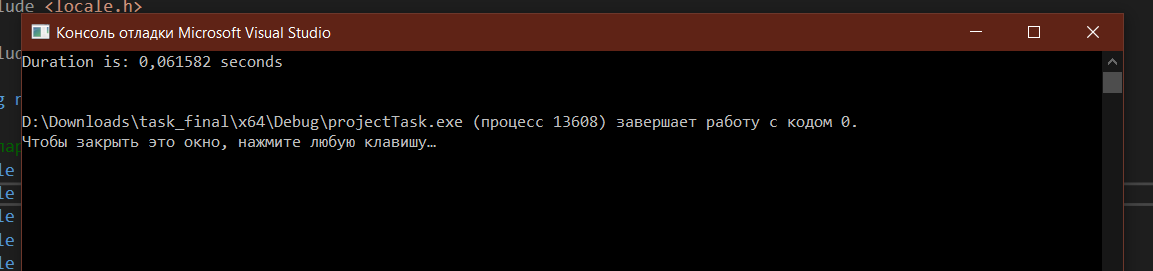
**Отчет по проектному заданию по курсу "Parallel Programming with IPS"**

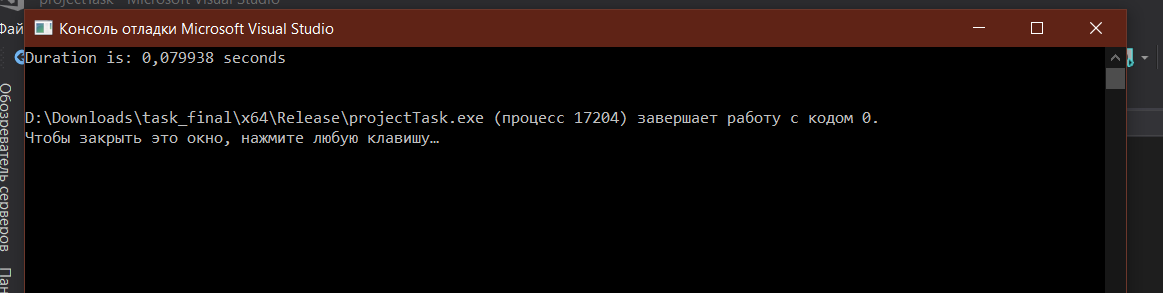
*Выполнила Зуева Ирина ИВТ-21М*

1. Определите время работы последовательной версии разработанной программы в двух режимах: **Debug**и **Release**. Сделайте скрины консоли, где отображается время работы для обоих случаев. Вставьте скрины в отчет к проекту, дав им соответствующие названия.

**Debug**

****

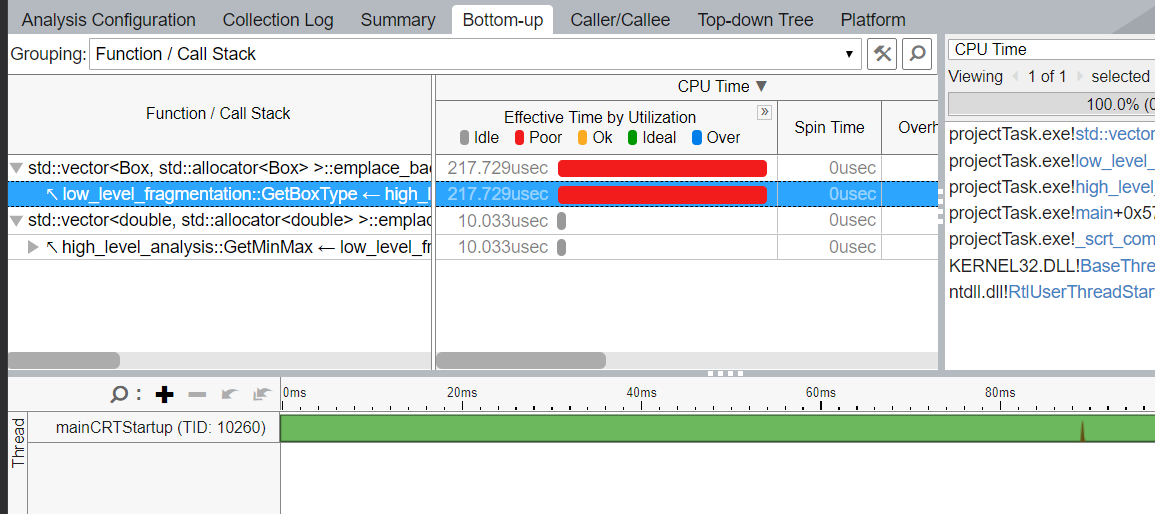
**Release**

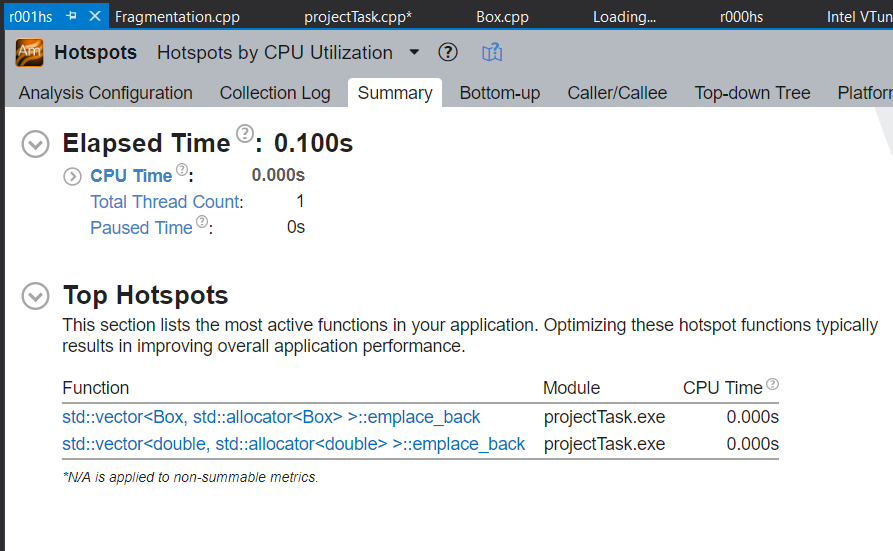
****

1. Постройте полученное рабочее пространство, используя скрипт *MATLAB* [PrintWorkspace.m](https://canvas.instructure.com/courses/1637194/files/78580218/download?wrap=1" \o "PrintWorkspace.m)

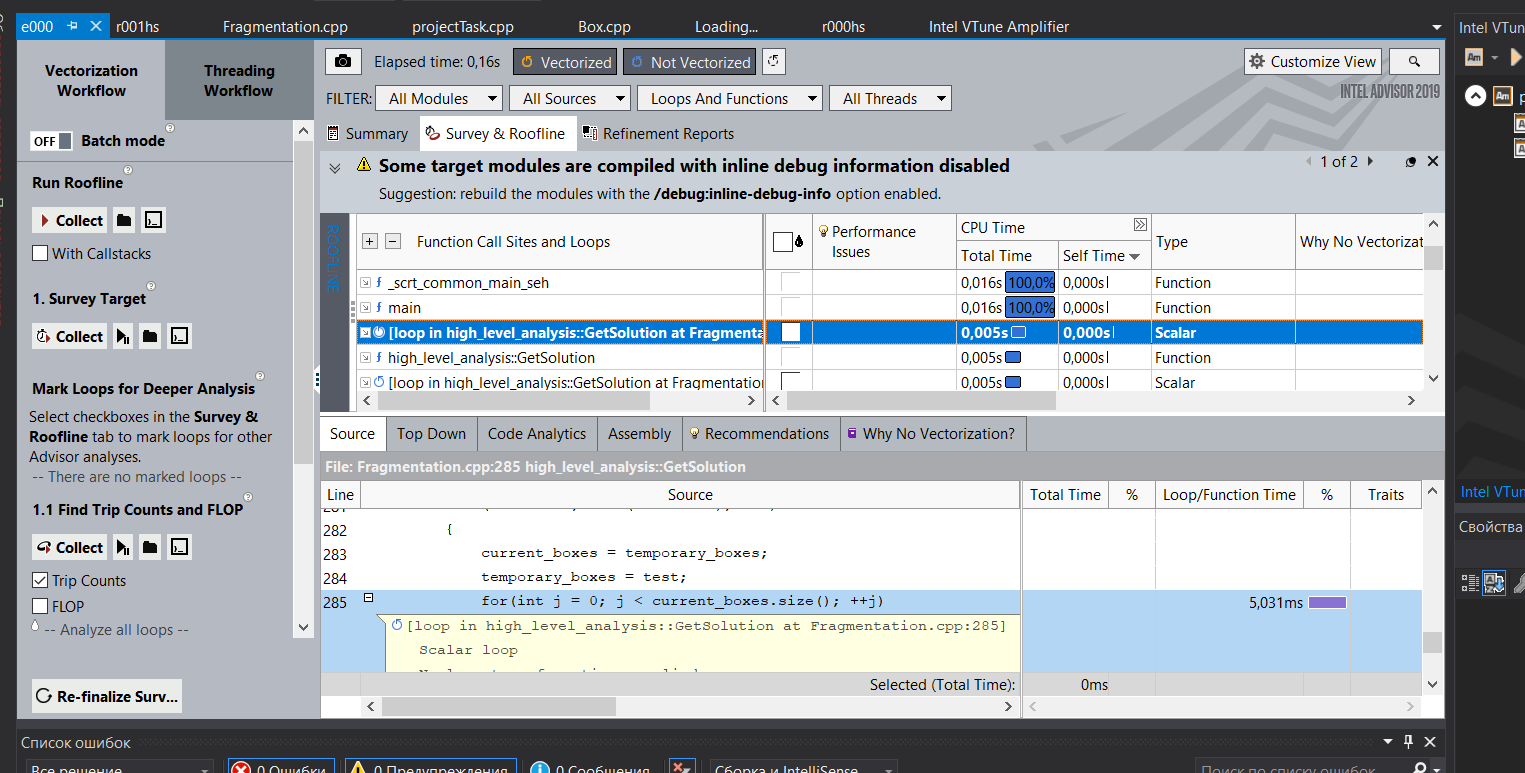


1. Использование ***Amplifier XE*** в целях определения наиболее часто используемых участков кода. Для этого закомментируйте строки кода, отвечающие за запись результатов в выходные файлы, выберите ***New Analysis*** из меню ***Amplifier XE*** на панели инструментов, укажите тип анализа ***Basic Hotspots***, запустите анализ. Сделайте скрин окна результатов анализа и вкладки **Bottom-up*.***В списке, представленном в разделе ***Top Hotspots***вкладки **Summary**должна фигурировать функция ***GetMinMax()***.

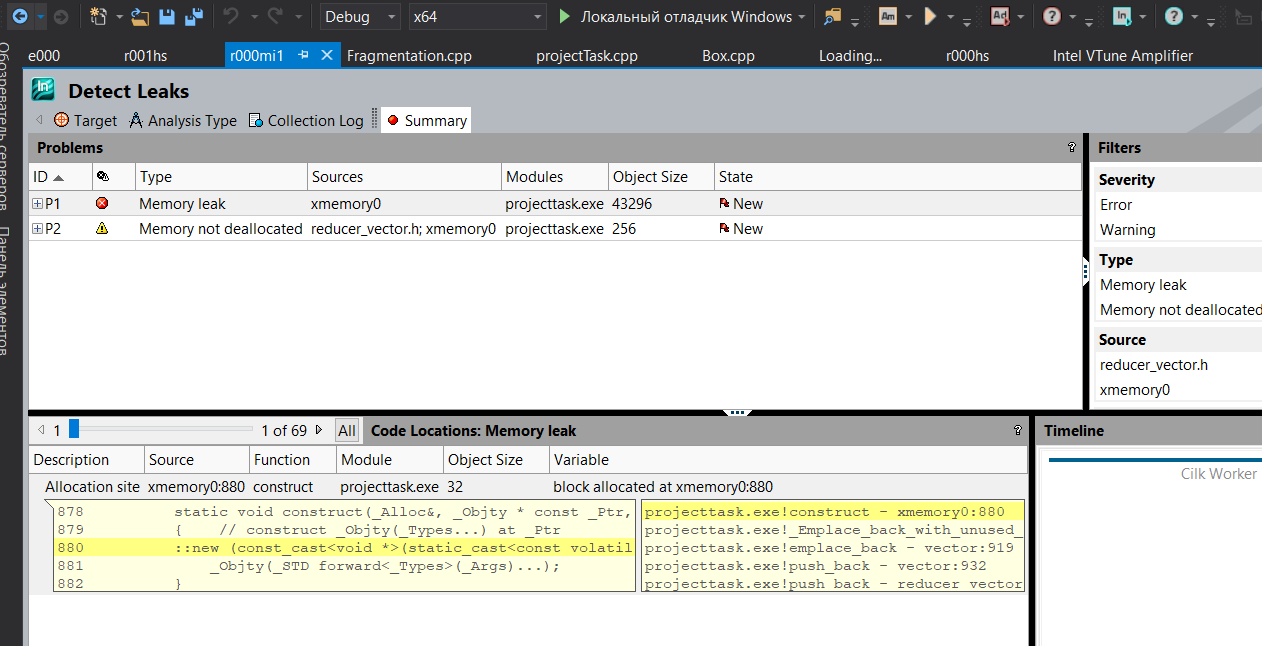


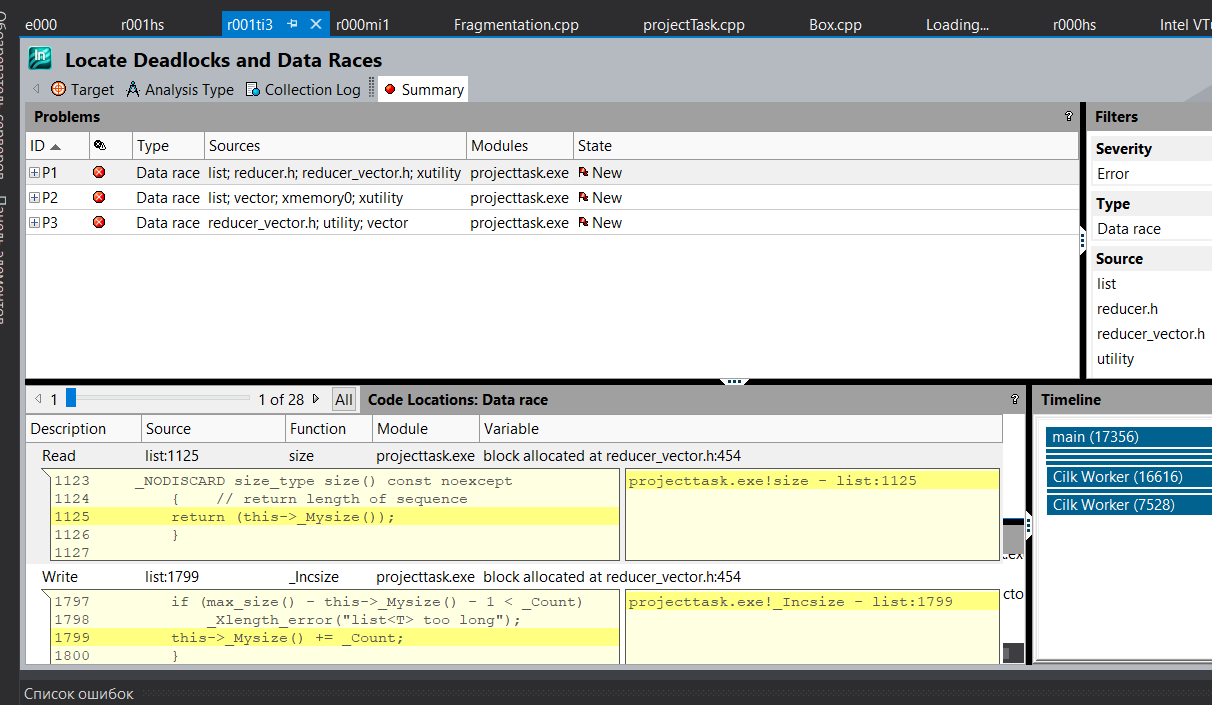


1. Использование ***Parallel Advisor*** с целью определения участков кода, которые требуют наибольшего времени исполнения. Переведите проект в режим **Release**и отключите всякую оптимизацию. Для этого следует выбрать свойства проекта, во вкладке ***С\С++*** перейти в раздел ***Оптимизация***, в пункте меню ***“Оптимизация”*** выбрать ***Отключено (/Od)***. Далее выберем ***Parallel Advisor*** на панели инструментов **Visual Studio** и запустим ***Survey Analysis***. По окончанию анализа Вы должны увидеть, что наибольшее время затрачивается в цикле функции ***GetSolution()***, двойным кликом по данной строке отчета можно перейти к участку исходного кода и увидеть, что имеется в виду цикл, в котором на каждой итерации вызывается функция ***GetBoxType()***. Сделайте скрины результатов ***Survey Analysis***, сохраните их, добавьте в отчет. Вернитесь в режим **Debug**.

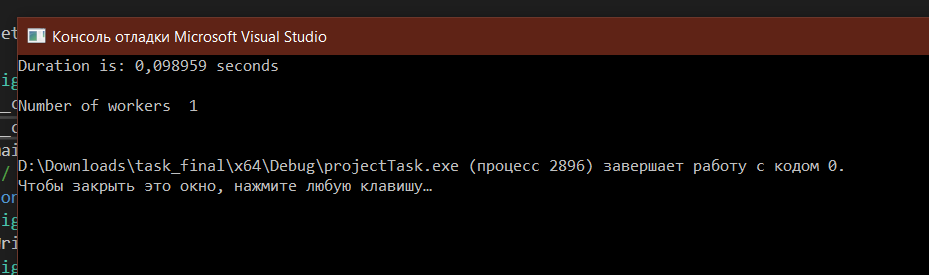


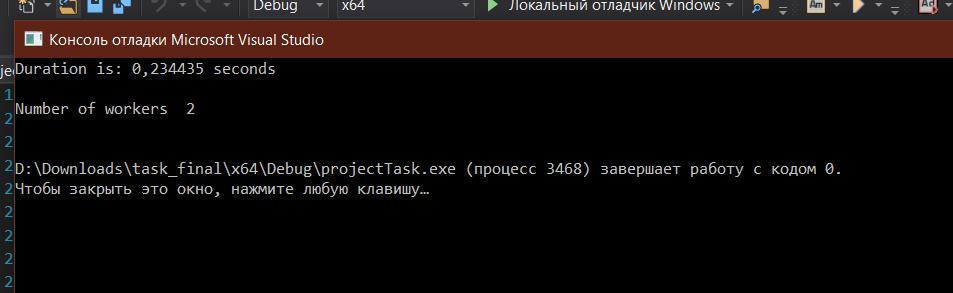
1. Определение ошибок после введения параллелизации. Запустите анализы  **Inspector XE**: **Memory Error Analysis** и **Threading Error Analysis** на различных уровнях (**Narrowest**, **Medium**, **Widest**). Приложите к отчету скрины результатов запуска перечисленных анализов. Исправьте обнаруженные ошибки, приложите новые скрины результатов анализов, в которых ошибки отсутствуют. Примечание: "глюки" **Intel** **Cilk Plus**исправлять не нужно.

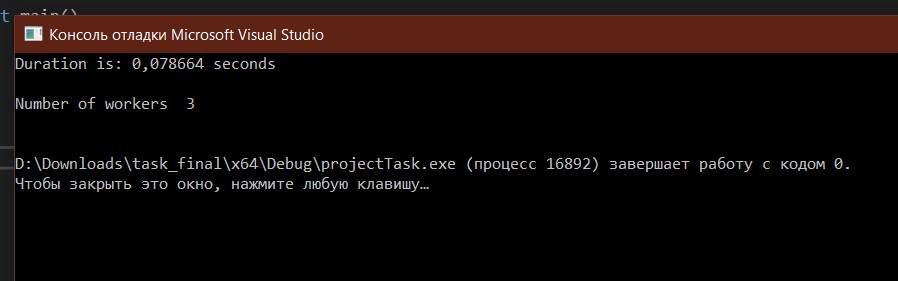


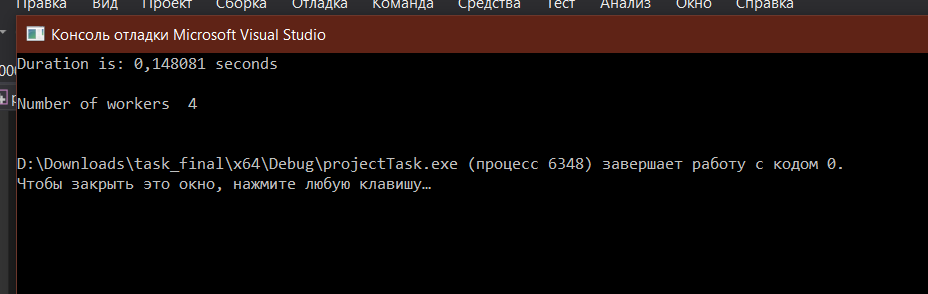


1. Изменяя Х, запускайте программу и фиксируйте время ее выполнения, каждый раз сохраняйте скрины консоли***,*** где должно быть отображено количество вычислителей (***cout << "Number of workers " << \_\_cilkrts\_get\_nworkers() << endl;***) и время работы программы.









1. Визуализация полученного решения. Поэкспериментируйте со входными параметрами программы и отобразите несколько версий полученного рабочего пространство робота.Рисунки приложите к отчету.



