GHC Runtime Linker

by Example

Artem Pianykh
Software Engineer @ FB
@artem_pyanykh

DNS is likely some of the most broadly deployed, yet most poorly understood software components deployed in the world today. What else is as common but less understood?

Timothy Perrett @timperrett
2:12am - 19 Apr 2019

Linkers twitter.com/timperrett/sta…

Gabriel Gonzalez @GabrielG439
1:32pm - 19 Apr 2019

Motivating Example

```haskell
module FibSlow where

fib :: Int -> Int
fib 0 = 1
fib 1 = 1
fib n = fib (n - 1) + fib (n - 2)
```

GHCi

```
λ> :set +s
λ> :load *haskell/FibSlow
[1 of 1] Compiling Fib                ( haskell/FibSlow.hs, interpreted )
λ> fib 35
9227465
(7.14 secs, 4,897,229,624 bytes)
λ> :load haskell/FibSlow
λ> fib 35
9227465
(0.50 secs, 2,150,044,856 bytes)
```

GHCi

```
λ> :set +s
λ> :load *haskell/FibSlow
[1 of 1] Compiling Fib                ( haskell/FibSlow.hs, interpreted )
λ> fib 35
9227465
(7.14 secs, 4,897,229,624 bytes)
λ> :load haskell/FibSlow
λ> fib 35
9227465
(0.50 secs, 2,150,044,856 bytes)
```

We will cover

→ What is linking

→ GHC + `-dynamic` and using **system linker**

→ GHC + `-static` and custom **RTS linker**

ld = link editor

file:c/magic.c

```c
int MAGIC = 42;

int magic(int x)
{
        return x + MAGIC;
}
```

```
c/magic.o:
(__TEXT,__text) section
_magic:
       0:   55  pushq   %rbp
       1:   48 89 e5    movq    %rsp, %rbp
       4:   03 3d 00 00 00 00   addl    _MAGIC(%rip), %edi
       a:   89 f8   movl    %edi, %eax
       c:   5d  popq    %rbp
       d:   c3  retq

SYMBOL TABLE:
0000000000000010 g       __DATA,__data  _MAGIC
0000000000000000 g     F __TEXT,__text  _magic

Contents of (__DATA,__data) section
0000000000000010    2a 00 00 00
```

```
c/magic.o:
(__TEXT,__text) section
_magic:
       0:   55  pushq   %rbp
       1:   48 89 e5    movq    %rsp, %rbp
       4:   03 3d 00 00 00 00   addl    _MAGIC(%rip), %edi
       a:   89 f8   movl    %edi, %eax
       c:   5d  popq    %rbp
       d:   c3  retq

SYMBOL TABLE:
0000000000000010 g       __DATA,__data  _MAGIC
0000000000000000 g     F __TEXT,__text  _magic

Contents of (__DATA,__data) section
0000000000000010    2a 00 00 00
```

```
c/magic.o:
(__TEXT,__text) section
_magic:
       0:   55  pushq   %rbp
       1:   48 89 e5    movq    %rsp, %rbp
       4:   03 3d 00 00 00 00   addl    _MAGIC(%rip), %edi
       a:   89 f8   movl    %edi, %eax
       c:   5d  popq    %rbp
       d:   c3  retq

SYMBOL TABLE:
0000000000000010 g       __DATA,__data  _MAGIC
0000000000000000 g     F __TEXT,__text  _magic

Contents of (__DATA,__data) section
0000000000000010    2a 00 00 00
```

```
4:  03 3d 00 00 00 00   addl    _MAGIC(%rip), %edi
```

Bytes	Value	Meaning
1	`03`	`ADD` opcode
2	`3D = b00_111_101`	addressing mode + register
3-6 (4)	0 stub	32bit offset from `%rip`

```
4:  03 3d 00 00 00 00    addl    _MAGIC(%rip), %edi
                           ==
4:  03 3d ?? ?? ?? ??    addl    _MAGIC(%rip), %edi
```

```
c/magic.o:
(__TEXT,__text) section
_magic:
       0:   55  pushq   %rbp
       1:   48 89 e5    movq    %rsp, %rbp
       4:   03 3d 00 00 00 00   addl    _MAGIC(%rip), %edi
       a:   89 f8   movl    %edi, %eax
       c:   5d  popq    %rbp
       d:   c3  retq

SYMBOL TABLE:
0000000000000010 g       __DATA,__data  _MAGIC
0000000000000000 g     F __TEXT,__text  _magic

Relocation information (__TEXT,__text) 1 entries
address  pcrel length extern type    scattered symbolnum/value
00000006 True  long   True   SIGNED  False     _MAGIC
```

file:c/array_magic.c

```c
extern int magic(int);

void array_magic(int* arr, int len)
{
        for (int i = 0; i < len; i++) {
                arr[i] = magic(arr[i]);
        }
}
```

```
void array_magic(int* arr, int len); // rdi <- arr; esi <- len
```

```
c/array_magic.o:
(__TEXT,__text) section
_array_magic:
0000000000000000
       a:   85 f6   testl   %esi, %esi
       c:   7e 27   jle 0x35
       e:   49 89 ff    movq    %rdi, %r15
      11:   41 89 f6    movl    %esi, %r14d
      14:   31 db   xorl    %ebx, %ebx
0000000000000016
      20:   41 8b 3c 9f     movl    _array_magic(%r15,%rbx,4), %edi
      24:   e8 00 00 00 00  callq   _magic
      29:   41 89 04 9f     movl    %eax, _array_magic(%r15,%rbx,4)
      2d:   48 ff c3    incq    %rbx
      30:   49 39 de    cmpq    %rbx, %r14
      33:   75 eb   jne 0x20
      35:   48 83 c4 08     addq    $8, %rsp
0000000000000039
c/array_magic.o:
SYMBOL TABLE:
0000000000000000 g     F __TEXT,__text  _array_magic
0000000000000000         *UND*  _magic
c/array_magic.o:
Relocation information (__TEXT,__text) 1 entries
address  pcrel length extern type    scattered symbolnum/value
00000025 True  long   True   BRANCH  False     _magic
```

```
void array_magic(int* arr, int len); // rdi <- arr; esi <- len

c/array_magic.o:
(__TEXT,__text) section
_array_magic:
                                                        ---prologue
         a: 85 f6    testl   %esi, %esi ; length = 0?
-        c: 7e 27    jle 0x35
|        e: 49 89 ff    movq    %rdi, %r15  ; r15 <- arr
|       11: 41 89 f6    movl    %esi, %r14d ; r14 <- length
|       14: 31 db   xorl    %ebx, %ebx      ; i <- 0
|   ...
|    ->  20: 41 8b 3c 9f     movl    _array_magic(%r15,%rbx,4), %edi ; edi <- arr[i]
|  /     24: e8 00 00 00 00  callq   _magic                          ; eax <~ magic(arr[i])
|  |     29: 41 89 04 9f     movl    %eax, _array_magic(%r15,%rbx,4) ; arr[i] <- eax
|  |     2d: 48 ff c3    incq    %rbx                                ; i++
|  |     30: 49 39 de    cmpq    %rbx, %r14                          ; i < length?
\  -     33: 75 eb   jne 0x20
  ->     35: 48 83 c4 08     addq    $8, %rsp
                                                        ---epilogue
```

```
void array_magic(int* arr, int len); // rdi <- arr; esi <- len
```

```
c/array_magic.o:
(__TEXT,__text) section
_array_magic:
                                                    ---prologue
         a: 85 f6   testl   %esi, %esi ; length = 0?
-        c: 7e 27   jle 0x35
|        e: 49 89 ff    movq    %rdi, %r15  ; r15 <- arr
|       11: 41 89 f6    movl    %esi, %r14d ; r14 <- length
|       14: 31 db   xorl    %ebx, %ebx      ; i <- 0
|   ...
|    ->  20: 41 8b 3c 9f     movl    _array_magic(%r15,%rbx,4), %edi ; edi <- arr[i]
|  /     24: e8 00 00 00 00  callq   _magic                          ; eax <~ magic(arr[i])
|  |     29: 41 89 04 9f     movl    %eax, _array_magic(%r15,%rbx,4) ; arr[i] <- eax
|  |     2d: 48 ff c3    incq    %rbx                                ; i++
|  |     30: 49 39 de    cmpq    %rbx, %r14                          ; i < length?
\  -     33: 75 eb   jne 0x20
  ->     35: 48 83 c4 08     addq    $8, %rsp
                                                    ---epilogue
```

```
void array_magic(int* arr, int len); // rdi <- arr; esi <- len
```

```
c/array_magic.o:
(__TEXT,__text) section
_array_magic:
                                                    ---prologue
         a: 85 f6   testl   %esi, %esi ; length = 0?
-        c: 7e 27   jle 0x35
|        e: 49 89 ff    movq    %rdi, %r15  ; r15 <- arr
|       11: 41 89 f6    movl    %esi, %r14d ; r14 <- length
|       14: 31 db   xorl    %ebx, %ebx      ; i <- 0
|   ...
|    ->  20: 41 8b 3c 9f     movl    _array_magic(%r15,%rbx,4), %edi ; edi <- arr[i]
|  /     24: e8 00 00 00 00  callq   _magic                          ; eax <~ magic(arr[i])
|  |     29: 41 89 04 9f     movl    %eax, _array_magic(%r15,%rbx,4) ; arr[i] <- eax
|  |     2d: 48 ff c3    incq    %rbx                                ; i++
|  |     30: 49 39 de    cmpq    %rbx, %r14                          ; i < length?
\  -     33: 75 eb   jne 0x20
  ->     35: 48 83 c4 08     addq    $8, %rsp
                                                    ---epilogue
```

```
void array_magic(int* arr, int len); // rdi <- arr; esi <- len
```

```
c/array_magic.o:
(__TEXT,__text) section
_array_magic:
                                                    ---prologue
         a: 85 f6   testl   %esi, %esi ; length = 0?
-        c: 7e 27   jle 0x35
|        e: 49 89 ff    movq    %rdi, %r15  ; r15 <- arr
|       11: 41 89 f6    movl    %esi, %r14d ; r14 <- length
|       14: 31 db   xorl    %ebx, %ebx      ; i <- 0
|   ...
|    ->  20: 41 8b 3c 9f     movl    _array_magic(%r15,%rbx,4), %edi ; edi <- arr[i]
|  /     24: e8 00 00 00 00  callq   _magic                          ; eax <~ magic(arr[i])
|  |     29: 41 89 04 9f     movl    %eax, _array_magic(%r15,%rbx,4) ; arr[i] <- eax
|  |     2d: 48 ff c3    incq    %rbx                                ; i++
|  |     30: 49 39 de    cmpq    %rbx, %r14                          ; i < length?
\  -     33: 75 eb   jne 0x20
   ->    35: 48 83 c4 08     addq    $8, %rsp
                                                    ---epilogue
```

```
c:  7e 27    jle 0x35
e:  49 89 ff    movq    %rdi, %r15
...
35: 48 83 c4 08     addq    $8, %rsp
...
```

→ `7e` is opcode for `JLE rel8`

→ `JLE` is short (1 byte) jump to `rel8` offset from **next** instruction address

→ `e + 27 = 35`

```
24: e8 00 00 00 00  callq   _magic
    ^  ^----------
    |            |
    |            32bit 0-stub
    |
    near call, 32bit displacement relative to next instr.

Relocation information (__TEXT,__text) 1 entries
address  pcrel length extern type    scattered symbolnum/value
00000025 True  long   True   BRANCH  False     _magic
```