#WWDC19

What's New in Clang and LLVM

Jessica Paquette, Compiler Engineer
JF Bastien, Compiler Engineer
Devin Coughlin, Program Analysis Engineer

Agenda

Agenda

New platform support

Agenda

New platform support

Low-level code size optimizations

Agenda

New platform support

Low-level code size optimizations

Language-level code size optimizations

Agenda

New platform support

Low-level code size optimizations

Language-level code size optimizations

Diagnostics

Agenda

New platform support

Low-level code size optimizations

Language-level code size optimizations

Diagnostics

Static Analyzer checks

New Platform Support


14:59
11:15AM TRAINING WITH KRISTA
52
68°
89
MON
3
UVI
3.6
1:09PM, +3HRS
NYC


14:59
11:15AM TRAINING WITH KRISTA
52
68°
89
MON
3
UVI
3.6
1:09PM, +3HRS
NYC
Mon
3

Bitcode

What is LLVM Bitcode?

Source Code

doggo.c

snek.m

birb.swift

pupper.cc

What is LLVM Bitcode?

Source Code

doggo.c

snek.m

birb.swift

pupper.cc

What is LLVM Bitcode?

Source Code

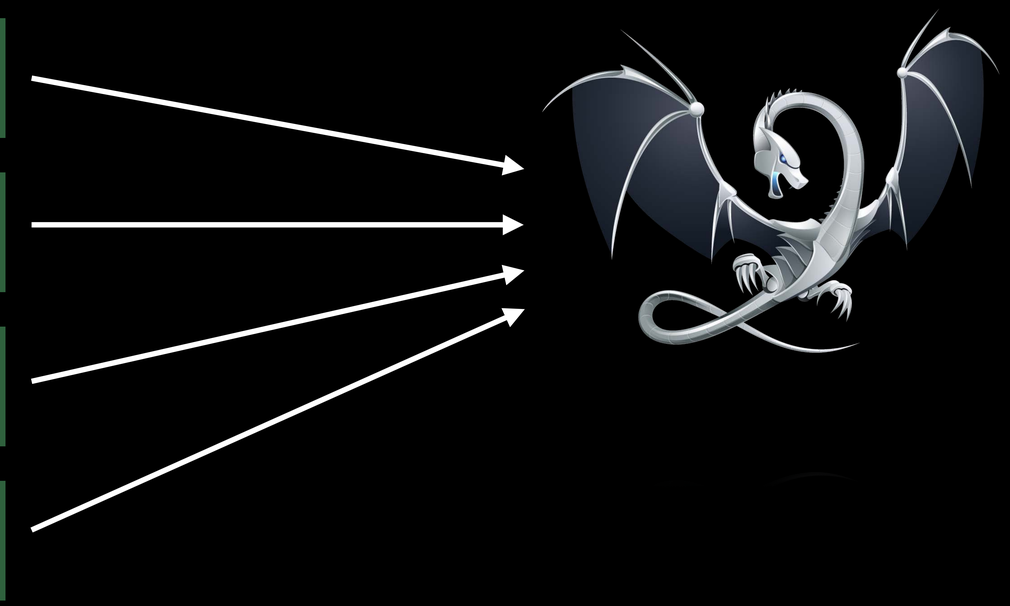
Compiler

doggo.c

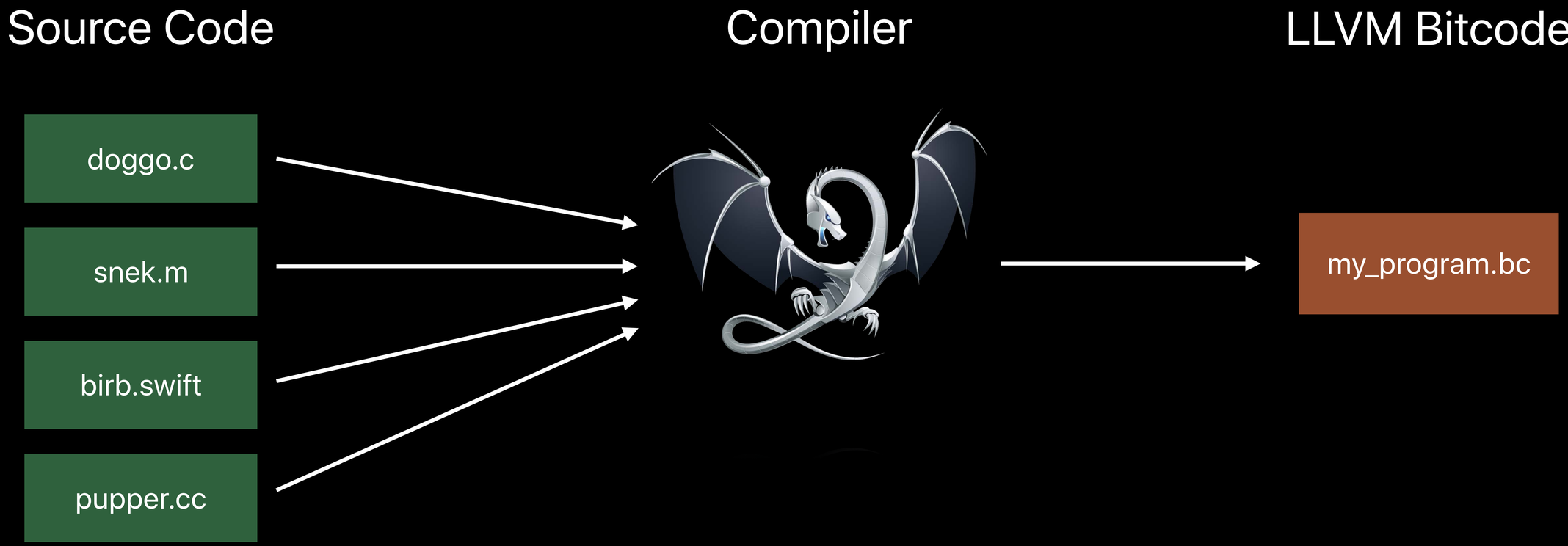
snek.m

birb.swift

pupper.cc

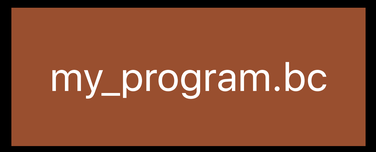


What is LLVM Bitcode?



What is LLVM Bitcode?

LLVM Bitcode



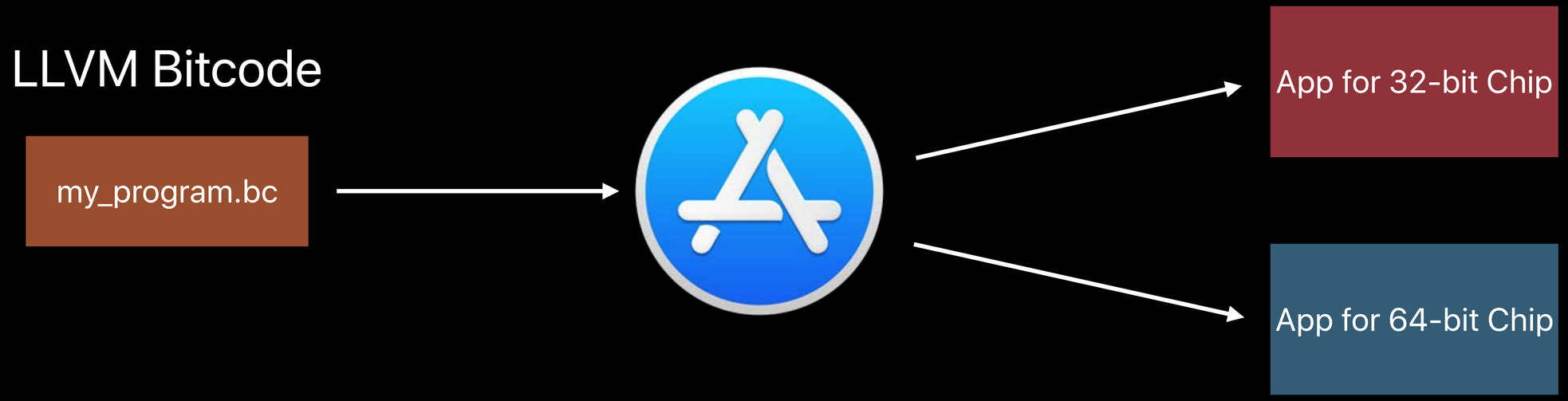
Serialization of internal compiler state

Producing Apps for Two Chips with Bitcode

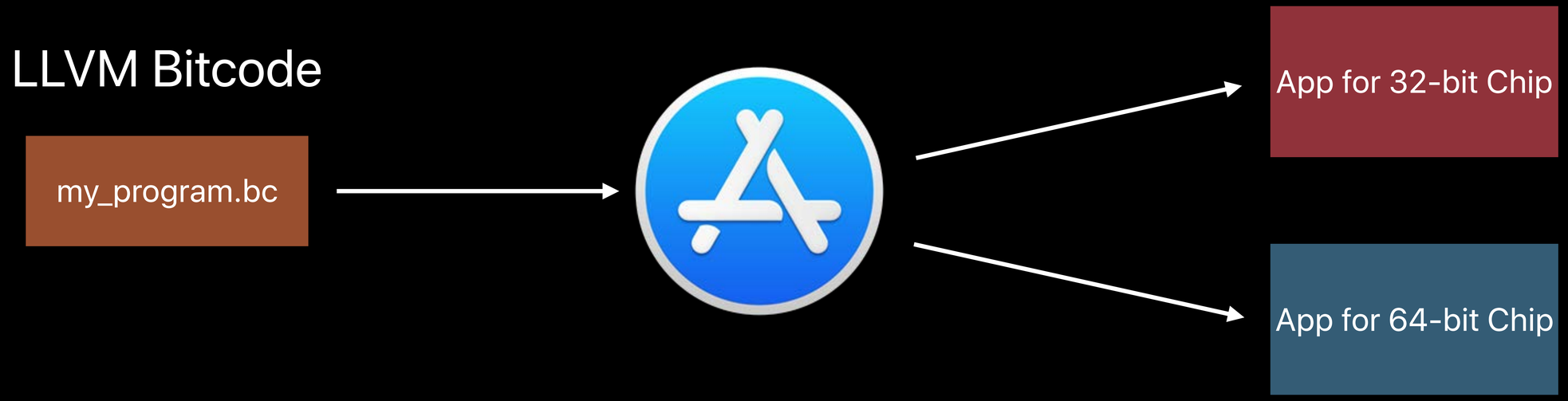
LLVM Bitcode

my_program.bc

Producing Apps for Two Chips with Bitcode



Producing Apps for Two Chips with Bitcode



Target-Tailored Bitcode = More Optimization

Bitcode for 32-bit Chip

Bitcode for 64-bit Chip

Code Size Improvements

Low-level code size optimizations

Why code size?

NEW

-Oz

-Oz

Jessica is Canadian, so she says “zed”

How does a compiler work?

```cpp
int collatz(int Num) {
    if (Num % 2 == 0)
        return Num / 2;
    return Num * 3 + 1;
}
```

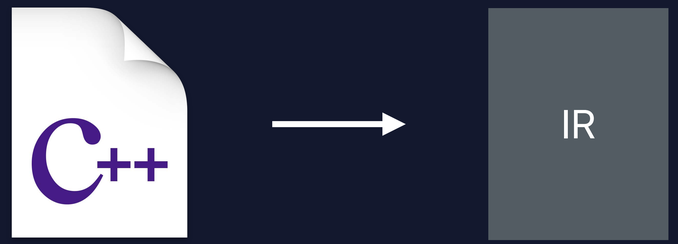
```
define i32 @collatz(i32) {
    if (Num % 2 == 0)
        return Num / 2;
    return Num * 3 + 1;
}
```

```
define i32 @collatz(i32) {
    %2 = and i32 %0, 1
    if (Num % 2 == 0)
        return Num / 2;
    return Num * 3 + 1;
}
```

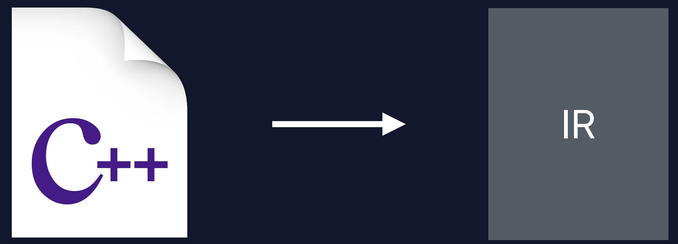
```
define i32 @collatz(i32) {
    %2 = and i32 %0, 1
    %3 = icmp eq i32 %2, 0
    %4 = sdiv i32 %0, 2
    return Num * 3 + 1;
}
```

```
define i32 @collatz(i32) {
    %2 = and i32 %0, 1
    %3 = icmp eq i32 %2, 0
    %4 = sdiv i32 %0, 2
    %5 = mul nsw i32 %0, 3
    %6 = add nsw i32 %5, 1
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```

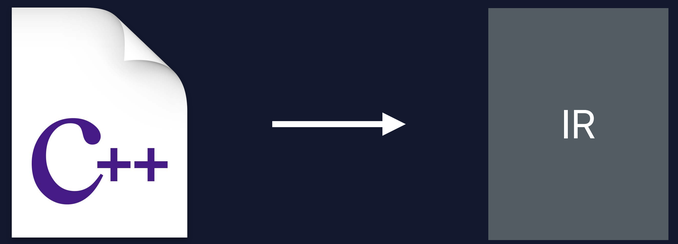
```
define i32 @collatz(i32) {
    %2 = and i32 %0, 1
    %3 = icmp eq i32 %2, 0
    %4 = sdiv i32 %0, 2
    %5 = mul nsw i32 %0, 3
    %6 = add nsw i32 %5, 1
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```



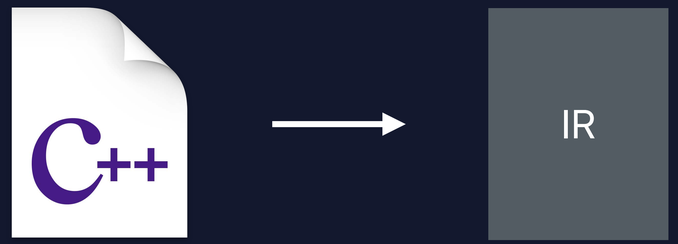
```
name:            collatz
body:             |
  bb.0:
    %2 = and i32 %0, 1
    %3 = icmp eq i32 %2, 0
    %4 = sdiv i32 %0, 2
    %5 = mul nsw i32 %0, 3
    %6 = add nsw i32 %5, 1
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```



```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %3 = icmp eq i32 %2, 0
    %4 = sdiv i32 %0, 2
    %5 = mul nsw i32 %0, 3
    %6 = add nsw i32 %5, 1
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```



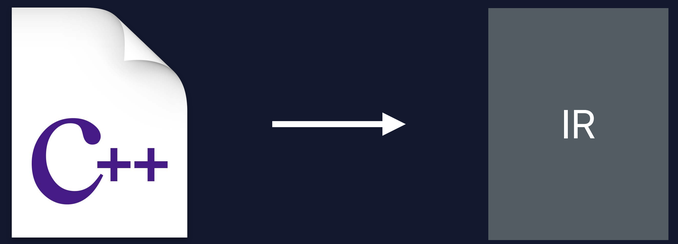
```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %5:gpr32 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %5 = mul nsw i32 %0, 3
    %6 = add nsw i32 %5, 1
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```



```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %5:gpr32 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %7:gpr32 = MOVi32imm 3
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %7 = select i1 %3, i32 %4, i32 %6
    ret i32 %7
}
```

```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %5:gpr32 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %7:gpr32 = MOVi32imm 3
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    ret i32 %7
}
```

```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %5:gpr32 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %7:gpr32 = MOVi32imm 3
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```



```
name:            collatz
body:             |
  bb.0:
    %0:gpr32 = COPY $w0
    %5:gpr32 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %7:gpr32 = MOVi32imm 3
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    %6:gpr32 = SDIVWr %0, %5
    %7:gpr32 = MOVi32imm 3
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    $w9 = MOVi32imm 3
    %6:gpr32 = SDIVWr %0, %5
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    $w9 = MOVi32imm 3
    $w8 = SDIVWr $w0, $w8
    %8:gpr32common = MADDWrrr %0, %7, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    $w9 = MOVi32imm 3
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    %10:gpr32 = CSELWr %6, %9, 1, implicit $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    $w9 = MOVi32imm 3
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET_ReallyLR implicit $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVi32imm 2
    $w9 = MOVi32imm 3
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET_ReallyLR $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVZWi 2, 0
    $w9 = MOVi32imm 3
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET_ReallyLR $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVZWi 2, 0
    $w9 = MOVZWi 3, 0
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET_ReallyLR $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVZWi 2, 0
    $w9 = MOVZWi 3, 0
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET $lr, $w0
```

```
name:            collatz
body:             |
  bb.0:
    $w8 = MOVZWi 2, 0
    $w9 = MOVZWi 3, 0
    $w8 = SDIVWr $w0, $w8
    $w9 = MADDWrrr $w0, $w9, $wzr
    $w9 = ADDWri $w9, 1, 0
    $w0 = CSELWr $w8, $w9, 1, $nzcv
    RET $lr, $w0
```

```
collatz:
    mov   w8, #2
    mov   w9, #3
    sdiv  w8, w0, w8
    mul   w9, w0, w9
    add   w9, w9, #1
    csel  w0, w8, w9, ne
    ret
```

Function Outlining

An -Oz code size optimization

```
// These two functions share some instructions

hasse:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret

kakutani:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

```
// Create a new function using the shared instructions…

hasse:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret

kakutani:
  …
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

```
OUTLINED_FUNCTION_0:
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

```
// Replace the repeated sequences with calls to the new function

hasse:
  …
  b OUTLINED_FUNCTION_0

kakutani:
  …
  b OUTLINED_FUNCTION_0
```

```
OUTLINED_FUNCTION_0:
  ldr   w0, [sp, #16]
  mul   w0, w1, w2
  add   sp, sp, #16
  ret
```

25%

Where do the savings come from? 🤔

```
// What does the compiler have to do to represent this function in assembly?

int ulam(int Num, int NumIters) {
    // TODO: Does this always converge?
    while (Num != 1) {
        Num = collatz(Num);
        ++NumIters;
    }
    return NumIters;
}
```

```
// What does the compiler have to do to represent this function in assembly?

int ulam(int Num, int NumIters) {
    // TODO: Does this always converge?
    while (Num != 1) {
        Num = collatz(Num);
        ++NumIters;
    }
    return NumIters;
}
```

```
ulam:
        stp     x20, x19, [sp, #-32]!
        stp     x29, x30, [sp, #16]
        add     x29, sp, #16
        b       LBB1_2
LBB1_1:
        bl      collatz
        add     w19, w19, #1
LBB1_2:
        cmp     w0, #1
        b.ne    LBB1_1
        mov     x0, x19
        ldp     x29, x30, [sp, #16]
        ldp     x20, x19, [sp], #32
        ret
```

```
// What does the compiler have to do to represent this function in assembly?
```

Prologue

Epilogue

```
ulam:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    add     x29, sp, #16
    b       LBB1_2
LBB1_1:
    bl      collatz
    add     w19, w19, #1
LBB1_2:
    cmp     w0, #1
    b.ne    LBB1_1
    mov     x0, x19
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret
```

```
// Compiler introduces similarity into code
```

```
collatz:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    …
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret

kakutani:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    …
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret
```

```
ulam:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    add     x29, sp, #16
    b       LBB1_2
LBB1_1:
    bl      collatz
    add     w19, w19, #1
LBB1_2:
    cmp     w0, #1
    b.ne    LBB1_1
    mov     x0, x19
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret
```

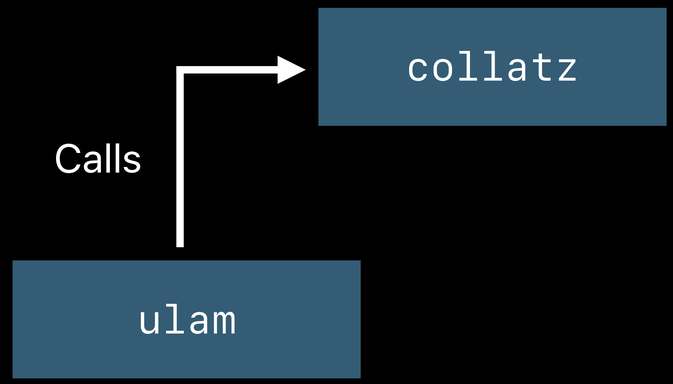
```
// Compiler introduces similarity into code

collatz:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    …
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret

kakutani:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    …
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret
```

```
ulam:
    stp     x20, x19, [sp, #-32]!
    stp     x29, x30, [sp, #16]
    add     x29, sp, #16
    b       LBB1_2
LBB1_1:
    bl      collatz
    add     w19, w19, #1
LBB1_2:
    cmp     w0, #1
    b.ne    LBB1_1
    mov     x0, x19
    ldp     x29, x30, [sp, #16]
    ldp     x20, x19, [sp], #32
    ret
```

Gotchas

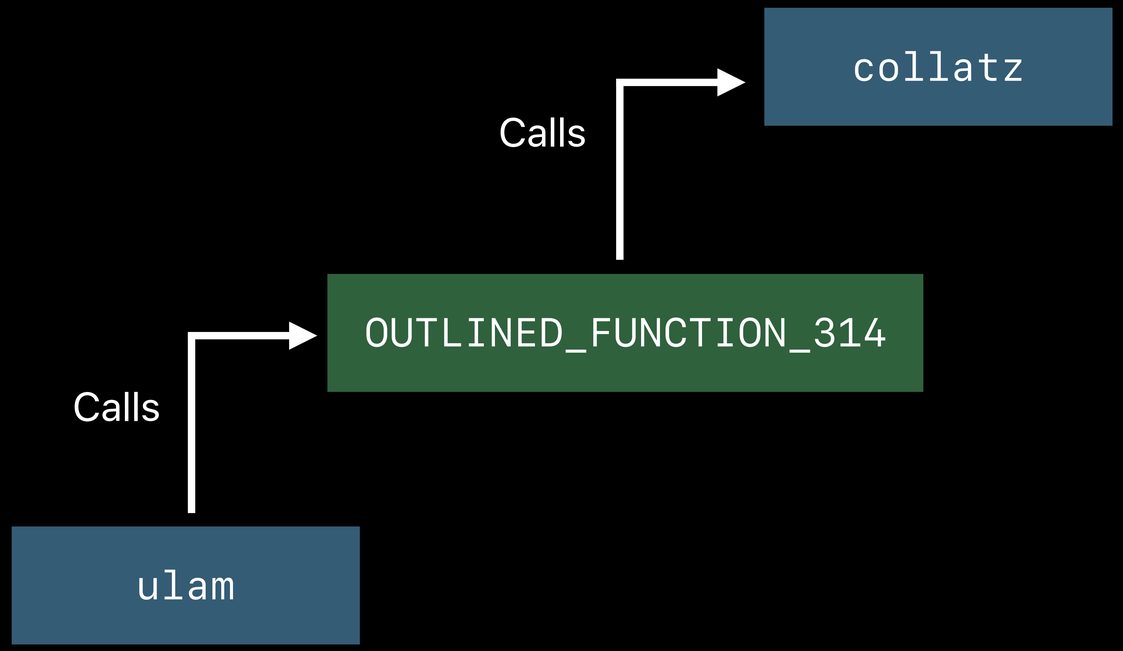
Outlining Changes Control Flow

```
ulam:
    …
    bl      collatz
    add     w19, w19, #1
    …
    ret
```



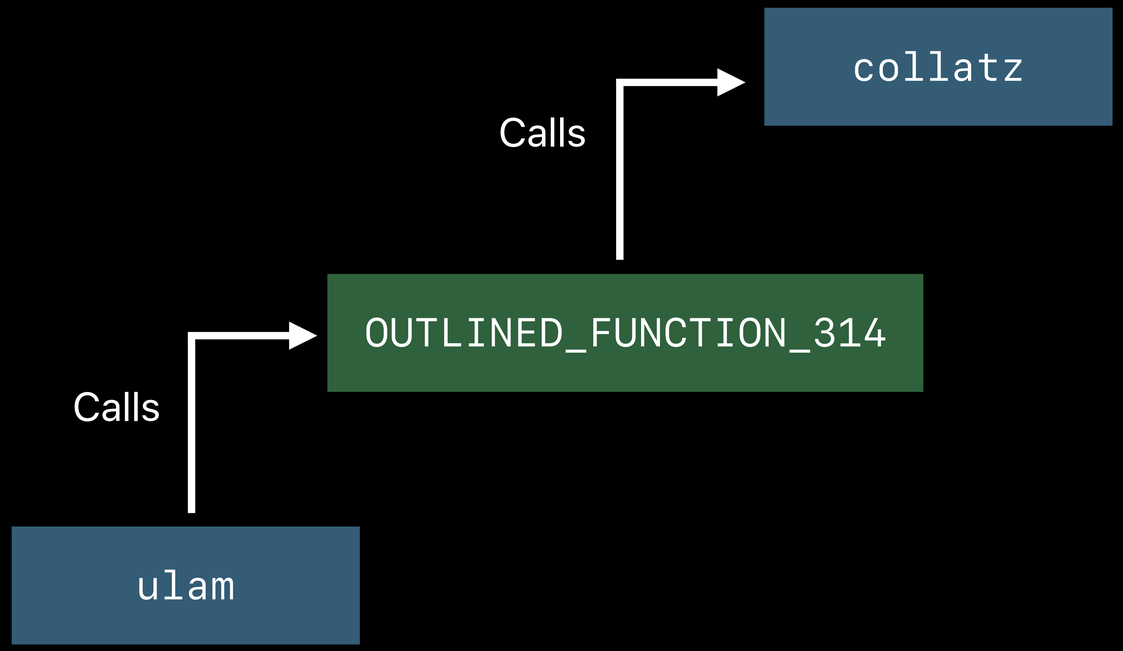
Outlining Changes Control Flow

```
ulam:
    …
    bl      OUTLINED_FUNCTION_314
    …
    ret
```

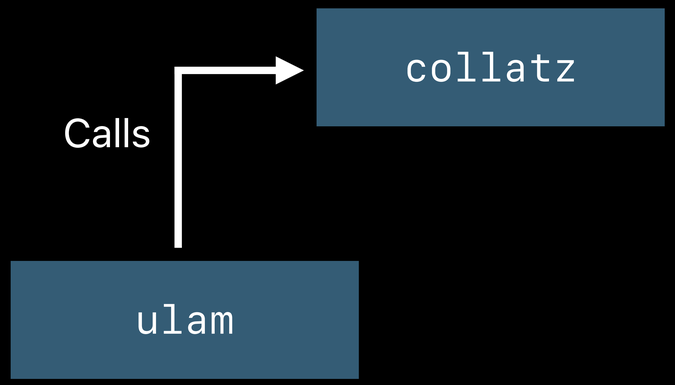


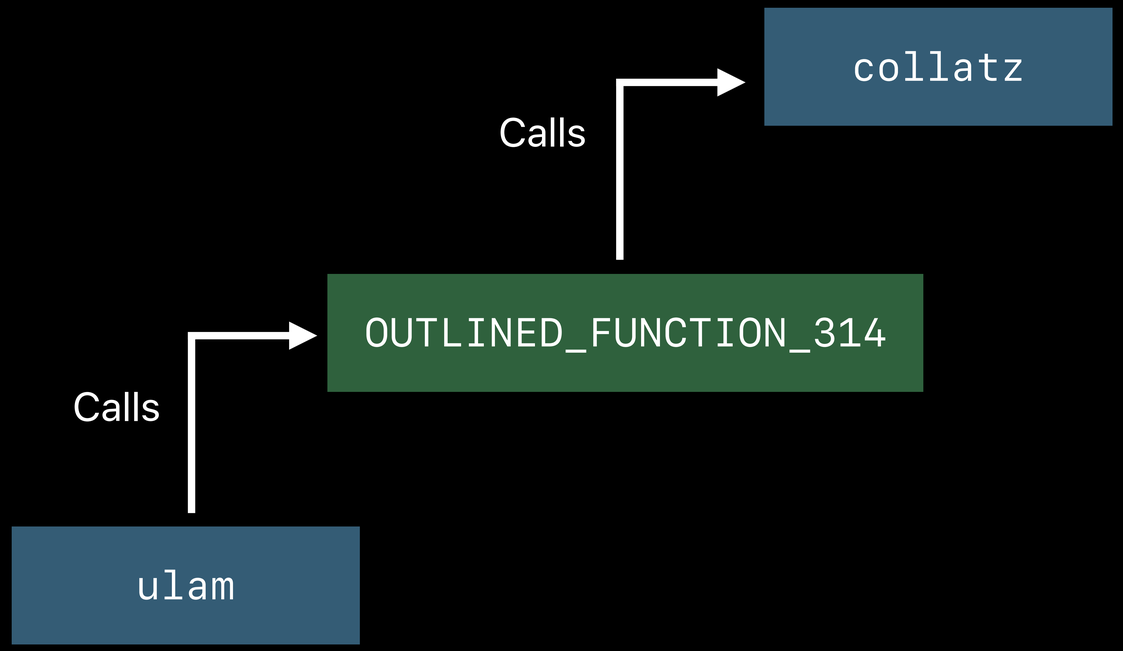
Outlining Changes Control Flow

```
ulam:
        …
        bl      OUTLINED_FUNCTION_314
        …
        ret
```

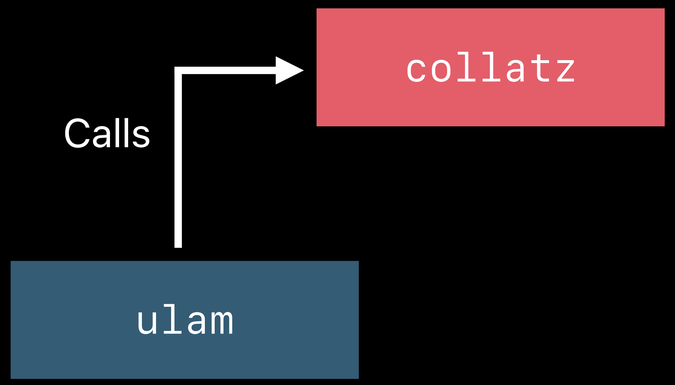


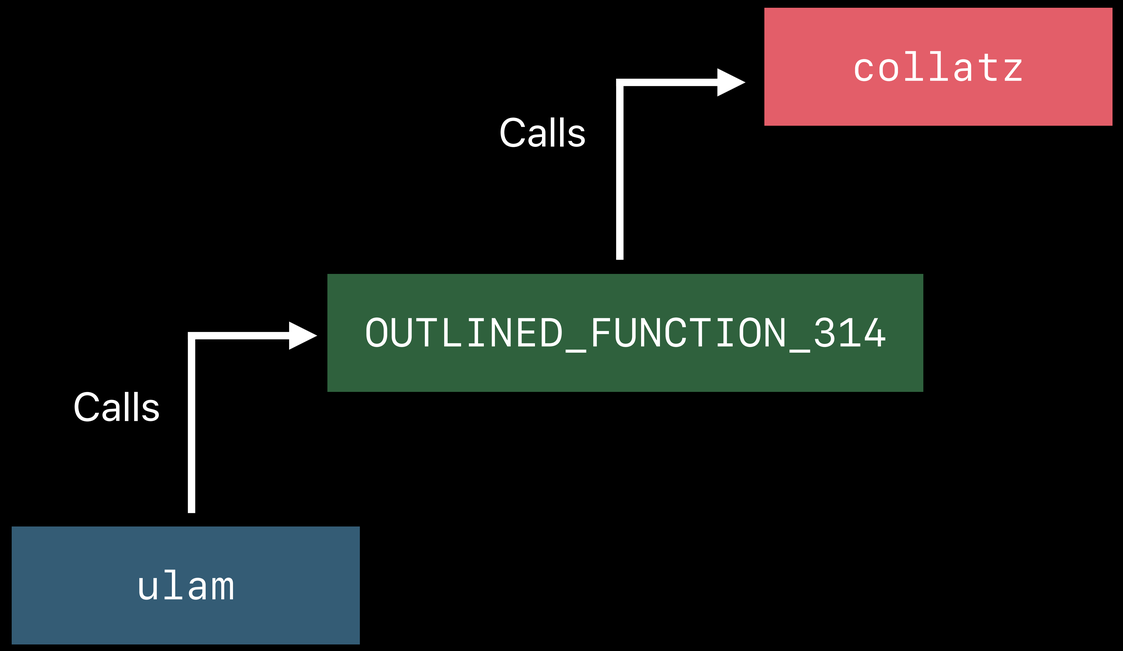
Crashing with Outlined Code





Crashing with Outlined Code





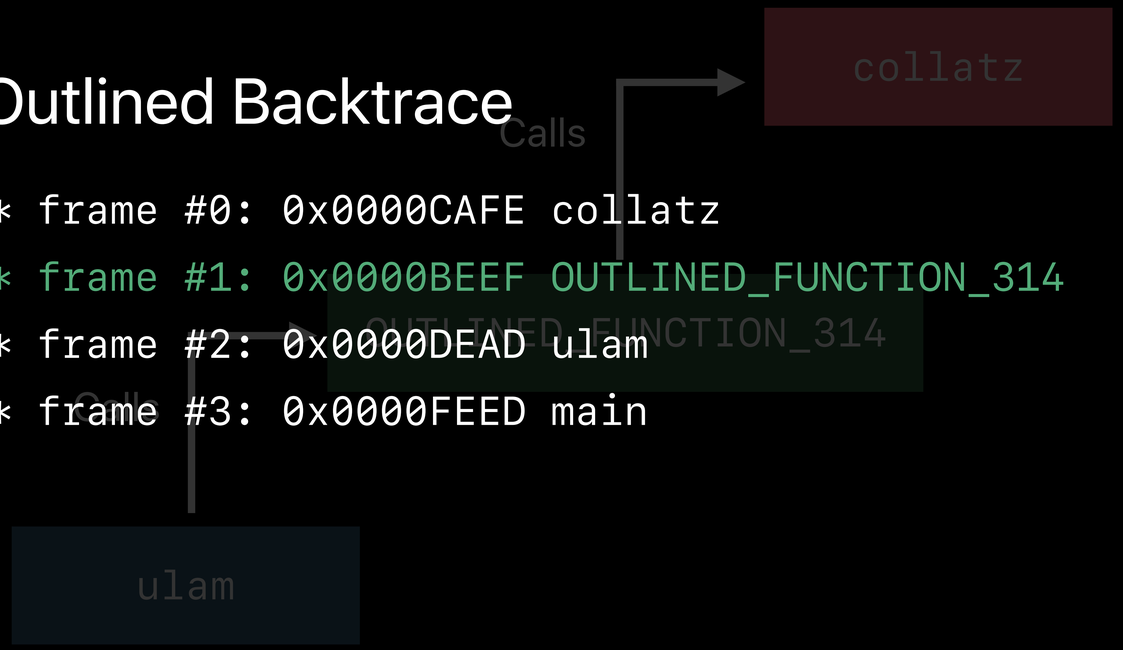
Outlining Can Impact Backtraces

Original Backtrace

```
* frame #0: 0x0000BEEF collatz
* frame #1: 0x0000DEAD ulam
* frame #2: 0x0000FEED main
```

Outlined Backtrace

```
* frame #0: 0x0000CAFE collatz
* frame #1: 0x0000BEEF OUTLINED_FUNCTION_314
* frame #2: 0x0000DEAD ulam
* frame #3: 0x0000FEED main
```



Outlining Can Increase Execution Time

Outlining Can Increase Execution Time

Calls can have execution time overhead

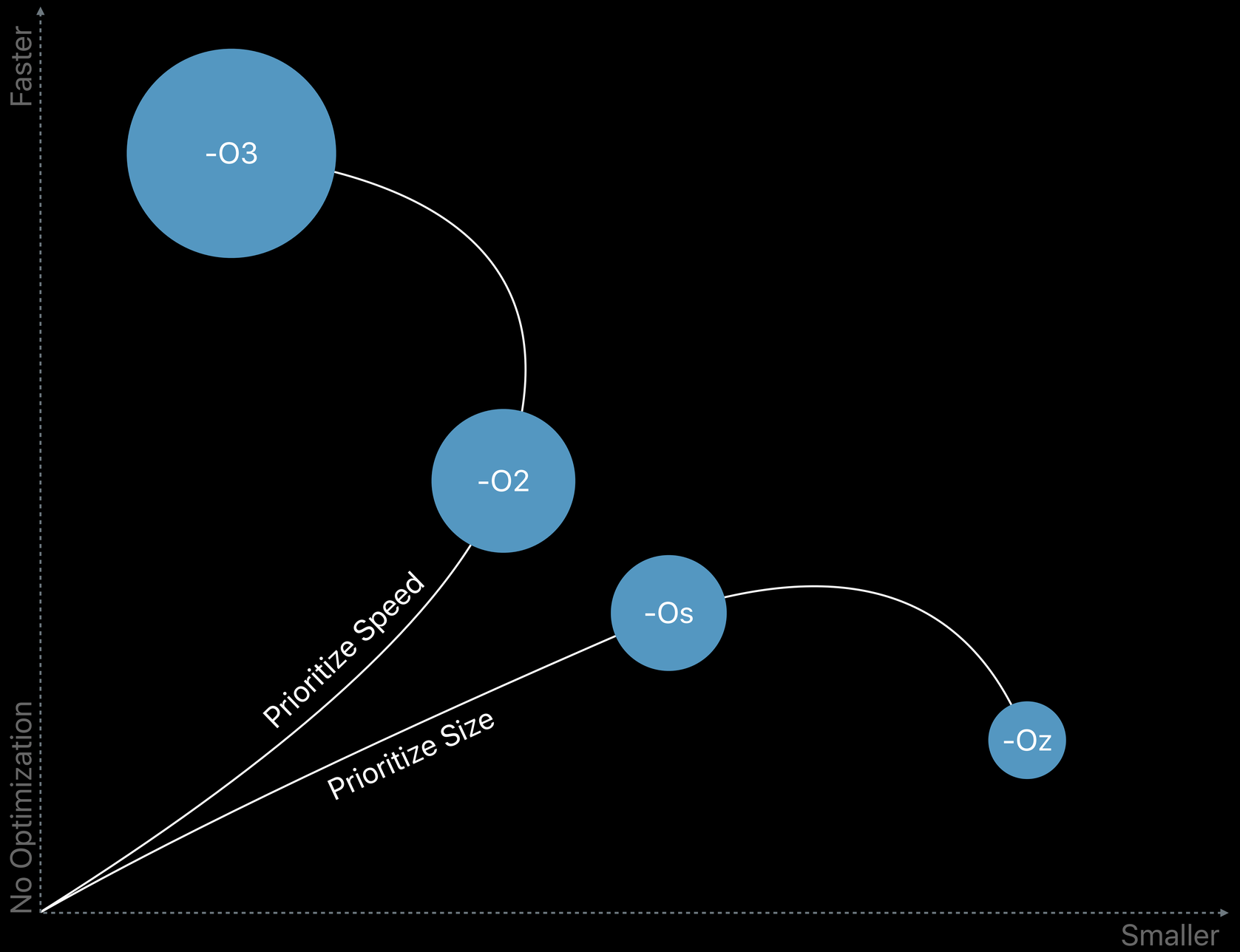
Outlining Can Increase Execution Time

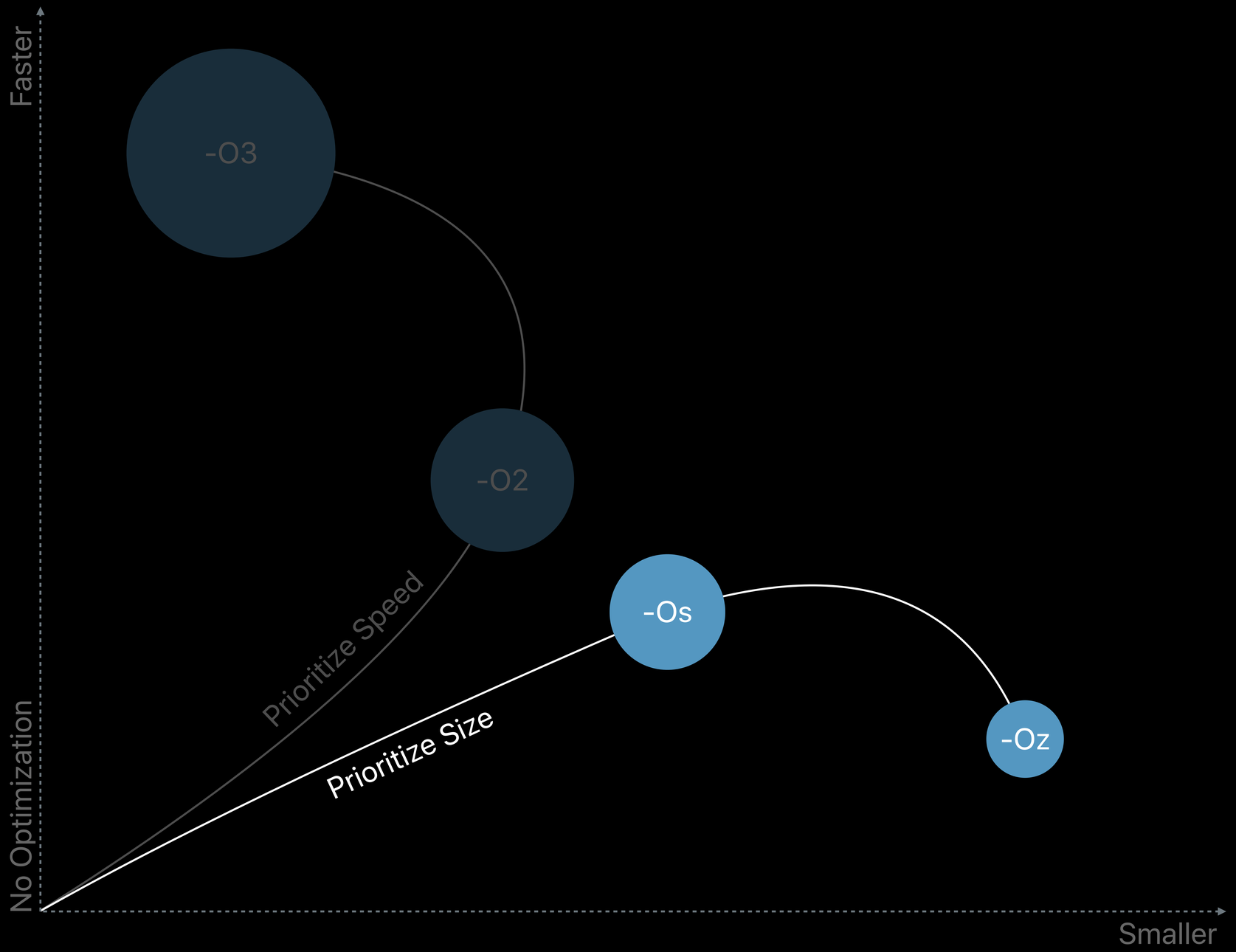
Calls can have execution time overhead

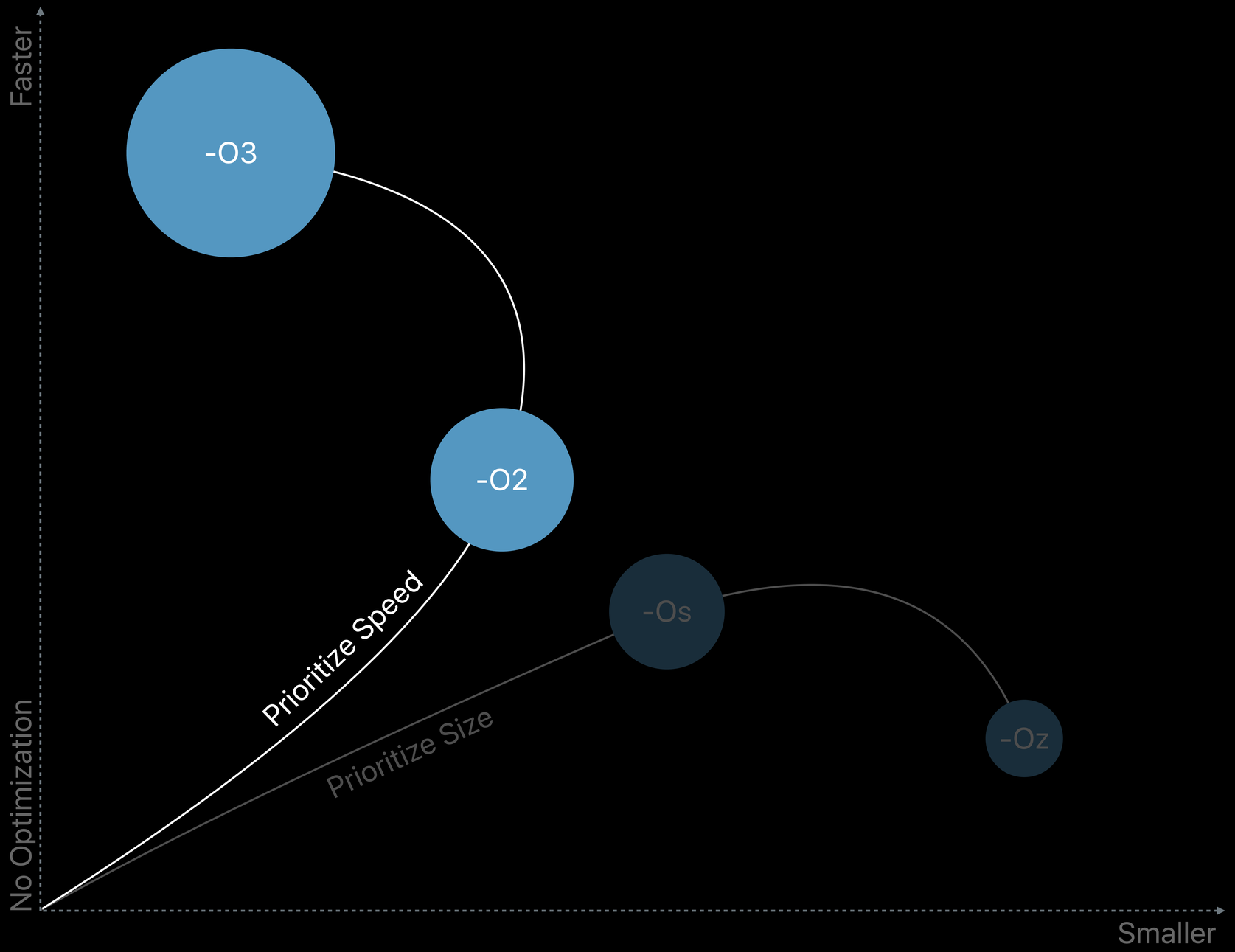
-Oz prioritizes size at all costs

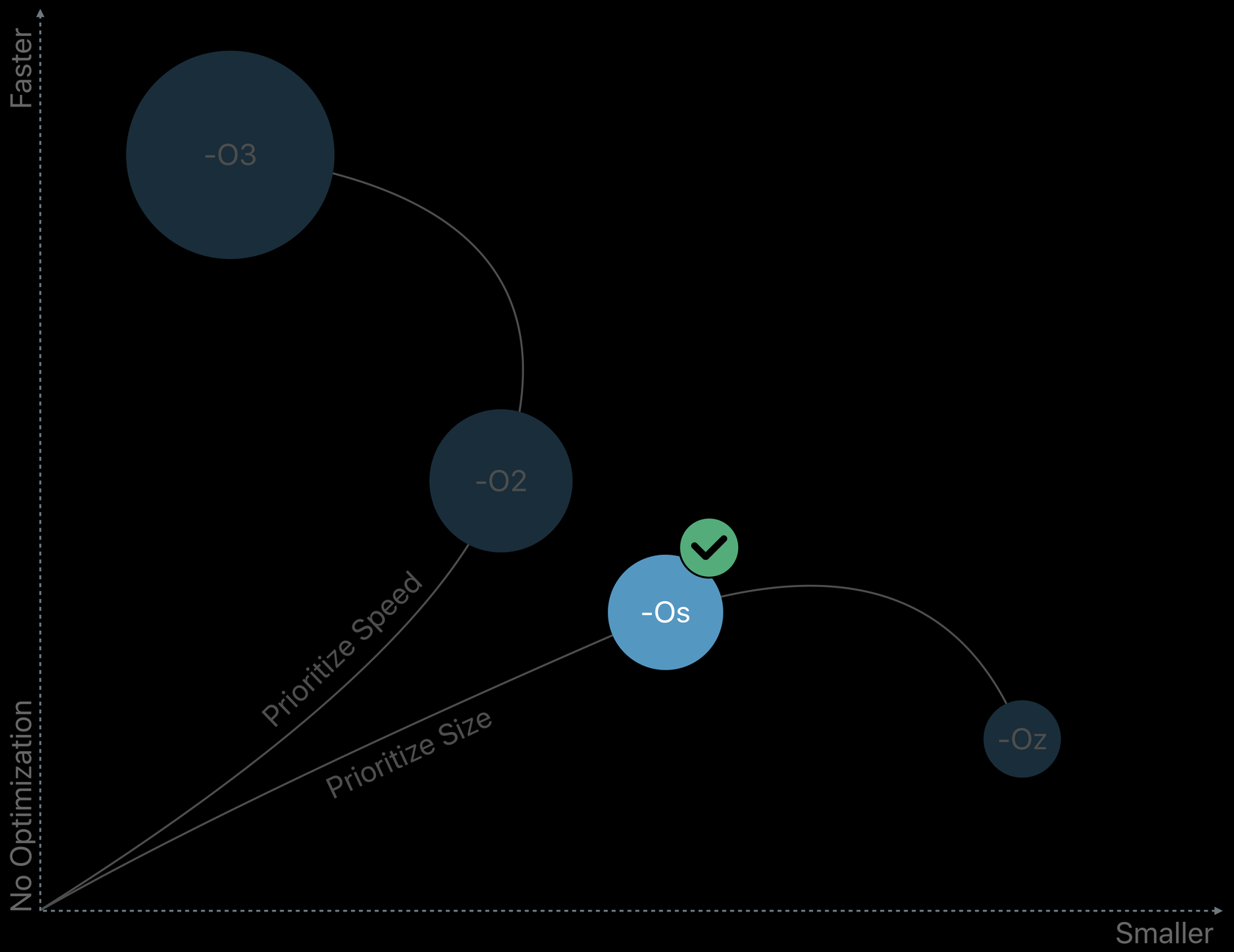
Don’t compile performance-sensitive code with -Oz!

Use Instruments!


Faster
-O3
-O2
-Os
-Oz
Prioritize Speed
Prioritize Size
No Optimization
Smaller


Faster
No Optimization
Smaller
-O3
-O2
-Os
-Oz
Prioritize Speed
Prioritize Size


Faster
-O3
-O2
Prioritize Speed
-Os
-Oz
Prioritize Size
No Optimization
Smaller


Faster
No Optimization
Smaller
-O3
-O2
-Os
-Oz
Prioritize Speed
Prioritize Size

Extra Optimizations

Extra Optimizations

PGO — Profile-guided optimization

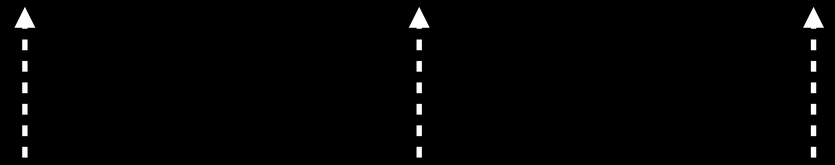
Extra Optimizations

PGO — Profile-guided optimization

LTO — Link-time optimization

Combining Optimizations

LTO + PGO + -O3



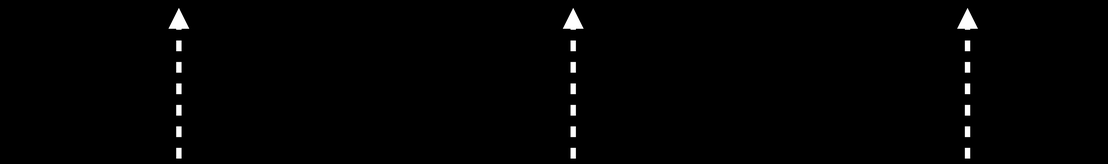
Wait until link-time to optimize

Use profiling information

Prioritize execution time

Combining Optimizations

LTO + PGO + -O3



Wait until link-time to optimize

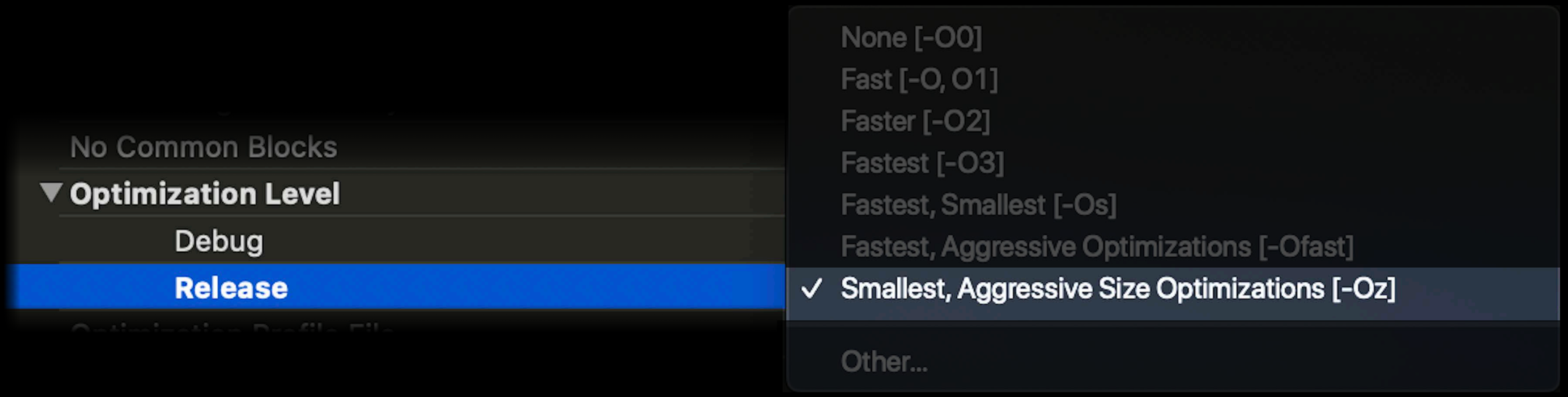
Use profiling information

Prioritize execution time

Enable -Oz in Your Project's Build Settings

C, C++, and Objective-C Projects

NEW



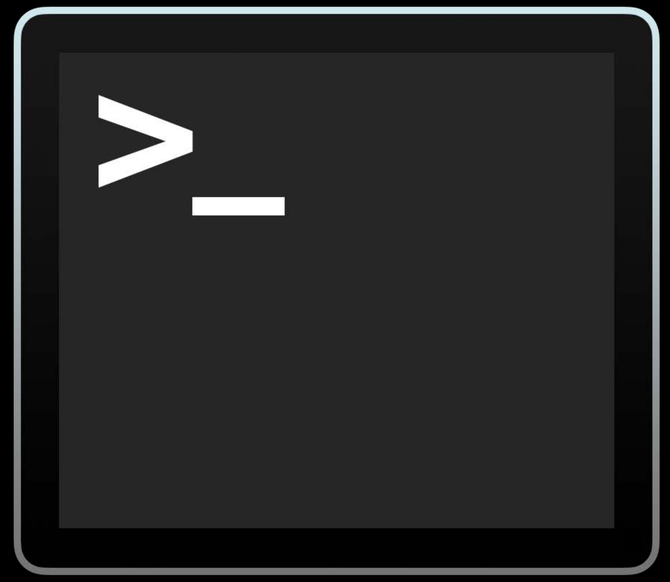
Enable -Oz on Specific Files

C, C++, and Objective-C Files

Build Phases > Compile Sources > Compiler Flags

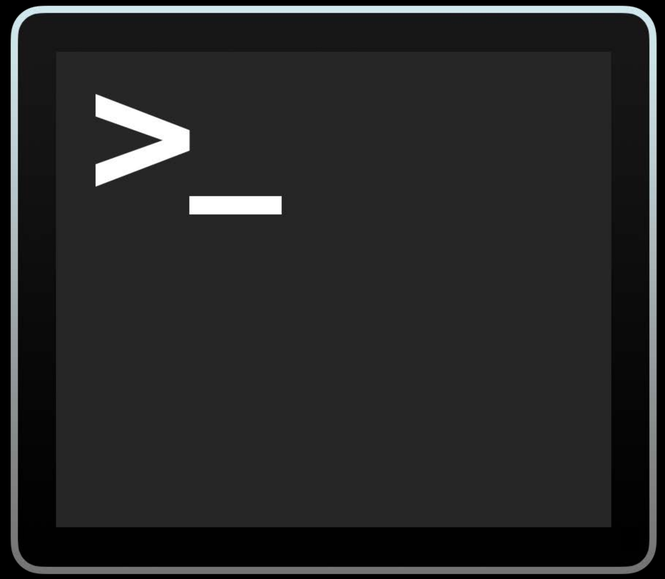
How does this impact app code size?

size



size

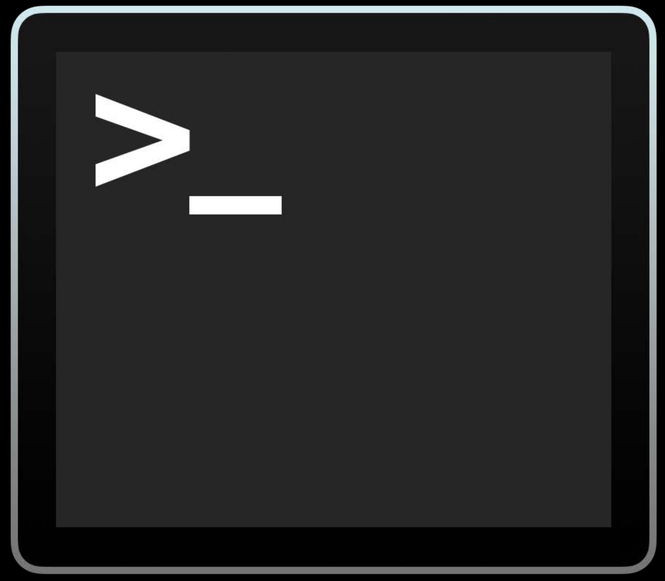
Executable binary code size information



`size`

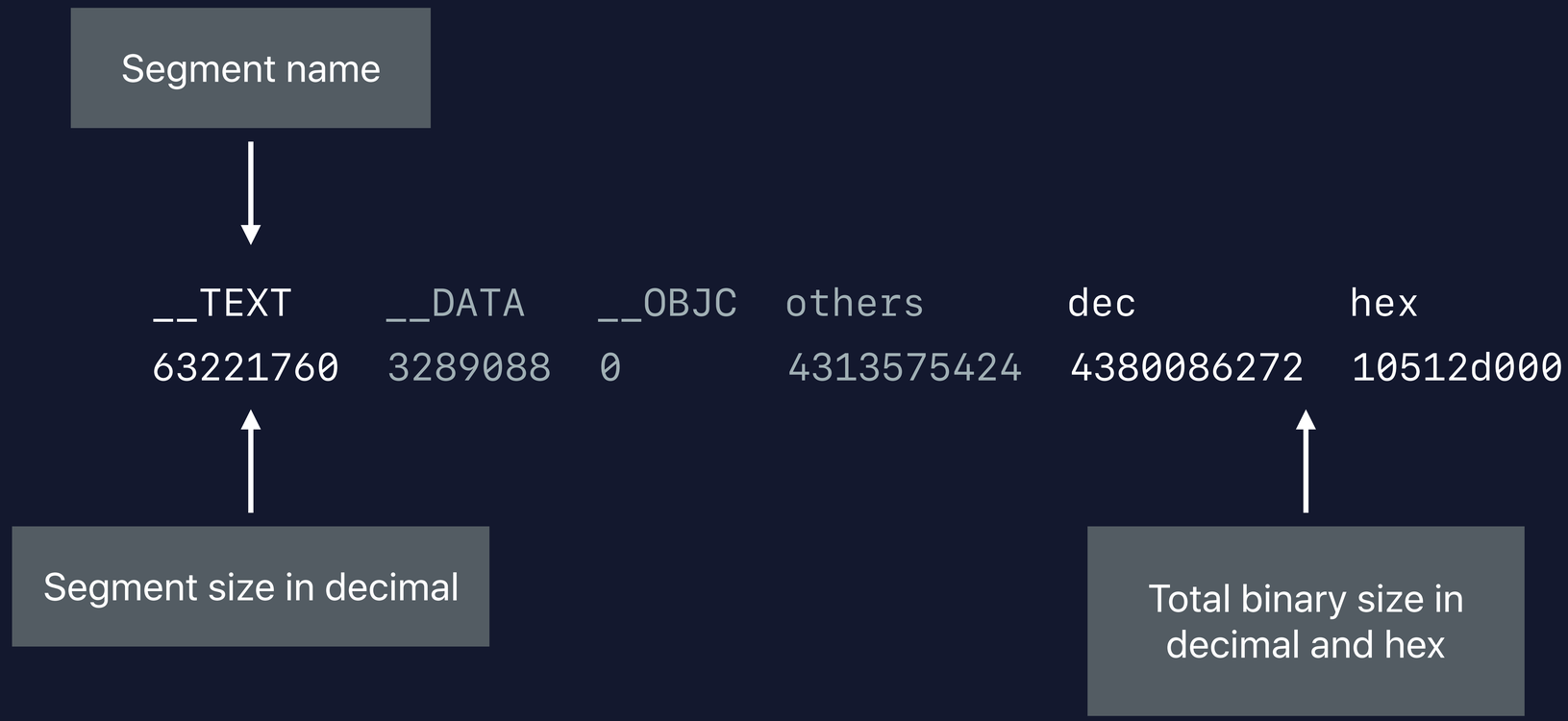
Executable binary code size information

⚠️ Not the total size of the app



```
# Example: Finding Code Size Info Using size

$ size ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```

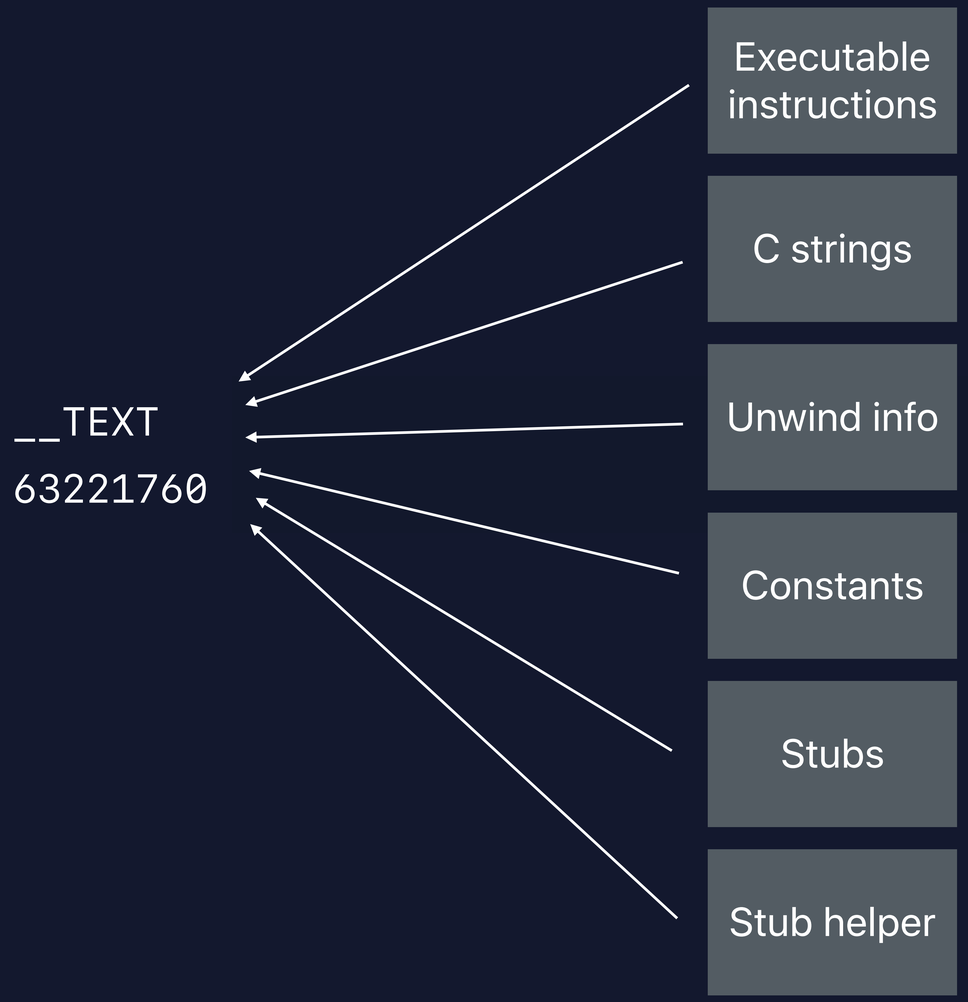
```
# Example: Finding Code Size Info Using size

$ size ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```



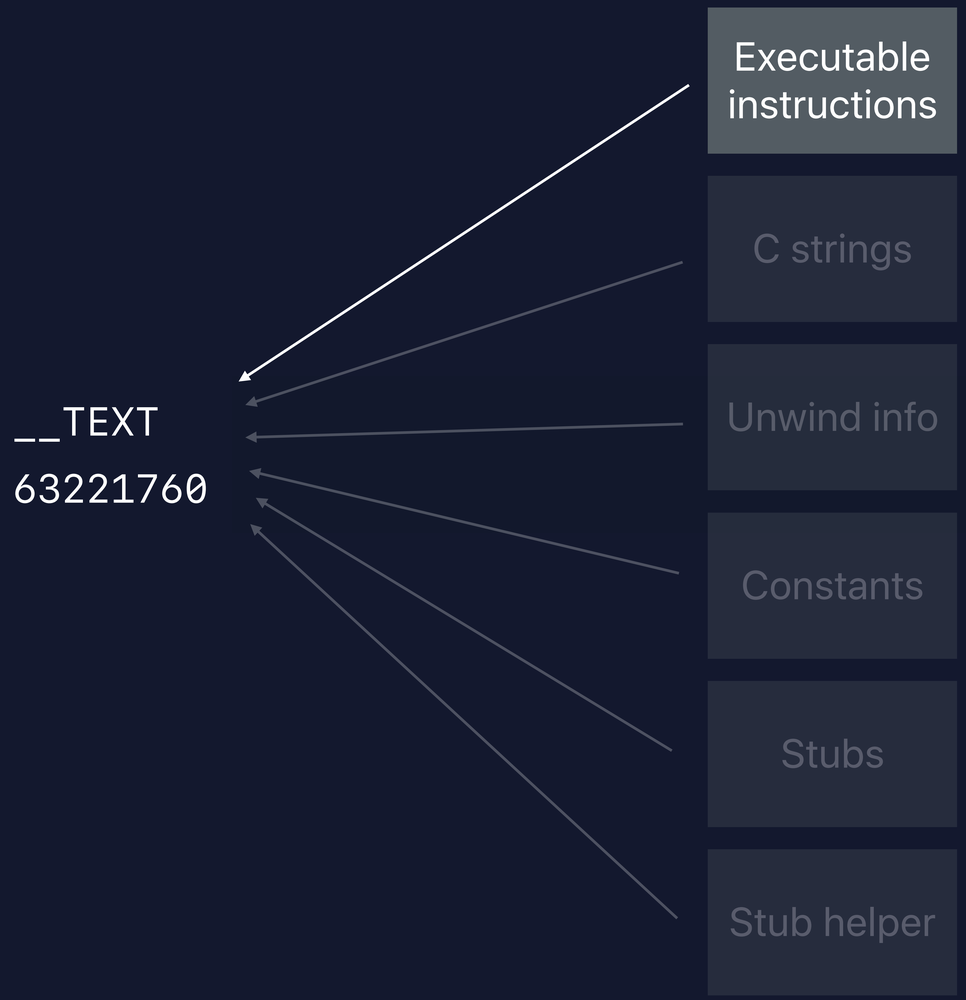
```
# Example: Finding Code Size Info Using size

$ size ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```

```
__TEXT
63221760
```

```
# Example: Finding Code Size Info Using size

$ size ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```

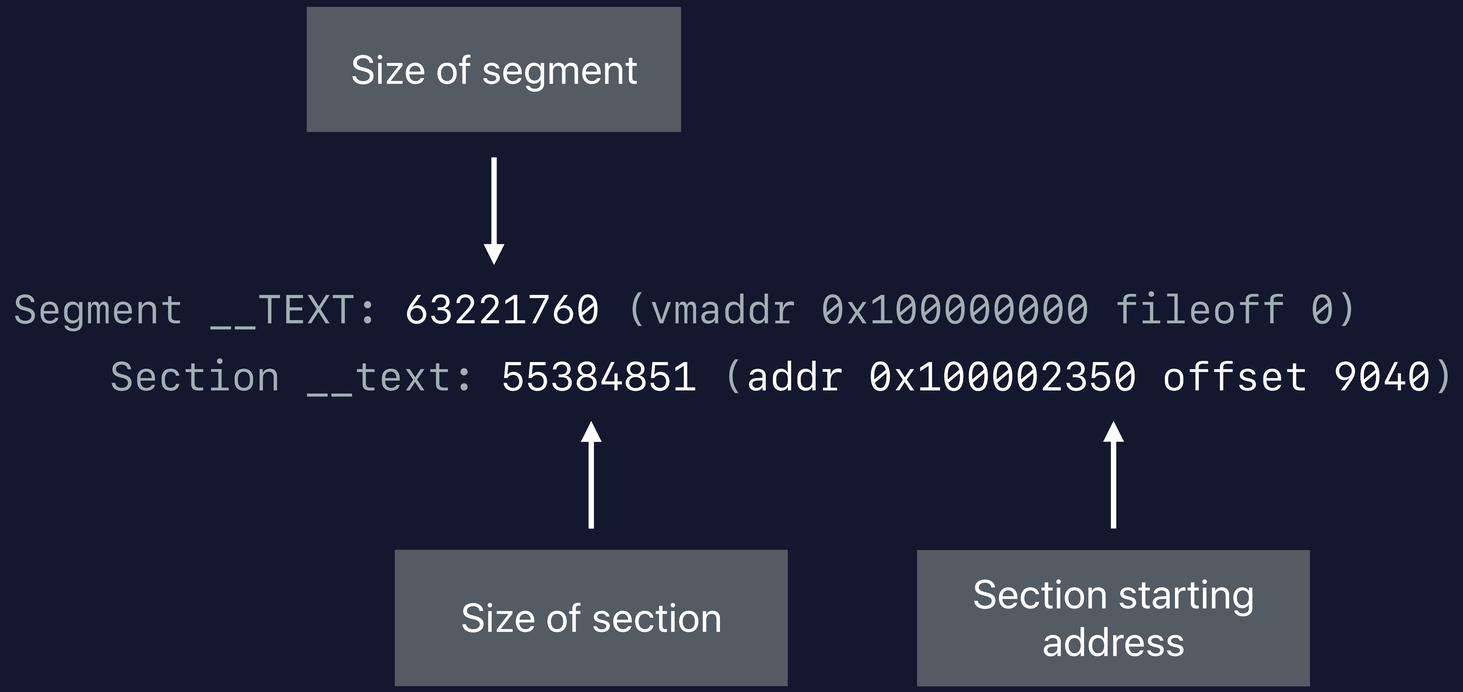


```
# Example: Finding Code Size Info Using size

$ size ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```



```
# Example: Detailed Code Size Info Using size

$ size -l -m ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```

```
# Example: Detailed Code Size Info Using size

$ size -l -m ~/Library/Developer/Xcode/…/Sniffo.app/Contents/MacOS/Sniffo
```



Code Size Improvements

Language-level optimizations

JF Bastien, Compiler Engineer

Merge redundant block helpers and metadata

```
// Merge redundant block helpers and metadata
```

```objc
// Merge redundant block helpers and metadata


- (void)neutron:(id)particle {
  [self fuseWithCallbackBlock:^ (id nuclei) {
    [nuclei collide:particle];
    [self upDownDown];
  }];
}

- (void)proton:(id)particle chargeQuantity:(double)charge {
  [self fuseWithCallbackBlock:^ (id nuclei) {
    [self checkCoulombForce:charge];
    [particle collide:nuclei];
  }];
}
```

```
// Merge redundant block helpers and metadata
```

```
struct Metadata {
  unsigned long reserved, block_size;
  void          *copy_helper;
  void          *destroy_helper;
  const char    *block_method_signature;
  uintptr_t      block_layout_info;
};
```

```
- (void)neutron:(id)particle {
  [self fuseWithCallbackBlock:^ (id nuclei) {
    [nuclei collide:particle];
    [self upDownDown];
  }];
}

- (void)proton:(id)particle chargeQuantity:(double)charge {
  [self fuseWithCallbackBlock:^ (id nuclei) {
    [self checkCoulombForce:charge];
    [particle collide:nuclei];
  }];
}
```

```
// Generated block metadata — as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

```
// Generated block metadata — as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void          *copy_helper             = ___copy_helper_block_ea8_32s40s;
  void          *destroy_helper          = ___destroy_helper_block_ea8_32s40s;
  const char    *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t      block_layout_info       = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void          *copy_helper             = ___copy_helper_block_ea8_32s40s;
  void          *destroy_helper          = ___destroy_helper_block_ea8_32s40s;
  const char    *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t      block_layout_info       = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

🗣️ **48 ≠ 52 ∴ can't merge**

```
// Generated block metadata — as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

```
// Generated block metadata — as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void         *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void         *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char   *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t     block_layout_info      = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

```
// Generated block metadata — as of Xcode 11

static const char  *__block_method_signature_v16_0_8 = "v16@?0@8";
static const struct { // neutron block metadata
  unsigned long reserved = 0, block_size = 48;
  void          *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void          *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char    *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t      block_layout_info      = 512;
} ___block_descriptor_48_ea8_32s40s_e8_v16?08l;
static const struct { // proton block metadata
  unsigned long reserved = 0, block_size = 52;
  void          *copy_helper            = ___copy_helper_block_ea8_32s40s;
  void          *destroy_helper         = ___destroy_helper_block_ea8_32s40s;
  const char    *block_method_signature = __block_method_signature_v16_0_8;
  uintptr_t      block_layout_info      = 512;
} ___block_descriptor_52_ea8_32s40s_e8_v16?08l;
```

```
___copy_helper_block_ea8_32s40s
```

```
___destroy_helper_block_ea8_32s40s
```

```
// Generated block helpers — as of Xcode 11

static void ___copy_helper_block_ea8_32s40s(void *block) {


}

static void ___destroy_helper_block_ea8_32s40s(void *block) {


}
```

```
// Generated block helpers — as of Xcode 11

static void ___copy_helper_block_ea8_32s40s(void *block) {
  objc_retain(*(id*)(((char*)block) + 32));
  objc_retain(*(id*)(((char*)block) + 40));
}

static void ___destroy_helper_block_ea8_32s40s(void *block) {
  objc_release(*(id*)(((char*)block) + 32));
  objc_release(*(id*)(((char*)block) + 40));
}
```

2–7%

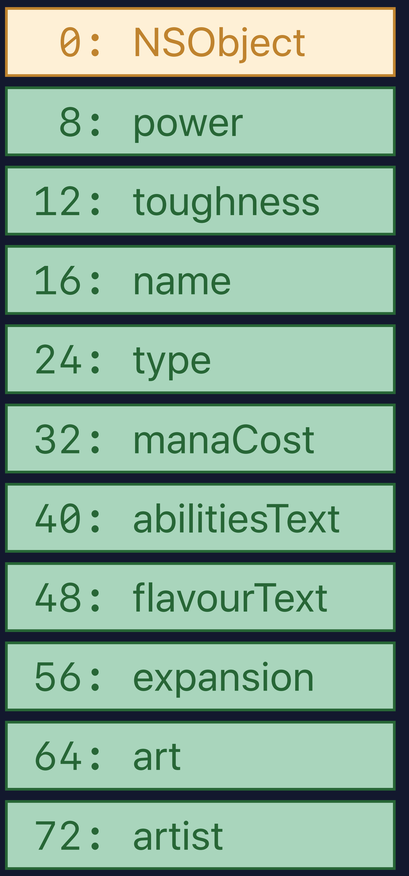
Typical code size reduction

Instance Variables of Direct Subclasses of NSObject

Offsets can be constant in the implementation

```
// Card.h — a direct subclass of NSObject

@interface Card : NSObject

@property (copy) NSString *name;
@property (copy) NSString *type;
@property (copy) NSDictionary *manaCost;
@property (copy) NSString *abilitiesText;
@property (copy) NSString *flavourText;
@property (copy) NSString *expansion;
@property int power;
@property int toughness;
@property UIImage *art;
@property (copy) NSString *artist;
- (instancetype)initWithName:(NSString *)name;

@end
```

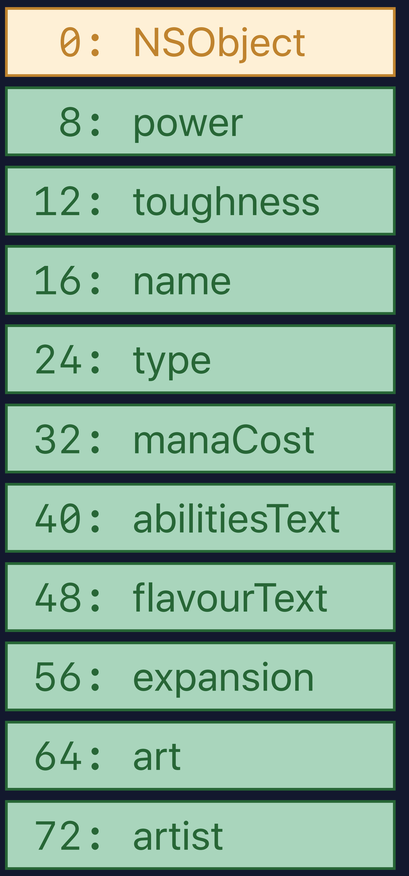
```objc
// Card.h — a direct subclass of NSObject

@interface Card : NSObject

@property (copy) NSString *name;
@property (copy) NSString *type;
@property (copy) NSDictionary *manaCost;
@property (copy) NSString *abilitiesText;
@property (copy) NSString *flavourText;
@property (copy) NSString *expansion;
@property int power;
@property int toughness;
@property UIImage *art;
@property (copy) NSString *artist;
- (instancetype)initWithName:(NSString *)name;

@end
```




0: NSObject
8: power
12: toughness
16: name
24: type
32: manaCost
40: abilitiesText
48: flavourText
56: expansion
64: art
72: artist


0: NSObject
8: power
12: toughness
16: name
24: type
32: manaCost
40: abilitiesText
48: flavourText
56: expansion
64: art
72: artist

```objc
// Card.mm — @implementation knows all offsets

#import "Card.h"

@implementation Card
- (instancetype)initWithName:(NSString *)name {
  if ((self = [super init])) {
    self.name = name;
    //   ...
    // ☀️💧💀🔥🌳
    //   ...
  }
  return self;
}
@end
```

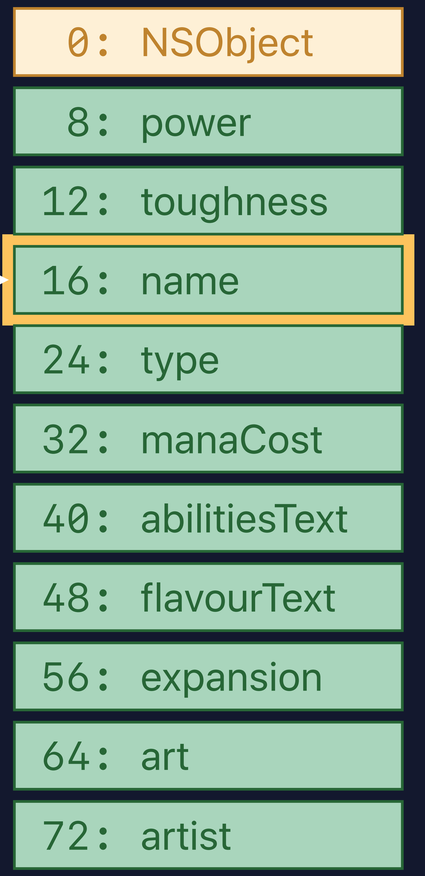


```objc
// Card.mm — @implementation knows all offsets

#import "Card.h"

@implementation Card
- (instancetype)initWithName:(NSString *)name {
  if ((self = [super init])) {
    self.name = name;
    //    ...
    // ☀️💧💀🔥🌳
    //    ...
  }
  return self;
}
@end
```

```
"-[Card setName:]":                     ; Before Xcode 11
  adrp    x8, _OBJC_IVAR_$_Card._name@PAGE
  ldrsw   x3, [x8, _OBJC_IVAR_$_Card._name@PAGEOFF]
  b       _objc_setProperty_atomic
```

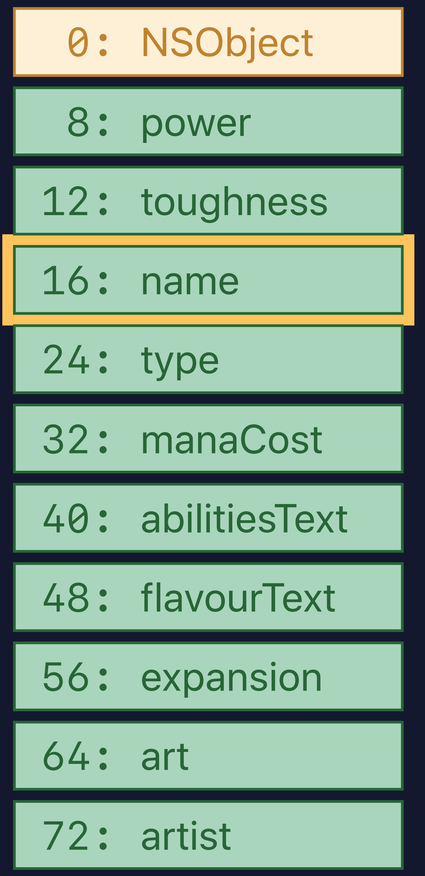
```objc
// Card.mm — @implementation knows all offsets

#import "Card.h"

@implementation Card
- (instancetype)initWithName:(NSString *)name {
  if ((self = [super init])) {
    self.name = name;
    //    ...
    // ☀️💧💀🔥🌳
    //    ...
  }
  return self;
}
@end
```

```
"-[Card setName:]":                    ; Before Xcode 11
  adrp    x8, _OBJC_IVAR_$_Card._name@PAGE
  ldrsw   x3, [x8, _OBJC_IVAR_$_Card._name@PAGEOFF]
  b       _objc_setProperty_atomic
```



```objc
// Card.mm — @implementation knows all offsets

#import "Card.h"

@implementation Card
- (instancetype)initWithName:(NSString *)name {
  if ((self = [super init])) {
    self.name = name;
    //    ...
    // ☀️💧💀🔥🌳
    //    ...
  }
  return self;
}
@end
```

```
"-[Card setName:]":                    ; Before Xcode 11
  adrp    x8, _OBJC_IVAR_$_Card._name@PAGE
  ldrsw   x3, [x8, _OBJC_IVAR_$_Card._name@PAGEOFF]
  b       _objc_setProperty_atomic

"-[Card setName:]":                    ; Xcode 11
  orr     w3, wzr, #0x10
  b       _objc_setProperty_atomic
```



2%

Typical code size reduction

Improved Debuggability of C++ Types

And associated code size wins

```
// Debugging around standard library code — print.cc

#include <cstdlib>
#include <iostream>
#include <string>
#include <vector>

int main(int argc, char** argv) {
    std::vector<std::string> args(argv + 1, argv + argc);
    std::vector<int> numbers;
    numbers.reserve(args.size());
    for (std::string const& arg : args) {
        int n = std::atoi(arg.c_str());
        numbers.push_back(n);
    }
    for (int i : numbers)
       std::cout << i << '\n';
}
```

```cpp
// Debugging around standard library code — print.cc

#include <cstdlib>
#include <iostream>
#include <string>
#include <vector>

int main(int argc, char** argv) {
    std::vector<std::string> args(argv + 1, argv + argc);
    std::vector<int> numbers;
    numbers.reserve(args.size());
    for (std::string const& arg : args) {
        int n = std::atoi(arg.c_str());
        numbers.push_back(n);                    // Let's set a breakpoint at line 12.
    }
    for (int i : numbers)
       std::cout << i << '\n';
}
```

$

```
$ lldb -- ./print 1 1 2 3 5
(lldb)
```

```
$ lldb -- ./print 1 1 2 3 5
(lldb)
```

```
$ lldb -- ./print 1 1 2 3 5
(lldb) b 12
Breakpoint 1: where = print`main + 420 at print.cc:12:17, address = 0x0000000000c0ffee
(lldb)
```

```
$ lldb -- ./print 1 1 2 3 5
(lldb) b 12
Breakpoint 1: where = print`main + 420 at print.cc:12:17, address = 0x0000000000c0ffee
(lldb) r
```

```
$ lldb -- ./print 1 1 2 3 5
(lldb) b 12
Breakpoint 1: where = print`main + 420 at print.cc:12:17, address = 0x0000000000c0ffee
(lldb) r
Process 1337 launched: '/Users/j_appleseed/print' (x86_64)
Process 1337 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
    frame #0: 0x0000000000c0ffee print`main(argc=6, argv=0x0000c0defefe0000) at print.cc:12:17
   9        numbers.reserve(args.size());
   10       for (std::string const& arg : args) {
   11           int n = std::atoi(arg.c_str());
-> 12           numbers.push_back(n);
   13       }
   14       for (int i : numbers)
   15           std::cout << i << '\n';
Target 0: (print) stopped.
```

```
$ lldb -- ./print 1 1 2 3 5
(lldb) b 12
Breakpoint 1: where = print`main + 420 at print.cc:12:17, address = 0x0000000000c0ffee
(lldb) r
Process 1337 launched: '/Users/j_appleseed/print' (x86_64)
Process 1337 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
    frame #0: 0x0000000000c0ffee print`main(argc=6, argv=0x0000c0defefe0000) at print.cc:12:17
   9        numbers.reserve(args.size());
   10       for (std::string const& arg : args) {
   11           int n = std::atoi(arg.c_str());
-> 12           numbers.push_back(n);
   13       }
   14       for (int i : numbers)
   15           std::cout << i << '\n';
Target 0: (print) stopped.
```

† in release configuration of codebases that heavily use the C++ Standard Library

7%

Code size reduction† and better debugging!

† in release configuration of codebases that heavily use the C++ Standard Library

C++ Static Destructor Suppression

```
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Another file — Game.cc
#include "logger.h"

class Game {
public:
  Game();
  ~Game();
  // ...
  //  🚀
  // ...
};
Game game;

Game::~Game() {



}
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Another file — Game.cc
#include "logger.h"

class Game {
public:
  Game();
  ~Game();
  // ...
  //  🚀
  // ...
};
Game game;

Game::~Game() {



}
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Another file — Game.cc
#include "logger.h"

class Game {
public:
  Game();
  ~Game();
  // ...
  //  🚀
  // ...
};
Game game;

Game::~Game() {
  logger.log("Thank you for playing ",
    game_name, "!");
}
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Another file — Game.cc
#include "logger.h"

class Game {
public:
  Game();
  ~Game();
  // ...
  //  🚀
  // ...
};
Game game;

Game::~Game() {
  logger.log("Thank you for playing ",
    game_name, "!");
}
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

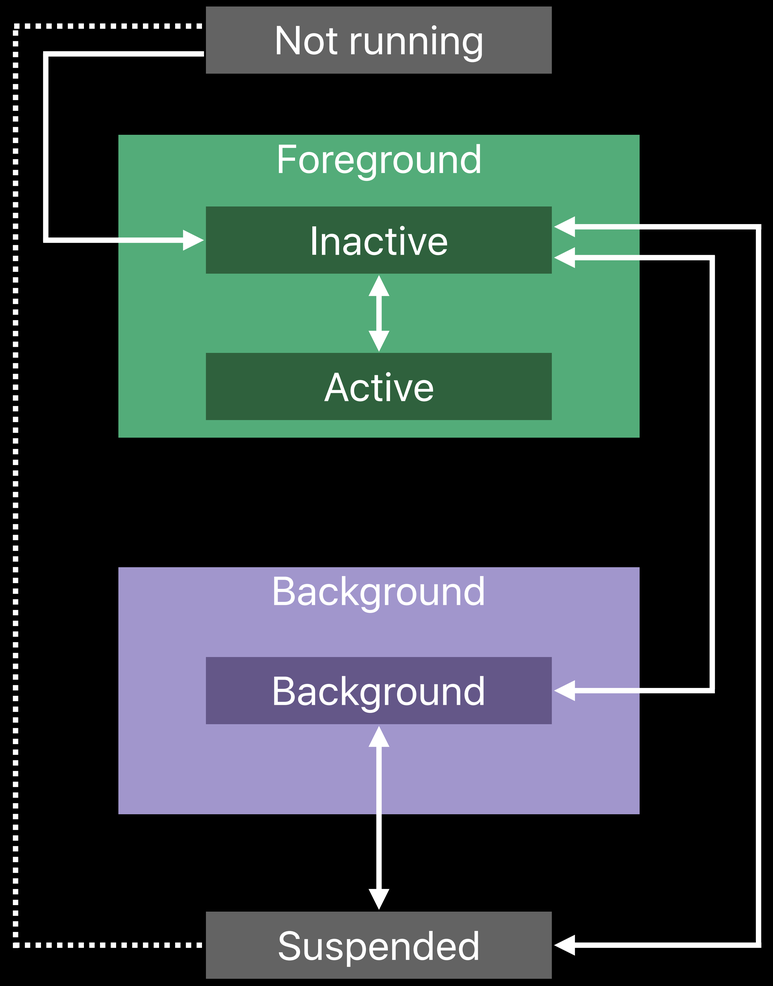
```cpp
// Another file — Game.cc
#include "logger.h"

class Game {
public:
  Game();
  ~Game();
  // ...
  //  🚀
  // ...
};
Game game;

Game::~Game() {



}
```

C++ Static Destructor Suppression

Application lifecycle

C++ Static Destructor Suppression

Application lifecycle

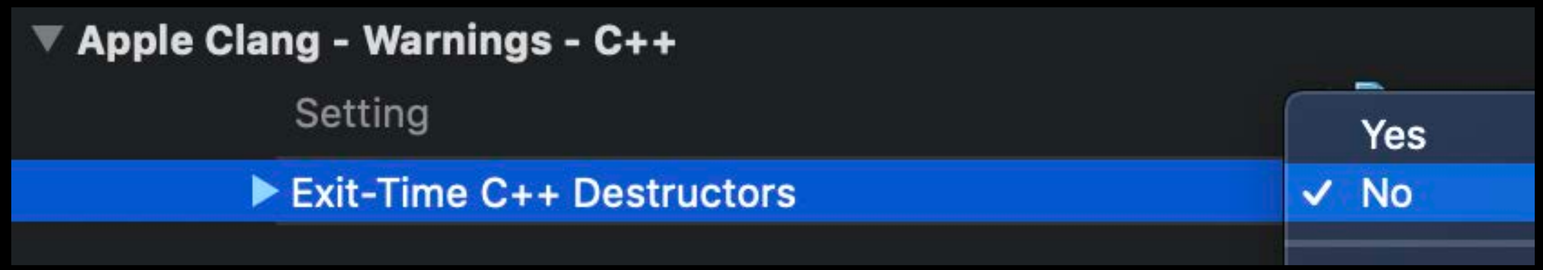


```
// Handling an application's lifecycle


@interface CowClicker : NSObject <UIApplicationDelegate>

- (void)applicationWillResignActive:(UIApplication *)application;
- (void)applicationDidBecomeActive:(UIApplication *)application;
- (void)applicationWillEnterForeground:(UIApplication *)application;
- (void)applicationDidEnterBackground:(UIApplication *)application;
- (void)applicationWillTerminate:(UIApplication *)application;

@end
```

```
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc


Logger logger;
```

```cpp
// Global with a destructor — logger.h

class Logger {
public:
  template <typename... Ts>
    void log(Ts&&... message);
  void flush();

private:
  std::vector<std::string> mBuffer;
};
extern Logger logger;

// Implementation file — logger.cc

[[clang::no_destroy]]
Logger logger;
```

C++ static Destructor Suppression



1%

Typical code size reduction

Diagnostics

```
// -Wcall-to-pure-virtual-from-ctor-dtor
```

```
// -Wcall-to-pure-virtual-from-ctor-dtor




class Table {


};
```

```
// -Wcall-to-pure-virtual-from-ctor-dtor



Grail* find(Knight*);
Shrubbery* find(Knight*);

class Table {
  virtual Knight* galahad() = 0;
  virtual ~Table() { find(galahad()); }
};
```

```
// -Wcall-to-pure-virtual-from-ctor-dtor



Grail* find(Knight*);
Shrubbery* find(Knight*);

class Table {
  virtual Knight* galahad() = 0;
  virtual ~Table() { find(galahad()); }
};

 warning: call to pure virtual member function 'galahad' has undefined behaviour;
          overrides of 'galahad' in subclasses are not available in the destructor of 'Table'
```

```
// -Wcall-to-pure-virtual-from-ctor-dtor



Grail* find(Knight*);
Shrubbery* find(Knight*);

class Table {
  virtual Knight* galahad() = 0;
  virtual ~Table() {                    }
};
```

```
// -Wcall-to-pure-virtual-from-ctor-dtor



Grail* find(Knight*);
Shrubbery* find(Knight*);

class Table {
  virtual Knight* galahad() = 0;
  virtual ~Table() { /* nothing! */   }
};
```

```
class RoundTable : public Table {
  Knight* galahad() final { /* 🐴 */ }
  ~RoundTable() override { find(galahad()); }
};
```

```
// -Wmemset-transposed-args
```

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
```

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
void me_after_vacation(struct Inbox* inbox) {

}
```

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
void me_after_vacation(struct Inbox* inbox) {
  memset(inbox, sizeof(struct Inbox), 0);
}
```

```
warning: 'size' argument to memset is '0';
         did you mean to transpose the last two arguments?
```

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
void me_after_vacation(struct Inbox* inbox) {
  memset(inbox, 0, sizeof(struct Inbox));
}
```

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
void me_after_vacation(struct Inbox* inbox) {
  memset(inbox, 0, sizeof(struct Inbox));
}
```

Avoid

memset

```
// -Wmemset-transposed-args

struct Inbox { int emails[1024]; };
void me_after_vacation(struct Inbox* inbox) {
  memset(inbox, 0, sizeof(struct Inbox));
}
```

Avoid

`memset`

Recommended

`<algorithm>` → `std::fill` or `std::fill_n`

```
// -Wmemset-transposed-args

struct Inbox {            int  1024  emails; };
void me_after_vacation(struct Inbox* inbox) {

}
```

Avoid

`memset`

Recommended

`<algorithm>` → `std::fill` or `std::fill_n`

```cpp
// -Wmemset-transposed-args

struct Inbox { std::array<int, 1024> emails; };
void me_after_vacation(struct Inbox* inbox) {
  std::fill(inbox->emails.begin(), inbox->emails.end(), 0);
}
```

Avoid

`memset`

Recommended

`<algorithm>` → `std::fill` or `std::fill_n`

```
// -Wreturn-std-move
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

Goat bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
warning: local variable 'chimæra' will be copied despite being returned by name
note:    call 'std::move' explicitly to avoid copying
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

Goat bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
warning: local variable 'chimæra' will be copied despite being returned by name
note:    call 'std::move' explicitly to avoid copying
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

Goat bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return           chimæra ;
}
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

Goat bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return std::move(chimæra);
}
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

       bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

Chimæra bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

                     bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
// -Wreturn-std-move

struct Lion  {       /* 🦁 */      };
struct Goat  { std::vector<int> v; };
struct Snake {       /* 🐍 */      };

struct Chimæra : Lion, Goat, Snake { /* 🔥 🦁 🐐 🐍 */ };

template <typename Creature> void slay(Creature&);

std::optional<Chimæra> bellerophon(Chimæra chimæra) {
  slay(chimæra);
  return chimæra;
}
```

```
// -Wsizeof-pointer-div
```

```
// -Wsizeof-pointer-div

void all_work() {
  int no_play[10];
  size_t array_elts = sizeof(no_play) / sizeof(no_play[0]);
  // ... 💻 ...
}
```

```
// -Wsizeof-pointer-div

void        (int       [10]) {
  size_t array_elts = sizeof(      ) / sizeof(      [0]);
  // ...    ...

}
```

```
// -Wsizeof-pointer-div

void pigeon(int array[10]) {
  size_t array_elts = sizeof(array) / sizeof(array[0]);  // Is this a 10? 🦋
  // ... 💁 ...
  // ...
}
```

```
      warning: 'sizeof(array)' will return the size of the pointer, not the array itself
```

```c
// -Wsizeof-pointer-div

void pigeon(int array[10]) {
  size_t array_elts = sizeof(array) / sizeof(array[0]);  // Is this a 10? 🦋
  // ... 🤦 ...
  // ...
}
```

```
      warning: 'sizeof(array)' will return the size of the pointer, not the array itself
```

Avoid

```c
sizeof(array) / sizeof(array[0])
```

```
// -Wsizeof-pointer-div

void pigeon(int array[10]) {
  size_t array_elts = sizeof(array) / sizeof(array[0]);  // Is this a 10? 🦋
  // ... 💁 ...
  // ...
}
```

```
      warning: 'sizeof(array)' will return the size of the pointer, not the array itself
```

Avoid

```
sizeof(array) / sizeof(array[0])
```

Recommended

C++17 `<iterator>` → `std::size`

```
// -Wsizeof-pointer-div

void          () {
  int         [10]
  size_t array_elts =
  // ...    ...
}
```

```
// -Wsizeof-pointer-div

void dull_boi() {
  int no_play[10];
  size_t array_elts = std::size(no_play);
  // ... 💻 ...
}
```

```
// -Wdefaulted-function-deleted
```

```
// -Wdefaulted-function-deleted

struct Aberration {
  float& eyestalks;
  int eye;
  int mouth;
  Aberration() = default;
};
```

```
warning: explicitly defaulted default constructor is implicitly deleted
note:    default constructor of 'Aberration' is implicitly deleted because
         field 'eyestalks' of reference type 'float &' would not be initialized
```

```
// -Wdefaulted-function-deleted

struct Aberration {
  float& eyestalks;
  int eye;
  int mouth;
  Aberration(               )
};
```

```
// -Wdefaulted-function-deleted

struct Aberration {
  float& eyestalks;
  int eye;
  int mouth;
  Aberration(float& eyestalks) : eyestalks(eyestalks) { }
};
```

```
// -Wdefaulted-function-deleted

struct Aberration {
  float eyestalks;
  int eye;
  int mouth;
  Aberration()
};
```

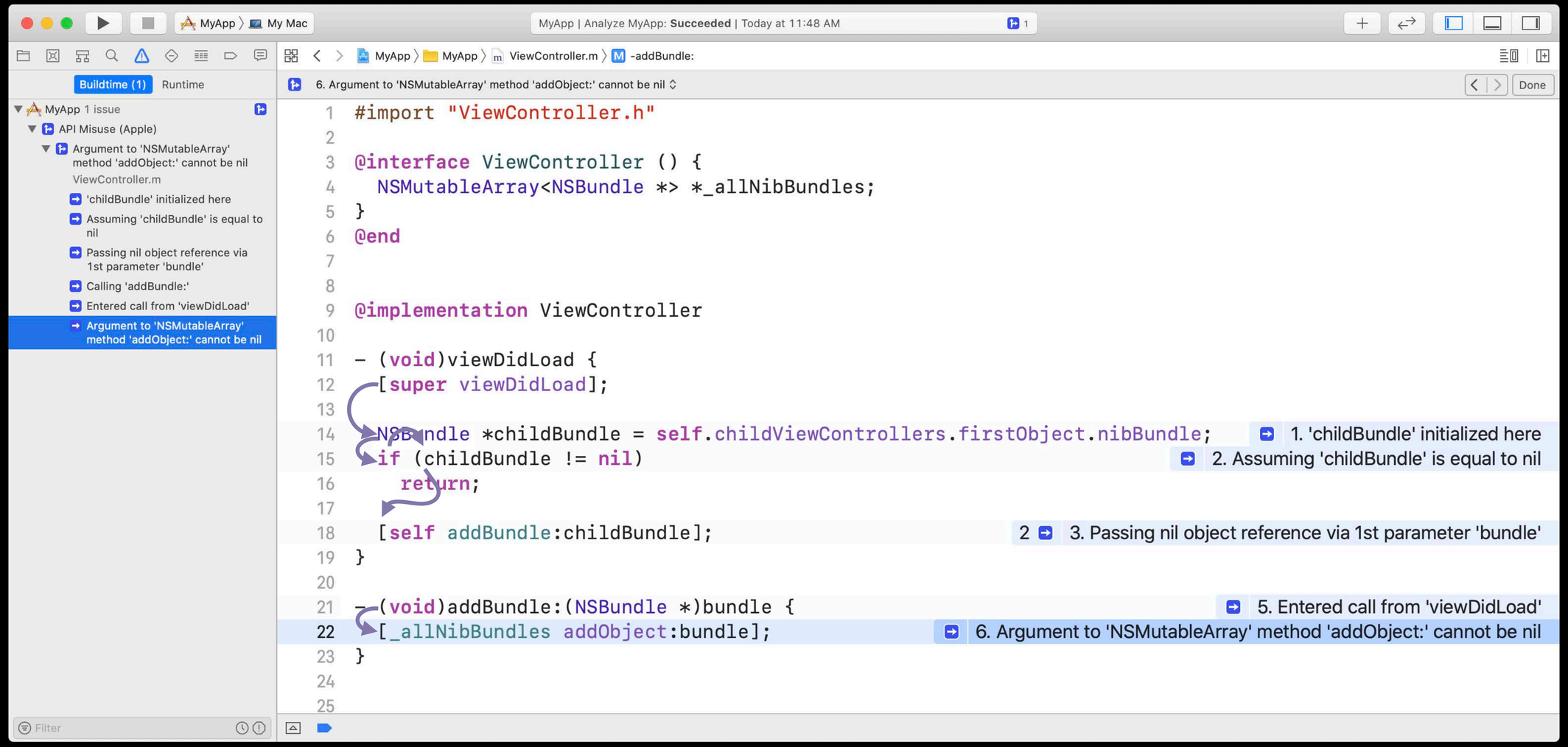
```
// -Wdefaulted-function-deleted

struct Aberration {
  float eyestalks;
  int eye;
  int mouth;
  Aberration() = default;
};
```

New Static Analyzer Checks

Devin Coughlin, Program Analysis Engineer

Finds Deep Bugs

Great at catching hard-to-produce, edge-case bugs



Three New C++ Checks

NEW

Use after move bugs

Dangling C string pointers with `std::string`

Reference-counting bugs in DriverKit and IOKit

NEW

Use After Move in C++

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

Moves from source variable rather than copying

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

Moves from source variable rather than copying

Can enforce unique ownership semantics

C++ Moves Avoid Unwanted Copies

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

Moves from source variable rather than copying

Can enforce unique ownership semantics

Leaves source in unspecified state

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));
```

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));

myNovel.spellCheck();
```

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));

myNovel.spellCheck();
```

May have unexpected results or even crash!

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));

myNovel.spellCheck();
```

Method called on moved-from object

May have unexpected results or even crash!

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

publish(std::move(myNovel));

myNovel.spellCheck();
```

Method called on moved-from object

May have unexpected results or even crash!

Common fix is to reorder code

Do Not Use Variable after Moving

```
Book myNovel("It was the best of times...");

myNovel.spellCheck();


publish(std::move(myNovel));
```

May have unexpected results or even crash!

Common fix is to reorder code

NEW

Dangling Pointers from `std::string`

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

`c_str()` returns inner pointer to buffer inside `std::string`

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

`c_str()` returns inner pointer to buffer inside `std::string`

Buffer deallocated when `std::string` goes out of scope

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Returning memory that will be deallocated

`c_str()` returns inner pointer to buffer inside `std::string`

Buffer deallocated when `std::string` goes out of scope

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Returning memory that will be deallocated

`c_str()` returns inner pointer to buffer inside `std::string`

Buffer deallocated when `std::string` goes out of scope

Using deallocated memory may crash!

Mixing C++ and C Strings Can Be Tricky!

```
const char *generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting.c_str();
}

printf("%s from WWDC!", generateGreeting("World"));
```

Returning memory that will be deallocated

Inner string pointer used after deallocation

`c_str()` returns inner pointer to buffer inside `std::string`

Buffer deallocated when `std::string` goes out of scope

Using deallocated memory may crash!

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Change scope of `std::string` to last as long as C string is used

Match Lifetimes of C++ and C Strings

```
std::string generateGreeting(const char *name) {
  std::string greeting = "Hello ";
  greeting.append(name);
  return greeting;
}
std::string greeting = generateGreeting("World");
printf("%s from WWDC!", greeting.c_str());
```

Change scope of `std::string` to last as long as C string is used

Often easier to stay within C++ world

NEW

Reference Counting in DriverKit and IOKit

System Extensions and DriverKit

WWDC 2019

Drivers Use Manual Retain/Release

Drivers Use Manual Retain/Release

`OSObject` uses retain/release for memory management

- Similar to CoreFoundation or Objective-C without ARC

Drivers Use Manual Retain/Release

`OSObject` uses retain/release for memory management

- Similar to CoreFoundation or Objective-C without ARC

Easy to over-release

- Memory used after deallocated

Drivers Use Manual Retain/Release

`OSObject` uses retain/release for memory management

- Similar to CoreFoundation or Objective-C without ARC

Easy to over-release

- Memory used after deallocated

Easy to under-release

- Memory leaked

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);

return true;
```

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);

return true;
```

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);

return true;
```

Array allocated and returned retained

`OSArray::withCapacity()` returns retained

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);

return true;
```

Array allocated and returned retained

`OSArray::withCapacity()` returns retained

Array will leak if not released

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);

return true;
```

Array allocated and returned retained

Leak of object stored into ‘devices’

`OSArray::withCapacity()` returns retained

Array will leak if not released

Allocated Objects Must Be Released

```
OSArray *devices = OSArray::withCapacity(2);
fillInDevices(devices);
setUpDevices(devices);
OSObjectRelease(devices);
return true;
```

Array allocated and returned retained

`OSArray::withCapacity()` returns retained

Array will leak if not released

Convention for Memory Management

Convention for Memory Management

Similar in spirit to rules for CoreFoundation and ObjC Manual Retain/Release

Convention for Memory Management

Similar in spirit to rules for CoreFoundation and ObjC Manual Retain/Release

Default convention is to return retained (+1)

- Clients must call `OSObjectRelease()`

Convention for Memory Management

Similar in spirit to rules for CoreFoundation and ObjC Manual Retain/Release

Default convention is to return retained (+1)

- Clients must call `OSObjectRelease()`

Except getters return unretained (+0)

- Clients must not call `OSObjectRelease()`

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Differing from Convention

```
OSObject *findFirstDevice()  {
    OSObject *result = devices->getObject(0);

    return result;
}
```

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Object with +0 retain count returned where +1 is expected

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Change behavior to follow convention

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);
   OSObjectRetain(result);
   return result;
}
```

Change behavior to follow convention

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Change behavior to follow convention

Rename method to follow convention

Differing from Convention

```
OSObject *getFirstDevice()   {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Change behavior to follow convention

Rename method to follow convention

Differing from Convention

```
OSObject *findFirstDevice()  {
   OSObject *result = devices->getObject(0);

   return result;
}
```

Change behavior to follow convention

Rename method to follow convention

Add annotation to tell readers and analyzer convention not followed

Differing from Convention

```
OSObject *findFirstDevice() DRIVERKIT_RETURNS_NOT_RETAINED;

OSObject *findFirstDevice()  {
  OSObject *result = devices->getObject(0);

  return result;
}
```

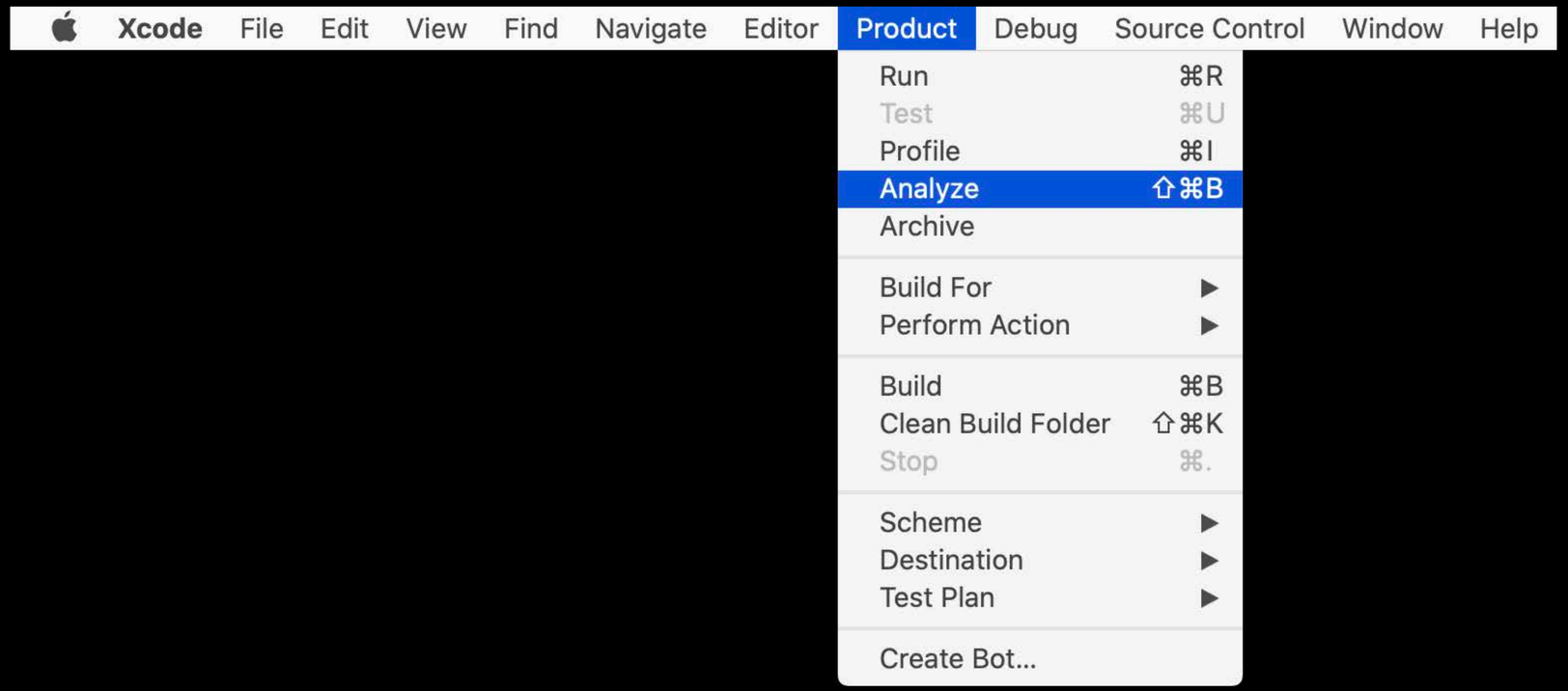
Change behavior to follow convention

Rename method to follow convention

Add annotation to tell readers and analyzer convention not followed

Run Analyzer on Your Code!

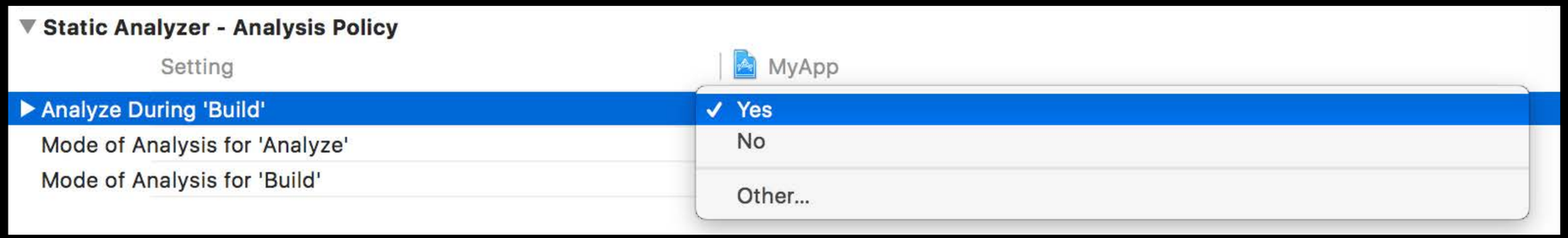
Supports Objective-C, C, C++



Run Analyzer on Your Code!

Supports Objective-C, C, C++

Analyze During Build



Summary

LLVM bitcode enabled seamless 64-bit transition for watchOS

Reduce code size with new compiler optimizations

Run the static analyzer on your code

More Information

developer.apple.com/wwdc19/409

LLVM Compiler, Objective-C, C++, and Linking Lab	Thursday, 9:00
Performance, Power, Crashes, and Debugging Lab	Thursday, 12:00


WWDC19