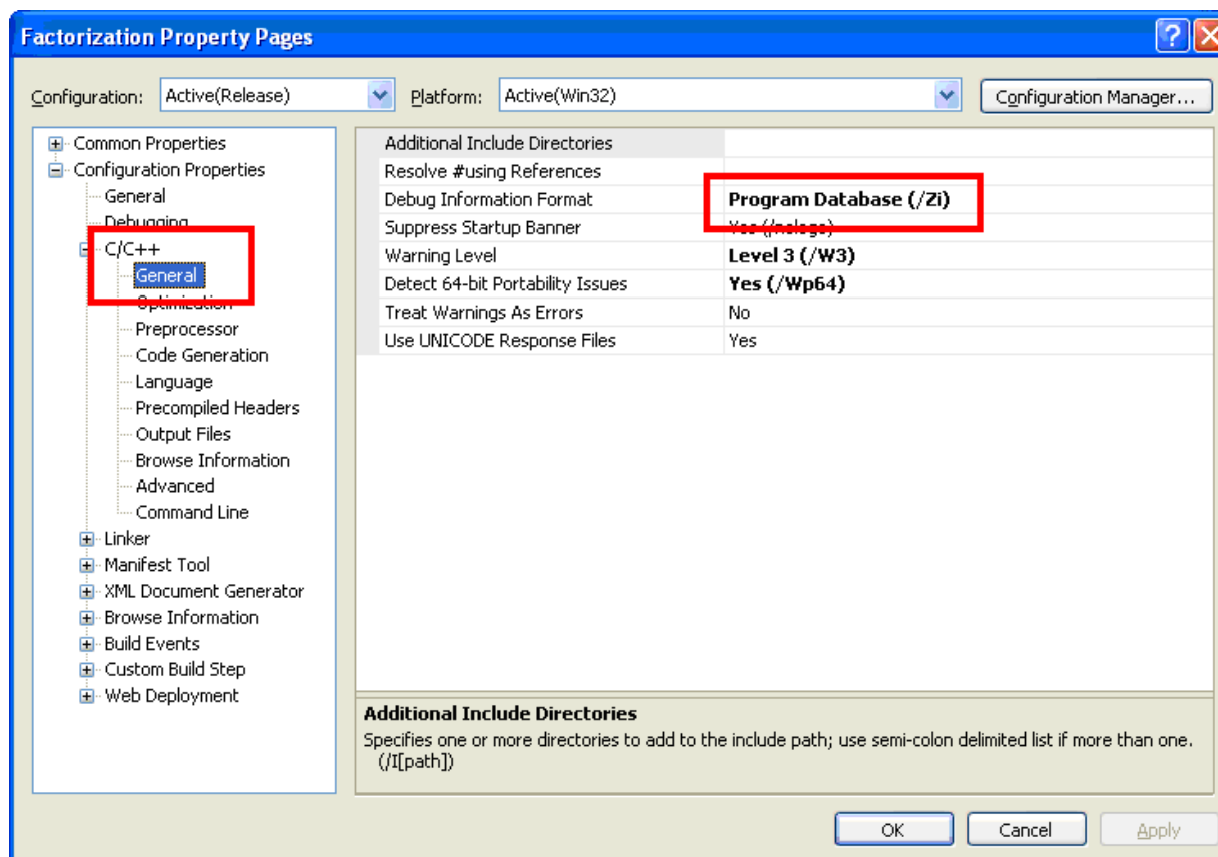
В меню **Build** выберите пункт **Configuration Manager...**, в открытом окне выберите режим **Release**.



Подготовка приложения для анализа

Задание 2

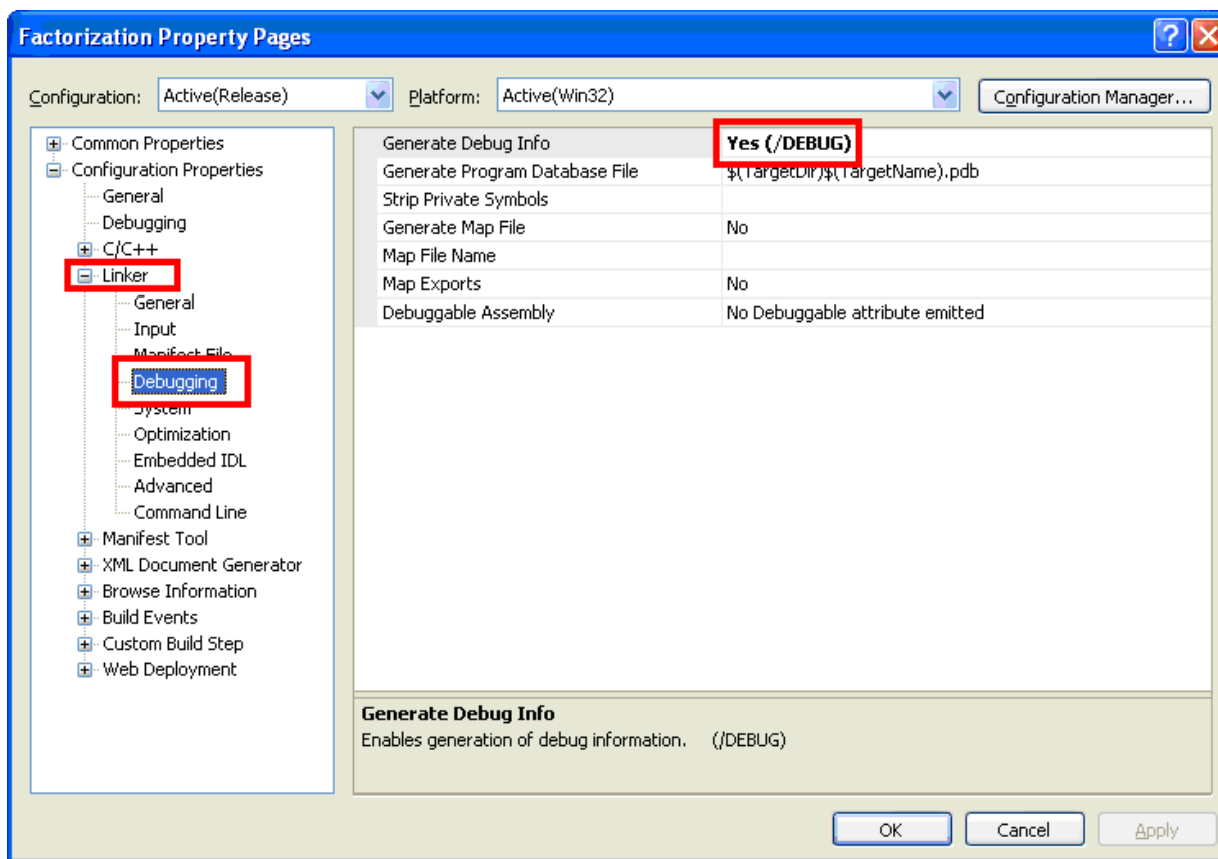
Укажите формат отладочной информации **Program Database (/Zi)**.



Подготовка приложения для анализа

Задание 2

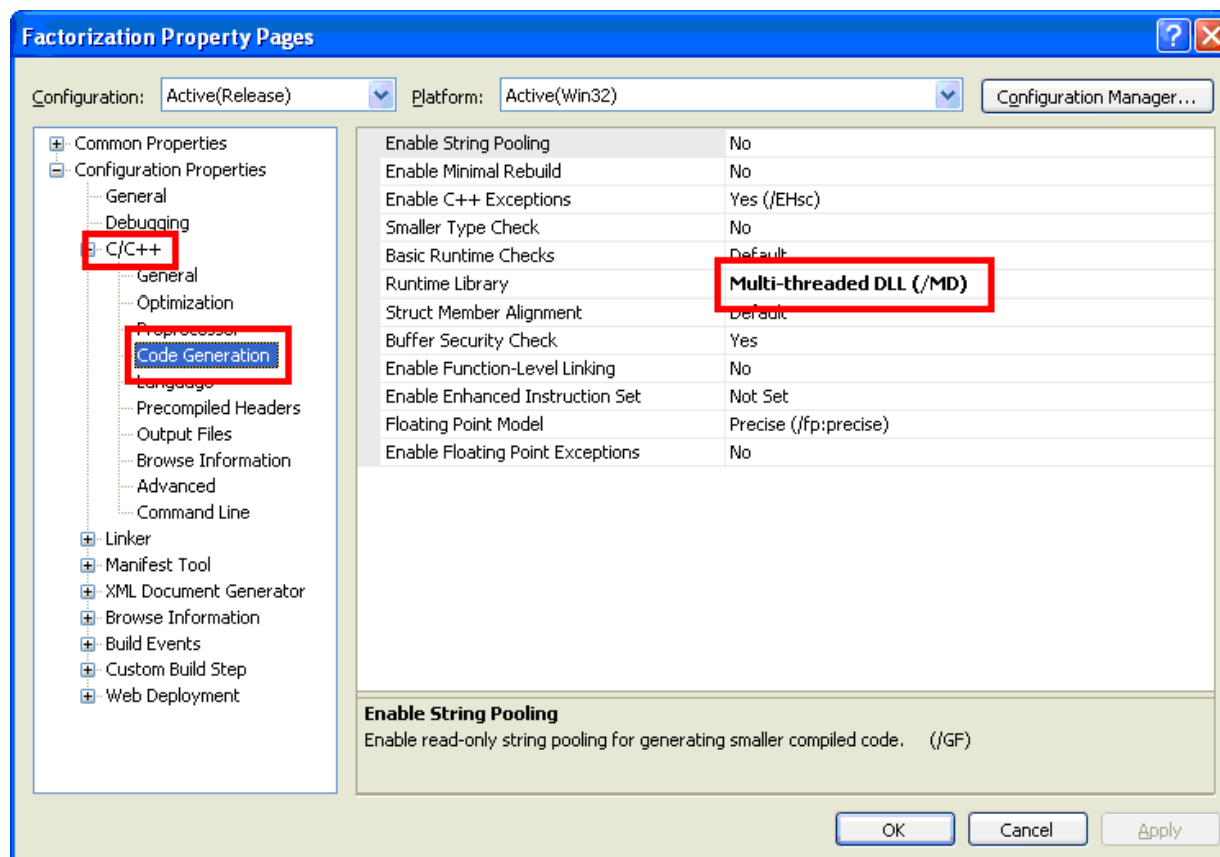
Включите генерацию отладочной информации.



Подготовка приложения для анализа

Задание 2

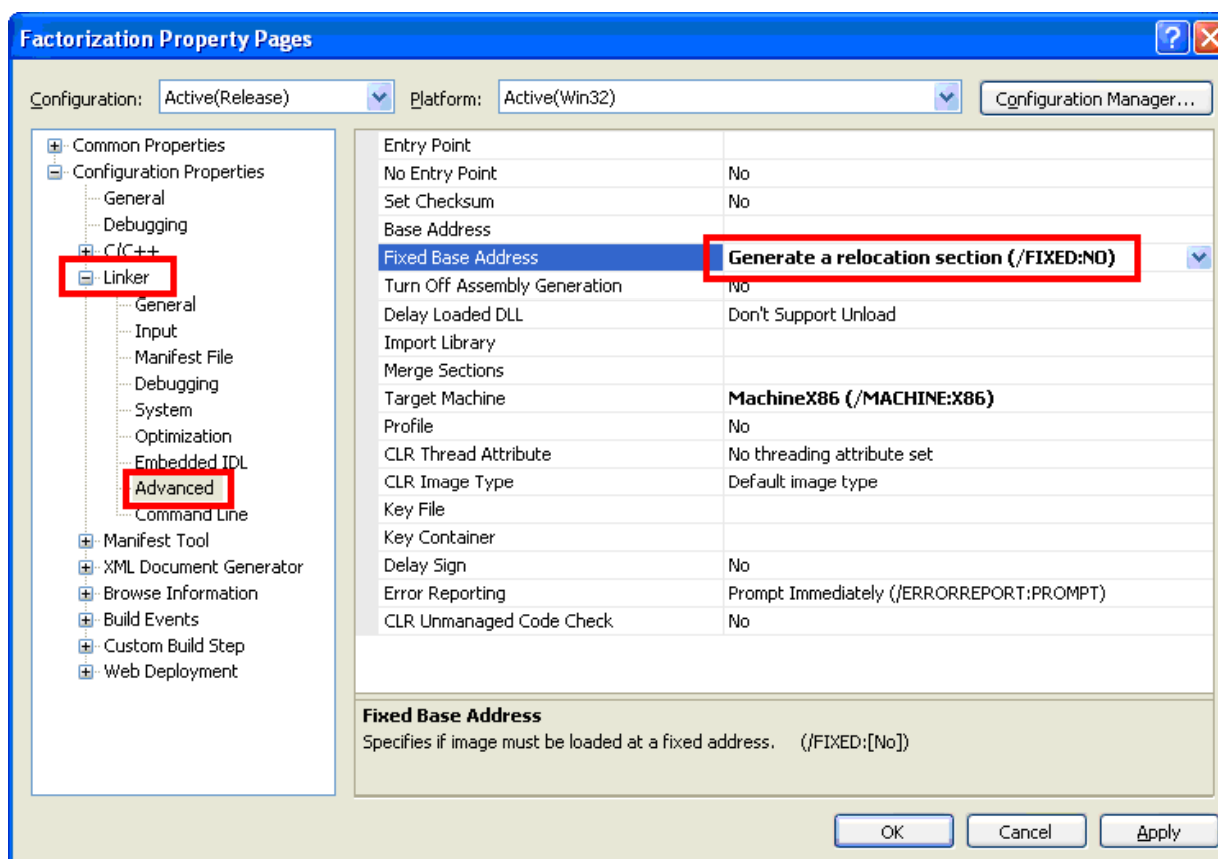
Использование потокобезопасных библиотек.



Подготовка приложения для анализа

Задание 2

Установите опцию компоновщика `/fixed:no`.





Создание проекта Intel® Thread Profiler

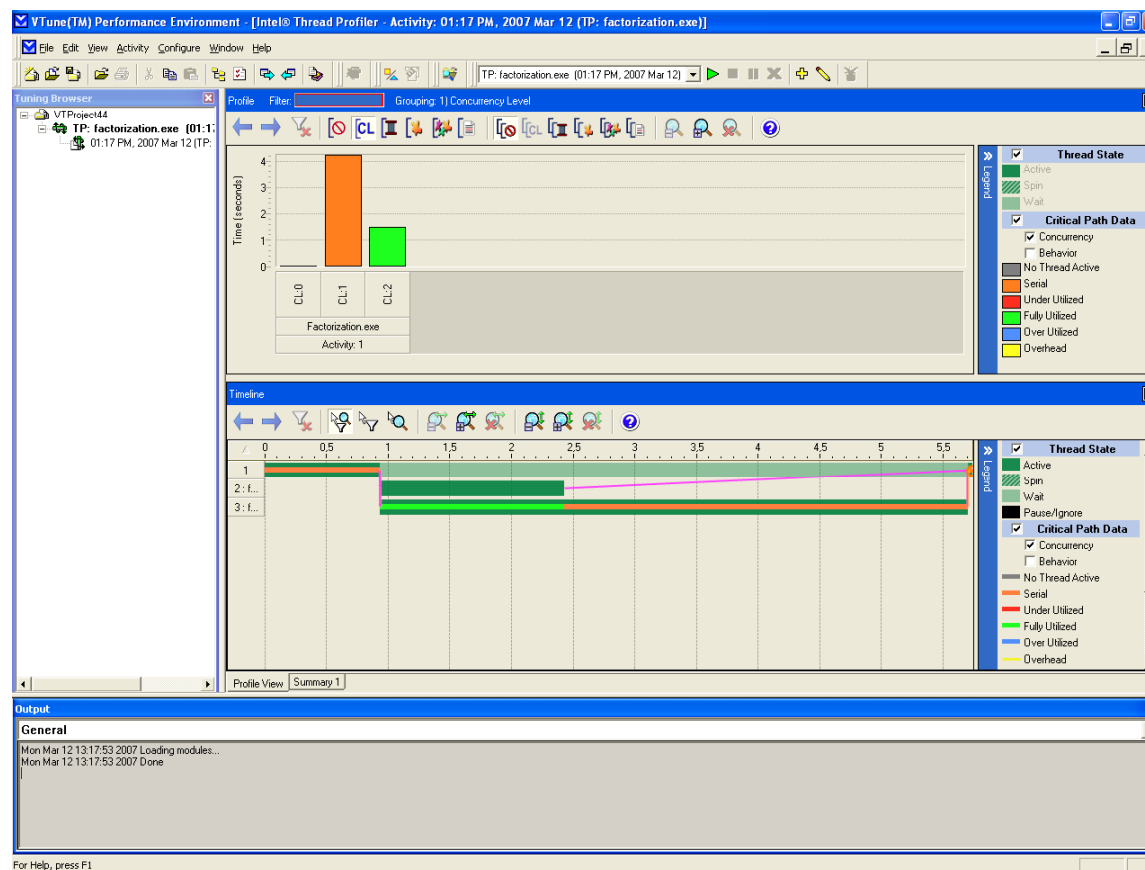
Задание 3

- ❑ Запустите Intel Thread Profiler. **Start→All programs→Intel(R) Software Development Tools→Intel(R) Thread Profiler 3.0→Intel(R) Thread Profiler.**
- ❑ В открывшемся окне нажмите на кнопку **New Project** .
- ❑ В новом окне выберите **Intel(R) Thread Profiler Wizard** и нажмите кнопку **OK**.
- ❑ Ниже надписи **Launch an application** имеется строка, в которой необходимо указать имя профилируемого приложения. Нажмите кнопку [...] и в открывшемся диалоговом окне укажите путь до приложения, подлежащего профилировке. В нашем случае это **C:\ITPLab\Factorization\release\Factorization.exe**.
- ❑ Нажмите кнопку **Finish**.



Профилирование

- ❑ ITP самостоятельно осуществляет инструментацию приложения и запускает процесс профилирования.



Профилирование

Окно Summary

The screenshot shows the Intel Thread Profiler Summary window for the application 'factorization.exe'. The window is titled 'VTune(TM) Performance Environment - [Intel® Thread Profiler - Activity: 07:56 PM, 2007 Apr 02 (TP: factorization.exe)]'. The Summary pane displays the following data:

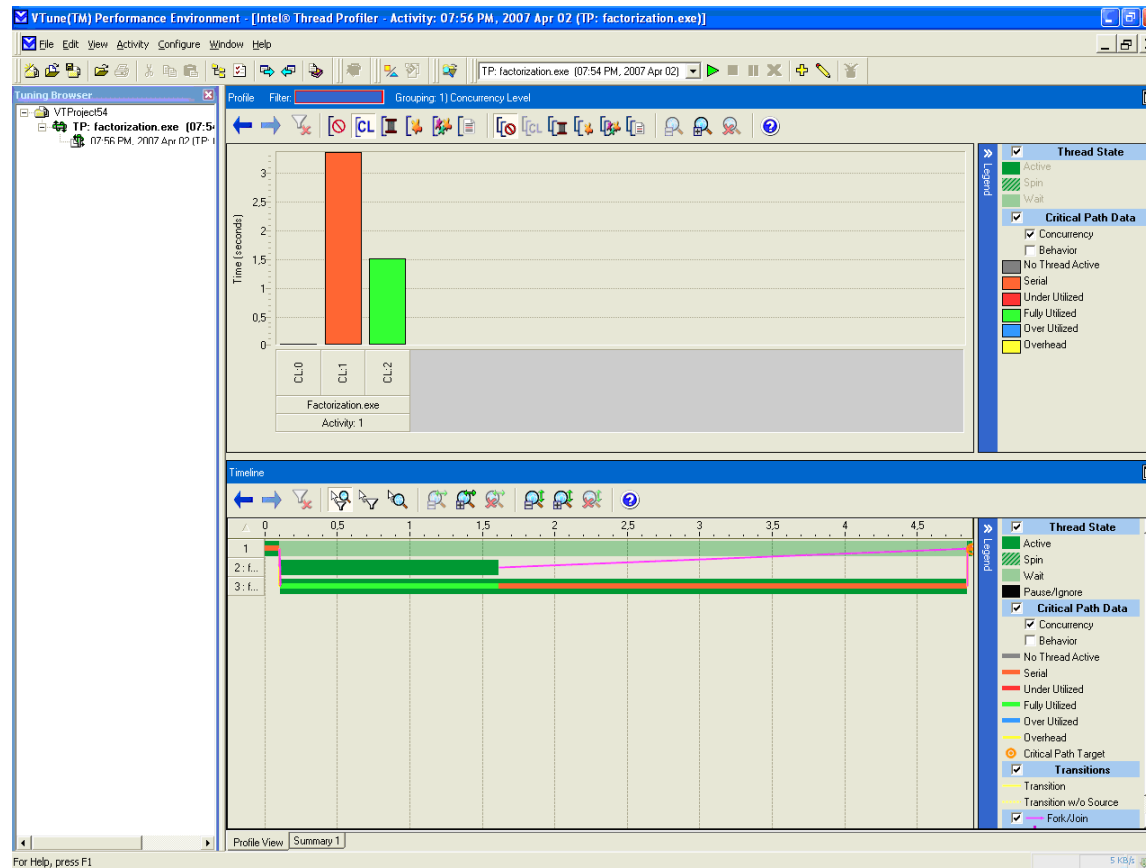
General Information	
Machine Name	WS-2K-1114-02
Executable Name	c:\VTPLab\Factorization\release\Factorization.exe
Hyper-Threading Technology Enabled	No
Intel® EM64T Enabled	No
Run Date/Time	Mon Apr 02 19:55:59 2007
Run Length (sec)	4.89
Average Concurrency	1,31
Logical Processors	2
Threads	3
Callsites	Yes
Callsite Depth	5
Transitions	5
Transitions Per Sec	1,02
Filtered Transitions	167
Filtered Transition Time	0,00
APIs	2158
APIs Per Sec	441,75

APIs	Count	Contended	Uncontended	Ignored Wait Time	Ignore All Filtered	Block Filtered	Acquire Filtered	Critical Section Filtered	Overhead Filtered
API	2158	1	2157	0,00246888	0	167	0	0	0
Critical Section	1924	0	1924	0	0	0	0	0	0
InitializeC...	16	0	16	0	0	0	0	0	0
InitializeC...	34	0	34	0	0	0	0	0	0
EnterCriti...	937	0	937	0	0	0	0	0	0
LeaveCriti...	937	0	937	0	0	0	0	0	0
File	169	0	169	0,00246888	0	167	0	0	0
WriteFile	169	0	169	0,00246888	0	167	0	0	0
I/O	6	0	6	0	0	0	0	0	0
GetSidHandle	6	0	6	0	0	0	0	0	0
Miscellaneous	50	0	50	0	0	0	0	0	0
GetProcAd...	50	0	50	0	0	0	0	0	0
Process	1	0	1	0	0	0	0	0	0
ExitProcess	1	0	1	0	0	0	0	0	0
Thread	7	0	7	0	0	0	0	0	0
ThreadEntry	3	0	3	0	0	0	0	0	0
ThreadExit	2	0	2	0	0	0	0	0	0
CreateThread	2	0	2	0	0	0	0	0	0
Wait	1	1	0	0	0	0	0	0	0
WaitForMult...	1	1	0	0	0	0	0	0	0



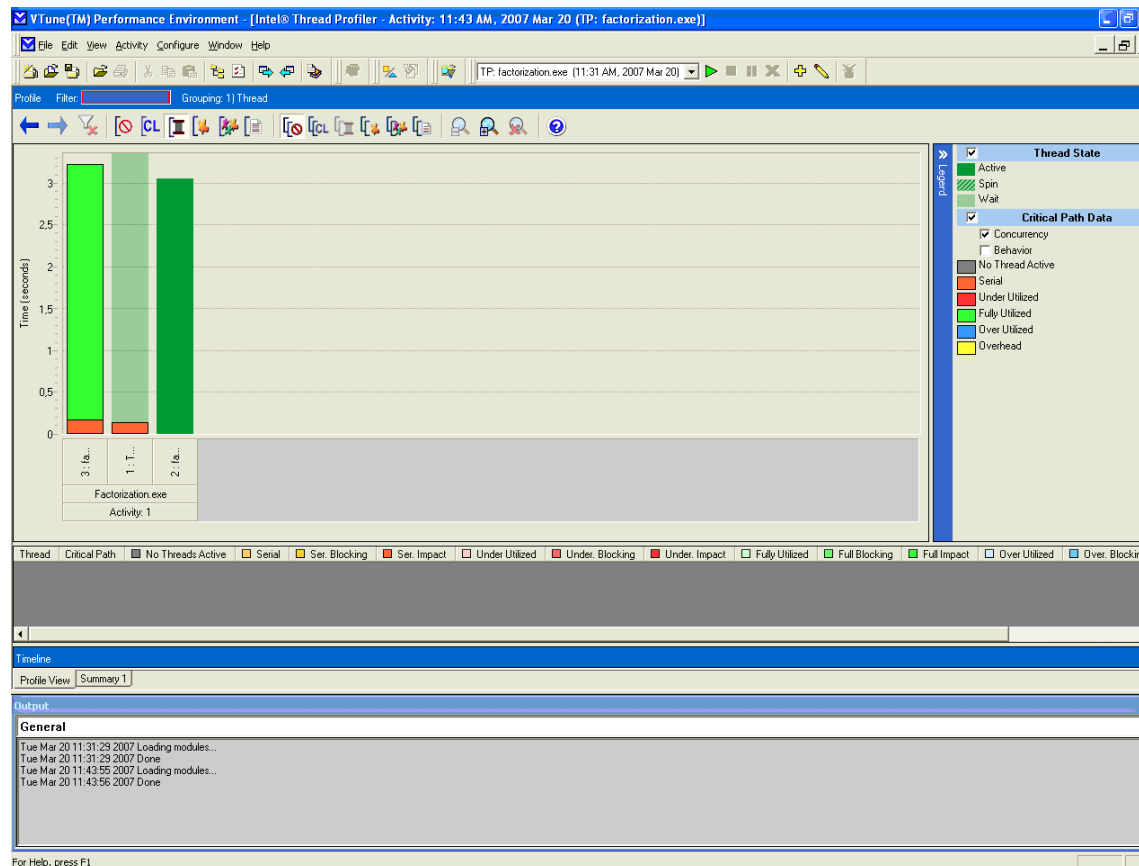
Профилирование

Окно Profile



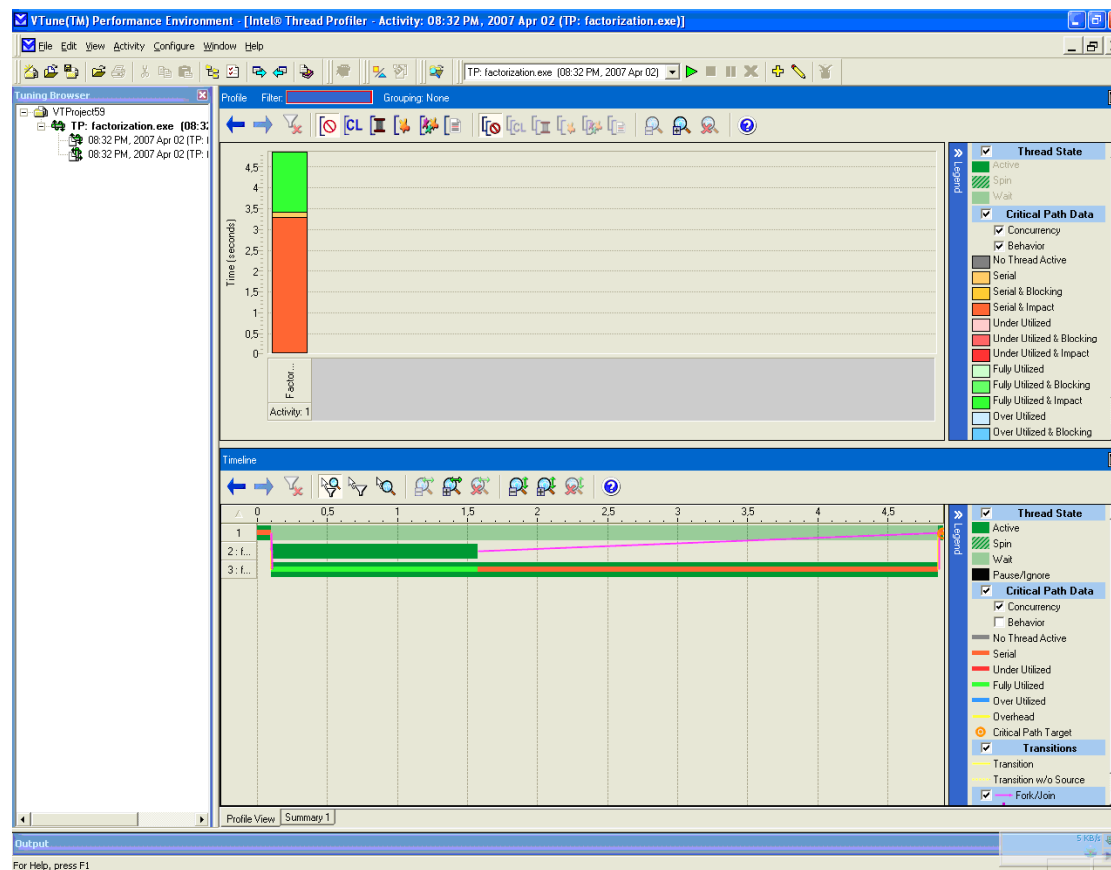
Профилирование

Окно Profile (группировка по потокам)



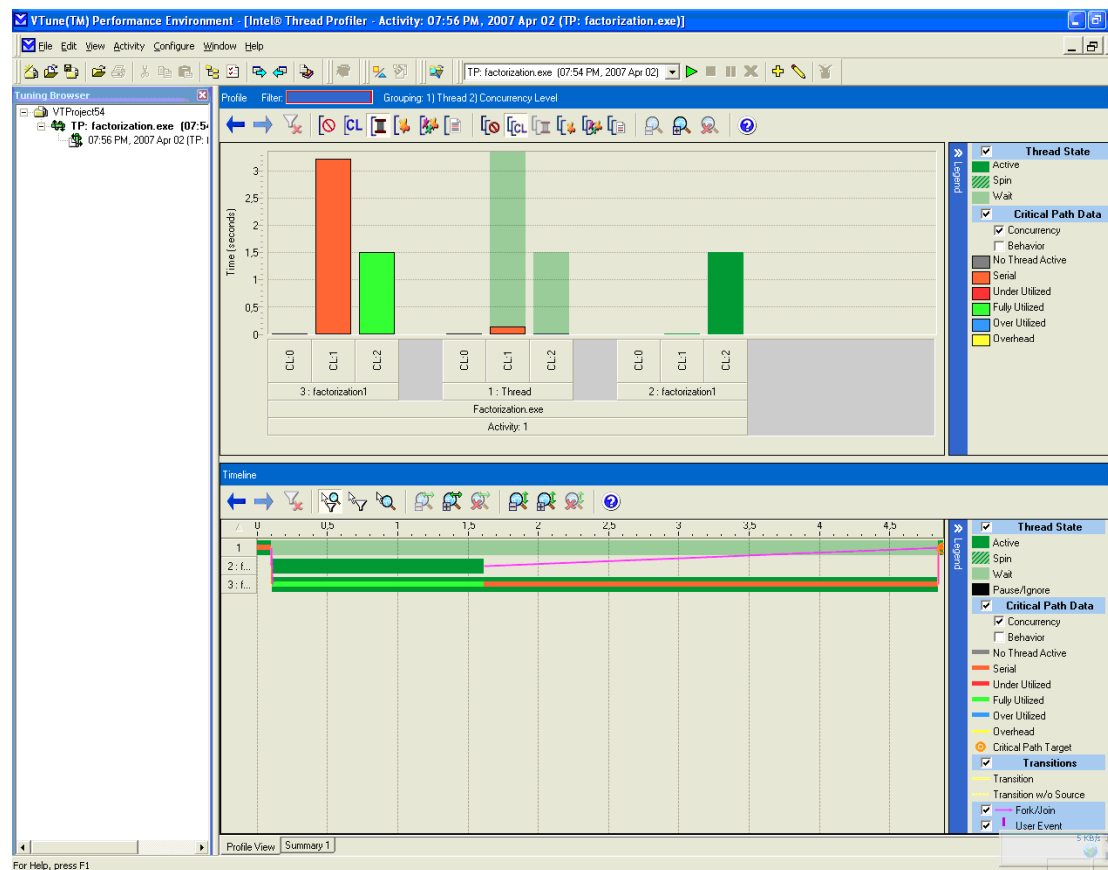
Профилирование

Окно Profile (без группировки)



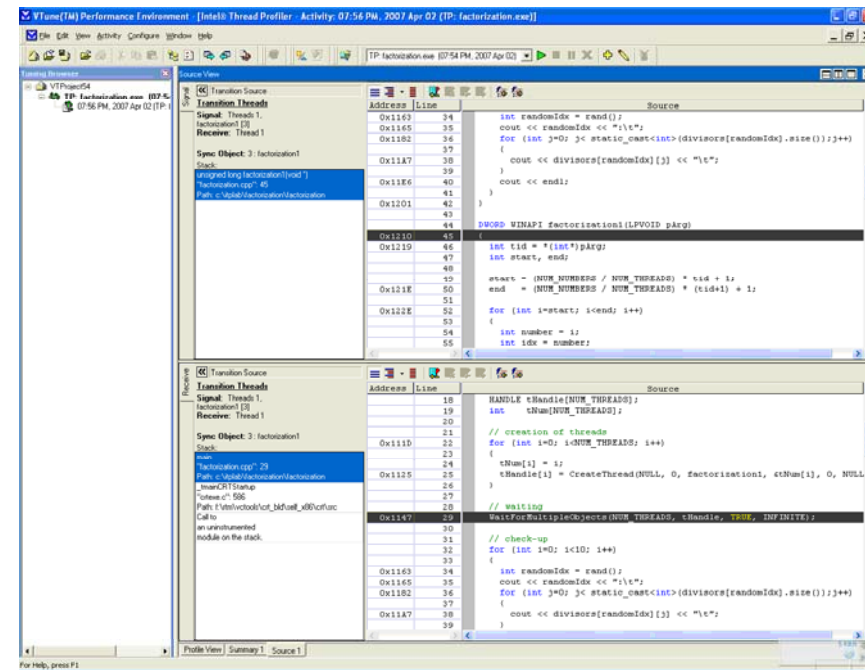
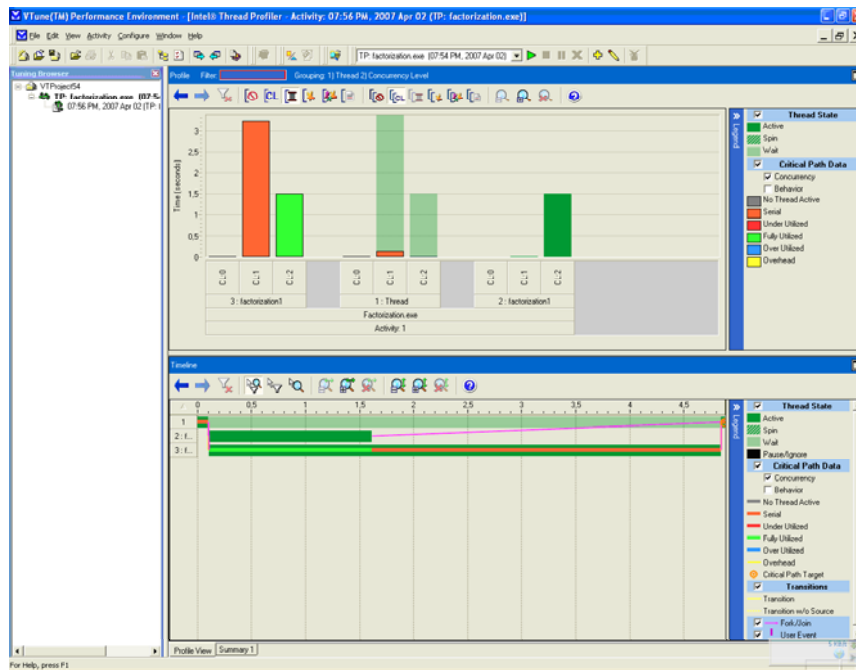
Профилирование

Окно Profile (двойная группировка)



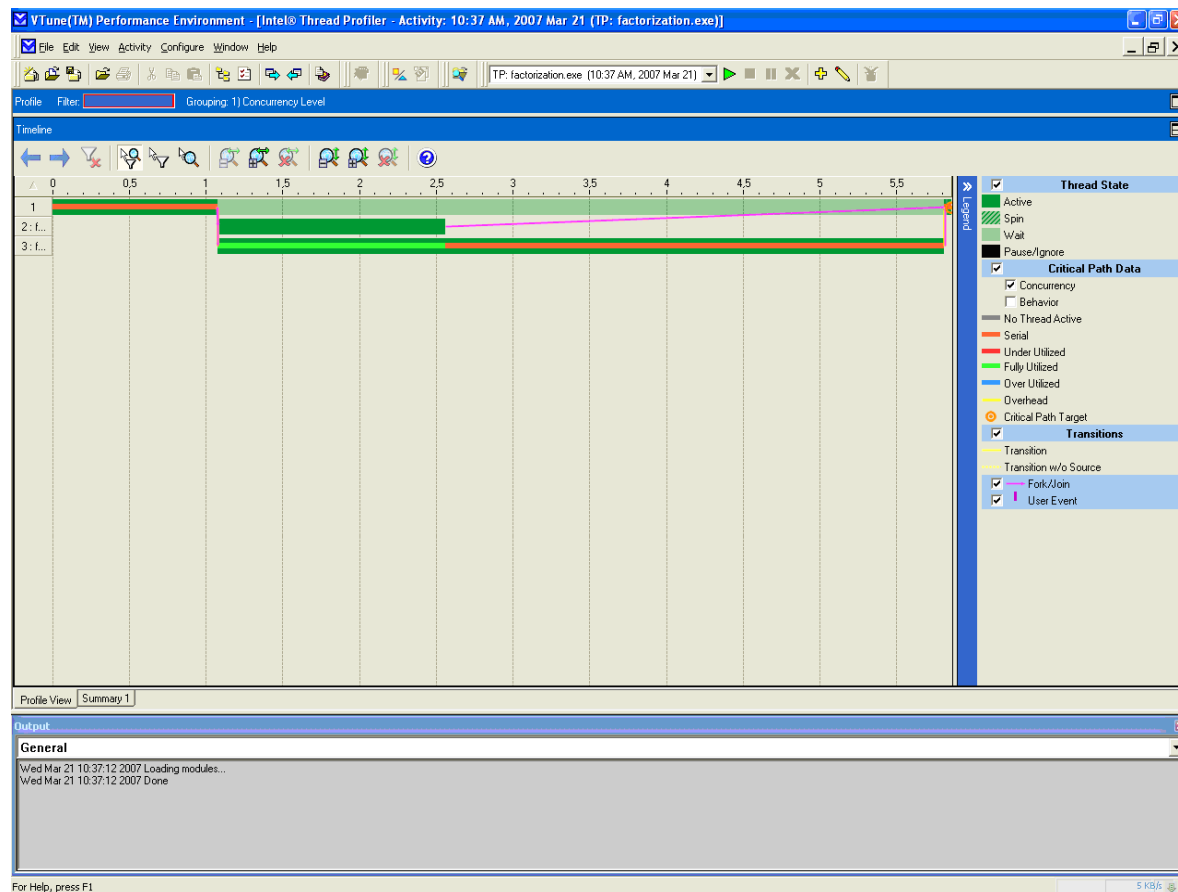
Профилрование

Окно Profile (обращение к коду)



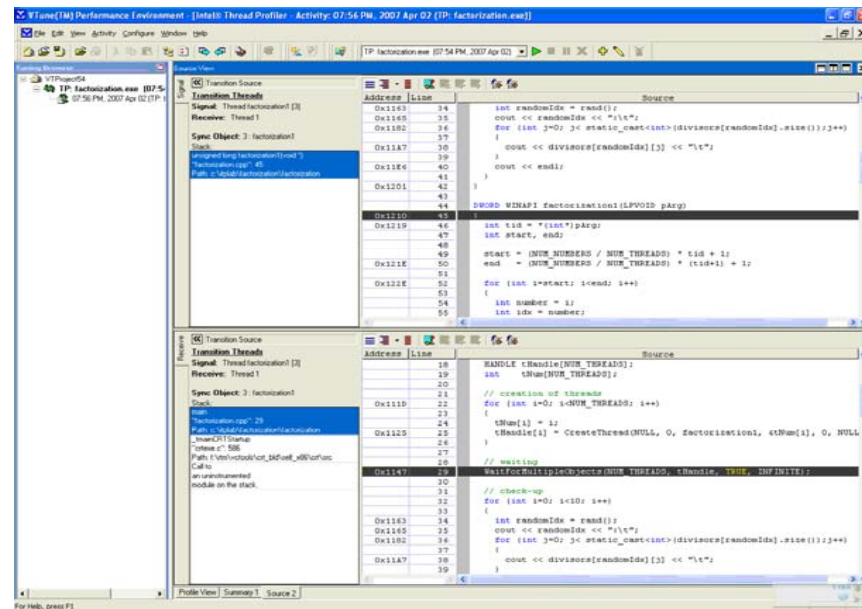
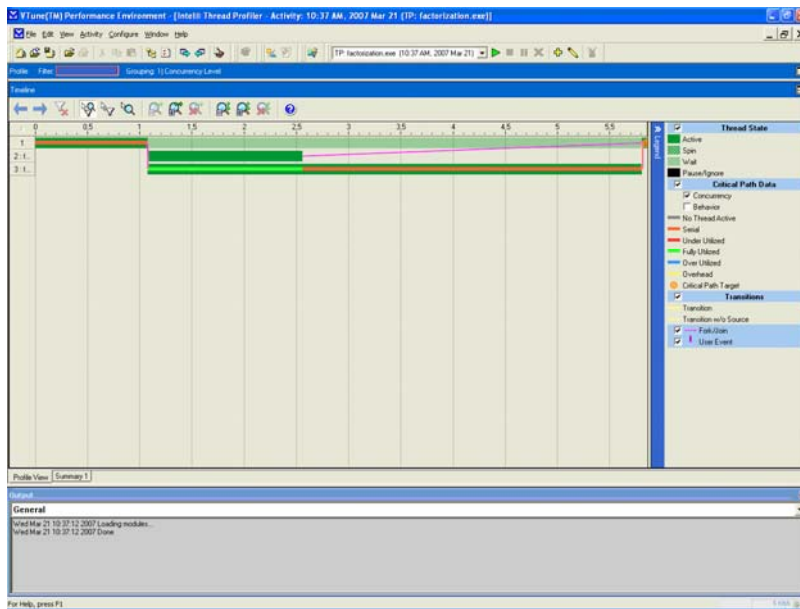
Профилирование

Окно Timeline



Профилирование

Окно Timeline (обращение к коду)



Самостоятельная работа

- ❑ В окне **Profile** установите первичную группировку по потокам, а вторичную по уровню параллелизма. Установите соответствие между столбцами в окне **Profile** и полосами в окне **Timeline**.
- ❑ Мы установили, что основная причина медленной работы нашего приложения – неравномерное распределение нагрузки между потоками. Предложите ваш способ решения данной проблемы.
- ❑ Ответьте на контрольные вопросы
 - Наличие каких цветов в критическом пути свидетельствует о проблемах с производительностью приложения?
 - В чем разница между группировкой по объектам и по типам объектов?



Источники информации

1. «Developing Multithreaded Applications: A Platform Consistent Approach», Intel Corporation, March 2003.
2. «Threading Methodology: Principles and Practices», Intel Corporation, March 2003.
3. «Multi-Core Programming for Academia», Student Workbook, by Intel.
4. «Multi-Core Programming», book by Sh. Akhter and J. Roberts, Intel Press 2006.
5. «Intel® Thread Profiler. Getting Started Guide».
6. «Intel® Thread Profiler. Guide to Sample Code».
7. «Intel® Thread Profiler Help».
8. «Intel® Thread Profiler – краткое описание», материалы по образовательному комплексу «Технологии разработки параллельных программ».

