

Используем AOT-компиляцию правильно

Елизавета Голенок

Ведущий разработчик MTS IT

golenok-ea5@narod.ru

@marmothetka

Введение

- Ahead-of-time (AOT) compilation
 - Что это такое и с чем его «готовить»
 - Зачем нужен AOT, когда есть JIT
- AOT и runtimes:
 - AOT для .NET
 - AOT для .NET Core
 - AOT для mono

Окружение

- Intel Core i5 7th Gen
- 8Гб ОП
- 512 SSD
- Windows 10 / Ubuntu 17.04

Ahead-of-time компиляция



ЗАЧЕМ НУЖНА АОТ-КОМПИЛЯЦИЯ, КОГДА ЕСТЬ JIT?



Зачем нужна AOT-компиляция, когда есть JIT?



Интерпретаторы



JIT



AOT

Зачем нужна AOT-компиляция, когда есть JIT?

	Интерпретатор	JIT	AOT
Время для сборки			
Дополнительная память			

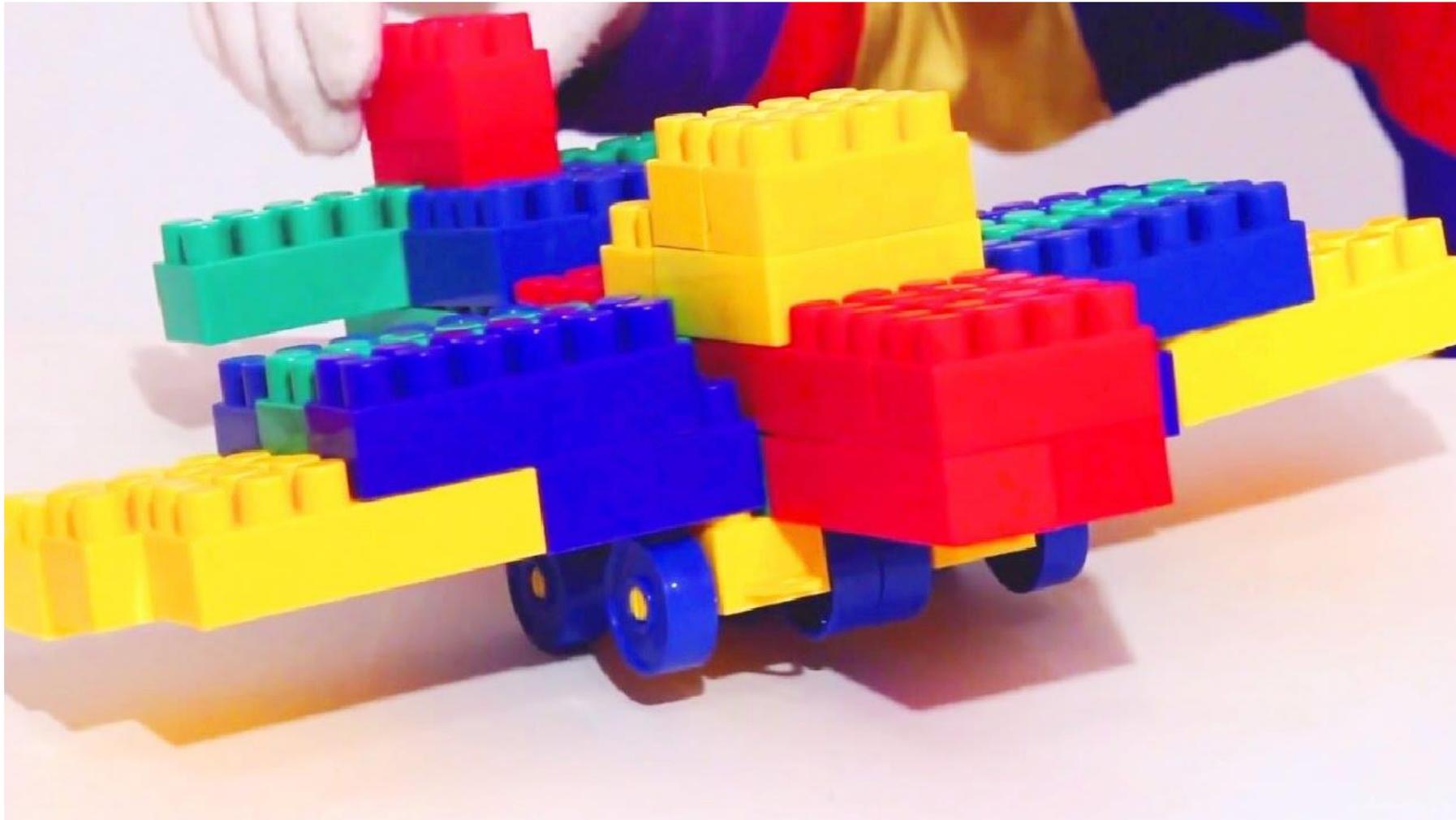
Сокращаем время запуска



Потенциально увеличиваем производительность

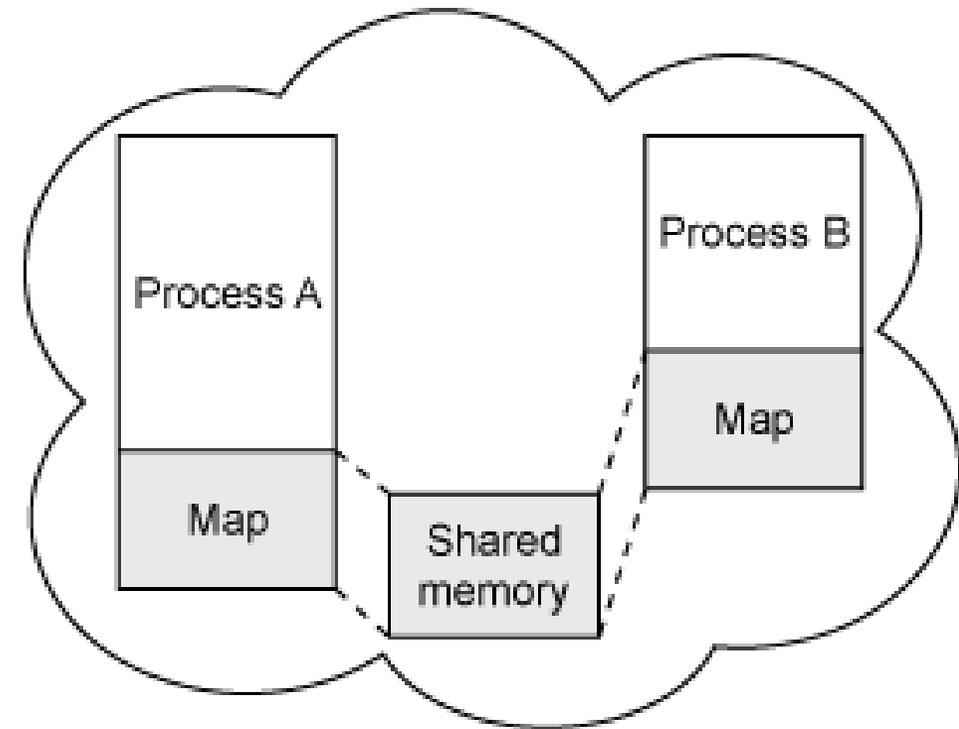


Используем shared-memory



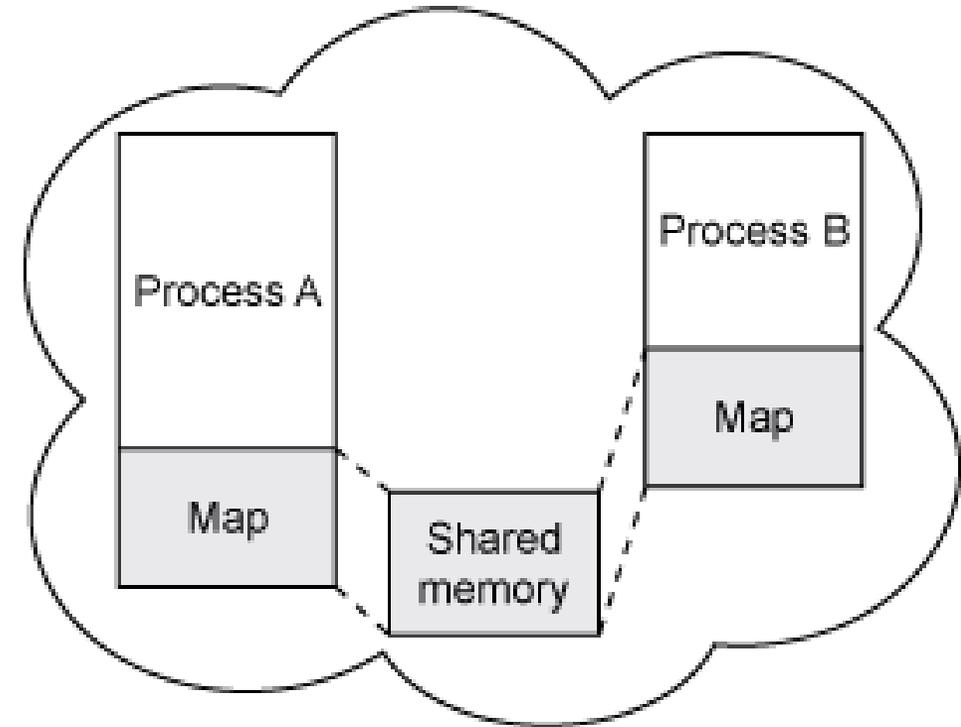
Сценарий использования разделяемой памяти

- Приложение получает доступ к разделяемой памяти, используя семафор.



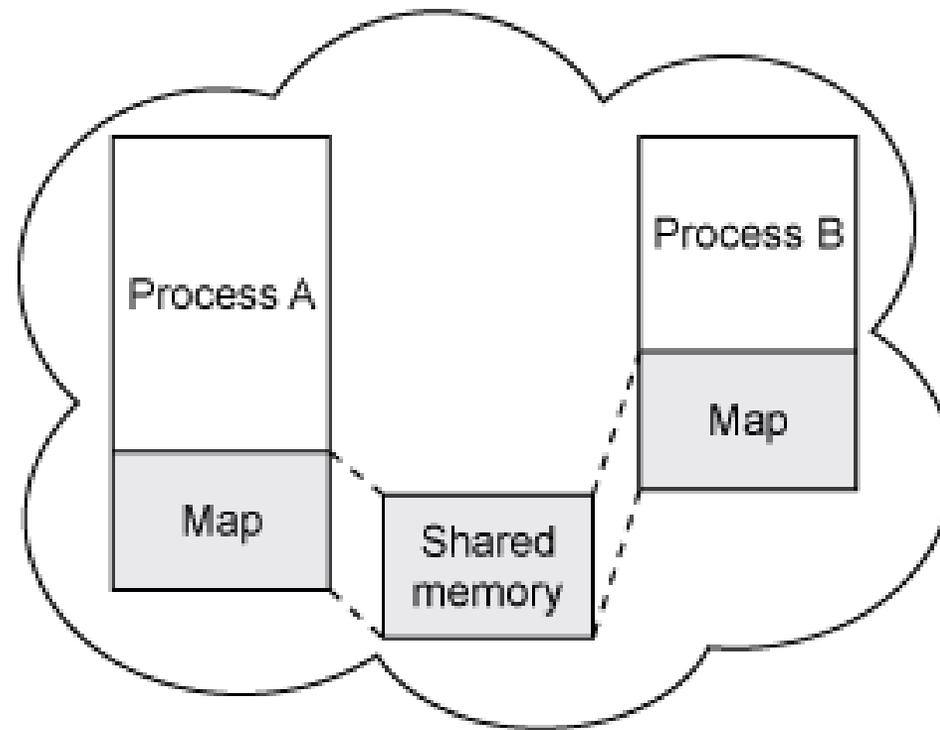
Сценарий использования разделяемой памяти

- Приложение получает доступ к разделяемой памяти, используя семафор.
- Приложение производит запись данных в разделяемую память.



Сценарий использования разделяемой памяти

- Приложение получает доступ к разделяемой памяти, используя семафор.
- Приложение производит запись данных в разделяемую память.
- После завершения записи данных приложение освобождает доступ к разделяемой памяти с помощью семафора.



Для чего нужна AOT-компиляция?

- Сокращение времени запуска
- Потенциальное улучшение производительности
- Низкоуровневые образы могут совместно использоваться процессами

-
- Априори поставленные задачи производителем (например, Apple iOs)

АОТ для .NET. NGen

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561
System.Xml.XPath.XDocument, Version=4.0.2.0, Culture=neutral, PublicKe
System.Xml.ReaderWriter, Version=4.1.0.0, Culture=neutral, PublicKeyTo
System.Xml.XmlDocument, Version=4.0.2.0, Culture=neutral, PublicKeyTok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
windowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
wpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

АОТ для .NET. NGen

- **NGen** – Native image generator – генератор образов в машинном коде

АОТ для .NET. NGen

- **NGen** – Native image generator – генератор образов в машинном коде
- NGen.exe создает образы в машинном коде и устанавливает их в кэш образов

АОТ для .NET. NGen

- **NGen** – Native image generator – генератор образов в машинном коде
- NGen.exe создает образы в машинном коде и устанавливает их в кэш образов
- Среда выполнения может использовать образы в машинном коде, находящиеся в кэше

АОТ для .NET. NGen

- Работает только в режиме полного доверия в соответствии с политикой code access security (CAS)

АОТ для .NET. NGen

- Работает только в режиме полного доверия в соответствии с политикой code access security (CAS)
- NGen считает ссылки на зависимые сборки.

АОТ для .NET. NGen

- Работает только в режиме полного доверия в соответствии с политикой code access security (CAS)
- NGen считает ссылки на зависимые сборки.
- При каждом изменении сборки необходимо пересоздавать

АОТ для .NET. NGen

- Работает только в режиме полного доверия в соответствии с политикой code access security (CAS)
- NGen считает ссылки на зависимые сборки.
- При каждом изменении сборки необходимо пересоздавать
- При обнаружении метода, отсутствующего в образе в машинном коде, выполняется JIT-компиляция.

АОТ для .NET. NGen

- Работает только в режиме полного доверия в соответствии с политикой code access security (CAS)
- NGen считает ссылки на зависимые сборки.
- При каждом изменении сборки необходимо пересоздавать
- При обнаружении метода, отсутствующего в образе в машинном коде, выполняется JIT-компиляция.
- Работает только под windows

Задача

- Подобрать метод для оптимизации запуска нескольких процессов

Задача

```
// Configure and start threads using ThreadPool.
for (int i = 0; i < FibonacciCalculations; i++)
{
    doneEvents[i] = new ManualResetEvent(false);
    Fibonacci f = new Fibonacci(rand.Next(20, 50), doneEvents[i]);
    fibArray[i] = f;
    ThreadPool.QueueUserWorkItem(f.ThreadPoolCallback, i);
}

// Wait for all threads in pool to calculate.
WaitHandle.WaitAll(doneEvents);
```

Варианты

- JIT
- NGen
- `RuntimeHelpers.PrepareMethod`

Первый вариант. JIT

- /p:Configuration=Release
- /t:Rebuild
- x64
- **Запуск исполняемого файла без VS и Rider**



Первый вариант. JIT

- /p:Configuration=Release
- /t:Rebuild
- x64
- **Запуск исполняемого файла без VS и Rider**



- **Получаем время работы программы**

Второй вариант. NGen

- /p:Configuration=Release
- /t:Rebuild
- X64
- Проверяем, что нет старого образа

```
P: C:\> ngen.exe display ExampleNGen
Microsoft (R) CLR Native Image Generator - version 4.7.2556.0
Copyright (c) Microsoft Corporation. All rights reserved.
Error: The specified assembly is not installed.
```

Второй вариант. NGen

- /p:Configuration=Release
- /t:Rebuild
- X64
- Проверяем, что нет старого образа

```
PS C:\> ngen.exe display ExampleNGen
Microsoft (R) CLR Native Image Generator - Version 4.7.2556.0
Copyright (c) Microsoft Corporation. All rights reserved.
Error: The specified assembly is not installed.
```

Второй вариант. NGen

- /p:Configuration=Release
- /t:Rebuild
- X64
- Проверяем, что нет старого образа
- Генерируем образ

```
PS C:\WINDOWS\system32> ngen.exe install C:\Users\ea\source\repos\
ExampleNGen\ExampleNGen\bin\x64\Release\ExampleNGen.exe
```

Второй вариант. NGen

- Образ генерируется...

```
PS C:\WINDOWS\system32> ngen.exe install C:\Users\ea\source\repos\
ExampleNGen\ExampleNGen\bin\x64\Release\ExampleNGen.exe
Microsoft (R) CLR Native Image Generator - Version 4.7.2556.0
Copyright (c) Microsoft Corporation. All rights reserved.
2>    Compiling assembly System.Data.SqlXml, Version=4.0.0.0, Cult
ure=neutral, PublicKeyToken=b77a5c561934e089 (CLR v4.0.30319) ...
1>    Compiling assembly C:\Users\ea\source\repos\ExampleNGen\Exam
pleNGen\bin\x64\Release\ExampleNGen.exe (CLR v4.0.30319) ...
3>    Compiling assembly System.Security, Version=4.0.0.0, Culture
=neutral, PublicKeyToken=b03f5f7f11d50a3a (CLR v4.0.30319) ...
PS C:\WINDOWS\system32>
```

Второй вариант. NGen

- Проверяем, что образ создан

```
NGEN Roots:
```

```
C:\Users\ea\source\repos\ExampleNGen\ExampleNGen\bin\x64\Release\ExampleNGen.exe
```

```
NGEN Roots that depend on "ExampleNGen":
```

```
C:\Users\ea\source\repos\ExampleNGen\ExampleNGen\bin\x64\Release\ExampleNGen.exe
```

```
Native Images:
```

```
ExampleNGen, Version=1.0.0.0, Culture=neutral,  
PublicKeyToken=null
```

Второй вариант. NGen

- /p:Configuration=Release
- /t:Rebuild
- X64
- Проверяем, что нет старого образа
- Генерируем образ
- Запускаем исполняемый файл

Второй вариант. NGen

- /p:Configuration=Release
- /t:Rebuild
- X64
- Проверяем, что нет старого образа
- Генерируем образ
- Запускаем исполняемый файл
- Получаем время работы программы

Вариант 3. RuntimeHelpers.PrepareMethod

- Получаем все методы, которые есть в сборке

Вариант 3. RuntimeHelpers.PrepareMethod

- Прогоняем все методы через PrepareMethod

```
foreach (var method in type.GetMethods(BindingFlags.DeclaredOnly |  
    BindingFlags.NonPublic |  
    BindingFlags.Public |  
    BindingFlags.Instance |  
    BindingFlags.Static))  
{  
    RuntimeHelpers.PrepareMethod(method.MethodHandle);  
}
```

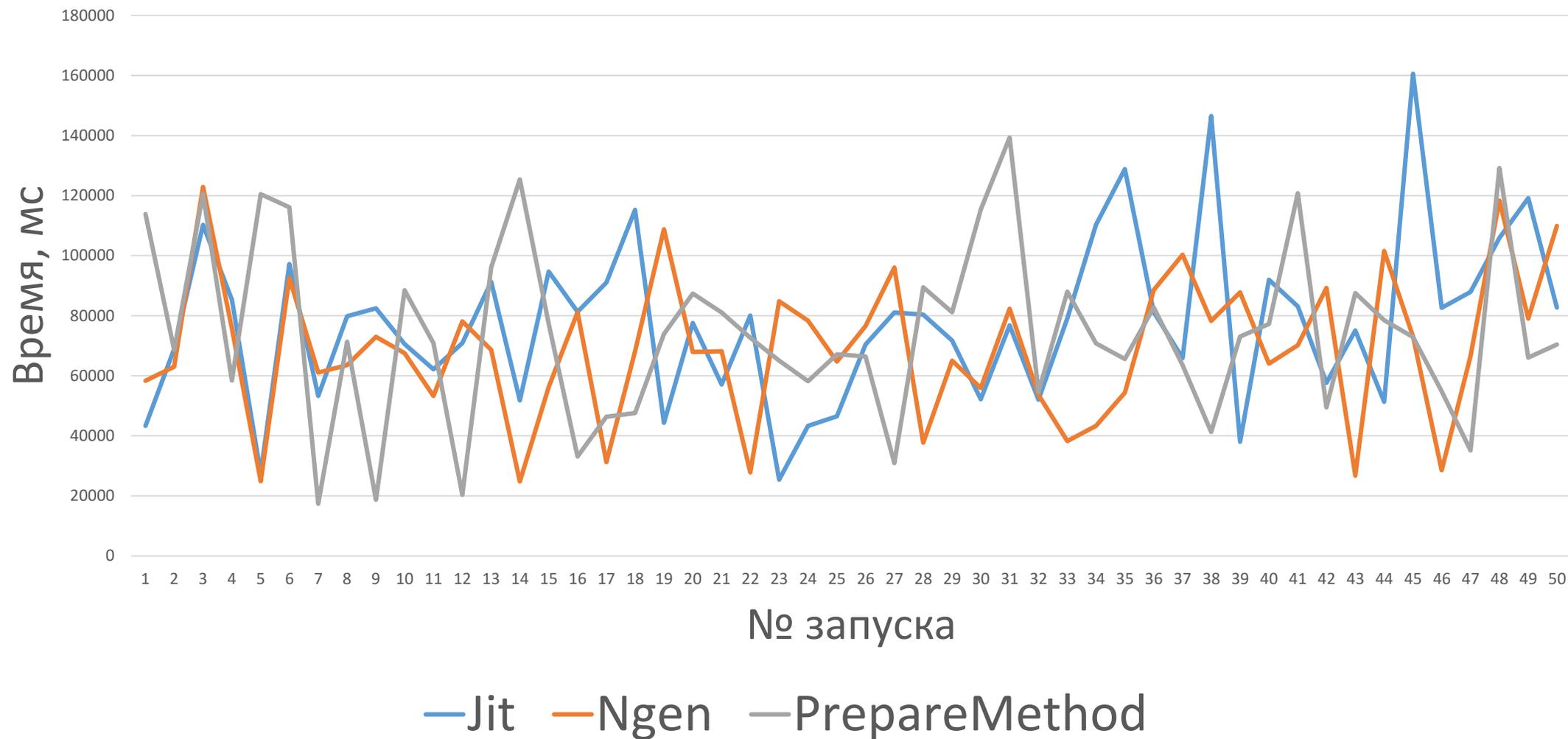
Вариант 3. RuntimeHelpers.PrepareMethod

- Получаем все методы, которые есть в сборке
- Прогоняем все методы через PrepareMethod
- Запускаем!

Вариант 3. RuntimeHelpers.PrepareMethod

- Получаем все методы, которые есть в сборке
- Прогоняем все методы через PrepareMethod
- Запускаем!
- Измеряем!

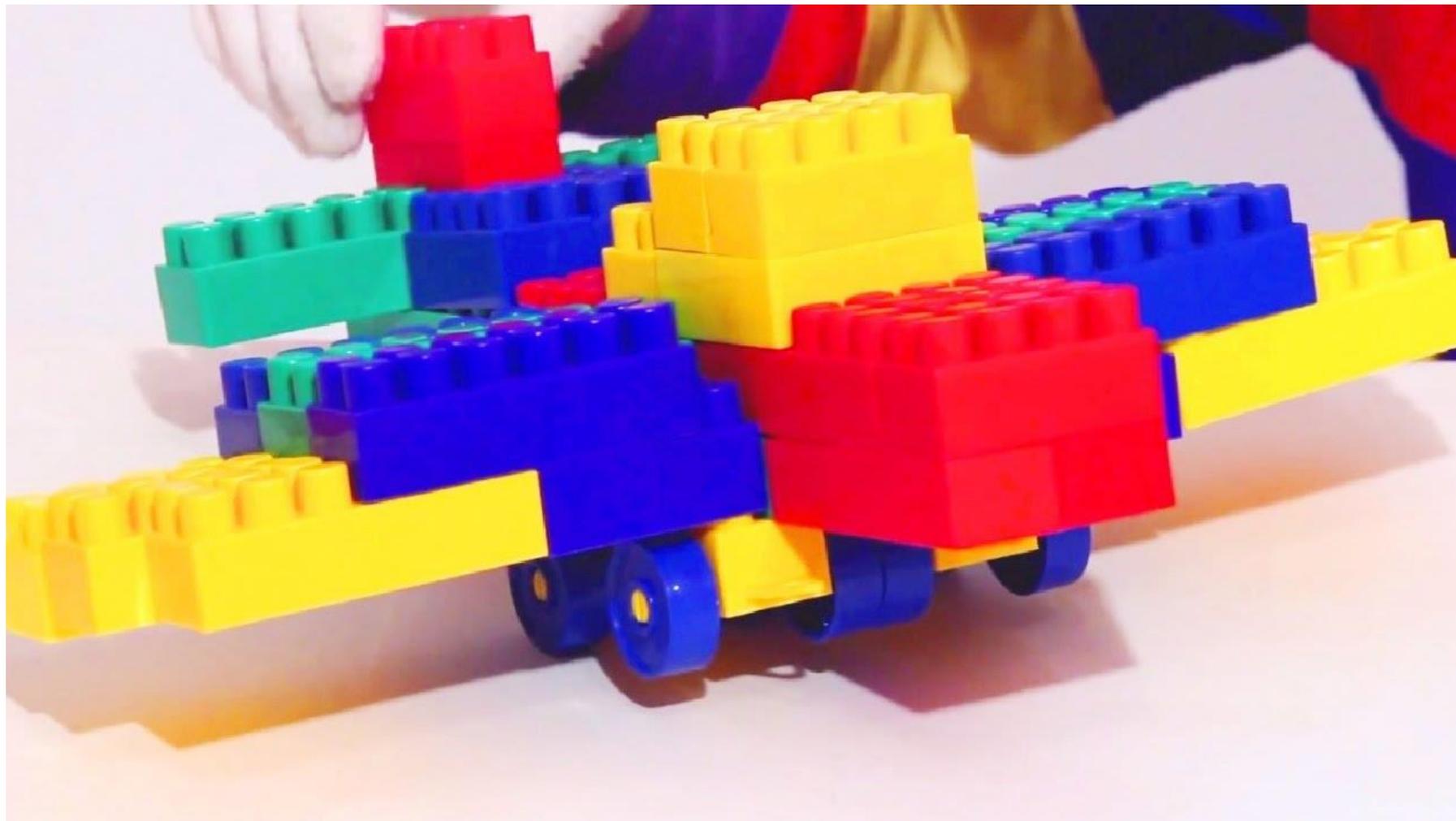
Время вычисления



Среднее время вычисления (мс)

JIT	NGen	Prepare Method
77 671,24	68 992,38	74 410,28

Shared memory и 2 приложения



Состояние памяти без NGen

 ConsoleApp1	46,5%	2,8 МБ
 Console Window Host	0%	2,8 МБ

Shared memory с NGen

 ConsoleApp1	56,9%	2,7 МБ
 Console Window Host	0%	0,9 МБ

С помощью Shared memory несколько процессов
используют одну память

$$2.8 + 2.8 \neq 2.7 + 0.9$$

**А если
приложение «ест»
в 100-100000
больше памяти?**

Минутка дизассемблера!

```
var before =  
    System.Diagnostics.Process  
        .GetCurrentProcess()  
        .VirtualMemorySize64;
```

Минутка дизассемблера! NGen

NGen

```

·····var before := System.Diagnostics.Process.GetCurrentProcess();
00007FFAD5D944B0 · push ······ r15 ···
00007FFAD5D944B2 · push ······ r14 ···
00007FFAD5D944B4 · push ······ r12 ···
00007FFAD5D944B6 · push ······ rdi ···
00007FFAD5D944B7 · push ······ rsi ···
00007FFAD5D944B8 · push ······ rbp ···
00007FFAD5D944B9 · push ······ rbx ···
00007FFAD5D944BA · sub ······ rsp, 20h ···
00007FFAD5D944BE · call ······ 00007FFAD5D920D0 ···
00007FFAD5D944C3 · mov ······ rcx, rax ···
00007FFAD5D944C6 · cmp ······ dword ptr [rcx], ecx ···
00007FFAD5D944C8 · call ······ 00007FFAD5D920D8 ···
00007FFAD5D944CD · mov ······ rsi, rax ···

```

JIT

```

·····var before := System.Diagnostics.Process.GetCurrentProcess();
00007FFA61F10486 · add ······ byte ptr [rax], al ···
00007FFA61F10488 · add ······ byte ptr [rax-73h], cl ···
00007FFA61F1048B · lods ······ byte ptr [rsi] ···
00007FFA61F1048C · and ······ al, 0E0h ···
00007FFA61F1048E · add ······ byte ptr [rax], al ···
00007FFA61F10490 · add ······ byte ptr [rax-73h], cl ···
00007FFA61F10493 · mov ······ ebp, 0FFFFFFF48h ···
00007FFA61F10498 · mov ······ ecx, 2Ch ···
00007FFA61F1049D · xor ······ eax, eax ···
00007FFA61F1049F · rep stos ······ dword ptr [rdi] ···
00007FFA61F104A1 · cmp ······ dword ptr [7FFA61E04588h], 0 ···
00007FFA61F104A8 · je ······ 00007FFA61F104AF ···
00007FFA61F104AA · call ······ 00007FFAC19CF820 ···
00007FFA61F104AF · call ······ 00007FFAAC45BD78 ···
00007FFA61F104B4 · mov ······ qword ptr [rbp-68h], rax ···
00007FFA61F104B8 · mov ······ rcx, qword ptr [rbp-68h] ···
00007FFA61F104BC · cmp ······ dword ptr [rcx], ecx ···
00007FFA61F104BE · call ······ 00007FFAAC476CA0 ···
00007FFA61F104C3 · mov ······ qword ptr [rbp-20h], rax ···
00007FFA61F104C7 · mov ······ rcx, qword ptr [rbp-20h] ···
00007FFA61F104CB · mov ······ qword ptr [rbp-10h], rcx ···

```

Минутка дизассемблера! NGen

```
·push ······ r15 ··  
·push ······ r14 ··  
·push ······ r12 ··  
·push ······ rdi ··  
·push ······ rsi ··  
·push ······ rbp ··  
·push ······ rbx ··  
·sub ······ rsp, 20h ··  
·call ······ 00007FFAD5D920D0 ··  
·mov ······ rcx, rax ··  
·cmp ······ dword ptr [rcx], ecx ··  
·call ······ 00007FFAD5D920D8 ··  
·mov ······ rsi, rax ··
```

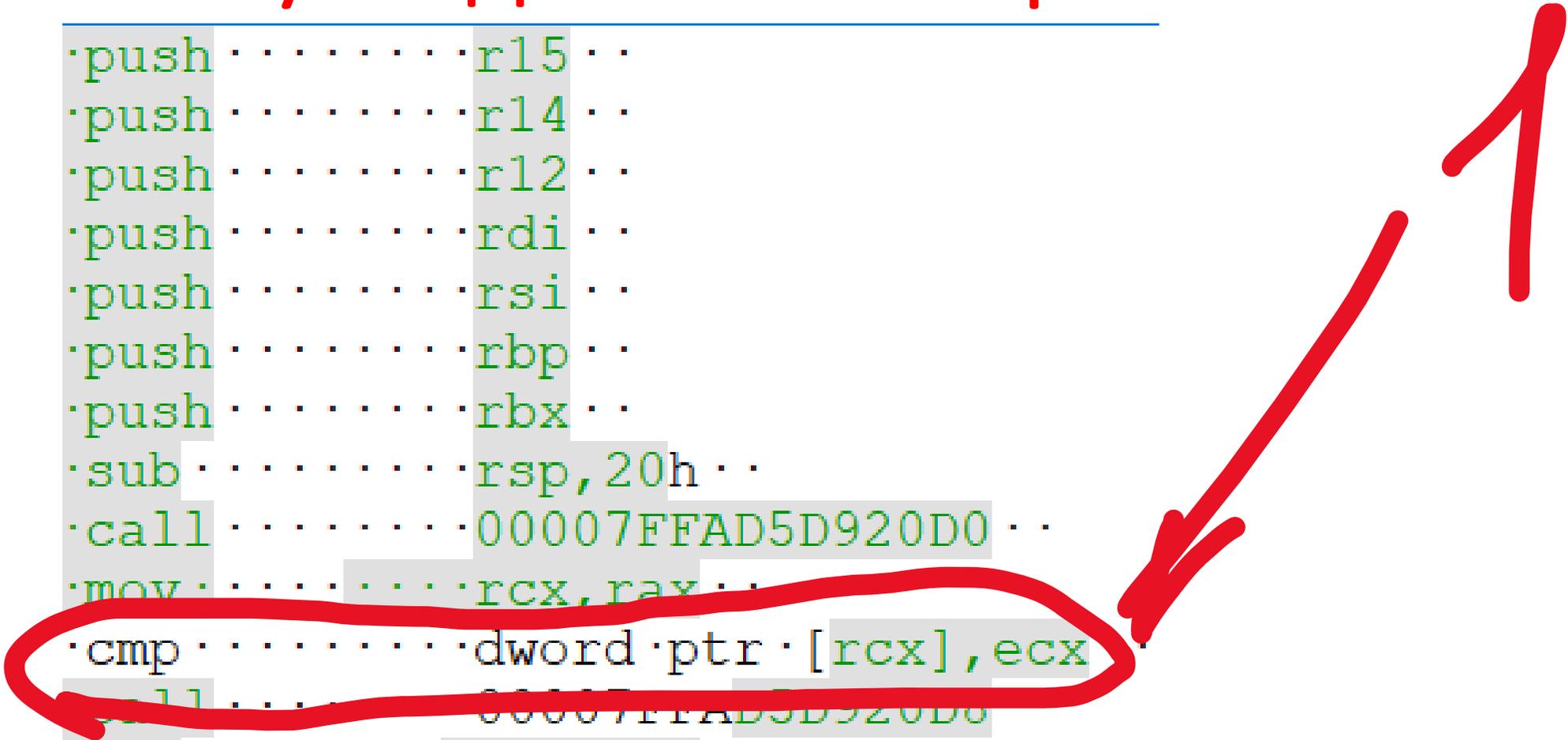
Минутка дизассемблера! NGen

```
·push ······ r15 ···  
·push ······ r14 ···  
·push ······ r12 ···  
·push ······ rdi ···  
·push ······ rsi ···  
·push ······ rbp ···  
·push ······ rbx ···  
·sub ······ rsp, 20h ···  
·call ······ 00007FFAD5D920D0 ···  
·mov ······ rcx, rax ···  
·cmp ······ dword ptr [rcx], ecx ···  
·call ······ 00007FFAD5D920D8 ···  
·mov ······ rsi, rax ···
```

← СТЕК

Минутка дизассемблера! NGen

```
·push ..... r15 ..  
·push ..... r14 ..  
·push ..... r12 ..  
·push ..... rdi ..  
·push ..... rsi ..  
·push ..... rbp ..  
·push ..... rbx ..  
·sub ..... rsp, 20h ..  
·call ..... 00007FFAD5D920D0 ..  
·mov ..... rcx, rax ..  
·cmp ..... dword ptr [rcx], ecx ..  
·call ..... 00007FFAD5D920D0 ..  
·mov ..... rsi, rax ..
```

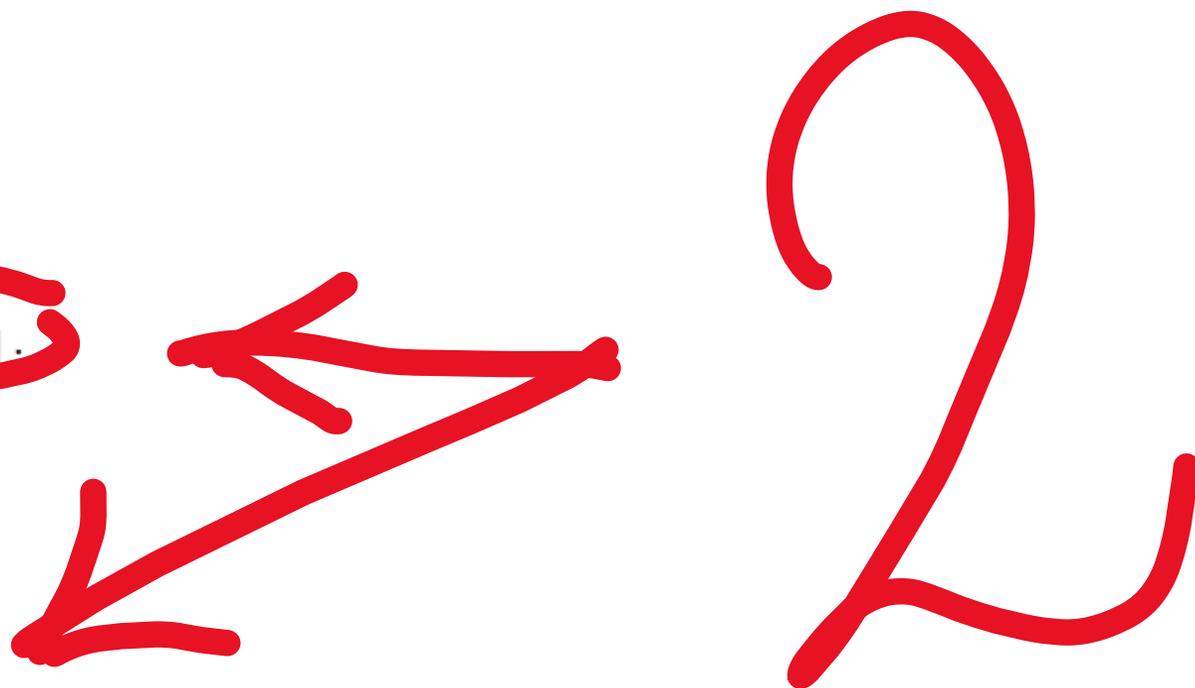


Минутка дизассемблера! JIT

```
·add ······ ·byte ·ptr ·[rax], al ··  
·add ······ ·byte ·ptr ·[rax-73h], cl ··  
·lods ······ ·byte ·ptr ·[rsi] ··  
·and ······ ·al, 0E0h ··  
·add ······ ·byte ·ptr ·[rax], al ··  
·add ······ ·byte ·ptr ·[rax-73h], cl ··  
·mov ······ ·ebp, 0FFFFFF48h ··  
·mov ······ ·ecx, 2Ch ··  
·xor ······ ·eax, eax ··  
·rep ·stos ··· ·dword ·ptr ·[rdi] ··  
·cmp ······ ·dword ·ptr ·[7FFA61E04588h], 0 ··  
·je ······ ·00007FFA61F104AF ··  
·call ······ ·00007FFAC19CF820 ··  
·call ······ ·00007FFAAC45BD78 ··  
·mov ······ ·qword ·ptr ·[rbp-68h], rax ··  
·mov ······ ·rcx, qword ·ptr ·[rbp-68h] ··  
·cmp ······ ·dword ·ptr ·[rcx], ecx ··  
·call ······ ·00007FFAAC476CA0 ··  
·mov ······ ·qword ·ptr ·[rbp-20h], rax ··  
·mov ······ ·rcx, qword ·ptr ·[rbp-20h] ··  
·mov ······ ·qword ·ptr ·[rbp-10h], rcx ··
```

Минутка дизассемблера! JIT

```
·add ······ ·byte·ptr·[rax],al··  
·add ······ ·byte·ptr·[rax-73h],cl··  
·lods ······ ·byte·ptr·[rsi]··  
·and ······ ·al,0E0h··  
·add ······ ·byte·ptr·[rax],al··  
·add ······ ·byte·ptr·[rax-73h],cl··  
·mov ······ ·ebp,0FFFFFF48h··  
·mov ······ ·ecx,2Ch··  
·xor ······ ·eax,eax··  
·mov ······ ·word·ptr·[rdi]··  
·cmp ······ ·dword·ptr·[7FFA61E04588h],0··  
·jz ······ 00007FFA61F104AF··  
·call ······ 00007FFAC19CF820··  
·call ······ 00007FFAAC45BD78··  
·mov ······ ·qword·ptr·[rbp-68h],rax··  
·mov ······ ·rcx,qword·ptr·[rbp-68h]··  
·cmp ······ ·dword·ptr·[rcx],ecx··  
·call ······ 00007FFAAC476CA0··  
·mov ······ ·qword·ptr·[rbp-20h],rax··  
·mov ······ ·rcx,qword·ptr·[rbp-20h]··  
·mov ······ ·qword·ptr·[rbp-10h],rcx··
```



Минутка дизассемблера!

NGen

```

·····var ·before := ·System.Diagnostics.Process.GetCurr
00007FFAD5D944B0 ·push ······r15··
00007FFAD5D944B2 ·push ······r14··
00007FFAD5D944B4 ·push ······r12··
00007FFAD5D944B6 ·push ······rdi··
00007FFAD5D944B7 ·push ······rsi··
00007FFAD5D944B8 ·push ······rbp··
00007FFAD5D944B9 ·push ······rbx··
00007FFAD5D944BA ·sub ······rsp,20h··
00007FFAD5D944BE ·call ······00007FFAD5D920D0··
00007FFAD5D944C3 ·mov ······rcx, rax··
00007FFAD5D944C6 ·cmp ······dword ptr ·[rcx], ecx··
00007FFAD5D944C8 ·call ······00007FFAD5D920D8··
00007FFAD5D944CD ·mov ······rsi, rax··

```

JIT

```

·······var ·before := ·System.Diagnostics.Process.GetCurrent
00007FFA61F10486 ·add ·······byte ptr ·[rax], al··
00007FFA61F10488 ·add ·······byte ptr ·[rax-73h], cl··
00007FFA61F1048B ·lods ·······byte ptr ·[rsi]··
00007FFA61F1048C ·and ·······al, 0E0h··
00007FFA61F1048E ·add ·······byte ptr ·[rax], al··
00007FFA61F10490 ·add ·······byte ptr ·[rax-73h], cl··
00007FFA61F10493 ·mov ·······ebp, 0FFFFFFF48h··
00007FFA61F10498 ·mov ·······ecx, 2Ch··
00007FFA61F1049D ·xor ·······eax, eax··
00007FFA61F1049F ·rep stos ·······dword ptr ·[rdi]··
00007FFA61F104A1 ·cmp ·······dword ptr ·[7FFA61E04588h], 0··
00007FFA61F104A8 ·je ······00007FFA61F104AF··
00007FFA61F104AA ·call ······00007FFAC19CF820··
00007FFA61F104AF ·call ······00007FFAAC45BD78··
00007FFA61F104B4 ·mov ·······qword ptr ·[rbp-68h], rax··
00007FFA61F104B8 ·mov ·······rcx, qword ptr ·[rbp-68h]··
00007FFA61F104BC ·cmp ·······dword ptr ·[rcx], ecx··
00007FFA61F104BE ·call ······00007FFAAC476CA0··
00007FFA61F104C3 ·mov ·······qword ptr ·[rbp-20h], rax··
00007FFA61F104C7 ·mov ·······rcx, qword ptr ·[rbp-20h]··
00007FFA61F104CB ·mov ·······qword ptr ·[rbp-10h], rcx··

```

NGen так мало кода генерирует...

Почему же мы его так редко
используем?

NGen может генерировать код
хуже!

Минутка дизассемблера!

```
for (int i = 0; i < FibonacciCalculations; i++)
```

Рассмотрим выход из цикла!

Минутка дизассемблера!

NGen

```

.....for·(int·i·:=·0;·i·<·FibonacciCalculations;·i++)
00007FFAD5D945FE·inc·.....eax·
00007FFAD5D94600·cmp·.....eax,1Eh·
00007FFAD5D94603·j1·.....00007FFAD5D945F6·
00007FFAD5D94605·jmp·.....00007FFAD5D94616·
00007FFAD5D94607·movsxd·.....rdx,eax·
00007FFAD5D9460A·mov·.....rdx,qword·ptr·[rbx+rdx*8+10h]·
00007FFAD5D9460F·inc·.....eax·
00007FFAD5D94611·cmp·.....eax,1Eh·
00007FFAD5D94614·j1·.....00007FFAD5D94607·

```

JIT

```

.....for·(int·i·:=·0;·i·<·FibonacciCalculations;·i++)
00007FFA61F106D4·mov·.....eax,dword·ptr·[rbp-18h]·
00007FFA61F106D7·inc·.....eax·
00007FFA61F106D9·mov·.....dword·ptr·[rbp-18h],eax·
00007FFA61F106DC·mov·.....eax,dword·ptr·[rbp-18h]·
00007FFA61F106DF·cmp·.....eax,1Eh·
00007FFA61F106E2·j1·.....00007FFA61F106B4·

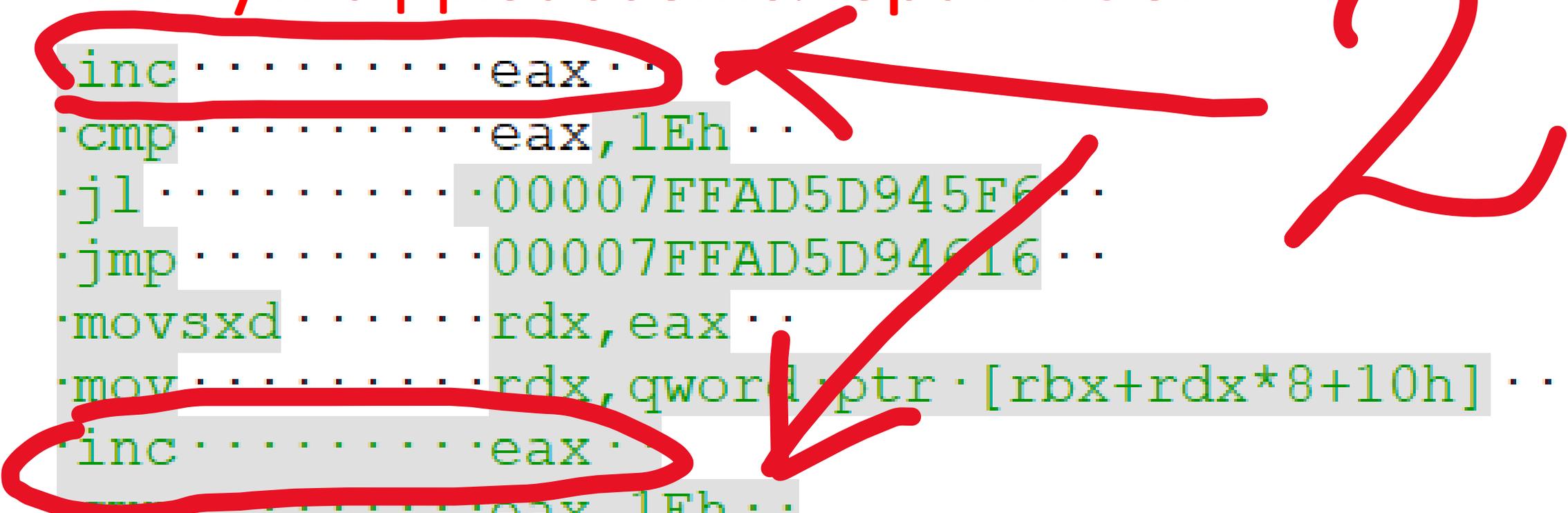
```

Минутка дизассемблера! NGen

```
· inc ······ · eax ··  
· cmp ······ · eax, 1Eh ··  
· jl ······ · 00007FFAD5D945F6 ··  
· jmp ······ · 00007FFAD5D94616 ··  
· movsxd ······ rdx, eax ··  
· mov ······ rdx, qword ptr · [rbx+rdx*8+10h] ··  
· inc ······ · eax ··  
· cmp ······ · eax, 1Eh ··  
· jl ······ · 00007FFAD5D94607 ··
```


Минутка дизассемблера! NGen

```
inc ..... eax .....  
cmp ..... eax, 1Eh .....  
j1 ..... 00007FFAD5D945F6 .....  
jmp ..... 00007FFAD5D94616 .....  
movsxd ..... rdx, eax .....  
mov ..... rdx, qword ptr [rbx+rdx*8+10h] .....  
inc ..... eax .....  
cmp ..... eax, 1Eh .....  
j1 ..... 00007FFAD5D94607 .....
```

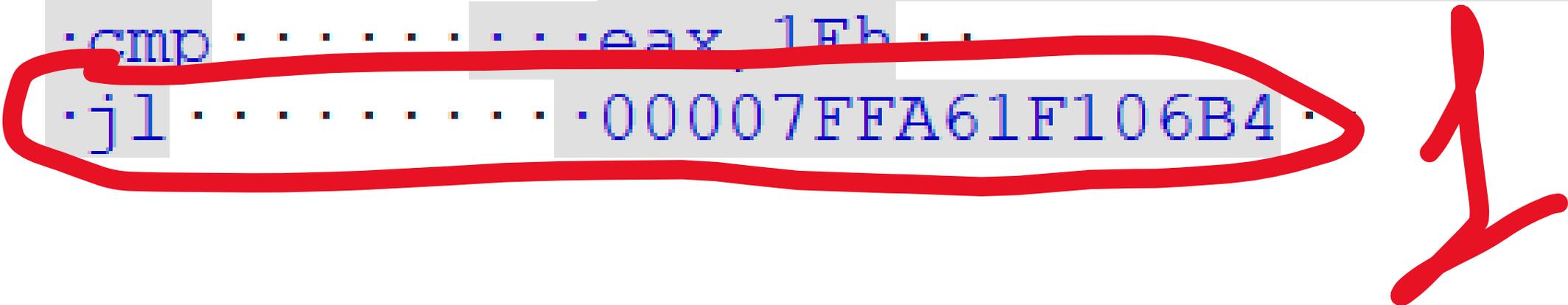


Минутка дизассемблера! JIT

```
·mov ······ ·eax, dword ·ptr ·[rbp-18h] ··  
·inc ······ ·eax ··  
·mov ······ ·dword ·ptr ·[rbp-18h], eax ··  
·mov ······ ·eax, dword ·ptr ·[rbp-18h] ··  
·cmp ······ ·· ·eax, 1Eh ··  
·jl ······ ·· ·00007FFA61F106B4 ··
```

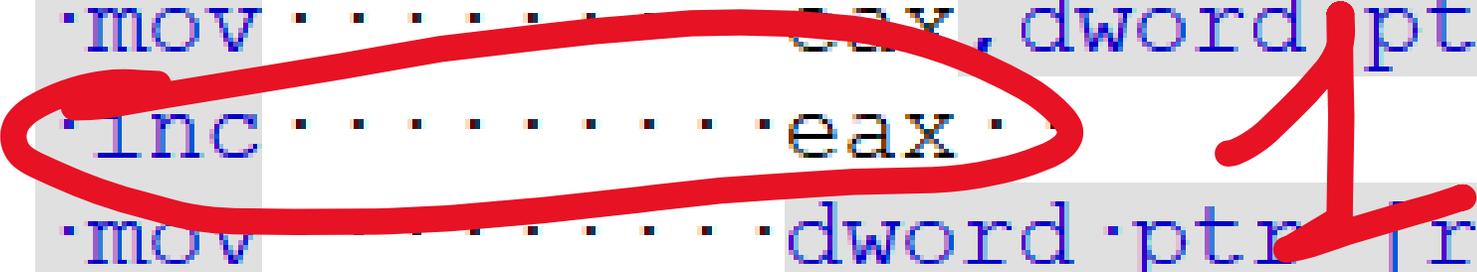
Минутка дизассемблера! JIT

```
·mov ······ ·eax, dword ·ptr ·[rbp-18h] ··  
·inc ······ ·eax ··  
·mov ······ ·dword ·ptr ·[rbp-18h], eax ··  
·mov ······ ·eax, dword ·ptr ·[rbp-18h] ··  
·cmp ······ ·eax, 1Fh ··  
·jl ······ ·00007FFA61F106B4 ··
```



Минутка дизассемблера! JIT

```
·mov ······ eax, dword ptr · [rbp-18h] ··  
·inc ······ eax ··  
·mov ······ dword ptr [rbp-18h], eax ··  
·mov ······ eax, dword ptr · [rbp-18h] ··  
·cmp ······ ·· · eax, 1Eh ··  
·jl ······ ·· · 00007FFA61F106B4 ··
```



Минутка дизассемблера!

NGen

```
.....for(int i:=0; i< FibonacciCalculations; i++)
00007FFAD5D945FE inc .....eax..
00007FFAD5D94600 cmp .....eax, 1Eh..
00007FFAD5D94603 j1 .....00007FFAD5D945F6..
00007FFAD5D94605 jmp .....00007FFAD5D94616..
00007FFAD5D94607 movsxd .....rdx, eax..
00007FFAD5D9460A mov .....rdx, qword ptr [rbx+rdx*8+10h]..
00007FFAD5D9460F inc .....eax..
00007FFAD5D94611 cmp .....eax, 1Eh..
00007FFAD5D94614 j1 .....00007FFAD5D94607..
```

JIT

```
.....for(int i:=0; i< FibonacciCalculations; i++)
00007FFA61F106D4 mov .....eax, dword ptr [rbp-18h]..
00007FFA61F106D7 inc .....eax..
00007FFA61F106D9 mov .....dword ptr [rbp-18h], eax..
00007FFA61F106DC mov .....eax, dword ptr [rbp-18h]..
00007FFA61F106DF cmp .....eax, 1Eh..
00007FFA61F106E2 j1 .....00007FFA61F106B4..
```

Выводы

- Для оптимизации работы нескольких процессов с памятью нам подходит NGen

Выводы

- Для оптимизации работы нескольких процессов с памятью нам подходит NGen
- На производительности NGen в ряде случаев так же окажется быстрее JIT
 - пример с VirtualMemorySize

Выводы. NGen

- Для оптимизации работы нескольких процессов с памятью нам подходит NGen
- На производительности NGen в ряде случаев так же окажется быстрее JIT
- Но!!! Необходимо учитывать длительность работы приложения и повторяющиеся операции.

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
 - Использовать не стоит, так как таким образом вы «жестко» привязываете сборку. Стоит ее переместить – и она больше недоступна.

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
 - Будут выполняться вызовы JIT

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
- `ni.dll/ni.exe` хранятся рядом
 - NGen генерируют файлы машинного образа `AssemblyName.ni.dll`, `FileName.ni.exe`

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
- `ni.dll/ni.exe` хранятся рядом
- Рассинхронизация NGen-файлов
 - Файлы NGen нужно обновлять после каждой сборки

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
- `ni.dll/ni.exe` хранятся рядом
- Рассинхронизация NGen-файлов
- Модификация адресов сборок `ni` и `не ni`
 - Если функции нет в машинном образе – необходимо сопоставить адрес вызова этой функции в `ni`-файле с `не ni`-файлом

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
- `ni.dll/ni.exe` хранятся рядом
- Рассинхронизация NGen-файлов
- Модификация адресов сборок `ni` и `не ni`
- То, что NGen не сможет скомпилировать, будет выполнять JIT

Ограничения. NGen

- `Assembly.LoadFrom(...)`, `Null-context` и `Assembly.LoadFile`
- `Reflection-emit`
- `ni.dll/ni.exe` хранятся рядом
- Рассинхронизация NGen-файлов
- Модификация адресов сборок `ni` и `не ni`
- То, что NGen не сможет скомпилировать, будет выполнять JIT
- Для `strongly-named` сборок выгоды не будет

AOT для .NET Core. CrossGen

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, culture=neutral, PublicKeyToken=b77a5c56
System.Xml, Version=4.0.0.0, culture=neutral, PublicKeyToken=b77a5c561
System.Xml.XPath.XDocument, Version=4.0.2.0, culture=neutral, Publicke
System.Xml.ReaderWriter, Version=4.1.0.0, culture=neutral, PublicKeyTo
System.Xml.XmlDocument, Version=4.0.2.0, culture=neutral, PublicKeyTok
System.Xml.Linq, Version=4.0.0.0, culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, culture=neutral, PublicKeyToken=
windowsBase, Version=4.0.0.0, culture=neutral, PublicKeyToken=31bf3856
wpfsurface, Version=15.0.0.0, culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core
- Инструмент, который генерирует из MSIL нативный код

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core
- Инструмент, который генерирует из MSIL нативный код
- CrossGen – часть **CoreCLR**

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core
- Инструмент, который генерирует из MSIL нативный код
- CrossGen – часть **CoreCLR**
- Чтобы использовать CoreCLR нужна сборка **System.Private.CoreLib.ni.dll** (она генерируется с помощью CrossGen из System.Private.CoreLib.dll)

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core
- Инструмент, который генерирует из MSIL нативный код
- CrossGen – часть **CoreCLR**
- Чтобы использовать CoreCLR нужна сборка **System.Private.CoreLib.ni.dll** (она генерируется с помощью CrossGen из System.Private.CoreLib.dll)
- Создает AssemblyName.ni.dll или ProjectName.ni.exe

АОТ для .NET Core. CrossGen

- Аналог NGen для .NET Core
- Инструмент, который генерирует из MSIL нативный код
- CrossGen – часть **CoreCLR**
- Чтобы использовать CoreCLR нужна сборка **System.Private.CoreLib.ni.dll** (она генерируется с помощью CrossGen из System.Private.CoreLib.dll)
- Создает AssemblyName.ni.dll или ProjectName.ni.exe
- Можно собрать проект под Mac, Linux, Windows

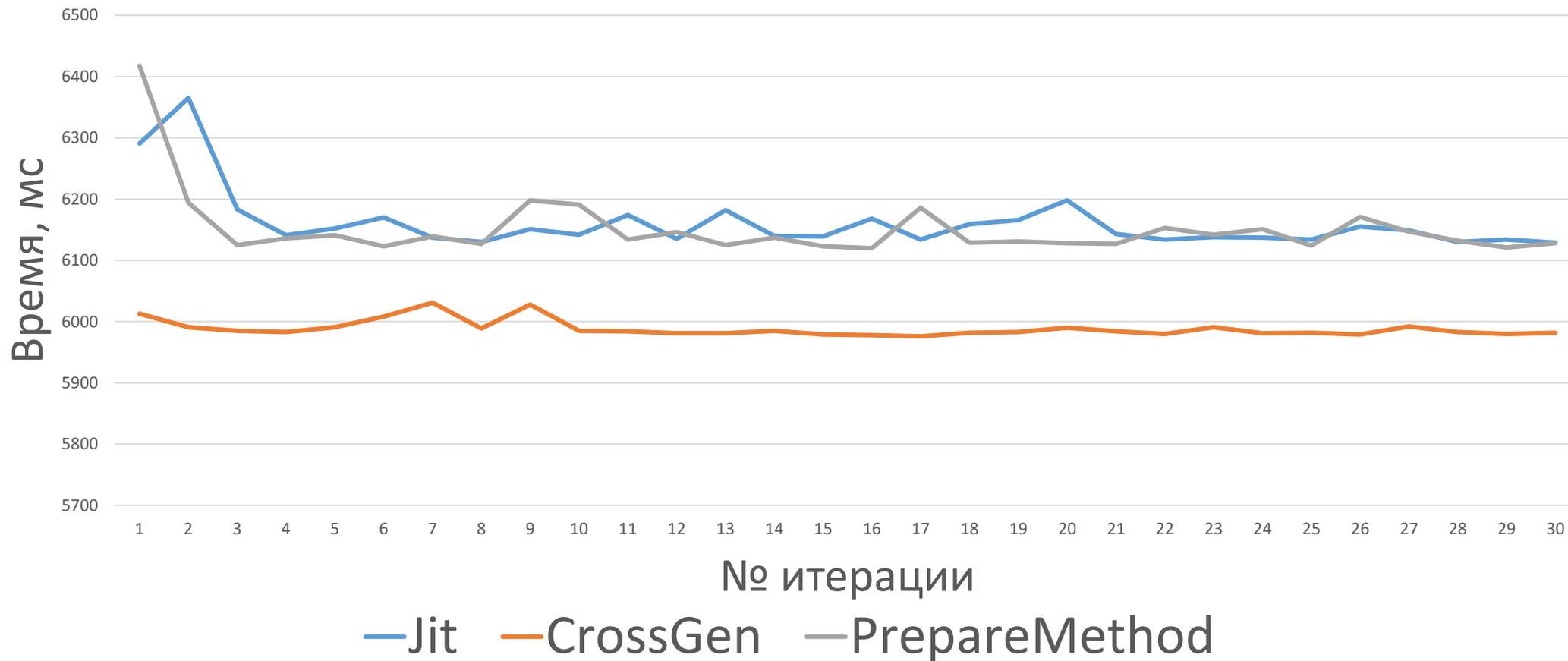
Демо

- Цель – оптимизировать процесс шифрования паролей
- Инструменты:
 - CryptoSharp

DEMO

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
Sys 61
Sys Ke
Sys To
Sys ok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
WindowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
WpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

JIT, CrossGen, PrepareMethod 1 итерация = 30 000 паролей



Среднее время вычисления (мс)

JIT	CrossGen	Prepare Method
6 161,333	5 988,567	6 151,567

Выводы. CrossGen

- CrossGen – достаточно мощный инструмент, который позволяет оптимизировать такие «тяжелые» вещи, как шифрование паролей/ключей, оптимизация вычислений

Выводы. CrossGen

- CrossGen – достаточно мощный инструмент, который позволяет оптимизировать такие «тяжелые» вещи, как шифрование паролей/ключей, оптимизация вычислений
- Однако, как и NGen имеет те же недостатки с пi-файлами

Ограничения. CrossGen

- Аналог NGen для .NET Core

Ограничения. CrossGen

- Аналог NGen для .NET Core
- ni.dll/ni.exe хранятся рядом

Ограничения. CrossGen

- Аналог NGen для .NET Core
- ni.dll/ni.exe хранятся рядом
- Рассинхронизация ni-файлов

Ограничения. CrossGen

- Аналог NGen для .NET Core
- ni.dll/ni.exe хранятся рядом
- Рассинхронизация ni-файлов
- Модификация адресов сборок ni и не ni

Ограничения. CrossGen

- Аналог NGen для .NET Core
- ni.dll/ni.exe хранятся рядом
- Рассинхронизация ni-файлов
- Модификация адресов сборок ni и не ni
- То, что CrossGen не сможет скомпилировать, будет выполнять RyuJIT

Is possible remove IL code from xxx.ni.dll compiled by CrossGen? #15347

Open

Petermarcu opened this issue on 2 Dec 2017 · 2 comments



Petermarcu commented on 2 Dec 2017

Member



@hzexe commented on [Sat Dec 02 2017](#)

To software protection, like .NET Native.



Petermarcu referenced this issue in [dotnet/core](#) on 2 Dec 2017

Is possible remove IL code from xxx.ni.dll compiled by CrossGen? #1125

Closed



jkotas commented on 2 Dec 2017 · edited

Member



There is no command line option to do that today, but it should not be hard to add one. We will be happy to accept PR to add it.

Of course, the features and tools that depend on the IL being present are not going to work with the IL removed.



1

Поговорим о UWP

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561
System.Xml.XPath.XDocument, Version=4.0.2.0, Culture=neutral, PublicKe
System.Xml.ReaderWriter, Version=4.1.0.0, Culture=neutral, PublicKeyTo
System.Xml.XmlDocument, Version=4.0.2.0, Culture=neutral, PublicKeyTok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
windowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
wpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

Поговорим о UWP

- UWP – Universal Windows Platform

Поговорим о UWP

- UWP – Universal Windows Platform
- Истинное воплощение Ahead-of-time компиляции

Поговорим о UWP

- UWP – Universal Windows Platform
- Истинное воплощение Ahead-of-time компиляции
- Все приложения UWP для Windows Store должны отвечать требованиям **.NET Native**

Поговорим о UWP

- UWP – Universal Windows Platform
- Истинное воплощение Ahead-of-time компиляции
- Все приложения UWP для Windows Store должны отвечать требованиям **.NET Native**
- При сборке Release-версии запускается инструмент для генерации машинного образа

Поговорим о UWP и .NET Native

- [.NET Native](#) – это технология предварительной компиляции, используемая при создании UWP в Visual Studio

Поговорим о UWP и .NET Native

- [.NET Native](#) – это технология предварительной компиляции, используемая при создании UWP в Visual Studio
- Инструменты .NET Native компилируют IL-библиотеки с управляемым кодом в нативные библиотеки.

Поговорим о UWP и .NET Native

- [.NET Native](#) – это технология предварительной компиляции, используемая при создании UWP в Visual Studio
- Инструменты .NET Native компилируют IL-библиотеки с управляемым кодом в нативные библиотеки.
- Приложения автоматически компилируются в нативный код прежде, чем они попадут на конечное устройство.

Поговорим о UWP и .NET Native

- [.NET Native](#) – это технология предварительной компиляции, используемая при создании UWP в Visual Studio
- Инструменты .NET Native компилируют IL-библиотеки с управляемым кодом в нативные библиотеки.
- Приложения автоматически компилируются в нативный код прежде, чем они попадут на конечное устройство.
- В основе .NET Native - CoreCLR

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта
- До 40% повышения скорости горячего старта

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта
- До 40% повышения скорости горячего старта
- Уменьшенное потребление памяти при компиляции в машинный код

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта
- До 40% повышения скорости горячего старта
- Уменьшенное потребление памяти при компиляции в машинный код
- Нет зависимости от десктопного .NET Runtime при установке

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта
- До 40% повышения скорости горячего старта
- Уменьшенное потребление памяти при компиляции в машинный код
- Нет зависимости от десктопного .NET Runtime при установке
- Debug - IL-код поверх CoreCLR, упакованного в ваше приложение

Поговорим о UWP и .NET Native

- До 60% повышения скорости холодного старта
- До 40% повышения скорости горячего старта
- Уменьшенное потребление памяти при компиляции в машинный код
- Нет зависимости от десктопного .NET Runtime при установке
- Debug - IL-код поверх CoreCLR, упакованного в ваше приложение
- Release - .NET Native

Hello, DotNext! I'm a UWP-App!



Как настроен UWP в Release

The screenshot shows the Visual Studio interface for a UWP application named "UWP_Demo". The "Build" tab is selected in the left-hand menu. The main area displays the "Configuration" and "Platform" settings, both set to "Active (Release)" and "Active (x64)" respectively. Below these, the "General" section is expanded, showing the following settings:

- Conditional compilation symbols: NETFX_CORE;WINDOWS_UWP;CODE_ANALYSIS
- Define DEBUG constant
- Define TRACE constant
- Platform target: x64
- Prefer 32-bit
- Compile with .NET Native tool chain
- Allow unsafe code
- Enable static analysis for .NET Native
- Optimize code

Как отладить UWP с .NET Native

Compile with .NET Native tool chain

Как отладить UWP с .NET Native

Compile with .NET Native tool chain

Optimize code



Поговорим о UWP. Делаем дамп

UWP_Demo - Microsoft Visual Studio

File Edit View Project Build Debug Team Design Format Tools Test R Tools Analyze Window Help

Release x86 Local Machine

Instances of StringMap16

Snapshot #1 Heap..._Demo.exe (54.46s) X MainPage.xaml.cs MainPage.g.cs MainPage.xaml

Snapshot does not reflect current program state. Values not available.

Instance	Size (Bytes)	Inclusive Size (Bytes)	
<0x3E45EF0>	24	7 628	
<0x3E45EC0>	24	6 308	
<0x3E45ED8>	24	2 924	
<0x3E03588>	24	2 420	
<0x3E035A0>	24	1 124	
<0x3E03600>	24	764	
<0x3E45F38>	24	500	
<0x3E035B8>	24	188	
<0x3E035D0>	24	24	
<0x3E035E8>	24	24	

Paths to Root | Referenced Objects

Instance	
StringMap16	<0x3E45EF0>
__Interop.McgCurrentModule	<0x3E45CE8> [Local Variable]

Diagnostics Tools

Select Tools Zoom In Zoom Out

Diagnostics session: 2:04 minutes (2:04 min selec...)

1:40min

Events

Process Memory (MB)

CPU (% of all processors)

Take Snapshot View Heap Delete

Heap Size (Diff)

ID	Time	Process	Heap Size (Diff)
1	54.46s	1 456 (n/a)	242,23 KB (n/a)

Solution Explorer

Search Solution Explorer (Ctrl+J)

Solution 'UWP_Demo' (1 project)

UWP_Demo (Universal Windows)

- Connected Services
- Properties
- References
- Assets
- App.xaml
- App.xaml.cs
- MainPage.xaml
- MainPage.xaml.cs
- Package.appxmanifest
- UWP_Demo_TemporaryKey.pfx

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Code Description Project File

Output

Ready Add to Source Control

Поговорим о UWP. Строки и адресация

<0x3E45EF0>	24	7 628	
<0x3E45EC0>	24	6 308	
<0x3E45ED8>	24	2 924	
<0x3E03588>	24	2 420	
<0x3E035A0>	24	1 124	
<0x3E03600>	24	764	
<0x3E45F38>	24	500	
<0x3E035B8>	24	188	
<0x3E035D0>	24	24	
<0x3E035E8>	24	24	

Минутка дизассемблера!

```
this.InitializeComponent();
```

Минутка дизассемблера!

.NET Native

JIT

```
.this.InitializeComponent();  
00007FFC28E1C4BF ··cmp·····byte·ptr·[rsi+68h],0··  
00007FFC28E1C4C3 ··jne·····00007FFC28E1C50E··  
00007FFC28E1C4C5 ··mov·····byte·ptr·[rsi+68h],1··  
00007FFC28E1C4C9 ··mov·····rcx,qword·ptr·[7FFC28DE5A48h]··  
00007FFC28E1C4D0 ··call·····qword·ptr·[7FFC28DE6EC8h]··  
00007FFC28E1C4D6 ··mov·····rbx,rax··  
00007FFC28E1C4D9 ··lea·····rdx,[7FFC28DE9620h]··  
00007FFC28E1C4E0 ··mov·····rcx,rax··  
00007FFC28E1C4E3 ··call·····qword·ptr·[7FFC28DE4040h]
```

```
·····this.InitializeComponent();  
00007FFBBA6594F2 ··mov·····rcx,rsi··  
00007FFBBA6594F5 ··call·····00007FFBBA655470··
```

Минутка дизассемблера! .NET Native

```
cmp . . . . . byte ptr [rsi+68h], 0 . .  
jne . . . . . 00007FFC28E1C50E . .  
mov . . . . . byte ptr [rsi+68h], 1 . .  
mov . . . . . rcx, qword ptr [7FFC28DE5A48h] . .  
call . . . . . qword ptr [7FFC28DE6EC8h] . .  
mov . . . . . rbx, rax . .  
lea . . . . . rdx, [7FFC28DE9620h] . .  
mov . . . . . rcx, rax . .  
call . . . . . qword ptr [7FFC28DE4040h]
```


Минутка дизассемблера! JIT

```
.....this.InitializeComponent();  
00007FFBBA6594F2 ··mov·····rcx,rsi··  
00007FFBBA6594F5 ··call·····00007FFBBA655470··
```

Минутка дизассемблера! JIT

```
.....this.InitializeComponent();  
00007FFBBA6594F2 ··mov·····rcx,rsi··  
00007FFBBA6594F5 ··call·····00007FFBBA655470··
```

compile

Минутка дизассемблера!

.NET Native

JIT

```
.this.InitializeComponent();  
00007FFC28E1C4BF ··cmp·····byte·ptr·[rsi+68h],0··  
00007FFC28E1C4C3 ··jne·····00007FFC28E1C50E··  
00007FFC28E1C4C5 ··mov·····byte·ptr·[rsi+68h],1··  
00007FFC28E1C4C9 ··mov·····rcx,qword·ptr·[7FFC28DE5A48h]··  
00007FFC28E1C4D0 ··call·····qword·ptr·[7FFC28DE6EC8h]··  
00007FFC28E1C4D6 ··mov·····rbx,rax··  
00007FFC28E1C4D9 ··lea·····rdx,[7FFC28DE9620h]··  
00007FFC28E1C4E0 ··mov·····rcx,rax··  
00007FFC28E1C4E3 ··call·····qword·ptr·[7FFC28DE4040h]
```

```
·····this.InitializeComponent();  
00007FFBBA6594F2 ··mov·····rcx,rsi··  
00007FFBBA6594F5 ··call·····00007FFBBA655470··
```

CoreRT

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561
System.Xml.XPath.XDocument, Version=4.0.2.0, Culture=neutral, PublicKe
System.Xml.ReaderWriter, Version=4.1.0.0, Culture=neutral, PublicKeyTo
System.Xml.XmlDocument, Version=4.0.2.0, Culture=neutral, PublicKeyTok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
windowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
wpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

АОТ для .NET Core. CoreRT. **WARNING**

- .NET Core runtime оптимизированный для АОТ

AOT для .NET Core. CoreRT. **WARNING**

- .NET Core runtime оптимизированный для AOT
- Компиляция приложения в 1 нативный исполняемый файл

АОТ для .NET Core. CoreRT. **WARNING**

- .NET Core runtime оптимизированный для АОТ
- Компиляция приложения в 1 нативный исполняемый файл
- Легко разворачивается

АОТ для .NET Core. CoreRT. **WARNING**

- .NET Core runtime оптимизированный для АОТ
- Компиляция приложения в 1 нативный исполняемый файл
- Легко разворачивается
- Представляет собой сборку ILCompiler

AOT для .NET Core. CoreRT. **WARNING**

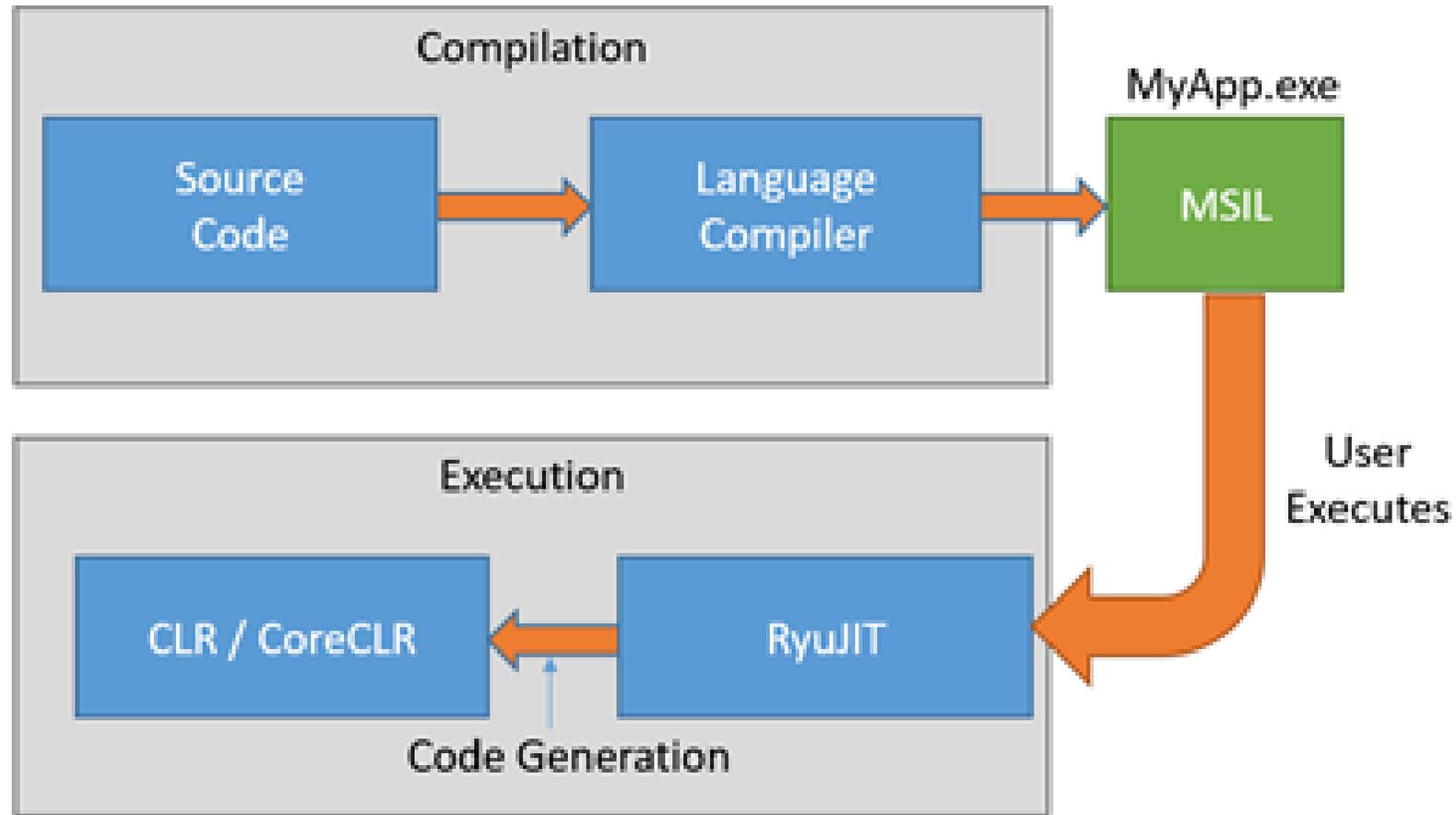
- .NET Core runtime оптимизированный для AOT на основе CoreCLR
- Компиляция приложения в 1 нативный исполняемый файл
- Легко разворачивается
- Представляет собой сборку ILCompiler
- Можно создать автономную статическую/динамическую библиотеку

AOT для .NET Core. CoreRT. **WARNING**

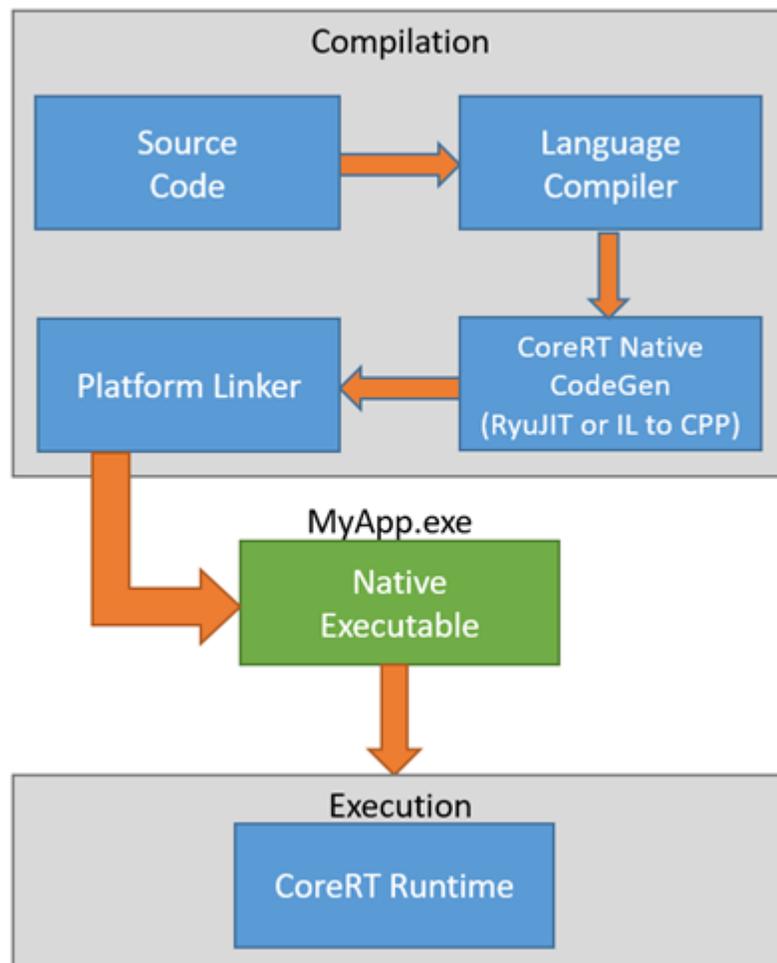
- .NET Core runtime оптимизированный для AOT на основе CoreCLR
- Компиляция приложения в 1 нативный исполняемый файл
- Легко разворачивается
- Представляет собой сборку ILCompiler
- Можно создать автономную статическую/динамическую библиотеку
- Используется в связке с RyuJIT

АОТ для .NET Core. CoreRT. **WARNING**

- .NET Core runtime оптимизированный для АОТ на основе CoreCLR
- Компиляция приложения в 1 нативный исполняемый файл
- Легко разворачивается
- Представляет собой сборку ILCompiler
- Можно создать автономную статическую/динамическую библиотеку
- Используется в связке с RyuJIT
- Можно собрать проект под MacOS, Linux, Windows

AOT для .NET Core. CoreRT (Jeffrey Fritz). [До](#)

АОТ для .NET Core. CoreRT (Jeffrey Fritz). После

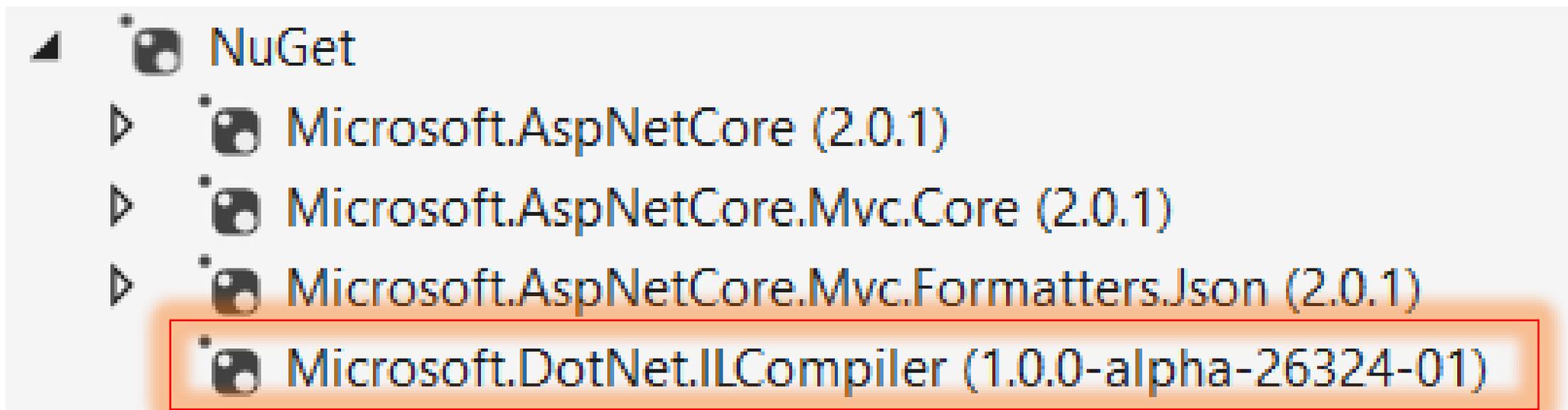


Что Вы нам сказки рассказываете,
у нас backend тормозит!!!



АОТ для .Net Core. CoreRT

- Подключаем ILCompiler



АОТ для .Net Core. CoreRT

- Возьмем простой пример с циклами

```
[HttpGet("/api/values")]  
public IEnumerable<string> Get()  
{  
    for (int i = 0; i < 10000; i++)  
        yield return $"string{i}";  
}
```

АОТ для .Net Core. CoreRT. 1 000 000 итераций. Только при старте приложения

- Duration (1й запрос!!!) для АОТ

Application Insights: Трассировка "Request finished in 844.012ms 200 application/json; charset=utf-8"

◆ Имя операции: GET Values/Get, Идентификатор операции: 1101796f-462648d93ee42075

- Duration (1й запрос!!!) для JIT

Application Insights: Трассировка "Request finished in 30754.4086ms 200 application/json; charset=utf-8"

◆ Имя операции: GET Values/Get, Идентификатор операции: 232f182e-4bb1b00420318496

МОЖНО ЛИ ЭТОМУ ВЕРИТЬ?



Демо

- Цель – разобраться, стоит ли использовать CoreRT для веб-проектов

Демо

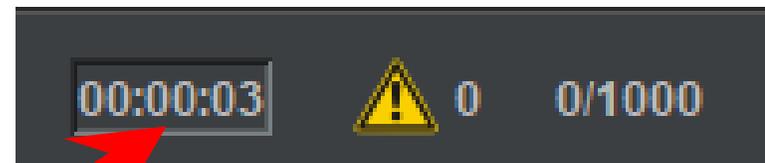
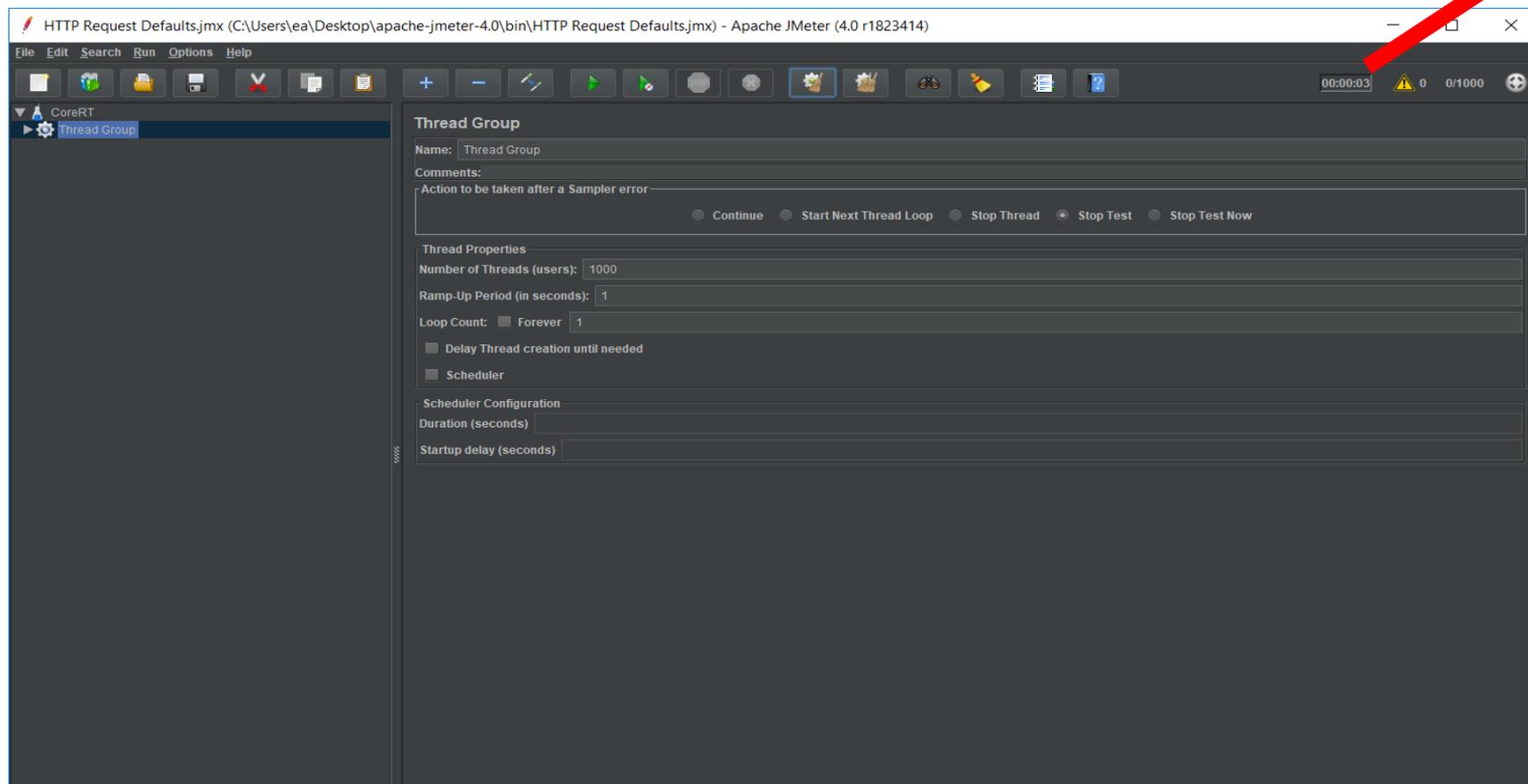
- Цель – разобраться, стоит ли использовать CoreRT для веб-проектов
- Попробовать CoreRT, несмотря на то, что он еще в альфе!

WARNINGS

DEMO

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
Sys 61
Sys Ke
Sys To
Sys ok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
WindowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
wpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

CoreRT, 1.0.0-alpha-26419-01,
1 000 итераций



JIT, 1 000 итераций, 3+ запуск

00:00:01



0

0/1000

The screenshot shows the Apache JMeter interface with the following configuration for an HTTP Request:

- Name:** HTTP Request
- Comments:** (empty)
- Basic Tab:**
 - Web Server:** Protocol [http]: http, Server Name or IP: localhost, Port Number: 5000
 - HTTP Request:** Method: GET, Path: /api/values/809, Content encoding: (empty)
 - Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data for POST, Browser-compatible headers
- Parameters Tab:** Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
-------	-------	---------	-----------------

A red arrow points from the top-right status bar to the JMeter window's title bar.

МИНУТКА АССЕМБЛЕРА!!!

CoreRT – все итерации

```
List<A> ·AList := ·new ·List<A> ();
00007FFBBE3E5A50 ·push ······rdi ··
00007FFBBE3E5A51 ·push ······rsi ··
00007FFBBE3E5A52 ·push ······rbp ··
00007FFBBE3E5A53 ·push ······rbx ··
00007FFBBE3E5A54 ·sub ······rsp,28h ··
00007FFBBE3E5A58 ·mov ······esi,edx ··
00007FFBBE3E5A5A ·mov ······rcx,7FFBBE536AC8h ··
00007FFBBE3E5A64 ·call ······00007FFC1DB74310 ··
00007FFBBE3E5A69 ·mov ······rdi,rax ··
00007FFBBE3E5A6C ·mov ······rcx,7FFBBD15338h ··
00007FFBBE3E5A76 ·xor ······edx,edx ··
00007FFBBE3E5A78 ·call ······00007FFC1DB25B00 ··
00007FFBBE3E5A7D ·mov ······rdx,212DFC5BFE0h ··
00007FFBBE3E5A87 ·mov ······rdx,qword ·ptr ·[rdx] ··
00007FFBBE3E5A8A ·lea ······rcx,[rdi+8] ··
00007FFBBE3E5A8E ·call ······00007FFC1DB72750 ··
```

JIT – после 30 запросов

```
··········List<A> ·AList := ·new ·List<A> ();
00007FFBBD805DA3 ·push ······rbx ··
00007FFBBD805DA4 ·sub ······rsp,28h ··
00007FFBBD805DA8 ·mov ······esi,edx ··
00007FFBBD805DAA ·mov ······rcx,7FFBBD955628h ··
00007FFBBD805DB4 ·call ······00007FFC1CF84310 ··
00007FFBBD805DB9 ·mov ······rdi,rax ··
00007FFBBD805DBC ·mov ······rcx,7FFBBD335338h ··
00007FFBBD805DC6 ·xor ······edx,edx ··
00007FFBBD805DC8 ·call ······00007FFC1CF35B00 ··
00007FFBBD805DCD ·mov ······rdx,1DCD311BFD8h ··
00007FFBBD805DD7 ·mov ······rdx,qword ·ptr ·[rdx] ··
00007FFBBD805DDA ·lea ······rcx,[rdi+8] ··
00007FFBBD805DDE ·call ······00007FFC1CF82750 ··
```

И тут я подумала...
Что CoreRT для веба не годится.....



UPDATE!!!

Description

Provides the toolchain to compile managed code to native
b459963c0b530f88905c4639b1948241486b8b40

When using NuGet 3.x this package requires at least version 3.4.

Version: 1.0.0-alpha-26421-01

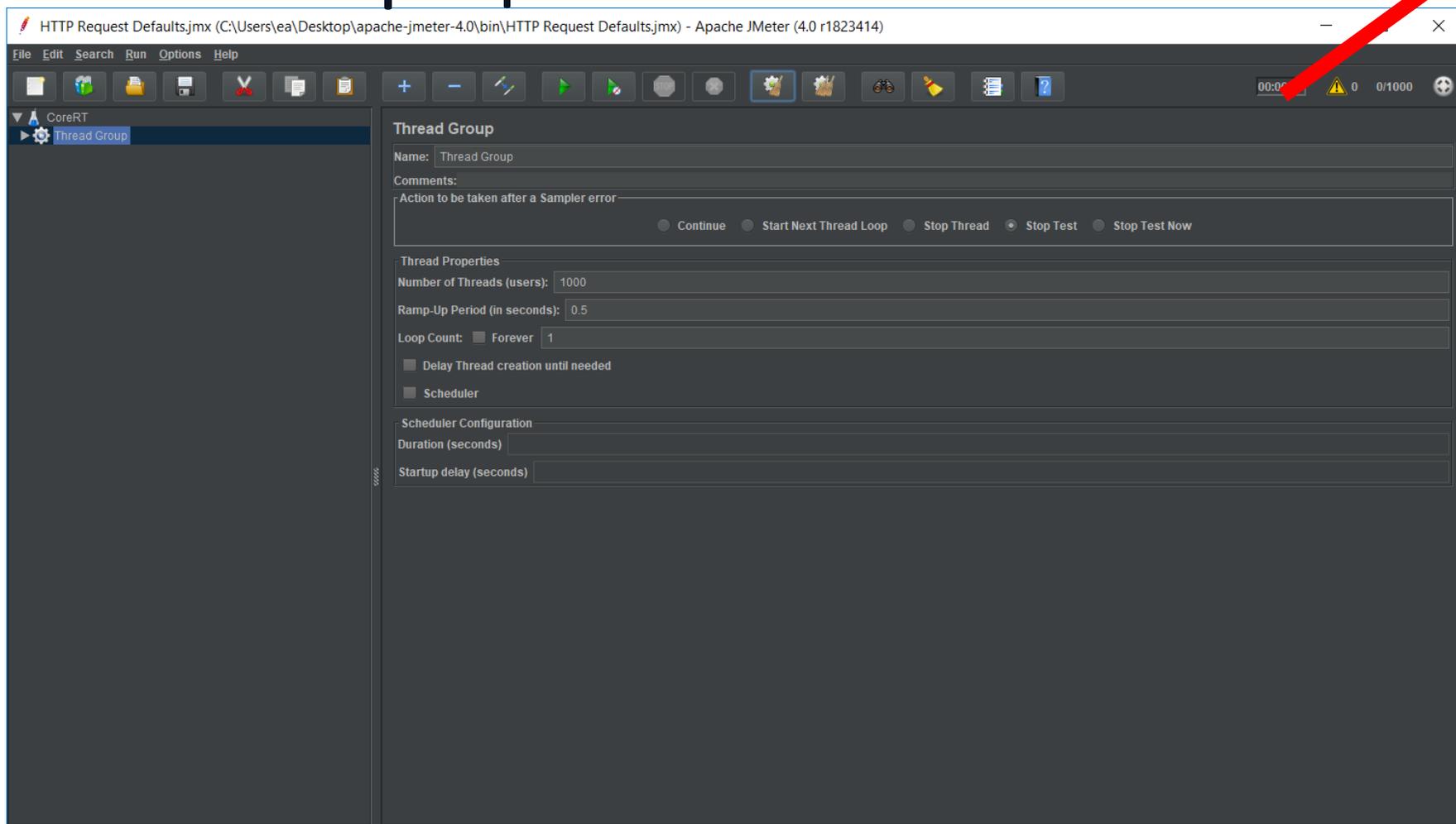
Author(s): Microsoft

License: <https://github.com/dotnet/coreclr/blob/master/LICENSE.TXT>

Date published: Saturday, April 21, 2018 (4/21/2018)

CoreRT, 1.0.0-alpha-26421-01, 1 000 итераций

00:00:01  0 0/1000



The screenshot shows the Apache JMeter GUI. The title bar reads "HTTP Request Defaults.jmx (C:\Users\ea\Desktop\apache-jmeter-4.0\bin\HTTP Request Defaults.jmx) - Apache JMeter (4.0 r1823414)". The menu bar includes "File", "Edit", "Search", "Run", "Options", and "Help". The toolbar contains various icons for file operations and test execution. The left sidebar shows a tree view with "CoreRT" and "Thread Group". The main panel displays the configuration for the "Thread Group":

- Name: Thread Group
- Comments: (empty)
- Action to be taken after a Sampler error: Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now
- Thread Properties:
 - Number of Threads (users): 1000
 - Ramp-Up Period (in seconds): 0.5
 - Loop Count: Forever 1
 - Delay Thread creation until needed
 - Scheduler
- Scheduler Configuration:
 - Duration (seconds): (empty)
 - Startup delay (seconds): (empty)

Много запросов. Используйте JIT?

- Версия CoreRT 1.0.0-alpha-26419-01 – отключены RyuJIT-оптимизации

Много запросов. Используйте JIT?

- Версия CoreRT 1.0.0-alpha-26419-01 – отключены RyuJIT-оптимизации
- Версия CoreRT 1.0.0-alpha-26421-01 – оптимизации включили!
 - CoreRT на старте даже быстрее, чем JIT!

Много запросов. Используйте JIT?

- Версия CoreRT 1.0.0-alpha-26419-01 – отключены RyuJIT-оптимизации
- Версия CoreRT 1.0.0-alpha-26421-01 – оптимизации включили!
 - CoreRT на старте даже быстрее, чем JIT!
- Но!!! В CoreRT много всего Not Implemented!!! **WARNING**

Много запросов. Используйте JIT?

- Версия CoreRT 1.0.0-alpha-26419-01 – отключены RyuJIT-оптимизации
- Версия CoreRT 1.0.0-alpha-26421-01 – оптимизации включили!
 - CoreRT на старте даже быстрее, чем JIT!
- Но!!! В CoreRT много всего Not Implemented!!! **WARNING**
- CoreRT находится в альфе и пока **НЕ стабилен**

Много запросов. Используйте JIT?

- Версия CoreRT 1.0.0-alpha-26419-01 – отключены RyuJIT-оптимизации
- Версия CoreRT 1.0.0-alpha-26421-01 – оптимизации включили!
 - CoreRT на старте даже быстрее, чем JIT!
- Но!!! В CoreRT много всего Not Implemented!!! **WARNING**
- CoreRT находится в альфе и пока **НЕ стабилен**

**А еще скоро можно будет делать красивые бенчмарки с помощью BenchmarkDotNet (уже есть превью!!!)*

samples/WebApi doesn't work with .NET Core 2.1 #5720

New issue

📌 Open markrendle opened this issue 4 days ago · 6 comments



markrendle commented 4 days ago



I encountered this on my own project, so I tried with the WebApi sample and got the same issue.

I tested before changing anything and everything worked.

I updated the `csproj` file to use 2.1 instead of 2.0, so it now looks like this:

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>netcoreapp2.1</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
    <RdXmlFile Include="rd.xml" />
  </ItemGroup>
  <ItemGroup>
    <Folder Include="wwwroot\" />
  </ItemGroup>
</Project>
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Ограничения. CoreRT

- Reflection и метаданные – преобразуются в статическое сопоставление значений полей

Ограничения. CoreRT

- Reflection и метаданные – преобразуются в статическое сопоставление значений полей
- По возможности, .NET Native/CoreRT старается удалить все метаданные

Ограничения. CoreRT

- Reflection и метаданные – преобразуются в статическое сопоставление значений полей
- По возможности, .NET Native/CoreRT старается удалить все метаданные
- Код не зависит от сторонних библиотек, код сторонних классов – является локальным для приложения

Ограничения. CoreRT

- Reflection и метаданные – преобразуются в статическое сопоставление значений полей
- По возможности, .NET Native/CoreRT старается удалить все метаданные
- Код не зависит от сторонних библиотек, код сторонних классов – является локальным для приложения
- Полностью заменяет CLR

Ограничения. CoreRT

- Reflection и метаданные – преобразуются в статическое сопоставление значений полей
- По возможности, .NET Native/CoreRT старается удалить все метаданные
- Код не зависит от сторонних библиотек, код сторонних классов – является локальным для приложения
- Полностью заменяет CLR
- Из приложения могут быть исключены: reflection, dynamic, late-bound calls, сериализация, COM interop.

AOT для mono

```
elizaveta@elizaveta-VirtualBox:~/RiderProjects/MonoAOT/MonoAOT$ mono --aot bin/Release/MonoAOT.exe
Mono Ahead of Time compiler - compiling assembly /home/elizaveta/RiderProjects/MonoAOT/MonoAOT/bin/Release/MonoAOT.exe
AOTID ECB98456-15BC-CA53-8F74-2B580E3D4EC0
Code: 437(34%) Info: 16(1%) EX Info: 64(5%) Unwind Info: 63(4%) Class Info: 53(4%) PLT: 100(7%) GOT
Info: 401(31%) Offsets: 136(10%) GOT: 368
Compiled: 3/3 (100%), No GOT slots: 1 (33%), Direct calls: 0 (0%)
Executing the native assembler: "as" --64 -o /tmp/mono_aot_IIfvHn.o /tmp/mono_aot_IIfvHn
Executing the native linker: "ld" -shared -o /home/elizaveta/RiderProjects/MonoAOT/MonoAOT/bin/Release/MonoAOT.exe.so.tmp /tmp/mono_aot_IIfvHn.o
JIT time: 1 ms, Generation time: 0 ms, Assembly+Link time: 91 ms.
elizaveta@elizaveta-VirtualBox:~/RiderProjects/MonoAOT/MonoAOT$
```

АОТ для mono

- Чтобы выполнить aot-компиляцию для mono-приложений, необходимо вызвать mono с флагом `-aot`

```
$ mono --aot MonoAOT.exe
```

- Для того, чтобы полностью отказаться от jit, используйте `--full-aot`

```
$ mono --full-aot MonoAOT.exe
```

- **Примечание: full-aot доступно только на AMD64/ARM**

AOT для mono

```
elizaveta@elizaveta-VirtualBox:~/RiderProjects/MonoAOT/MonoAOT/bin/Release$ mono --aot MonoAOT.exe
Mono Ahead of Time compiler - compiling assembly /home/elizaveta/RiderProjects/MonoAOT/MonoAOT/bin/Release/MonoAOT.exe
AOTID 45CF1E81-E9C1-16CA-81A4-20FBD7AEA094
Code: 437(34%) Info: 16(1%) Ex Info: 64(5%) Unwind Info: 63(4%) Class
Info: 53(4%) PLT: 100(7%) GOT Info: 401(31%) Offsets: 136(10%) GOT: 36
8
Compiled: 3/3 (100%), No GOT slots: 1 (33%), Direct calls: 0 (0%)
```

АОТ для mono

- Сбор методов для компиляции

АОТ для mono

- Сбор методов для компиляции
- JIT-компиляция

АОТ для mono

- Сбор методов для компиляции
- JIT-компиляция
- Вывод результата работы JIT и дополнительной информации

АОТ для mono

- Сбор методов для компиляции
- JIT-компиляция
- Вывод результата работы JIT и дополнительной информации
- Получение итогового файла .so

AOT для mono

- После выполнения AOT-компиляции появляется .so-файл
- Все, что было прекомпилировано AOT – в этом файле

```
ease$ ls  
MonoAOT.exe  MonoAOT.exe.so  MonoAOT.pdb
```

АОТ для mono

МоноАОТ.exe



МоноАОТ.exe.so

AOT для mono

- В Full-AOT не поддерживается наследование от интерфейсов для generic

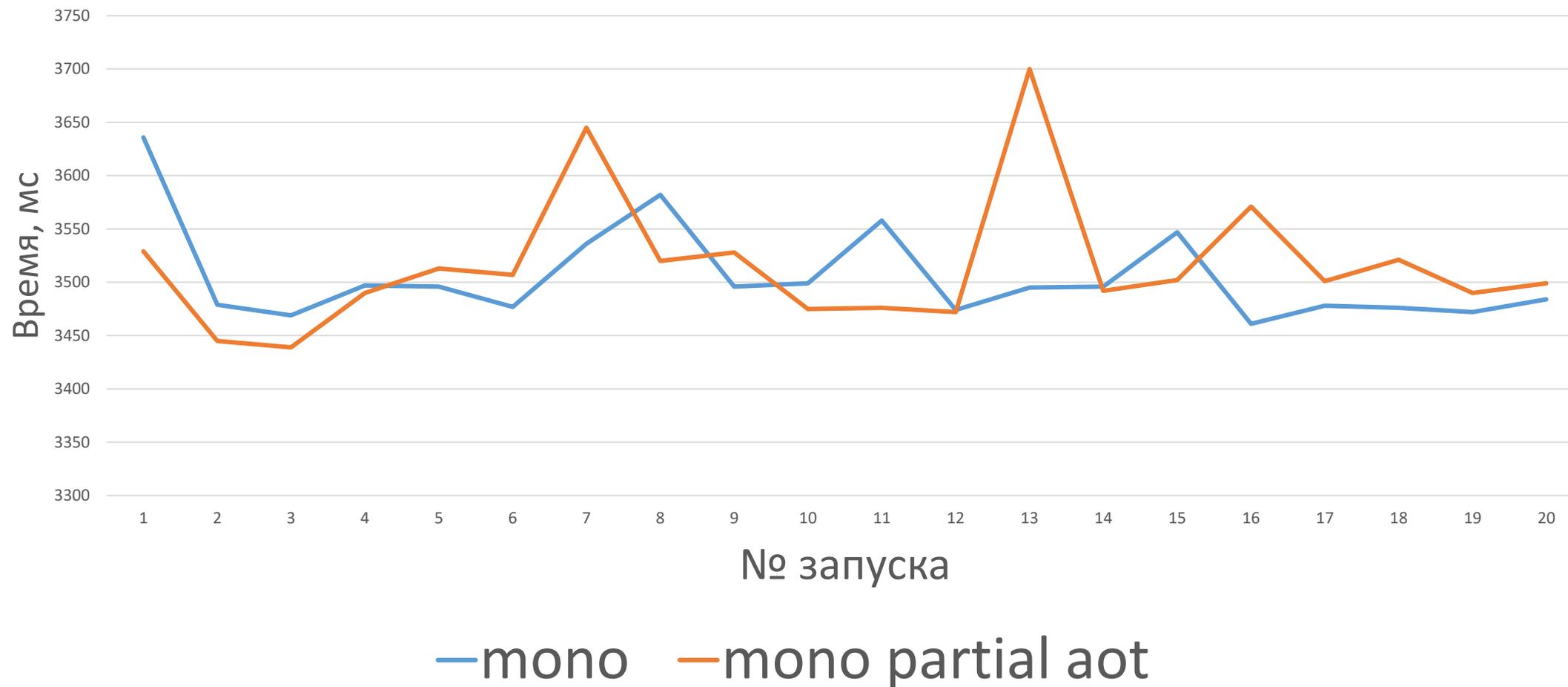
```
interface IDotNext<T>
{
    void CalculateTax();
}
```

АОТ для mono

- Код всех Generic-классов генерируется для всех ValueType автоматически - работает только на iOs

```
public class Room<T>
{
    public void Print(T t) { }
}
```

mono или mono --aot?



mono или mono --aot?

Среднее время работы программы (мс)

mono	mono --aot
3505,4	3515,75

Выводы

- Чтобы использовать AOT в mono более эффективно – желательно отказаться от использования интерфейсов, виртуализации и reflection.

Выводы

- Чтобы использовать AOT в mono более эффективно – желательно отказаться от использования интерфейсов, виртуализации и reflection.
- **В mono --aot ОЧЕНЬ дорогие биндинги адресов между сборками**

Когда использовать AOT?

```
elizaveta@elizaveta-VirtualBox:~/RiderProjects/MonoAOT/MonoAOT$ mono --aot bin/Release/MonoAOT.exe
Mono Ahead of Time compiler - compiling assembly /home/elizaveta/RiderProjects/MonoAOT/MonoAOT/bin/
Release/MonoAOT.exe
AOTID ECB98456-15BC-CA53-8F74-2B580E3D4EC0
Code: 437(34%) Info: 16(1%) EX Info: 64(5%) Unwind Info: 63(4%) Class Info: 53(4%) PLT: 100(7%) GOT
Info: 401(31%) Offsets: 136(10%) GOT: 368
Compiled: 3/3 (100%), No GOT slots: 1 (33%), Direct calls: 0 (0%)
Executing the native assembler: "as" --64 -o /tmp/mono_aot_IIfvHn.o /tmp/mono_aot_IIfvHn
Executing the native linker: "ld" -shared -o /home/elizaveta/RiderProjects/MonoAOT/MonoAOT/bin/Rel
ease/MonoAOT.exe.so.tmp /tmp/mono_aot_IIfvHn.o
JIT time: 1 ms, Generation time: 0 ms, Assembly+Link time: 91 ms.
elizaveta@elizaveta-VirtualBox:~/RiderProjects/MonoAOT/MonoAOT$
```

Когда стоит использовать AOT? Подведем итоги

- Необходимость использования библиотеки несколькими приложениями

Когда стоит использовать AOT? Подведем итоги

- Необходимость использования библиотеки несколькими приложениями
- Необходимо ускорить запуск приложения и уменьшить количество используемой памяти.

Когда стоит использовать AOT? Подведем итоги

- Необходимость использования библиотеки несколькими приложениями
- Необходимо ускорить запуск приложения и уменьшить количество используемой памяти.
- Применение к большим программам дает больший эффект, чем применение к маленьким программам.

Когда стоит использовать AOT? Подведем итоги

- Необходимость использования библиотеки несколькими приложениями
- Необходимо ускорить запуск приложения и уменьшить количество используемой памяти.
- Применение к большим программам дает больший эффект, чем применение к маленьким программам.
- Для программ с небольшим временем выполнения

AOT вокруг нас!

```
Windows PowerShell
System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=
System.Xaml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561
System.Xml.XPath.XDocument, Version=4.0.2.0, Culture=neutral, Publicke
System.Xml.ReaderWriter, Version=4.1.0.0, Culture=neutral, PublicKeyTo
System.Xml.XmlDocument, Version=4.0.2.0, Culture=neutral, PublicKeyTok
System.Xml.Linq, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a
System.Xml.XPath, Version=4.0.2.0, Culture=neutral, PublicKeyToken=b03
TaskScheduler, Version=10.0.0.0, Culture=neutral, PublicKeyToken=31bf3
TlbExp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
TlbImp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50
UIAutomationProvider, Version=4.0.0.0, Culture=neutral, PublicKeyToken
UIAutomationTypes, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31
vbc, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
VBCSCompiler, Version=2.6.0.0, Culture=neutral, PublicKeyToken=31bf385
vsdebugeng.manimpl, Version=15.0.0.0, Culture=neutral, PublicKeyToken=
windowsBase, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856
wpfSurface, Version=15.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f
wsdl, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
xsd, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
PS C:\>
```

Используем AOT-компиляцию правильно

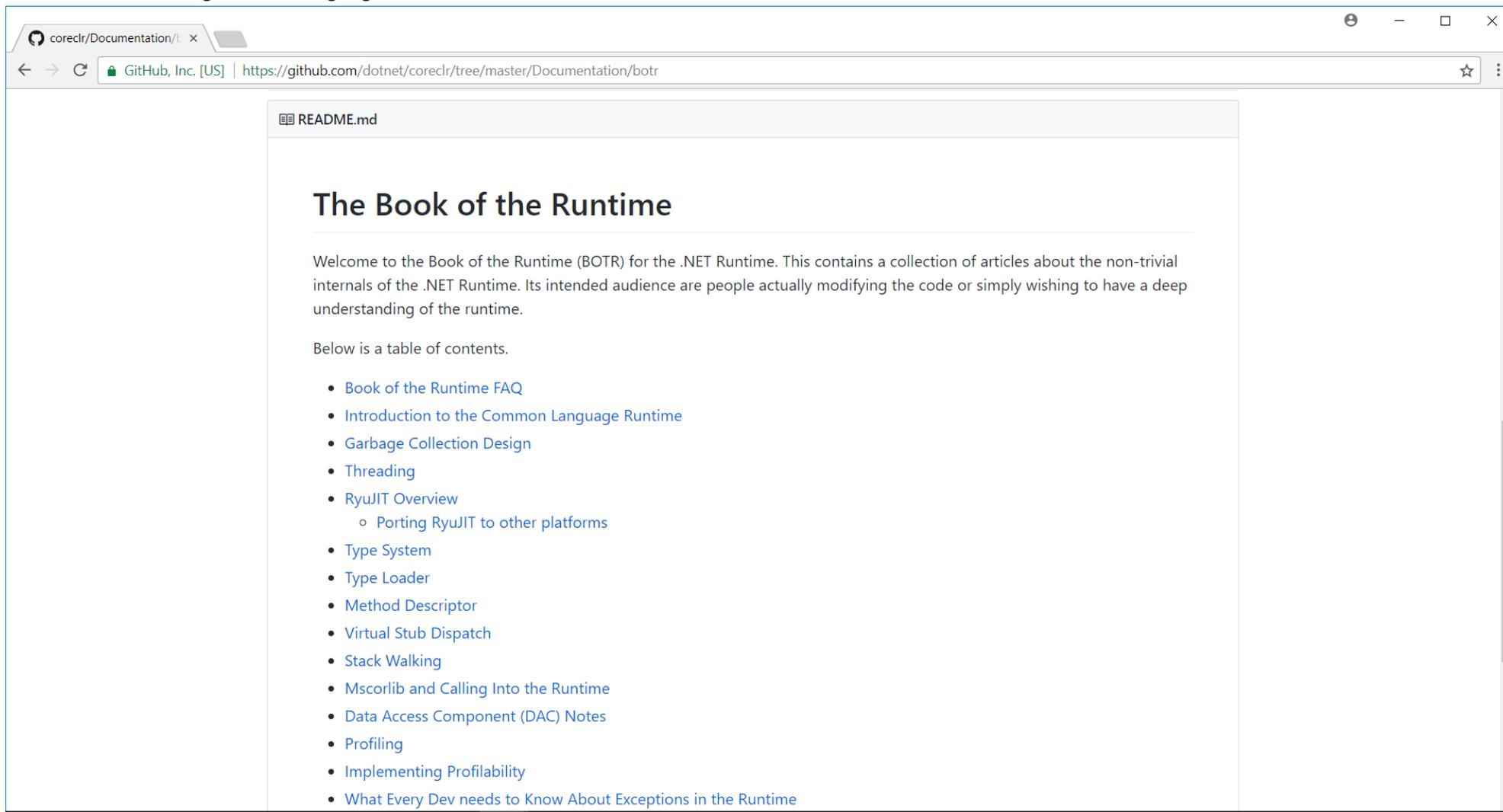
Елизавета Голенок

Ведущий разработчик MTS IT

golenok-ea5@narod.ru

@marmothetka

Литература



The screenshot shows a web browser window with the address bar displaying the URL `https://github.com/dotnet/coreclr/tree/master/Documentation/botr`. The page content is a README file titled "The Book of the Runtime". The text on the page reads: "Welcome to the Book of the Runtime (BOTR) for the .NET Runtime. This contains a collection of articles about the non-trivial internals of the .NET Runtime. Its intended audience are people actually modifying the code or simply wishing to have a deep understanding of the runtime. Below is a table of contents." followed by a bulleted list of links to various articles.

coreclr/Documentation/t x

GitHub, Inc. [US] | <https://github.com/dotnet/coreclr/tree/master/Documentation/botr>

README.md

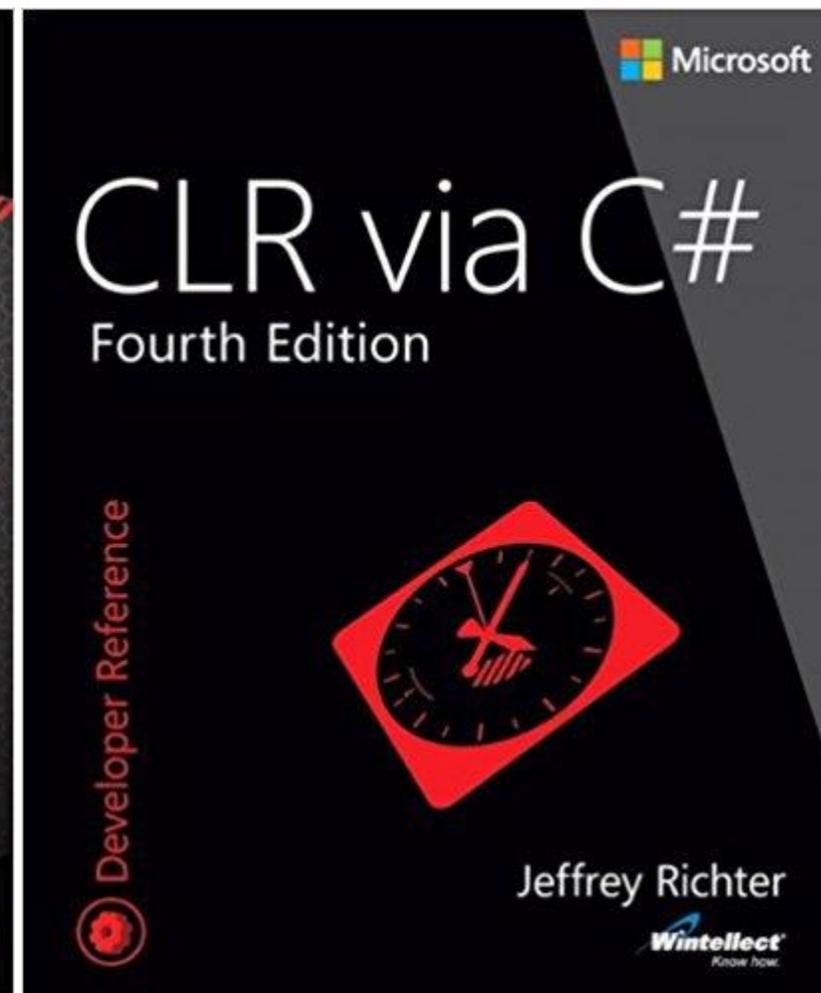
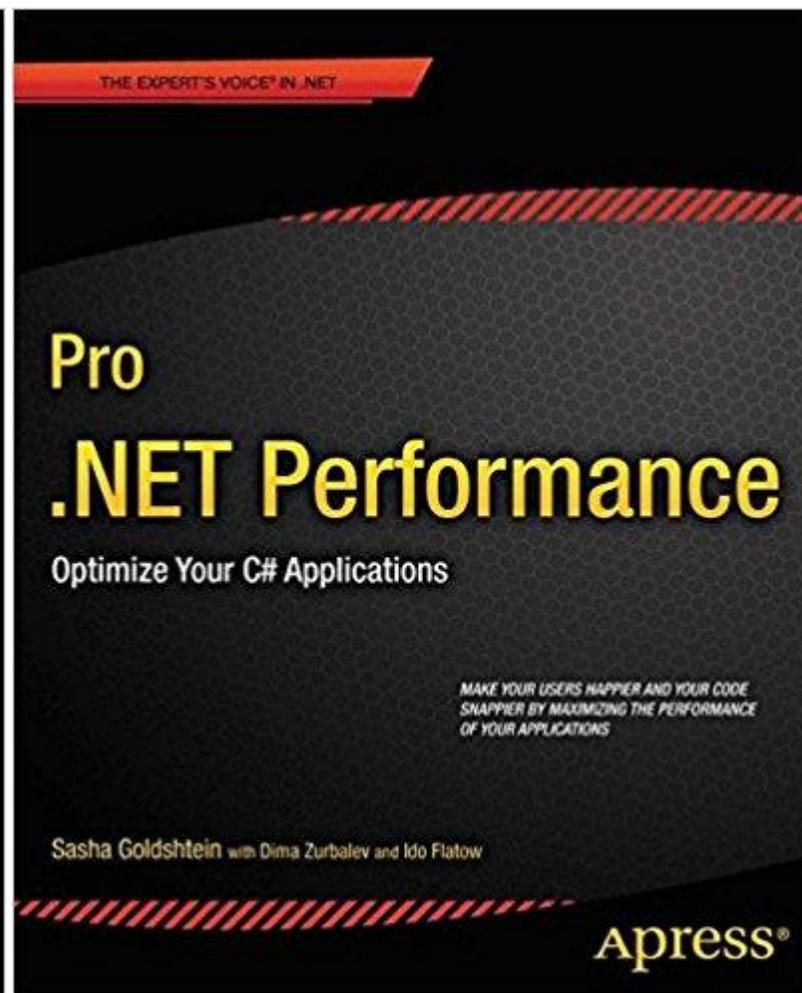
The Book of the Runtime

Welcome to the Book of the Runtime (BOTR) for the .NET Runtime. This contains a collection of articles about the non-trivial internals of the .NET Runtime. Its intended audience are people actually modifying the code or simply wishing to have a deep understanding of the runtime.

Below is a table of contents.

- [Book of the Runtime FAQ](#)
- [Introduction to the Common Language Runtime](#)
- [Garbage Collection Design](#)
- [Threading](#)
- [RyuJIT Overview](#)
 - [Porting RyuJIT to other platforms](#)
- [Type System](#)
- [Type Loader](#)
- [Method Descriptor](#)
- [Virtual Stub Dispatch](#)
- [Stack Walking](#)
- [Mscorlib and Calling Into the Runtime](#)
- [Data Access Component \(DAC\) Notes](#)
- [Profiling](#)
- [Implementing Profiability](#)
- [What Every Dev needs to Know About Exceptions in the Runtime](#)

Литература



Спасибо за внимание!

Елизавета Голенок

Ведущий разработчик MTS IT

golenok-ea5@narod.ru

@marmothetka