

World of... MONO_DEBUG

Speaker: Elizaveta Golenok
Company: GoTech Software

mail: golenok-ea5@narod.ru
twitter: @marmothetka

```
elizaveta — lldb /Library/Frameworks/Mono.framework/Versions/Current/bin/mono — 99x16
frame #9: 0x0000000100002529 mono`main(argc=4, argv=<unavailable>) at main.c:339 [opt]
frame #10: 0x0000000100002434 mono`start + 52
(lldb) thread list
Process 49194 stopped
* thread #1: tid = 0x99b889, 0x00000001005dfba4, name = 'tid_307', queue = 'com.apple.main-thread',
stop reason = EXC_BREAKPOINT (code=EXC_I386_BPT, subcode=0x0)
    thread #2: tid = 0x99b89e, 0x00007ffffbe587bf2 libsystem_kernel.dylib`__psynch_cvwait + 10, name =
'SGen worker'
    thread #3: tid = 0x99b89f, 0x00007ffffbe587bf2 libsystem_kernel.dylib`__psynch_cvwait + 10, name =
'SGen worker'
```

О чём поговорим?

- ❖ С чего всё начиналось
- ❖ Удалённая отладка
- ❖ Mono Embedding и расширения
- ❖ Работа с дампами
- ❖ Дедлоки

Окружение

- ❖ Mac OS Sierra
- ❖ Ubuntu 16.04, 17.04

План действий

- ❖ Берем задачу
- ❖ Выбираем нужный инструмент
- ❖ Решаем задачу
- ❖ Делаем выводы

MONO_DEBUG=1



Talk about...

Mono Hard Debugger

- ❖ Консольный отлачик
- ❖ Можно отлаживать Unsafe код
- ❖ Удалённая отладка недоступна



Debugging symbols



Talk about...

Debugging Symbols



MDB

Debugging Symbols

MONO 5+

Debugging Symbols



PDB

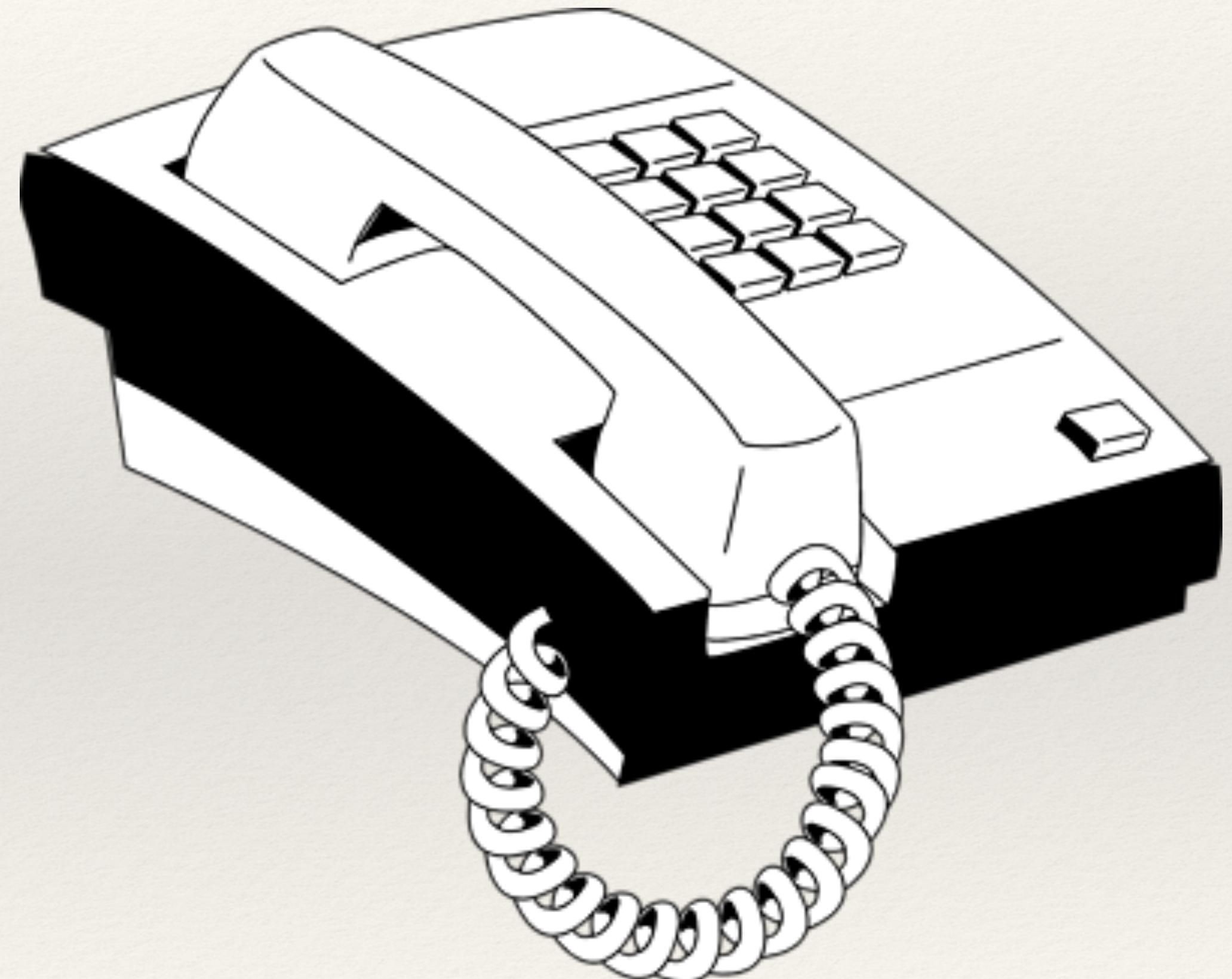
Remote Debug



Talk about...

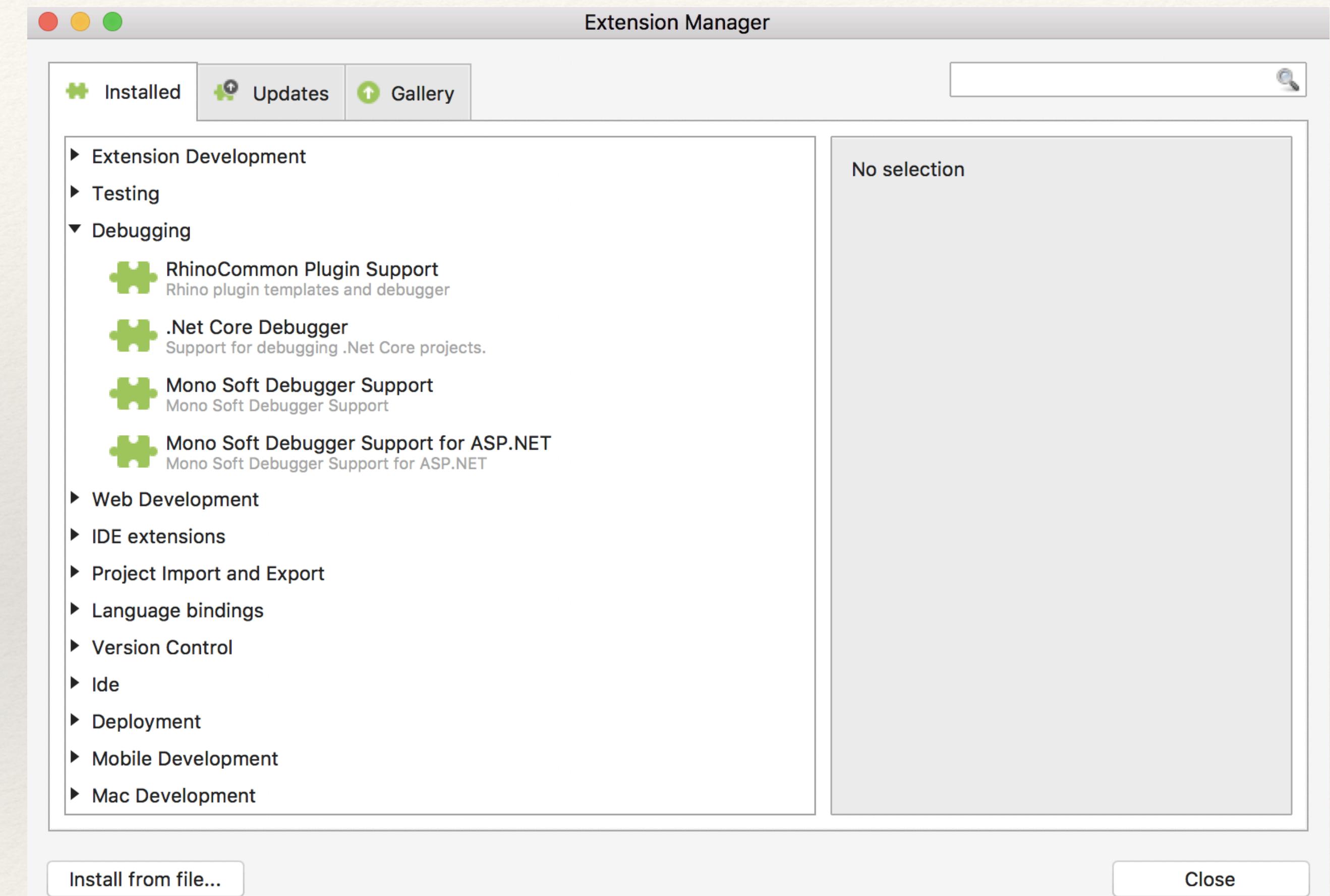
Remote Debug

- ❖ Платежи по кредитам не проходят
- ❖ Да, автоплатёж не работает
- ❖ Ваши шаблоны не работают
- ❖ Возвращаю задачу в Open
- ❖ А в прошлой версии всё работало



Remote Debug

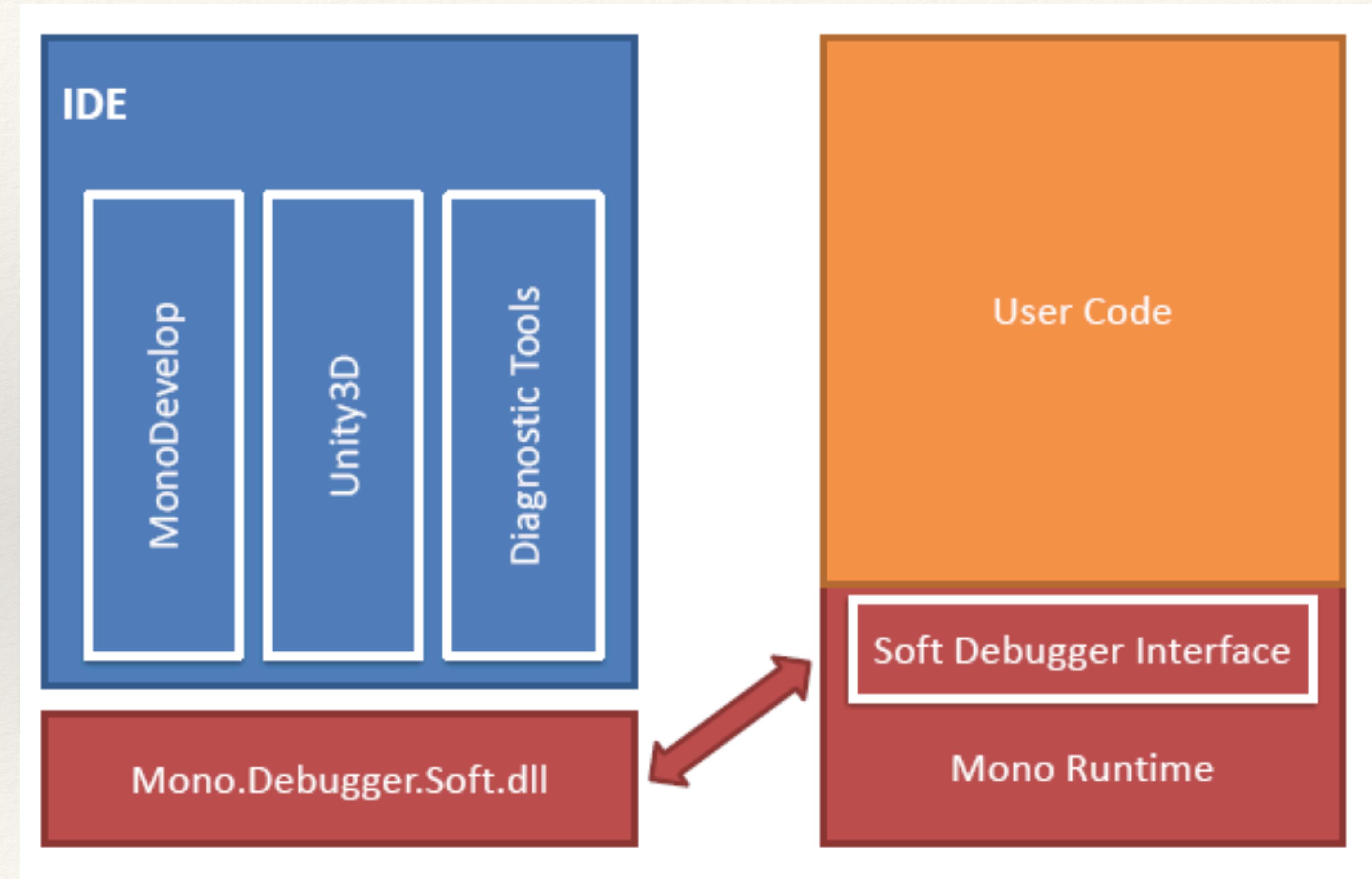
- ❖ Открываем Visual Studio for Mac
- ❖ Устанавливаем Mono Soft Debugger Support
- ❖ Устанавливаем Mono Soft Debugger Support for ASP.NET



Soft Debugger

- ❖ Встроен в MonoDevelop/Xamarin/VS for Mac
- ❖ Так же можно дебажить из консоли
- ❖ Mono.Debugger.Soft.dll
- ❖ Нельзя отлаживать unsafe-код
- ❖ Умеет отлаживать Remote

Soft Debugger



Remote Debug

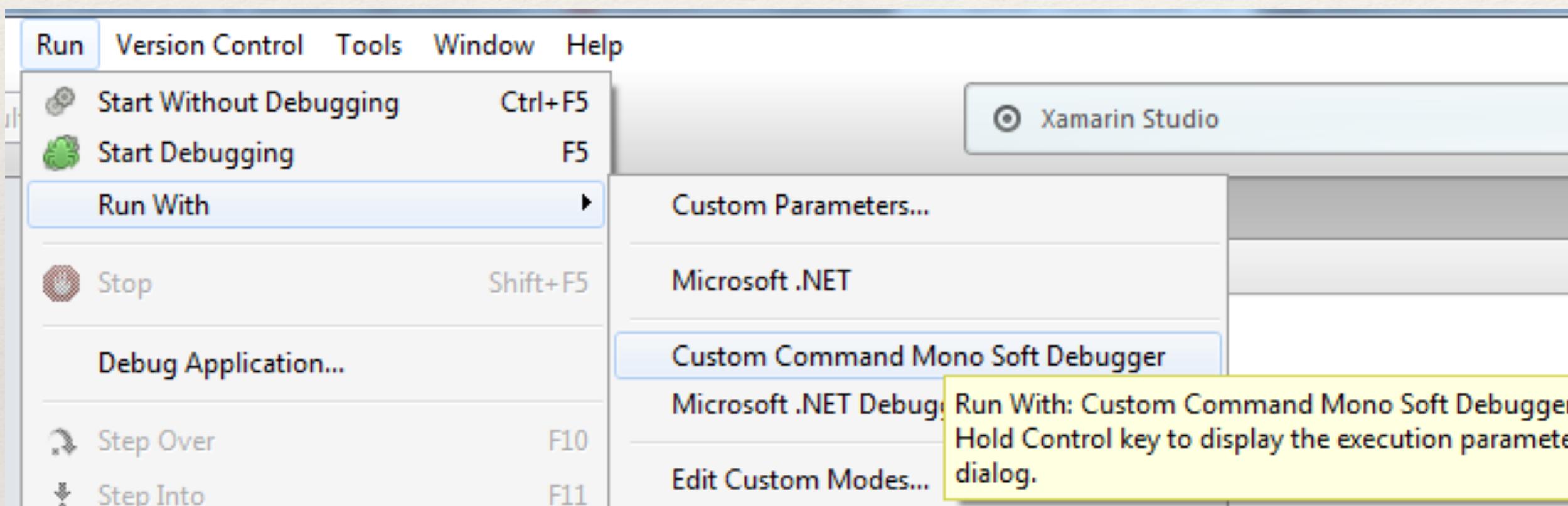
- ❖ Запускаем приложение

```
mono --debugger-agent=address=192.168.99.100:12345,server=y,transport=dt_socket  
--debug ~/Projects/DemoTest/DemoTest/bin/Debug/DemoTest.exe
```

Remote Debug

- ❖ Запускаем отладчик

Run -> Run With -> Custom Command Mono Soft Debugger



Remote Debug

```
case Operations:: Divide...  
    *p /= *p;  
break;
```

System.DivideByZeroException has been thrown ×

Attempted to divide by zero.

[Show Details](#)

Remote Debug. Выводы

- ❖ Для удалённой отладки необходимо запустить debugger-agent
- ❖ Удалённая отладка средствами IDE крайне ограничена
- ❖ Для удалённой отладки можно использовать Soft Debugger Console

Who said
UNSAFE?????



Talk about...

Who said unsafe???

- ❖ Кто сказал, что можно на ноль делить?
- ❖ И вообще ОТКУДА он пришел?
- ❖ Как это понять...
- ❖ **Зачем вообще нужен был UNSAFE?????????????**

Who said UNSAFE?

Имя	Значение	Тип
<input checked="" type="radio"/> p	0xffffffffbffb94ac	int*
*p	⚠ Expression not supported.	

⚠ *p = {} Operator * is not supported
⚠ &p = {} Operator & is not supported
⚠ (IntPtr) p = {} Invalid cast.
► ≡ \$exception = {System.DivideByZeroException} {System.DivideByZeroException: Attempted to divide by zero.\n
[10] op = {DemoTest.MainClass.Operations} Div
[10] p = {int*} 0x7fff59bdd200

LLDB started! Welcome!

- ❖ Высокопроизводительный отладчик
- ❖ Основан на LLVM
- ❖ На Mac OS - отладчик по умолчанию
- ❖ Удобно отлаживать многопоточные приложения

Unsafe & Debug

❖ Запускаем mono

```
elizaveta — lldb /Library/Frameworks/Mono.framework/Versions/Current/bin/mono - mono — 99x16
(lldb) call mono_method_get_class((MonoMethod *)0x1005e3ba9)
(MonoClass *) $6 = 0x481024448b480000
(lldb) q
Quitting LLDB will kill one or more processes. Do you really want to proceed: [Y/n] Y
iMac-Elizaveta:~ elizaveta$ lldb /Library/Frameworks/Mono.framework/Versions/Current/bin/mono
"monobt" command installed
(lldb) target create "/Library/Frameworks/Mono.framework/Versions/Current/bin/mono"
Current executable set to '/Library/Frameworks/Mono.framework/Versions/Current/bin/mono' (x86_64).
(lldb) run --break 'DemoTest.MainClass:Execute' ~/Projects/DemoTest/DemoTest/bin/Debug/DemoTest.exe
Process 49194 launched: '/Library/Frameworks/Mono.framework/Versions/Current/bin/mono' (x86_64)
warning: (x86_64) /Library/Frameworks/Mono.framework/Versions/5.4.0/lib/mono/4.5/mscorlib.dll.dylib
empty dSYM file detected, dSYM was created with an executable with no debug info.
Start process 49194
45
Process 49194 stopped
```

Unsafe & Debug

- ❖ Запускаем приложение и устанавливаем Breakpoint

```
iMac-Elizaveta:~ elizaveta$ lldb /Library/Frameworks/Mono.framework/Versions/Current/bin/mono  
"monobt" command installed  
(lldb) target create "/Library/Frameworks/Mono.framework/Versions/Current/bin/mono"  
Current executable set to '/Library/Frameworks/Mono.framework/Versions/Current/bin/mono' (x86_64).  
(lldb) run --break 'DemoTest.MainClass:Execute' ~/Projects/DemoTest/DemoTest/bin/Debug/DemoTest.exe  
  
Process 49194 launched: '/Library/Frameworks/Mono.framework/Versions/Current/bin/mono' (x86_64)  
warning: (x86_64) /Library/Frameworks/Mono.framework/Versions/5.4.0/lib/mono/4.5/mscorlib.dll.dylib  
empty dSYM file detected, dSYM was created with an executable with no debug info.  
Start process 49194  
45  
Process 49194 stopped  
* thread #1, name = 'tid_307', queue = 'com.apple.main-thread', stop reason = EXC_BREAKPOINT (code=EXC_I386_BPT, subcode=0x0)  
    frame #0: 0x00000001005dfba4  
->  0x1005dfba4: movq    %r15, 0x10(%rsp)
```

Unsafe & Debug

```
elizaveta — lldb /Library/Frameworks/Mono.framework/Versions/Current/bin/mono ➜ mono — 99x16
frame #9: 0x0000000100002529 mono`main(argc=4, argv=<unavailable>) at main.c:339 [opt]
frame #10: 0x0000000100002434 mono`start + 52
(lldb) thread list
Process 49194 stopped
* thread #1: tid = 0x99b889, 0x00000001005dfba4, name = 'tid_307', queue = 'com.apple.main-thread',
stop reason = EXC_BREAKPOINT (code=EXC_I386_BPT, subcode=0x0)
    thread #2: tid = 0x99b89e, 0x00007fffbe587bf2 libsystem_kernel.dylib`__psynch_cvwait + 10, name =
'SGen worker'
    thread #3: tid = 0x99b89f, 0x00007fffbe587bf2 libsystem_kernel.dylib`__psynch_cvwait + 10, name =
'SGen worker'
    thread #4: tid = 0x99b8a0, 0x00007fffbe580386 libsystem_kernel.dylib`semaphore_wait_trap + 10, na
me = 'Finalizer'
    thread #5: tid = 0x99b8a1, 0x00007fffbe58844e libsystem_kernel.dylib`__workq_kernreturn + 10
    thread #6: tid = 0x99b8a2, 0x00007fffbe58844e libsystem_kernel.dylib`__workq_kernreturn + 10
    thread #7: tid = 0x99b8a3, 0x00007fffbe58844e libsystem_kernel.dylib`__workq_kernreturn + 10
(lldb)
```

Unsafe & Debug

- ❖ Но наше приложение уже запущено...
- ❖ **ЧТО ДЕЛАТЬ???**

Remote & Debug

- ❖ Присоединиться к процессу

```
"monobt" command installed
(elldb) process attach --pid 49918
Process 49918 stopped
* thread #1, name = 'tid_307', queue = 'com.apple.main-thread', stop reason = signal SIGSTO
P
    frame #0: 0x00007fffbe589246 libsystem_kernel.dylib`read + 10
libsystem_kernel.dylib`read:
-> 0x7fffbe589246 <+10>: jae    0x7fffbe589250          ; <+20>
    0x7fffbe589248 <+12>: movq   %rax, %rdi
    0x7fffbe58924b <+15>: jmp    0x7fffbe580cd4          ; cerror
    0x7fffbe589250 <+20>: retq
Target 0: (mono) stopped.

Executable module set to "/Library/Frameworks/Mono.framework/Versions/Current/bin/mono".
Architecture set to: x86_64h-apple-macosx.
(elldb) c
```

Unsafe & Remote Debug

- ❖ Проверяем backtrace: monobt

```
elizaveta — lldb — sudo — 91x16
frame #10: 0x000000100516f6b mono`process_breakpoint(tls=<unavailable>, from_signal=<unavailable>) at debugger-agent.c:4978 [opt]
frame #11: 0x000000100516a2e mono`debugger_agent_breakpoint_from_context(ctx=0x00007fff5f7c4000) at debugger-agent.c:5316 [opt]
frame #12: 0x1008cb07f<0x1008cb07f - sdb_breakpoint_trampoline trampoline>
frame #13: 0x102f9f664 DemoTest.MainClass:Execute (int*,DemoTest.MainClass/Operations){0x7faea1417a78} + 0x74 (0x102f9f5f0 0x102f9f7b6) [0x7faea1602bf0 - DemoTest.exe]
frame #14: 0x00000010044d784 mono`mono_jit_runtime_invoke(method=<unavailable>, obj=<unavailable>, params=0x00007fff5f7c46a8, exc=0x000000100aeea90, error=<unavailable>) at mini-runtime.c:2791 [opt]
frame #15: 0x000000100645398 mono`do_runtime_invoke(method=0x00007faea16044a8, obj=0x0000000000000000, params=0x00007fff5f7c46a8, exc=0x0000000000000000, error=0x00007fff5f7c46e8) at object.c:2862 [opt]
frame #16: 0x0000001006489a9 mono`do_exec_main_checked [inlined] mono_runtime_invoke_checked(method=<unavailable>, obj=<unavailable>, error=<unavailable>) at object.c:3016 [opt]
frame #17: 0x000000100648968 mono`do_exec_main_checked(method=0x00007faea16044a8, args
```

Unsafe & Remote Debug

- ❖ Устанавливаем breakpoint по адресу:
 - ❖ **b 0x102f9f664**

Unsafe & Remote Debug

- ❖ Продолжаем отладку:

```
hecked(method=<unavailable>, obj=<unavailable>, error=<unavailable>) at object.c:3016 [opt]
    frame #17: 0x000000100648968 mono`do_exec_main_checked(method=0x00007faea16044a8, args
=<unavailable>, error=0x00007fff5f7c46e8) at object.c:4690 [opt]
        frame #18: 0x0000001004b64bf mono`mono_jit_exec(domain=<unavailable>, assembly=<unavai
lable>, argc=1, argv=0x00007fff5f7c4a18) at driver.g.c:1036 [opt]
        frame #19: 0x0000001004b8cf3 mono`mono_main [inlined] main_thread_handler at driver.g.
c:1105 [opt]
        frame #20: 0x0000001004b8cc3 mono`mono_main(argc=<unavailable>, argv=<unavailable>) at
driver.g.c:2226 [opt]
        frame #21: 0x00000010043d53d mono`main [inlined] mono_main_with_options(argc=<unavaila
ble>, argv=<unavailable>) at main.c:46 [opt]
        frame #22: 0x00000010043d529 mono`main(argc=4, argv=<unavailable>) at main.c:339 [opt]
        frame #23: 0x00000010043d434 mono`start + 52
(lldb) c
Process 49918 resuming
(lldb)
```

Unsafe & Remote Debug

- ❖ Проверяем входные параметры

```
(lldb) p mono_debug_print_vars ((void *) $rip, 0)
```

```
[elizaveta — llbd — sudo — 91x16]
0x102f9f66a: cmpl    $0x3, %r13d
0x102f9f66e: jae     0x102f9f73f
(lldb) p mono_debug_print_vars ((void *) 0x0000000102f9f664, 0)
(lldb) register read
General Purpose Registers:
rbx = 0x00007faea14171a0
rbp = 0x00007fff5f7c42b0
rsp = 0x00007fff5f7c41a0
r12 = 0x00007faea1602bf0
r13 = 0x0000000000000000
r14 = 0x00007fff5f7c4200
r15 = 0x0000000100aeea90
rip = 0x0000000102f9f664
13 registers were unavailable.

(lldb) p mono_debug_print_vars ((void *) $rip, 0)
```

```
[mono] (lldb) p mono_debug_print_vars ((void *) $rip, 0
```

```
Debugger > Console → " Debug Output → " Parallel Stacks →  /Library/Frameworks/Mono.framework/Versions/Current/bin/mono --debug --debugger-agent=transport=dt_socket,server=y,suspend=y,address=127.0.0.1:55555,setpgid=y /Users/elizaveta/Projects/DemoTest/DemoTest/bin/Debug/DemoTest.exe  
Start process 49918  
45  
45  
●: Arg p (0) in register %r14  
Arg op (1) in memory: base register %rsp + 48
```

Unsafe & Remote Debug

- ❖ Проверяем входные параметры

```
Arg p (0) in register %r14
Arg op (1) in memory: base register %rsp + 48
```

Unsafe & Remote Debug

- ❖ Так вот для чего регистры нужны... А мы-то думали.....

```
ble>, argv=<unavailable>) at main.c:46 [opt]
  frame #22: 0x000000010043d529 mono`main(argc=4, argv=<unavailable>) at main.c:339 [opt]
  frame #23: 0x000000010043d434 mono`start + 52
(lldb) register read
General Purpose Registers:
  rbx = 0x00007faea14171a0
  rbp = 0x00007fff5f7c42b0
  rsp = 0x00007fff5f7c41a0
  r12 = 0x00007faea1602bf0
  r13 = 0x0000000000000000
  r14 = 0x00007fff5f7c4200
  r15 = 0x0000000100aeea90
  rip = 0x0000000102f9f664
13 registers were unavailable.

(lldb)
```

Unsafe & Remote Debug

- ❖ Так вот для чего регистры нужны... А мы-то думали.....

```
ble>, argv=<unavailable>) at main.c:46 [opt]
  frame #22: 0x00000010043d529 mono`main(argc=4, argv=<unavailable>) at main.c:339 [opt]
  frame #23: 0x00000010043d434 mono`start + 52
(lldb) register read
General Purpose Registers:
  rbx = 0x00007faea14171a0
  rbp = 0x00007fff5f7c42b0
  rsp = 0x00007fff5f7c41a0
  r12 = 0x00007faea1602bf0
  r13 = 0x0000000000000000
  r14 = 0x00007fff5f7c4200
  r15 = 0x000000100aeea90
  rip = 0x0000000102f9f664
13 registers were unavailable.

(lldb)
```

Unsafe & Remote Debug

- ❖ Так какое значение у переменной p?

```
elizaveta — lldb — sudo — 82×13
rsp = 0x00007fff5f7c41a0
r12 = 0x00007faea1602bf0
r13 = 0x0000000000000000
r14 = 0x00007fff5f7c4200
r15 = 0x000000100aeea90
rip = 0x0000000102f9f664
13 registers were unavailable.

[(lldb) register read $r14
    r14 = 0x00007fff5f7c4200
[(lldb) p *((int *) $r14)
(int) $19 = 0
(lldb)
```

Unsafe & Remote Debug & Выводы

- ❖ Soft Debugger и Unsafe - не работает
- ❖ IDE позволяют отлаживать unsafe с помощью GDB/LLDB
- ❖ **Если в ваших руках указатель - получите адрес и идите в консоль!**

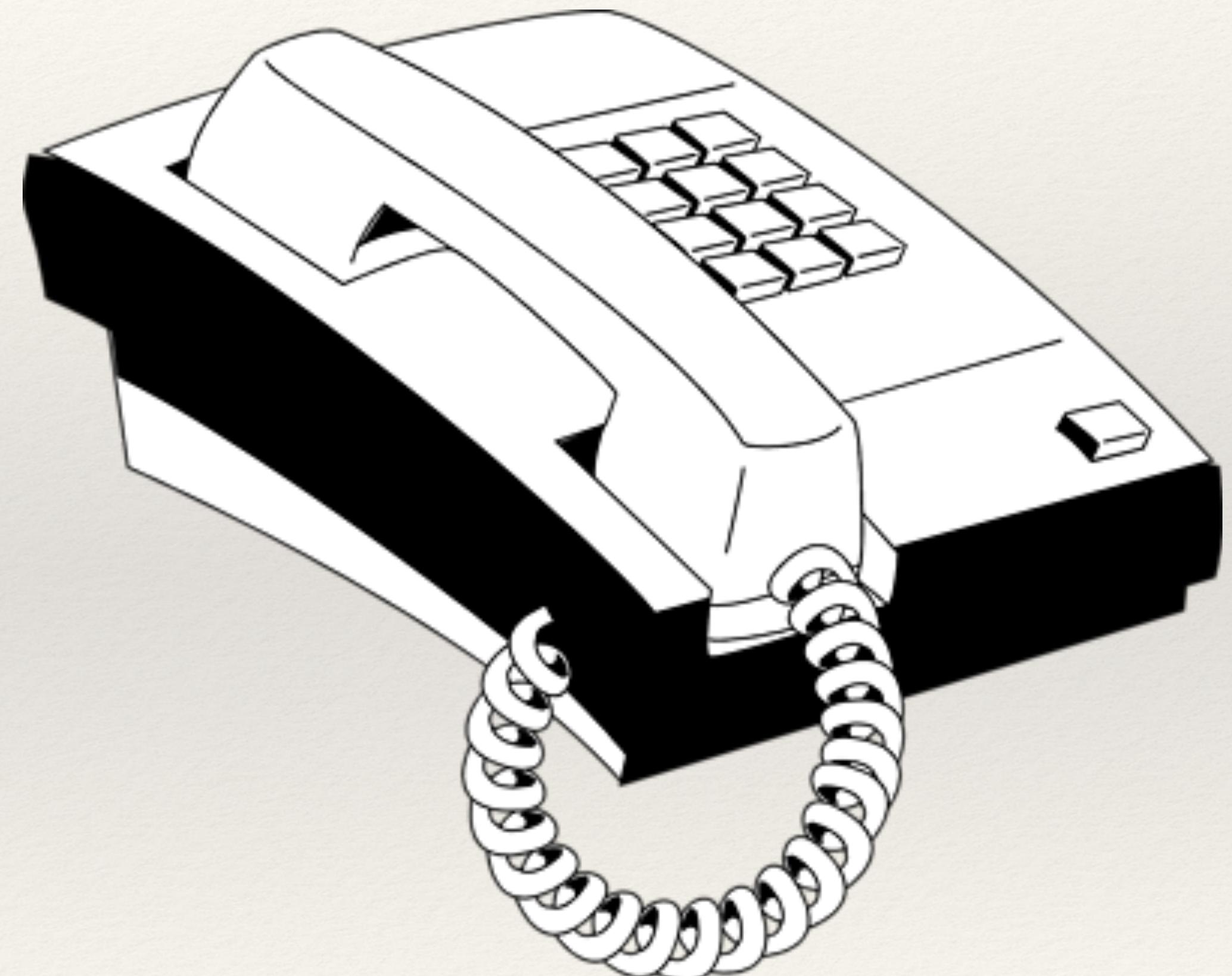
Deadlock?
Dead Lock!!!



Talk about...

Deadlock

- ❖ Ваша программа зависла
- ❖ Опять ничего не работает
- ❖ А в прошлой версии всё работало
- ❖ И зачем эти обновления...
- ❖ Кнопка не нажимается снова
- ❖ Надоели.....



Deadlock

- ❖ Soft Debugger: присоединяемся к процессу

```
Inferior process '2523' ('DemoDotnext.exe') suspended
#0 [0x00000026] TestDeadlock.reverse_order at
/home/elizaveta/Projects/DemoDotnext/DemoDotnext/Program.cs:24
lock (lockA) {
```

Deadlock

- ❖ Смотрим backtrace

```
(sdb) bt
#0 [0x00000026] TestDeadlock.reverse_order at /home/elizaveta/Projects/DemoDotnext/DemoDotnext/
Program.cs:24
lock (lockA) {
#1 [0x00000022] System.Threading.ThreadHelper.ThreadStart_Context at /build/mono-oWqW1z/mono-
4.2.1.102+dfsg2/external/referencesource/mscorlib/system/threading/thread.cs:68
#2 [0x00000083] System.Threading.ExecutionContext.RunInternal at /build/mono-oWqW1z/mono-
4.2.1.102+dfsg2/external/referencesource/mscorlib/system/threading/executioncontext.cs:581
```

Deadlock

- ❖ Смотрим backtrace

```
(sdb) bt
#0 [0x00000026] TestDeadlock.reverse_order at /home/elizaveta/Projects/DemoDotnext/DemoDotnext/
Program.cs:24
lock (lockA) {
#1 [0x00000022] System.Threading.ThreadHelper.ThreadStart_Context at /build/mono-oWqW1z/mono-
4.2.1.102+dfsg2/external/referencesource/mscorlib/system/threading/thread.cs:68
#2 [0x00000083] System.Threading.ExecutionContext.RunInternal at /build/mono-oWqW1z/mono-
4.2.1.102+dfsg2/external/referencesource/mscorlib/system/threading/executioncontext.cs:581
```

Deadlock

- ❖ Проверяем состояние threads

```
(sdb) thread list
Thread #3 "
[0x00000026] TestDeadlock.reverse_order at /home/elizaveta/Projects/DemoDotnext/DemoDotnext/Program.cs:
24
lock (lockA) {
Thread #4 'Finalizer'
Thread #1 "
[0x00000032] System.Threading.Thread.JoinInternal
Thread #2 "
[0x00000026] TestDeadlock.normal_order at /home/elizaveta/Projects/DemoDotnext/DemoDotnext/Program.cs:14
lock (lockB) {
```

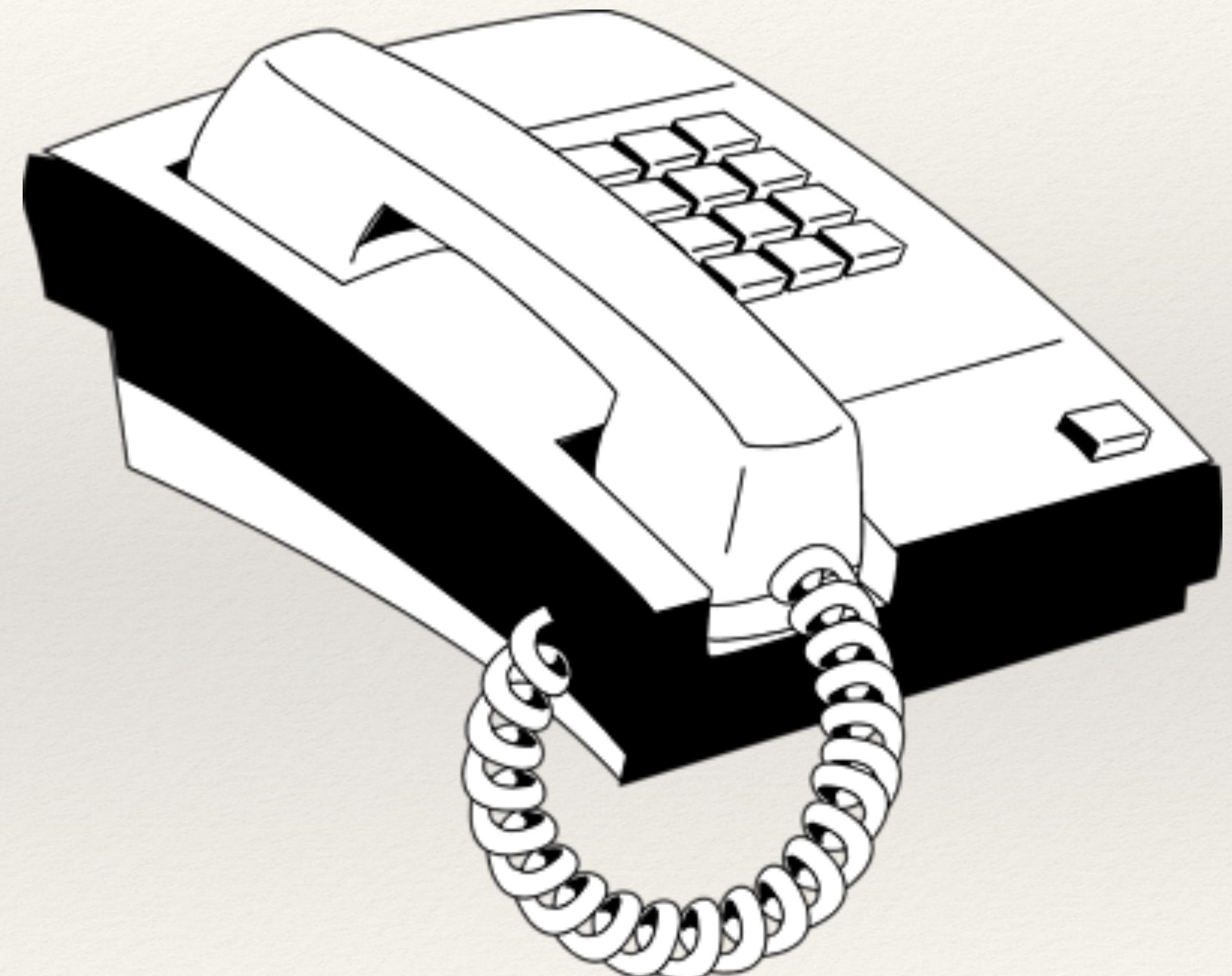
Deadlocks & PLinq & LLDB



Talk about...

Deadlock

- ❖ Ваша программа зависла
- ❖ Опять ничего не работает
- ❖ А в прошлой версии всё работало
- ❖ И зачем эти обновления...
- ❖ Кнопка не нажимается снова
- ❖ Надоели.....



Deadlock

-
- ❖ Делаем дамп ядра зависшего процесса

```
lldb -p 3737
process save-core.3737
```

Deadlock

- ❖ Работаем по инструкции

<https://youtu.be/cYqN8-T695s>



Deadlock

- ❖ Исследуем thread

<https://youtu.be/R141B3Dnc4M>



Deadlock

❖ Немного кода

```
public void Execute(ManualResetEventSlim mre){  
    Enumerable.Range(0, Environment.ProcessorCount * 100).AsParallel().ForAll((j) =>  
    {  
        if (j == Environment.ProcessorCount)  
        {  
            Console.WriteLine("Set on {0} with value of {1}", Thread.CurrentThread.ManagedThreadId, j);  
            mre.Set();  
        }  
        else  
        {  
            Console.WriteLine("Waiting on {0} with value of {1}", Thread.CurrentThread.ManagedThreadId, j);  
            mre.Wait();  
        }  
    });  
}
```

Deadlock. Выводы

- ❖ Для простого дедлока достаточно логов или Soft Debugger-а
- ❖ LLDB & GDB умеют работать с дампами памяти
- ❖ LLDB и GDB с помощью mono embedding позволяют оперативно найти объекты, попавшие в блокировку
- ❖ Но бывают ситуации, где мы вынуждены вручную анализировать backtraces.....

WinDBG & SOS против GDB/LLDB & mono



Talk about...

WinDBG&SOS против GDB/LLDB&mono

WinDBG&SOS против GDB/LLDB&mono

Описание	WinDBG & SOS	GDB & mono	LLDB & mono
Информация о куче	<code>!eeheap</code>	<code>mono_gc_max_generation</code> , <code>mono_gc_get_generation(MonoObject *object)</code> , <code>mono_gc_get_heap_size</code> , <code>mono_gc_get_used_size</code> , <code>mono_gc_walk_heap</code> ,	
Точки останова	<code>!bmpd name.dll</code> <code>Namespace.Class.Method</code>	<code>b 'Namespace.Class:Method'</code> <code>--break 'Namespace.Class:Method'</code>	
Блокировки	<code>!syncblock</code> <code>!analyze -hang -v -f</code>	<code>call mono_locks_dump</code> <code>call mono_object_describe</code> <code>call mono_class_describe_statics</code>	

*Расширение для VTables:

<https://monoruntime.wordpress.com/2009/04/22/interface-method-dispatch-im-table-and-thunks/>

SDB? GDB?
LLDB?



Talk about...

SDB? GDB? LLDB?

Возможности	SDB	GDB	LLDB
Присоединение по PID	-	+	+
Удалённая отладка	+	+	+
Отладка managed	+	С ограничениями	С ограничениями
Unsafe, P/Invoke	-	+	+

SDB? GDB? LLDB?

Возможности	SDB	GDB	LLDB
Точки останова, трассировочные точки	+	+	С ограничениями
Дизассемблер	-	+	+
Backtrace	Только managed	+	+
Расширения	[Command] в .cs	Python	Python

Подведём итоги



Talk about...

Подведём итоги

- ❖ Приложения на topo отлаживать **можно и нужно!**
 - ❖ SDB и IDE спасут Вас от 70% проблем
 - ❖ GDB/LLDB покроют ещё 20% проблем с unsafe, дедлоками и дампами
 - ❖ Valgrind/CLang Sanitizers/HeapShot покроют оставшиеся 9% проблем: <https://youtu.be/KXHMaxymlb4>

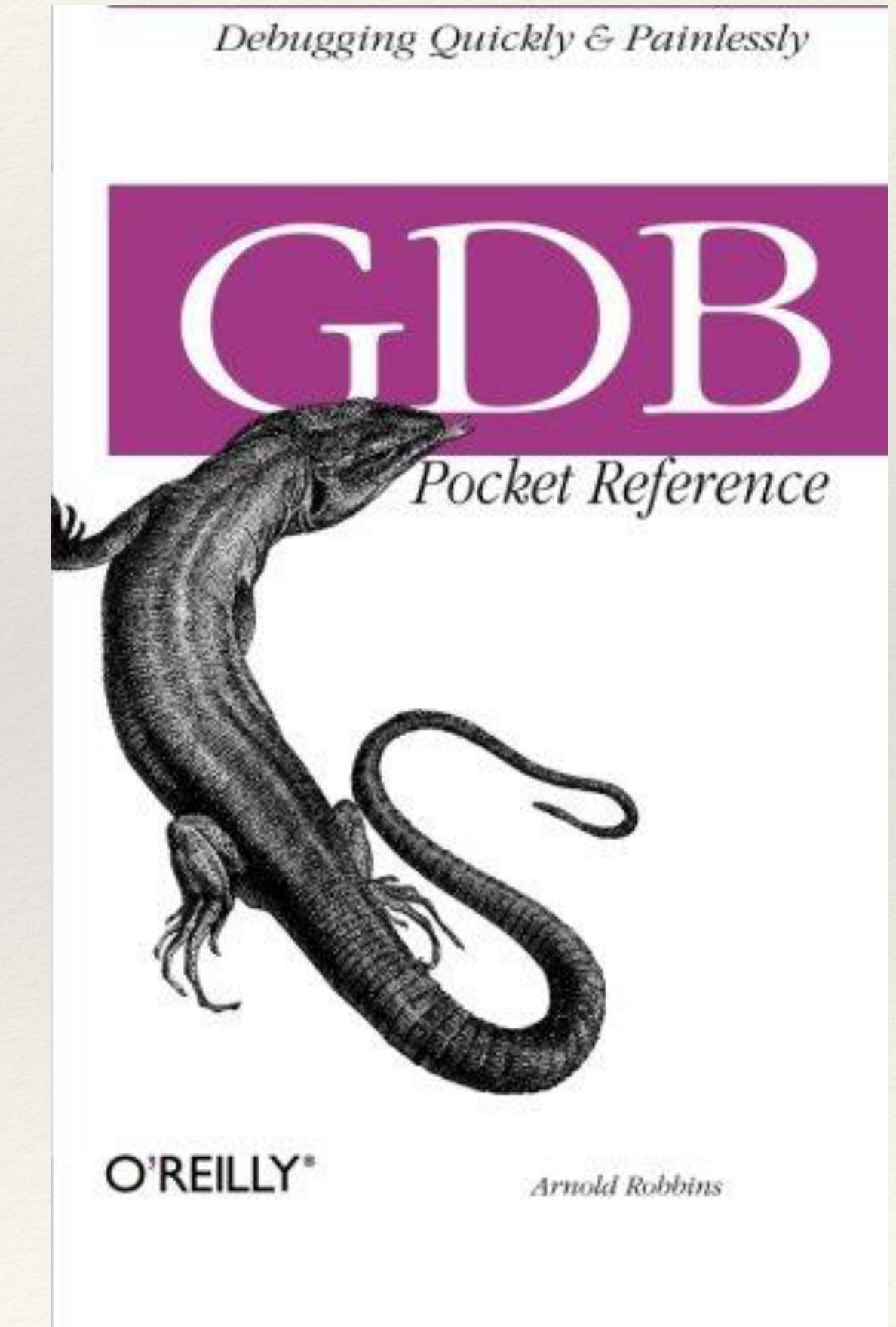
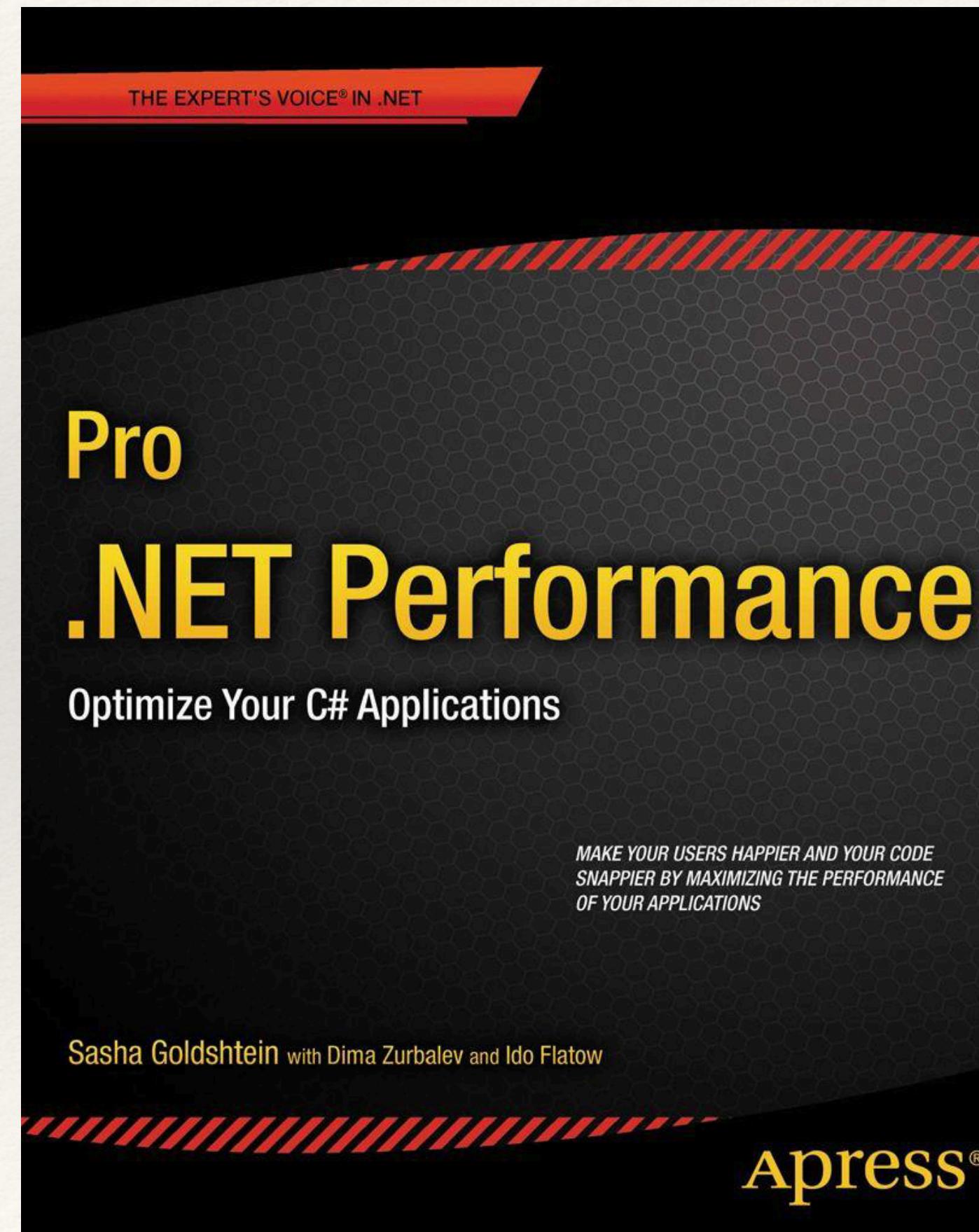
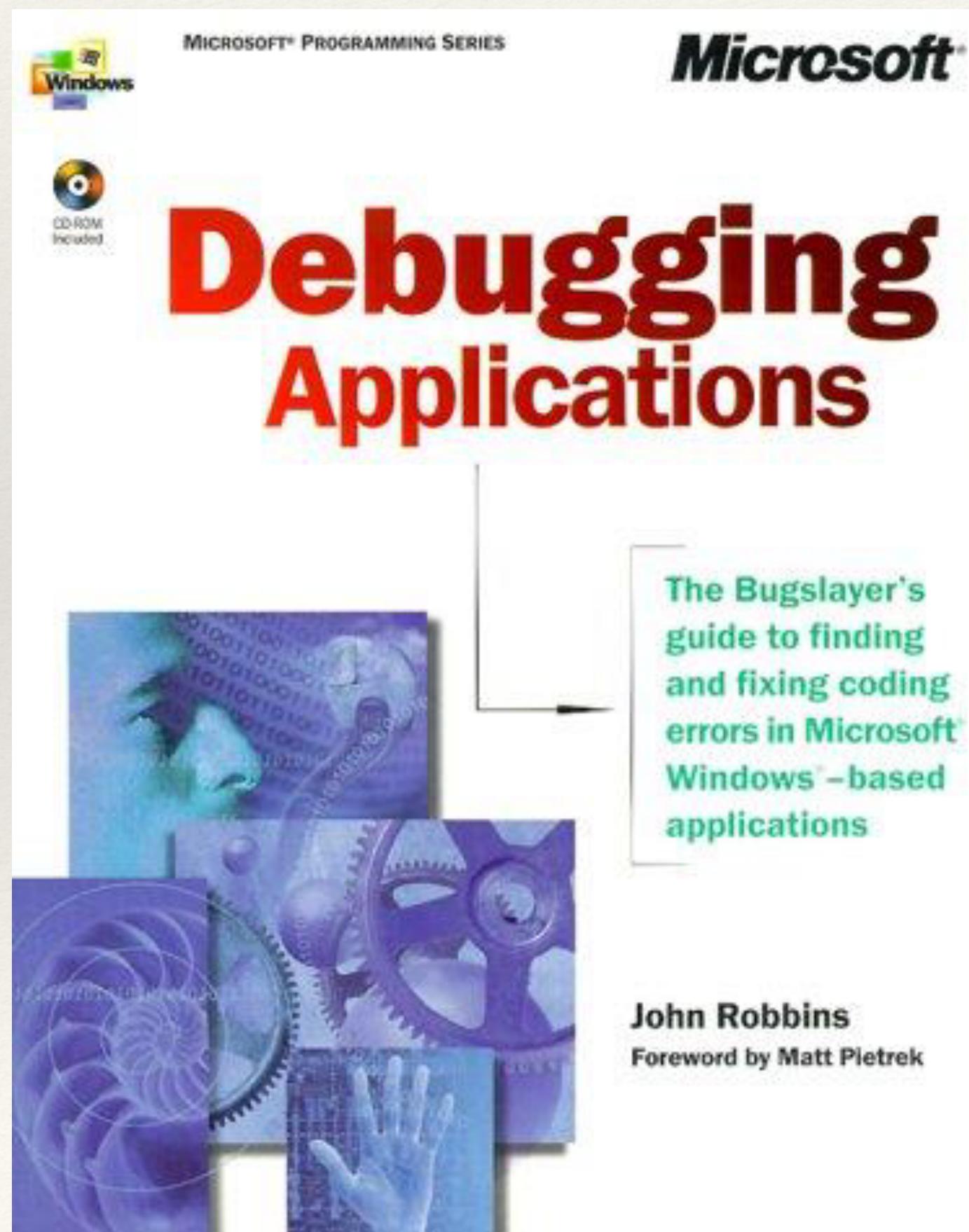
Литература



Talk about...

Литература

- ❖ <http://www.mono-project.com/docs/>



Ваши вопросы?

Елизавета Голенок

golenok-ea5@narod.ru

@marmothetka

