ELF Binary Comparison

Overview

Binaries ↩

old: /home/zheyuan/elf_diff/tests/x86_64/sqlelf_diff_test_old

new: /home/zheyuan/elf_diff/tests/x86_64/sqlelf_diff_test_new

Statistics ↩

Program Storage

	Old/ bytes	New/ bytes	Delta/ bytes
overall	2059	2059	0
text	1523	1523	0
data	536	536	0

Static RAM

	Old/ bytes	New/ bytes	Delta/ bytes
overall	544	544	0
data	536	536	0
bss	8	8	0

Legend

text	instructions
data	initialized global or static variables
bss	uninitialized global or static variables

Symbol Classes

Class	Entities
Old	12
New	12
Persisting	8
Disappeared	4
Appeared	4
Similar	10
Migrated	0

Symbol Selection

	Old Binary	New Binary
Total	12	12
Selected	12	12
Dropped	0	0
Selection Regex	.*	.*
Exclusion Regex		

Symbols ↩

Persisting Symbols ↩

Symbol	Type	Old Size/ bytes	New Size/ bytes	Delta/ bytes
_IO_stdin_used	R	4	4	0
persisting1(int) ⓘ	T	14	14	0
persisting2(int) ⓘ	T	14	14	0
__abi_tag	r	32	32	0
_start ⓘ	T	38	38	0
completed.0	b	1	1	0
main ⓘ	T	15	15	0
var	D	4	4	0

Columns

Symbol	The symbol name (possibly mangled)
Type	The symbol type (see the documentation of binutils tool nm for more information)
Old Size	The old symbol size either in RAM or program memory
New Size	The new symbol size either in RAM or program memory
Delta	The change to symbol size

Disappeared Symbols ↩

Symbol	Type	Size/ bytes
`Test::g(float, float)` ⓘ	T	25
`Test::f(int, int)` ⓘ	T	17
`func(int)` ⓘ	T	14
`Test::m_`	D	4

Columns

Symbol	The symbol name (possibly demangled)
Type	The symbol type (see the documentation of binutils tool nm for more information)
Size	The symbol size either in RAM or program memory

New Symbols ↩

Symbol	Type	Size/ bytes
`Test1::g(float, float)` ⓘ	T	25
`Test1::f(int, int)` ⓘ	T	17
`func(double)` ⓘ	T	16
`Test1::m_`	D	4

Columns

Symbol	The symbol name (possibly demangled)
Type	The symbol type (see the documentation of binutils tool nm for more information)
Size	The symbol size either in RAM or program memory

Similar Symbols ↩

Id	Symbols	Types	Sizes/ bytes	Deltas/ bytes	Sig. Sim./ %	Instr. Sim./ %

0
ⓘ
	Test::f(int, int)
	Test1::f(int, int)
		T
		T
			17
			17
				0	88.9	100.0

1
ⓘ
	Test::g(float, float)
	Test1::g(float, float)
		T
		T
			25
			25
				0	88.9	100.0

2	Test::m_ Test1::m_	D D	4 4	0	88.0	100.0

3
ⓘ
	func(int)
	func(double)
		T
		T
			14
			16
				2	87.5	85.1

4	Test::f(int, int) Test1::m_	T D	17 4	-13	69.2	0.0
5	Test::g(float, float) Test1::m_	T D	25 4	-21	69.2	0.0

6
ⓘ
	Test::f(int, int)
	Test1::g(float, float)
		T
		T
			17
			25
				8	66.7	75.1

7
ⓘ
	Test::g(float, float)
	Test1::f(int, int)
		T
		T
			25
			17
				-8	66.7	75.1

8	Test::m_ Test1::g(float, float)	D T	4 25	21	61.5	0.0
9	Test::m_ Test1::f(int, int)	D T	4 17	13	61.5	0.0

Columns

ID	Integer id assigned to each symbol pair
Symbols	The two similar symbol names (possibly mangled)
Types	The symbol types (see the documentation of binutils tool nm for more information)
Sizes	The sizes of the symbols either in RAM or program memory
Deltas	The difference in symbol size
Sig. Sim.	Lexicographic symbol signature similarity
Instr. Sim.	Instruction similarity of the symbols' assembly code

Symbol Details ↩

Persisting Symbols ↩

Persisting symbol persisting1(int) : old size: 14 bytes, new size: 14 bytes, delta: 0 bytes

```
Old source: ?
New source: ?
```

	Old				New		
f	1	push	%rbp	f	1	push	%rbp
	2	mov	%rsp,%rbp		2	mov	%rsp,%rbp
	3	mov	%edi,-0x4(%rbp)		3	mov	%edi,-0x4(%rbp)
t	4	mov	$0x2b,%eax	t	4	mov	$0x2a,%eax
	5	pop	%rbp		5	pop	%rbp
	6	ret			6	ret	
	7	xchg	%ax,%ax		7	xchg	%ax,%ax
	8				8		

Persisting symbol persisting2(int) : old size: 14 bytes, new size: 14 bytes, delta: 0 bytes

```
Old source: ?
New source: ?
```

	Old				New		
f	1	push	%rbp	f	1	push	%rbp
	2	mov	%rsp,%rbp		2	mov	%rsp,%rbp
	3	mov	%edi,-0x4(%rbp)		3	mov	%edi,-0x4(%rbp)
t	4	mov	$0x2b,%eax	t	4	mov	$0x2a,%eax
	5	pop	%rbp		5	pop	%rbp
	6	ret			6	ret	
	7	xchg	%ax,%ax		7	xchg	%ax,%ax
	8				8		

Persisting symbol _start : old size: 38 bytes, new size: 38 bytes, delta: 0 bytes

```
Old source: ?
New source: ?
```

Instructions unchanged

Persisting symbol main : old size: 15 bytes, new size: 15 bytes, delta: 0 bytes

```
Old source: ?
New source: ?
```

Instructions unchanged

Disappeared Symbols ↩

Disappeared symbol func(int) : size: 14 bytes

```
Source: ?
```

```
push    %rbp
mov     %rsp,%rbp
mov     %edi,-0x4(%rbp)
mov     $0x2a,%eax
pop     %rbp
ret
xchg    %ax,%ax
```

Disappeared symbol Test::f(int, int) : size: 17 bytes

```
Source: ?
```

```
push    %rbp
mov     %rsp,%rbp
mov     %edi,-0x4(%rbp)
mov     %esi,-0x8(%rbp)
mov     $0x2a,%eax
pop     %rbp
ret
cs nopw 0x0(%rax,%rax,1)
nopl    0x0(%rax,%rax,1)
```

Disappeared symbol Test::g(float, float) : size: 25 bytes

```
Source: ?
```

```
push    %rbp
mov     %rsp,%rbp
mov     %rdi,-0x8(%rbp)
movss   %xmm0,-0xc(%rbp)
movss   %xmm1,-0x10(%rbp)
mov     $0x1,%eax
pop     %rbp
ret
nopl    0x0(%rax)
```

New Symbols ↩

Appeared symbol func(double) : size: 16 bytes

```
Source: ?
```

```
push   %rbp
mov    %rsp,%rbp
movsd  %xmm0,-0x8(%rbp)
mov    $0x2a,%eax
pop    %rbp
ret
```

Appeared symbol Test1::f(int, int) : size: 17 bytes

Source: ?

```
push   %rbp
mov    %rsp,%rbp
mov    %edi,-0x4(%rbp)
mov    %esi,-0x8(%rbp)
mov    $0x2a,%eax
pop    %rbp
ret
cs nopw 0x0(%rax,%rax,1)
nopl   0x0(%rax,%rax,1)
```

Appeared symbol Test1::g(float, float) : size: 25 bytes

Source: ?

```
push   %rbp
mov    %rsp,%rbp
mov    %rdi,-0x8(%rbp)
movss  %xmm0,-0xc(%rbp)
movss  %xmm1,-0x10(%rbp)
mov    $0x1,%eax
pop    %rbp
ret
nopl   0x0(%rax)
```

Similar Symbols ↩

Similar pair 0 : old size: 17 bytes, new size: 17 bytes, delta: 0 bytes, sig. sim.: 88.9 %, instr. sim.: 100.0 %

```
Old: Test::f(int, int) [?]
New: Test1::f(int, int) [?]
```

Instructions unchanged

Similar pair 1 : old size: 25 bytes, new size: 25 bytes, delta: 0 bytes, sig. sim.: 88.9 %, instr. sim.: 100.0 %

```
Old: Test::g(float, float) [?]
New: Test1::g(float, float) [?]
```

Instructions unchanged

Similar pair 3 : old size: 14 bytes, new size: 16 bytes, delta: 2 bytes, sig. sim.: 87.5 %, instr. sim.: 85.1 %

```
Old: func(int) [?]
New: func(double) [?]
```

	Old				New		
f	1	push	%rbp	f	1	push	%rbp
	2	mov	%rsp,%rbp		2	mov	%rsp,%rbp
n	3	mov	%edi,-0x4(%rbp)	n	3	movsd	%xmm0,-0x8(%rbp)
	4	mov	$0x2a,%eax		4	mov	$0x2a,%eax
	5	pop	%rbp		5	pop	%rbp
	6	ret			6	ret	
t	7	xchg	%ax,%ax	t			
	8				7		

Similar pair 6 : old size: 17 bytes, new size: 25 bytes, delta: 8 bytes, sig. sim.: 66.7 %, instr. sim.: 75.1 %

```
Old: Test::f(int, int) [?]
New: Test1::g(float, float) [?]
```

	Old					New		
f	1	push	%rbp	f	1	push	%rbp
	2	mov	%rsp,%rbp		2	mov	%rsp,%rbp
n	3	mov	%edi,-0x4(%rbp)	n	3	mov	%rdi,-0x8(%rbp)
	4	mov	%esi,-0x8(%rbp)		4	movss	%xmm0,-0xc(%rbp)
					5	movss	%xmm1,-0x10(%rbp)
	5	mov	$0x2a,%eax		6	mov	$0x1,%eax
	6	pop	%rbp		7	pop	%rbp
	7	ret			8	ret	
t	8	cs nopw 0x0(%rax,%rax,1)		t			
	9	nopl	0x0(%rax,%rax,1)		9	nopl	0x0(%rax)
	10				10		

Similar pair 7 : old size: 25 bytes, new size: 17 bytes, delta: -8 bytes, sig. sim.: 66.7 %, instr. sim.: 75.1 %

Old: Test::g(float, float) [?]
New: Test1::f(int, int) [?]

	Old					New		
f	1	push	%rbp	f	1	push	%rbp
	2	mov	%rsp,%rbp		2	mov	%rsp,%rbp
n	3	mov	%rdi,-0x8(%rbp)	n	3	mov	%edi,-0x4(%rbp)
	4	movss	%xmm0,-0xc(%rbp)		4	mov	%esi,-0x8(%rbp)
	5	movss	%xmm1,-0x10(%rbp)				
	6	mov	$0x1,%eax		5	mov	$0x2a,%eax
	7	pop	%rbp		6	pop	%rbp
	8	ret			7	ret	
t				t	8	cs nopw 0x0(%rax,%rax,1)	
	9	nopl	0x0(%rax)		9	nopl	0x0(%rax,%rax,1)
	10				10		

Generated 2023-11-28 18:40:14 by elf_diff f07eee2916e9741448df447f829beea10aa98365 (https://github.com/noseglasses/elf_diff)

Using sortable tables from kryogenix.org