

Практическая работа

Отладка параллельного приложения с использованием инструмента Parallel Task

1. Создадим новое консольное приложение в Visual Studio, с названием "ParralelTaskAppllication":

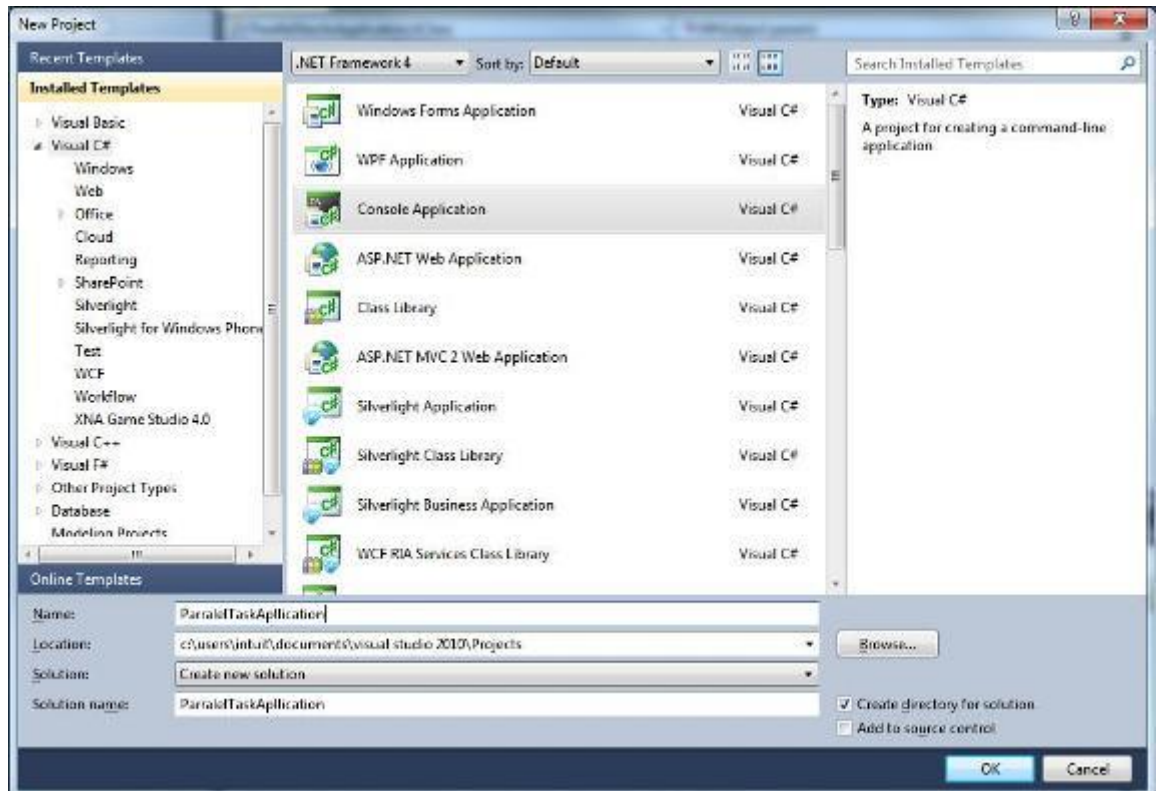


Рис. 1.

2. Скопируем, заранее созданный код, в проект:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Diagnostics;
namespace ParralelTaskAppllication
{
    class XClass
    {
        public void MA(object param) { MB(param); }
        public void MB(object param) { MC(param); }
        object s1 = new object();
        object s2 = new object();
        public void MC(object param)
        {
            if (param == "1")
            {
                MD();
            }
            if (param == "2")
            {
                ME();
            }
        }
    }
}
```

```

    }
    if (param == "3")
    {
        MF();
    }
}
public void MD()
{
    ME();
}
public void ME()
{
    while (true) Thread.SpinWait(int.MaxValue / 20);
}
public void MF()
{
    ML();
}
public void MG(object param)
{
    MH(param);
}
public void MH(object param)
{
    MI(param);
}
public void MI(object param)
{
    if (param == "4")
    {
        MJ();
    }
    else
    {
        MK();
    }
}
public void MJ()
{
    Monitor.Enter(s1); Thread.SpinWait(int.MaxValue / 20); Monitor.Enter(s2);
}
public void MK()
{
    Monitor.Enter(s2);
    Thread.SpinWait(int.MaxValue / 10);
    Monitor.Enter(s1);
}
public void ML()
{
    MM();
}
public void MM()

```

```

    {
        while (true)
        { Thread.SpinWait(int.MaxValue / 3);
          Debugger.Break();
        };
    }
}
class Program
{
    static XClass obj = null;
    static void Main(string[] args)
    {
        obj = new XClass();
        Task.Factory.StartNew(obj.MA, "1");
        Task.Factory.StartNew(obj.MA, "2");
        Task.Factory.StartNew(obj.MA, "3");
        Task.Factory.StartNew(obj.MG, "4");
        Task.Factory.StartNew(obj.MG, "5");
        Console.ReadLine();
    }
}
}

```

3. Запустите отладку программы с помощью кнопки "Start Debbing" из меню Visual Studio - "Debug" или с помощью клавиши "F5":

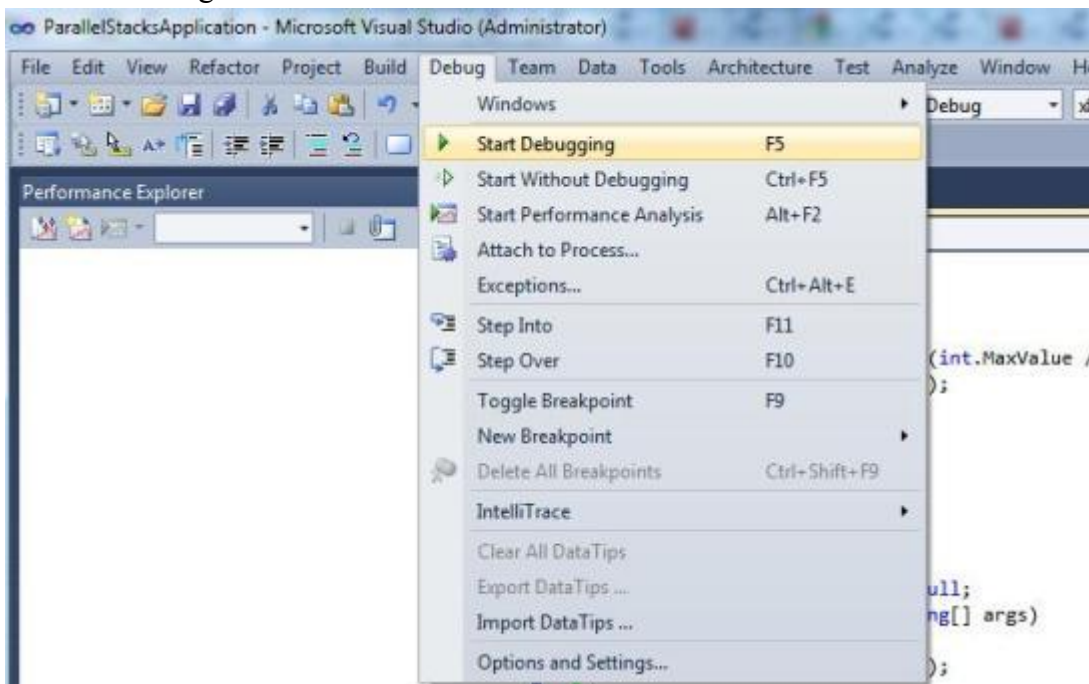


Рис. 2.

4. Через несколько секунд "Курсор отладки" должен остановиться на методе `Debugger.Break()`:

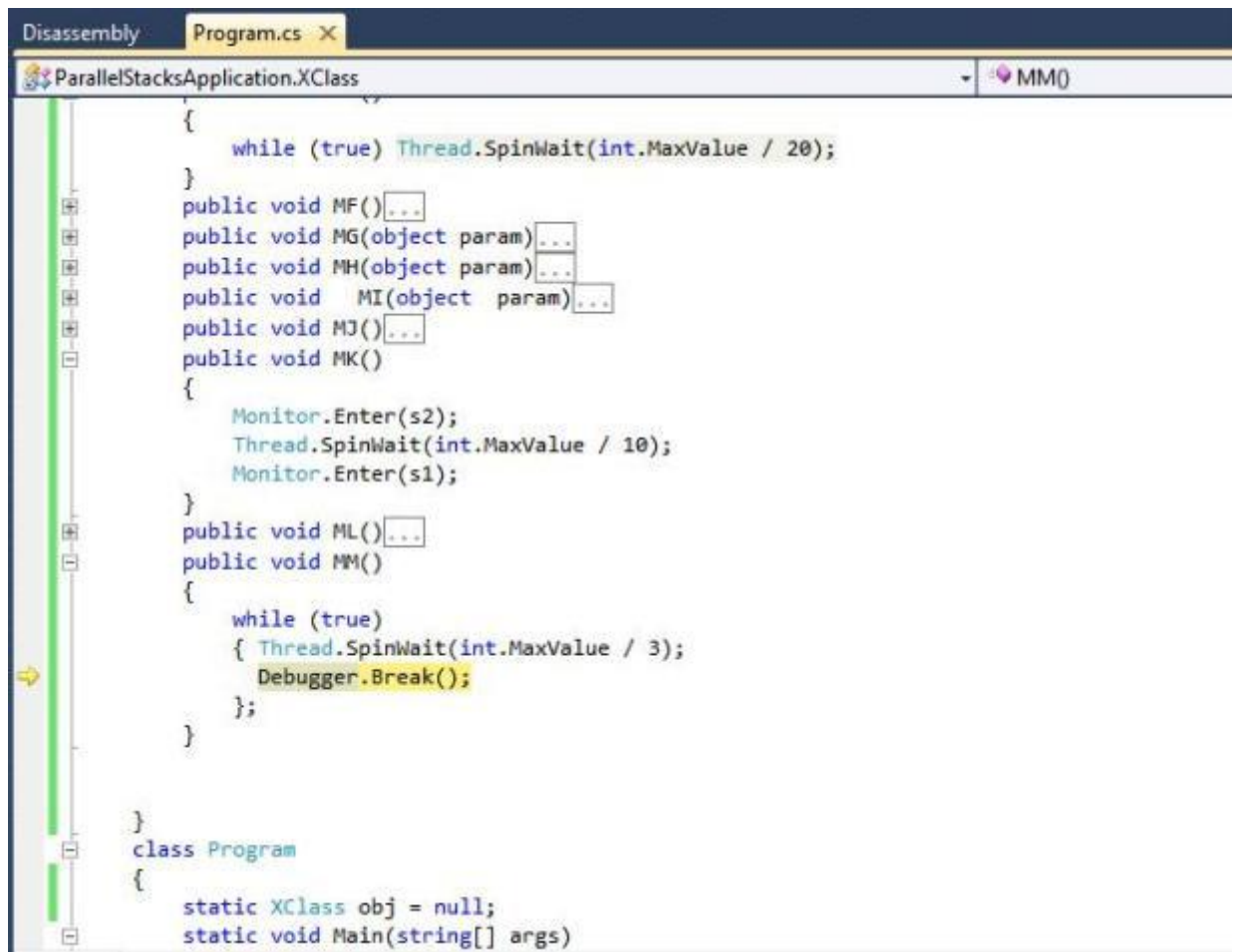


Рис. 3.

5. Запустите окно параллельных задач (Parallel Task) из меню "Debug -> Windows" или с помощью сочетания клавиш "Ctrl+Shift+D,K":

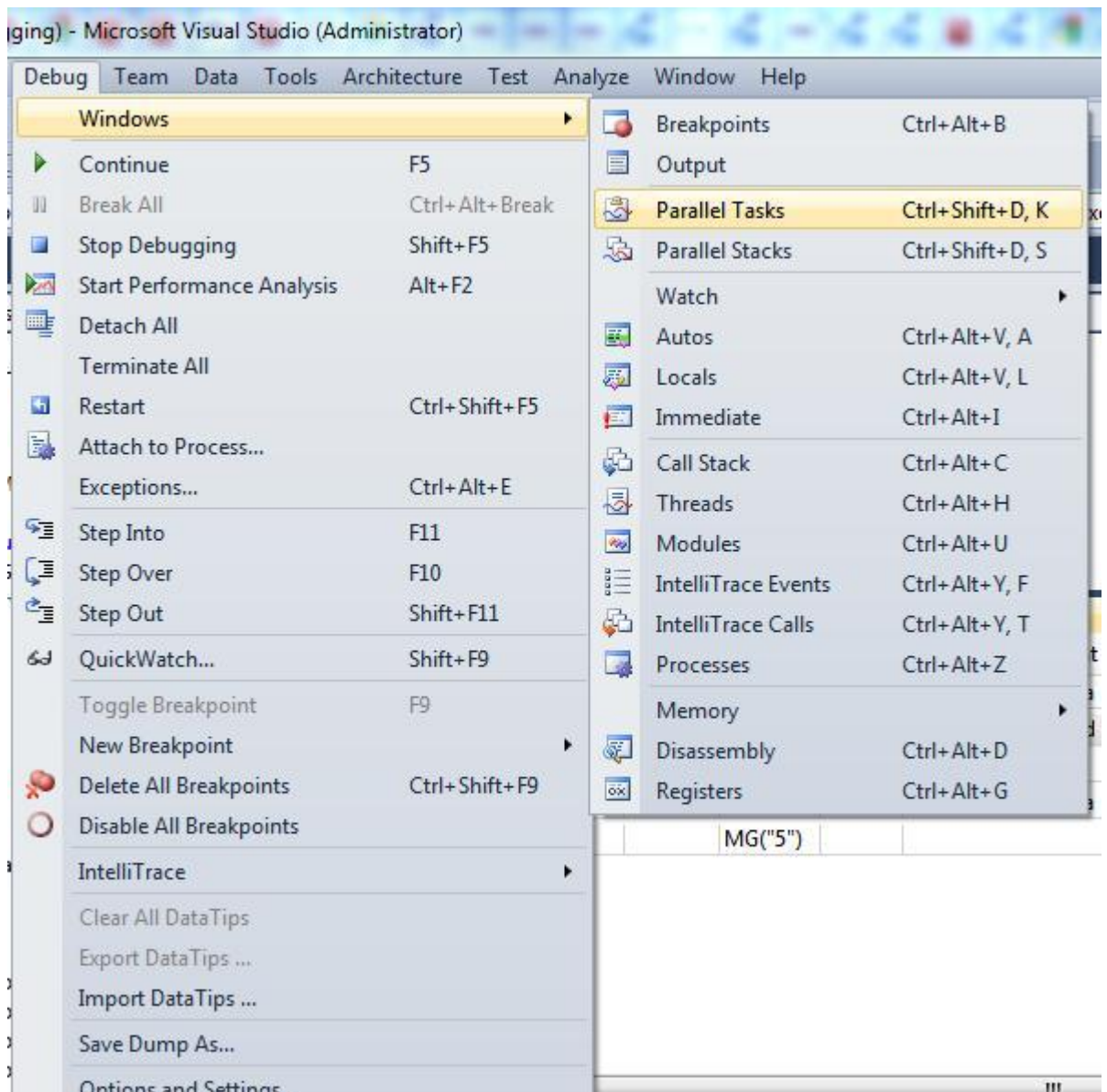


Рис. 4.

- Откроется окно параллельных задач (Parallel Task):

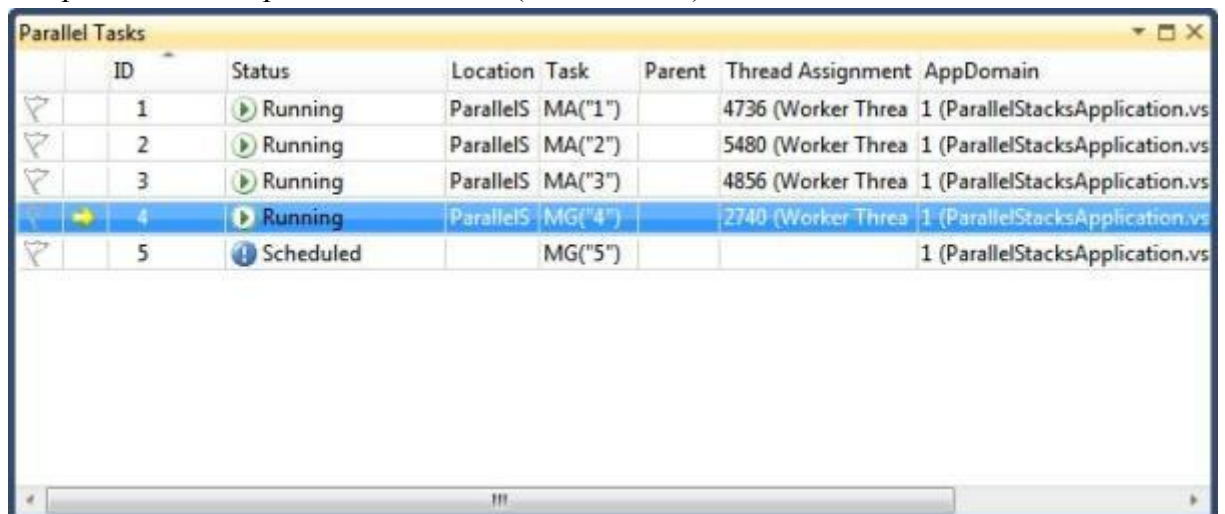


Рис. 5.

В данном окне, по каждой запущенной задаче можно увидеть: уникальный идентификатор задачи, имя потока, который запускает задача, ее расположение.

7. Для того, что бы "заморозить" выполнение конкретного потока, щелкните на нужном потоке в списке - правой кнопкой мыши и выберите пункт "Freeze Assigned Thread":

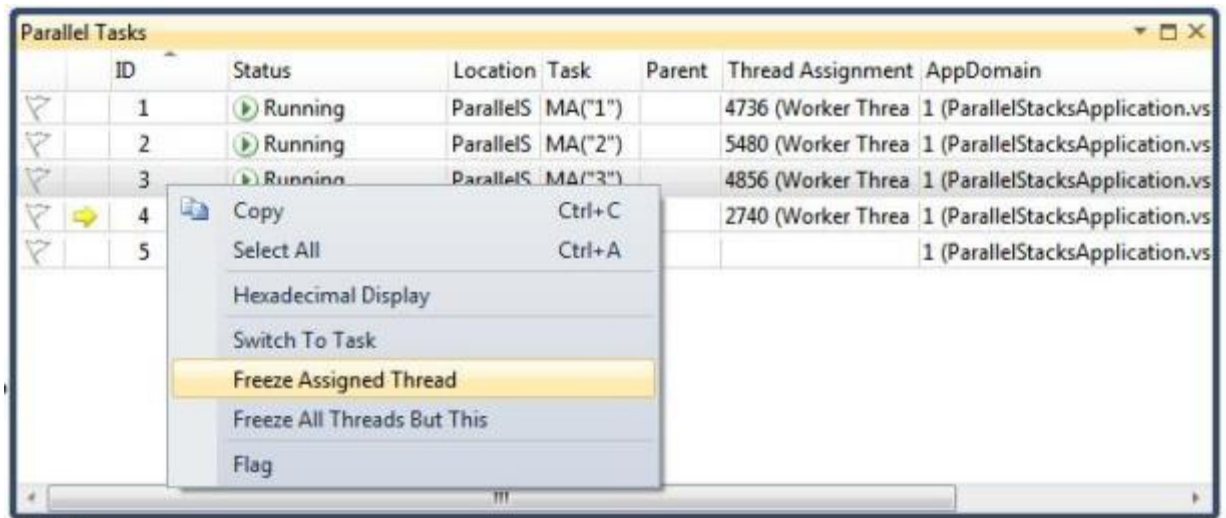


Рис. 6.

Напротив потока должен отображаться соответствующий значок:

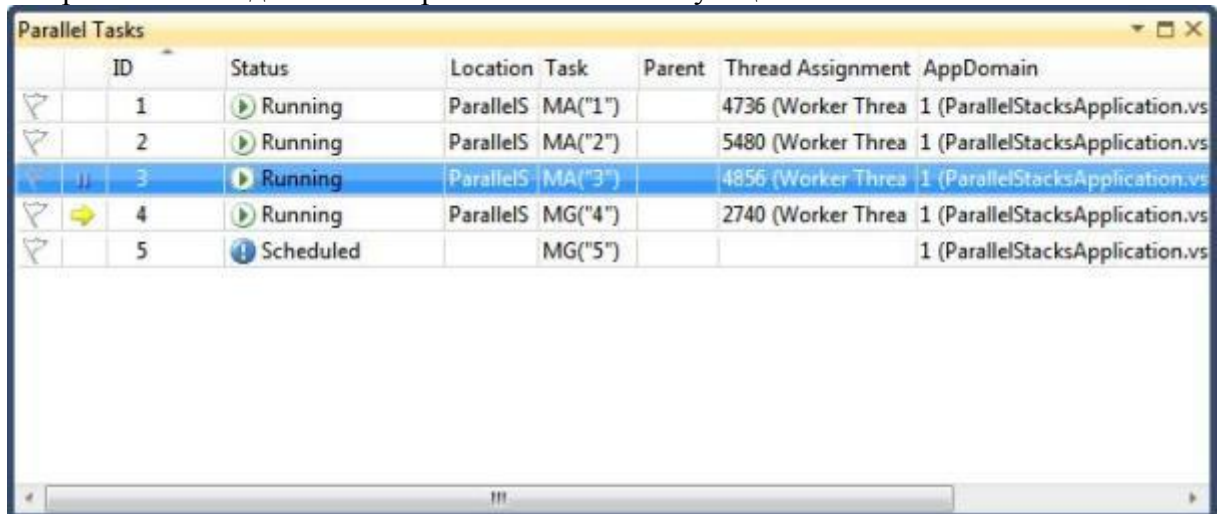


Рис. 7.

Если поток "замораживается", то он не будет выполняться при проходе по коду после текущей точки останова.

8. Для перехода на нужный поток используется пункт меню "Switch To Task". При этом программа перейдет на участок кода соответствующий данному потоку. В нашем случае метод с ID = 2 соответствует методу public void ME():

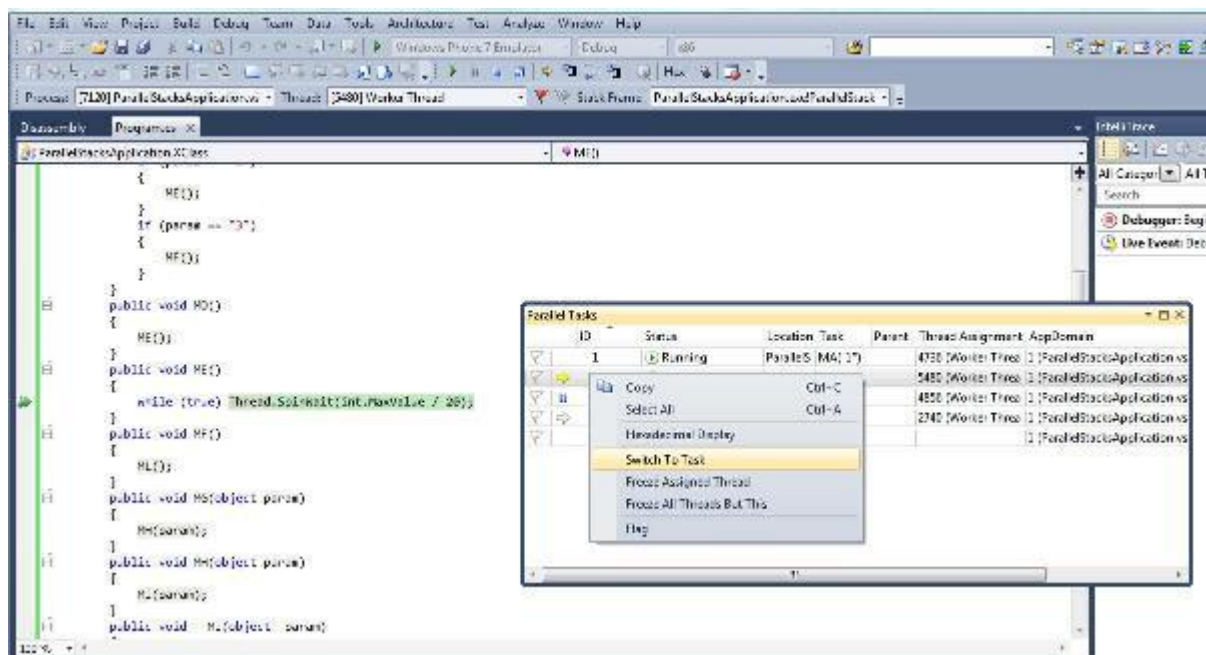


Рис. 8.

9. Для завершения процесса отладки используется пункт из меню "Debug" - "Stop Debugging" или сочетание клавиш "Shift+F5":

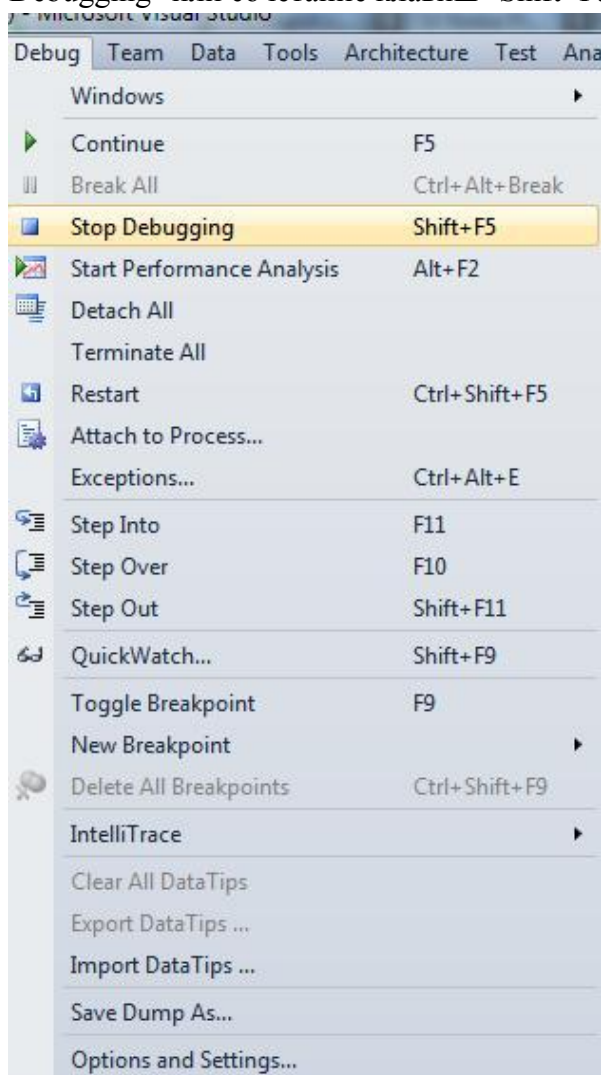


Рис. 9.