

CLRMD

как написать свой инструмент отладки

Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

Немного об CLRMD








- Появилась в 2013 г.
- Microsoft.Diagnostics.Runtime.dll
- Реализация поверх **mscordacwks.dll**
- Позволяет подключаться к живым процессам и memory dump
- Позволяет автоматизировать задачи и получать доступ к отладочной информации
- Писать скрипты автоматизации

Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

Начало работы с CLRMD

DataTarget.

 PlatformFunctions	PlatformFunctions
 LoadCrashDump	DataTarget
 AttachToProcess	DataTarget
 CreateFromDataReader	DataTarget
 CreateFromDebuggerInterface	DataTarget
 CreateSnapshotAndAttach	DataTarget
 LoadCoreDump	DataTarget

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Attach to process

1. Invasive

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Attach to process

1. Invasive
2. NonInvasive

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Attach to process

1. Invasive
2. NonInvasive
3. Passive

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Attach to process

1. Invasive
2. NonInvasive
3. Passive

Prerequisites

1. Загрузить правильную версию DAC с сервера

Начало работы с CLRMD

DataTarget.

PlatformFunctions	PlatformFunctions
LoadCrashDump	DataTarget
AttachToProcess	DataTarget
CreateFromDataReader	DataTarget
CreateFromDebuggerInterface	DataTarget
CreateSnapshotAndAttach	DataTarget
LoadCoreDump	DataTarget

Memory dump

Путь к memory dump

Prerequisites

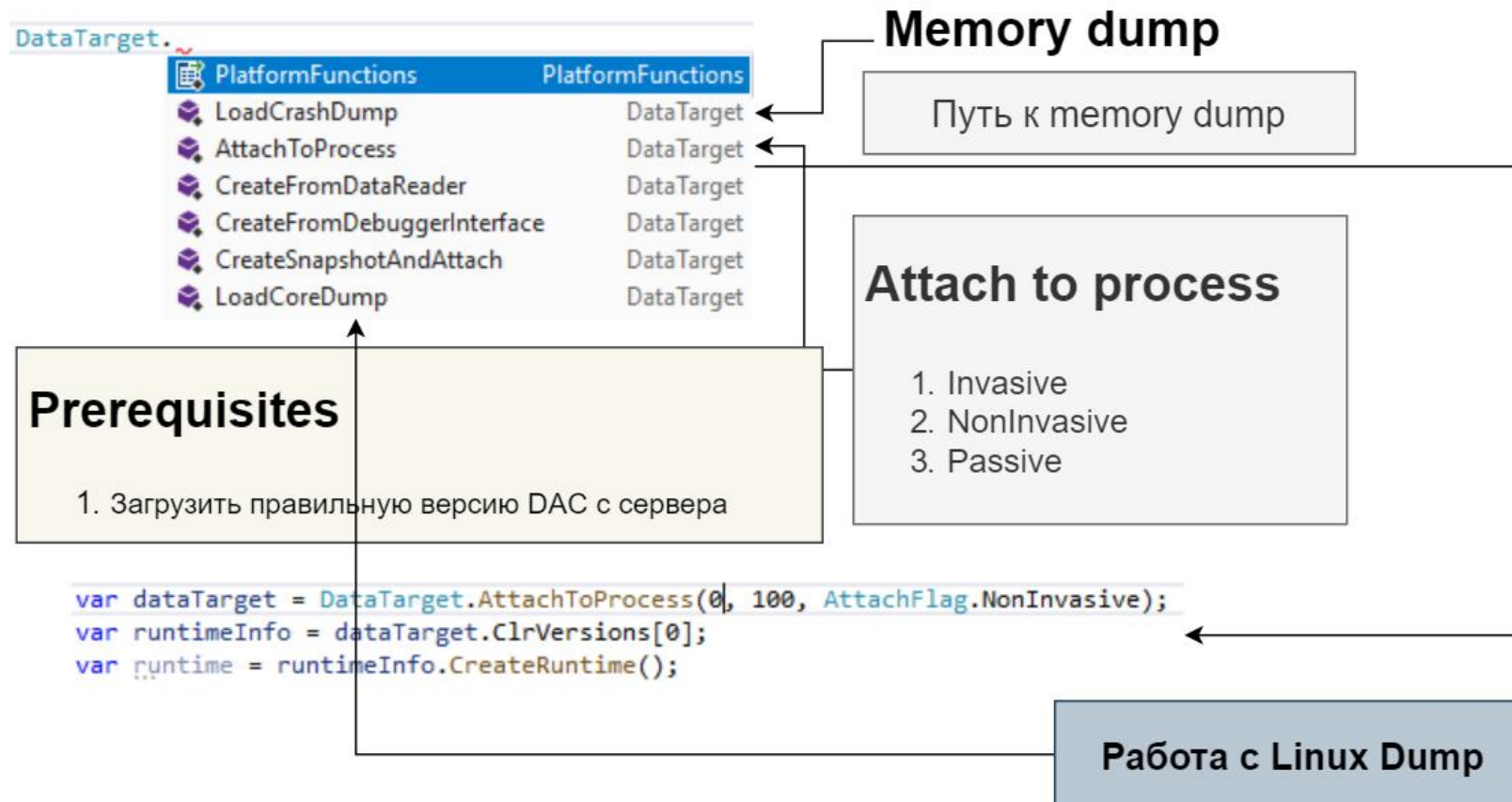
1. Загрузить правильную версию DAC с сервера

Attach to process

1. Invasive
2. NonInvasive
3. Passive

```
var dataTarget = DataTarget.AttachToProcess(0, 100, AttachFlag.NonInvasive);  
var runtimeInfo = dataTarget.ClrVersions[0];  
var runtime = runtimeInfo.CreateRuntime();
```

Начало работы с CLRMD



Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

Перечисление AppDomains

```
foreach (ClrAppDomain domain in runtime.AppDomains)
{
    writer.WriteLine("ID:      {0}", domain.Id);
    writer.WriteLine("Name:    {0}", domain.Name);
    writer.WriteLine("Address: {0}", domain.Address);
}
```

Перечисление AppDomains

```
foreach (ClrAppDomain domain in runtime.AppDomains)
{
    writer.WriteLine("ID:      {0}", domain.Id);
    writer.WriteLine("Name:    {0}", domain.Name);
    writer.WriteLine("Address: {0}", domain.Address);
}
```

Output:

ID: 1

Name: clrhost

Address: 2055986641184

Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

Detect Boxing and Unboxing

На стеке

```
int a = 123;
```

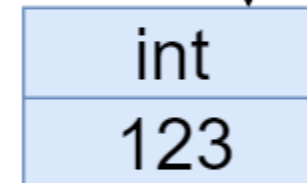
Detect Box and Unboxing

На стеке

```
int a = 123;
```

На куче

```
object a = 123;
```

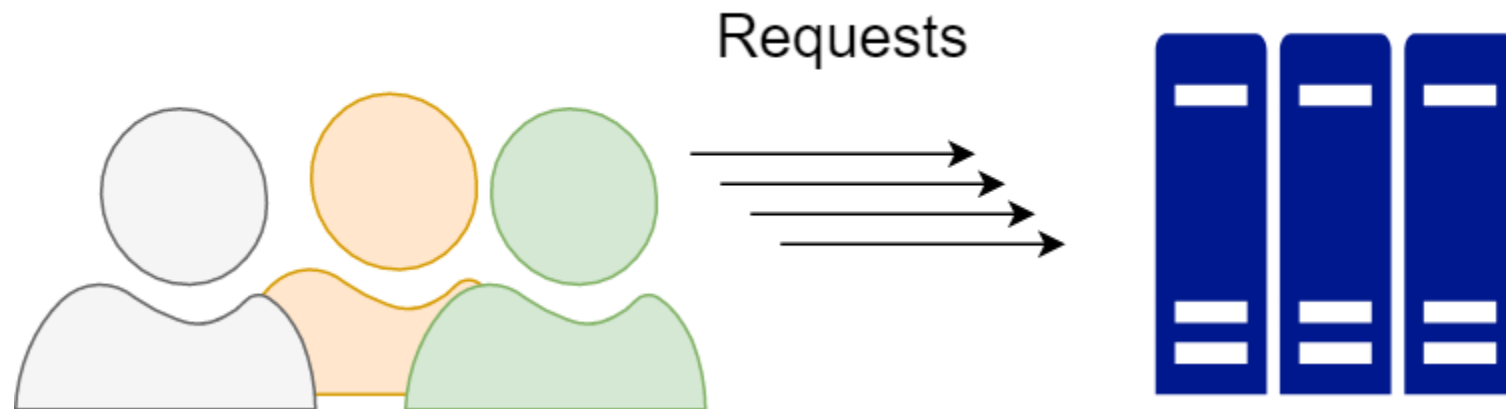


```
*/ IL_0033: ldloc.1  
*/ IL_0034: ldc.i4.s 10  
*/ IL_0036: box      [System.Runtime]System.Int32
```

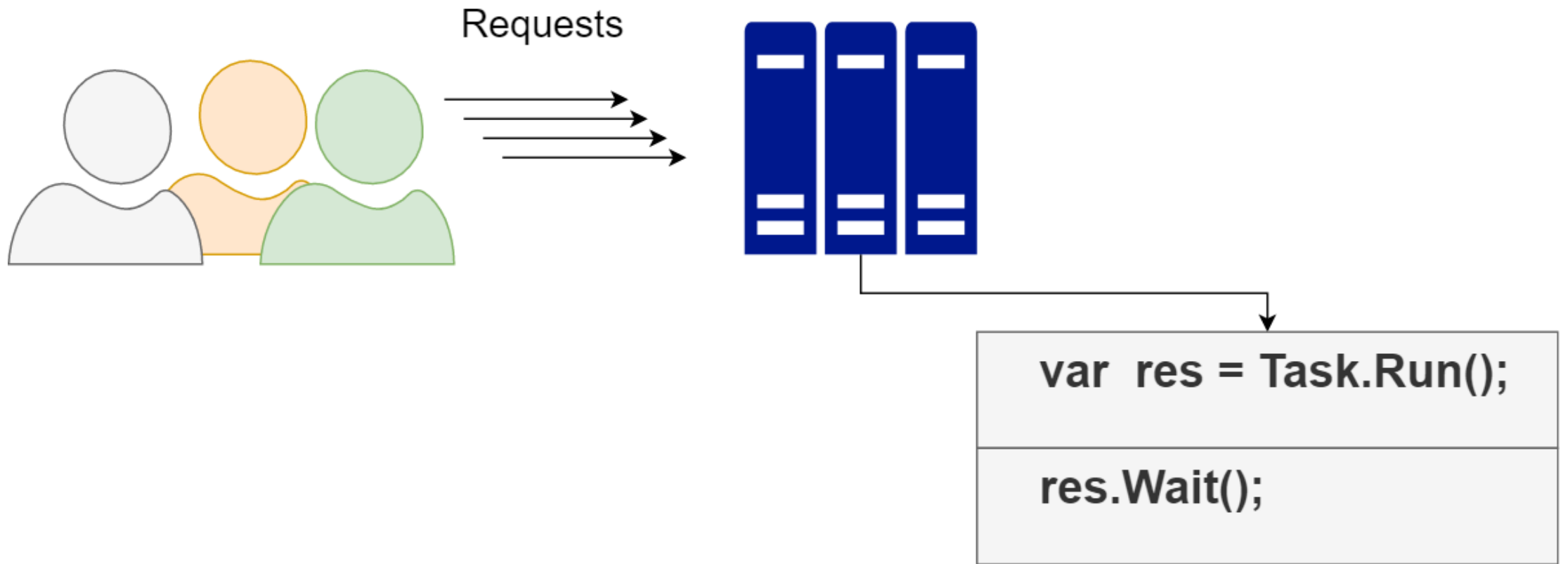
Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

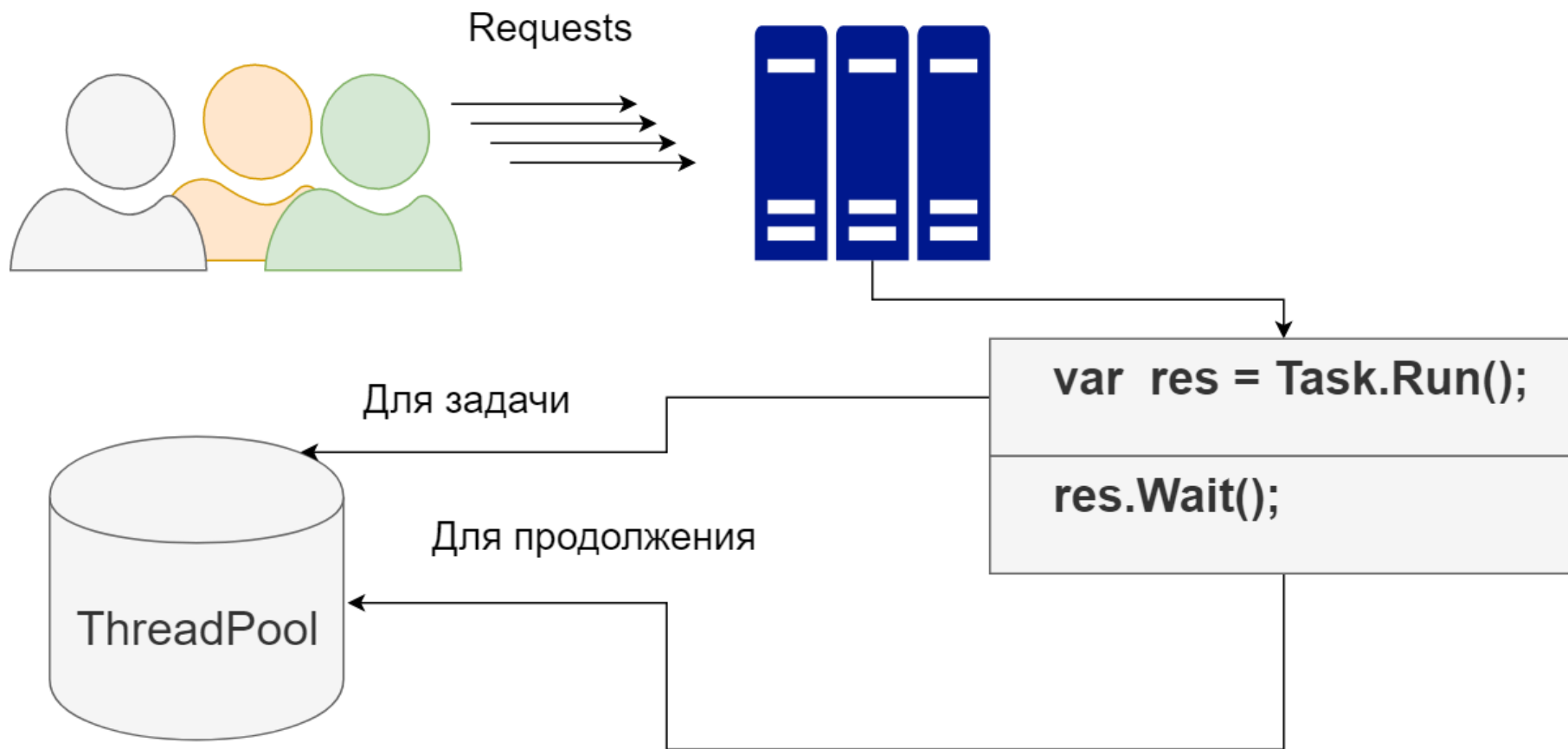
How to find a deadlock



How to find a deadlock



How to find a deadlock



How to find a deadlock

!syncblk

Address	Type	Locked	Owner(s)	Waiter(s)
000001deb4096370	Monitor	1	000001decc9f3bd0	
000001deb40a2d50	Monitor	1	000001decc9f3bd0	
...				
000001deb41fa3d0	MonitorWait	1		000001decc9f3bd0
000001deb416a368	MonitorWait	1		000001decc4721c0

How to find a deadlock

!clrstack

[GCFrame]

[HelperMethodFrame_1OBJ]SpecialSystem.Threading.Monitor.ObjWait(Boolean, Int32, System.Object)System.Threading.ManualResetEventSlim.Wait(Int32,

.....

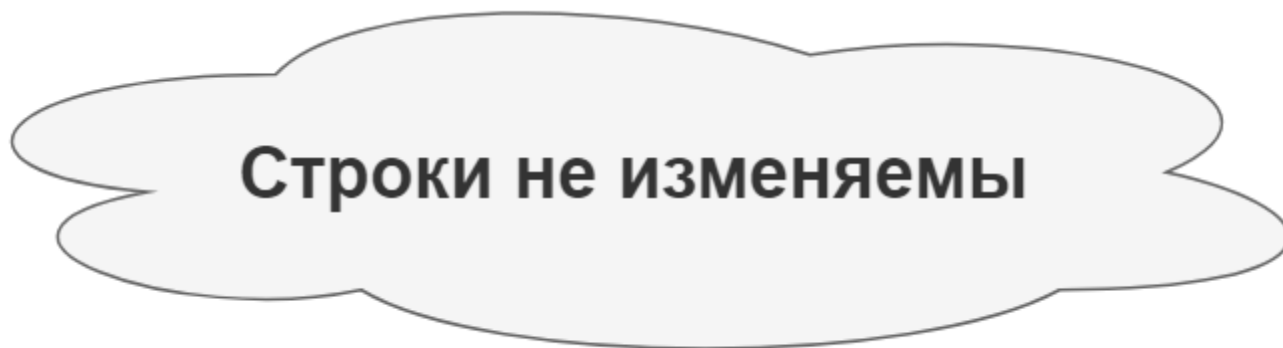
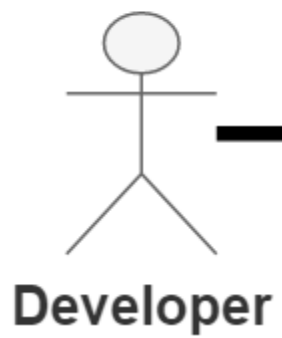
System.Threading.CancellationToken)System.Threading.Tasks.Task.Wait()ProcessDeadlock.Program+
<Process>d__2.MoveNext()System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext, System.Threading.ContextCallback, System.Object)System.Runtime.CompilerServices.AsyncTaskMethodBuilder`1+AsyncStateMachineBox`1[[System.Threading.Tasks.VoidTaskResult, System.Private.CoreLib],[System.__Canon]

.....

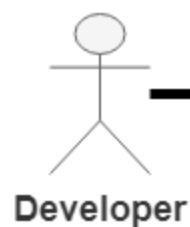
[000001decc9f3bd0]

Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод







Строки не изменяемы

```
string str = "Hello World";
```

Строковый литерал

Storage Stream #US
в assembly.dll

```
string str = "Hello" + " World";
```

Динамические строки

Small Object Heap

Large Object Heap

Agenda

- История CLRMD и проблематика использования
- Как начать использовать CLRMD
- Рассмотрим возможности использования библиотеки на примерах
 - Перечисление AppDomains
 - Detect Boxing and Unboxing
 - How to find a deadlock
 - String Interning
- Вывод

Вывод

- Рассмотрели как начать работать с CLRMD
- Разобрали на примерах как извлечь полезную информацию:
 - Создание DataTarget для работы с memory dump
 - Как обнаружить boxing and unboxing
 - Detect deadlock
 - String Interning

Полезные ссылки

- Документация —
 - <https://github.com/Microsoft/clrmd/blob/master/Documentation/GettingStarted.md>
 - <https://docs.microsoft.com/en-us/dotnet/standard/garbage-collection/>
- SuperDump (сервис для автоматизации crush dump)
<https://github.com/Dynatrace/superdump>
- msos - <https://github.com/goldshtn/msos/wiki>
- dnSpy - <https://github.com/0xd4d/dnSpy#dnspy>
- MemAnalyzer - <https://aloiskraus.wordpress.com/2017/08/17/memalyzer-v2-5-released/>
- BenchmarkDotnet
- DynaMD - <https://github.com/kevingosse/DynaMD>

Спасибо!!!

Вопросы