Учебная практика 03.02

Яковлев Данил ИС-31

<https://github.com/BeerZavr/VOODOOsh>

1. Открываем проект для отладки в Visual Studio и устанавливаем точки останова в приложении в точках, где вы хотите проверить загрузку ЦП.

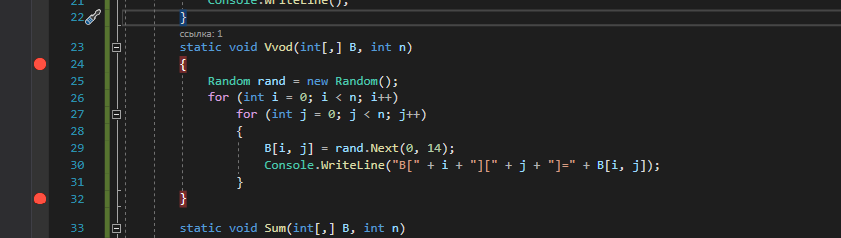


Рис. 1

2. Открываем средства диагностики с помощью Отладка > Окна > Показать средства диагностики.

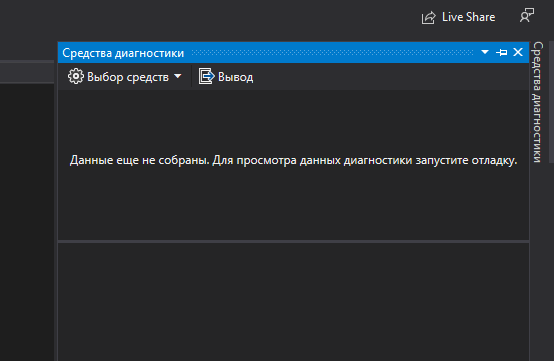


Рис. 2

3. С помощью параметра Выбор средств на панели инструментов включаем выбираем для просмотра Использование памяти или Загрузка ЦП (либо оба средства)

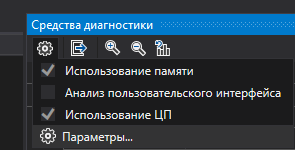


Рис. 3

4. Начинаем отладку через меню Отладка > Начать отладку (Запустить на панели инструментов или F5)

По завершении загрузки приложения отображается представление "Сводка" средств диагностики. Если вам нужно открыть окно, щелкните Отладка > Окна > Показать средства диагностики.

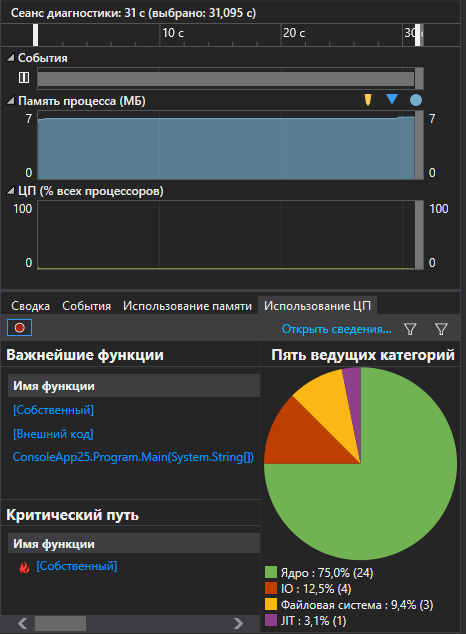


Рис. 4

5. Запустите сценарий, который вызвал срабатывание первой точки останова.

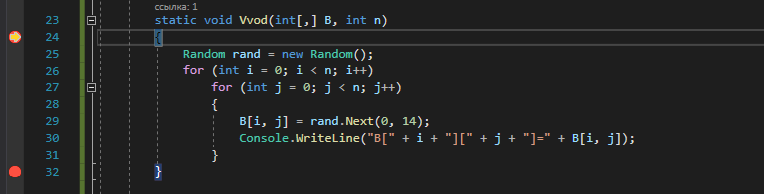


Рис. 5

6. Приостановив отладчик, включаем сбор данных о загрузке ЦП, а затем открываем вкладку Загрузка ЦП.

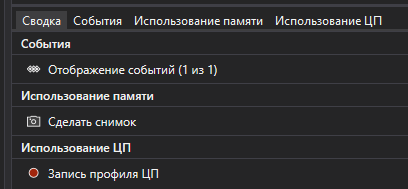


Рис. 6

При выборе пункта Запись профиля ЦП Visual Studio начнет записывать функции и сведения о времени их выполнения. Эти собранные данные можно просматривать только в том случае, если приложение останавливается в точке останова.

7. Нажимаем клавишу F5, чтобы запустить приложение до второй точки останова.

Теперь у вас есть данные о производительности приложения именно для той области кода, где выполняется между двумя точками останова.

Профилировщик начинает подготавливать данные потока. Дождитесь завершения этой операции.

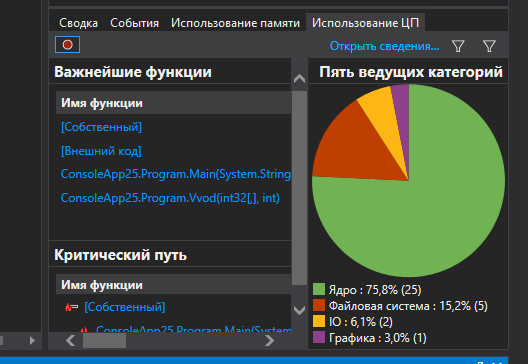


Рис. 7

8. В списке функций изучите функции, которые выполняют большую часть работы.

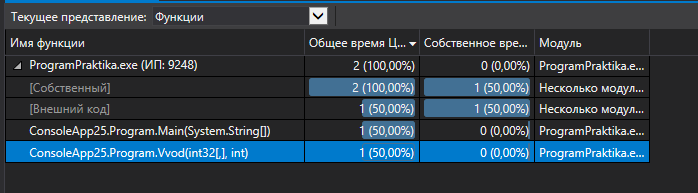


Рис. 8

9. В списке функций дважды щелкаем одну из функций нашего приложения, которая выполняет много работы.

При двойном щелчке функции в левой панели откроется представление Вызывающий/Вызываемый.

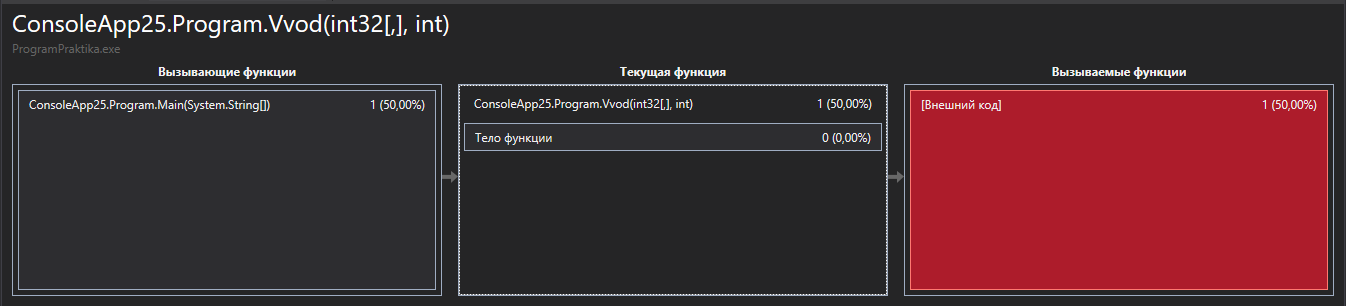


Рис. 9

10. Чтобы увидеть более обобщенное представление, показывающее порядок, в котором вызываются функции, выберите в раскрывающемся списке в верхней части панели пункт Дерево вызовов.

Каждая нумерованная область на рисунке соответствует определенному шагу в процедуре.

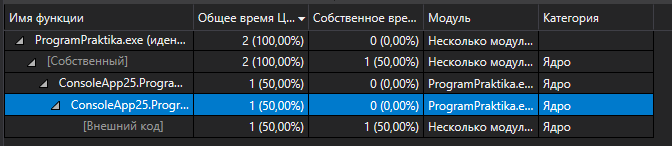


Рис. 10

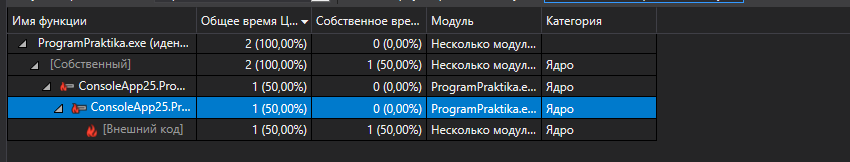


Рис. 11

Ниже приведены дополнительные сведения по значениям столбца.

* Значение Общее время ЦП указывает, какой объем работы был выполнен некоторой функцией и всеми вызванными ей функциями. Высокие значения общего времени ЦП указывают на функции, которые в целом являются наиболее ресурсоемкими.
* Значение **Собственное время** указывает, какой объем работы был выполнен кодом в теле функции, за исключением работы, выполненной вызванными ей функциями. Высокие значения собственного времени ЦП могут свидетельствовать о проблемах производительности внутри самой функции.
* **Модули**. Имя модуля, содержащего функцию, или количество модулей, содержащих функции в узле [Внешний код].

11. **Внешний код** – это функции в компонентах системы и платформы, которые исполняются вашим кодом. Внешний код включает функции, которые запускают и останавливают приложение, управляют потоками и предоставляют приложению другие низкоуровневые службы. В большинстве случаев внешний код вас интересовать не будет, поэтому средство "Загрузка ЦП" собирает внешние функции пользовательского метода в один узел **[Внешний код]**.

**Показать внешний код** > **Представление фильтра** > **Применить**.

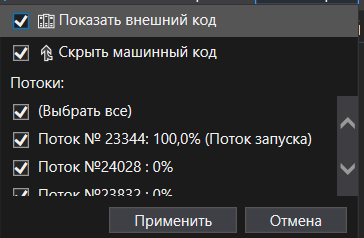


Рис. 12