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	6
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	OBJECT	4	4	0
__abi_tag	OBJECT	32	32	0
_start ⓘ	FUNC	38	38	0
completed.0	OBJECT	1	1	0
main ⓘ	FUNC	15	15	0
persisting1(int) ⓘ	FUNC	14	14	0
persisting2(int) ⓘ	FUNC	14	14	0
var	OBJECT	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)` ⓘ	FUNC	25
`Test::f(int, int)` ⓘ	FUNC	17
`func(int)` ⓘ	FUNC	14
`Test::m_`	OBJECT	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)` ⓘ	FUNC	25
`Test1::f(int, int)` ⓘ	FUNC	17
`func(double)` ⓘ	FUNC	16
`Test1::m_`	OBJECT	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::g(float, float)
	Test1::g(float, float)
		FUNC
		FUNC
			25
			25
				0	97.7	100.0

1
ⓘ
	Test::f(int, int)
	Test1::f(int, int)
		FUNC
		FUNC
			17
			17
				0	97.1	100.0

2	Test::m_ Test1::m_	OBJECT OBJECT	4 4	0	94.1	100.0

3
ⓘ
	Test::f(int, int)
	Test1::g(float, float)
		FUNC
		FUNC
			17
			25
				8	61.5	59.9

4
ⓘ
	Test::g(float, float)
	Test1::f(int, int)
		FUNC
		FUNC
			25
			17
				-8	61.5	59.9

5
ⓘ
	func(int)
	func(double)
		FUNC
		FUNC
			14
			16
				2	57.1	87.1

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 _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

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 rbp, rsp		2 mov rbp, rsp
	3 mov dword ptr [rbp - 4], edi		3 mov dword ptr [rbp - 4], edi
t	4 mov eax, 0x2b	t	4 mov eax, 0x2a
	5 pop rbp		5 pop rbp
	6 ret		6 ret
	7 nop		7 nop
	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 rbp, rsp		2	mov rbp, rsp
	3	mov dword ptr [rbp - 4], edi		3	mov dword ptr [rbp - 4], edi
t	4	mov eax, 0x2b	t	4	mov eax, 0x2a
	5	pop rbp		5	pop rbp
	6	ret		6	ret
	7	nop		7	nop
	8			8	

Disappeared Symbols ↩

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

```
Source: ?
```

```
push rbp
mov rbp, rsp
mov dword ptr [rbp - 4], edi
mov dword ptr [rbp - 8], esi
mov eax, 0x2a
pop rbp
ret
nop word ptr cs:[rax + rax]
```

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

```
Source: ?
```

```
push rbp
mov rbp, rsp
mov qword ptr [rbp - 8], rdi
movss dword ptr [rbp - 0xc], xmm0
movss dword ptr [rbp - 0x10], xmm1
mov eax, 1
pop rbp
ret
nop dword ptr [rax]
```

Disappeared symbol func(int) : size: 14 bytes

Source: ?

```
push rbp
mov rbp, rsp
mov dword ptr [rbp - 4], edi
mov eax, 0x2a
pop rbp
ret
nop
```

New Symbols ↩

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

Source: ?

```
push rbp
mov rbp, rsp
mov dword ptr [rbp - 4], edi
mov dword ptr [rbp - 8], esi
mov eax, 0x2a
pop rbp
ret
nop word ptr cs:[rax + rax]
```

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

```
Source: ?
```

```
push rbp
mov rbp, rsp
mov qword ptr [rbp - 8], rdi
movss dword ptr [rbp - 0xc], xmm0
movss dword ptr [rbp - 0x10], xmm1
mov eax, 1
pop rbp
ret
nop dword ptr [rax]
```

Appeared symbol func(double) : size: 16 bytes

```
Source: ?
```

```
push rbp
mov rbp, rsp
movsd qword ptr [rbp - 8], xmm0
mov eax, 0x2a
pop rbp
ret
push rbp
```

Similar Symbols ↩

Similar pair 0 : old size: 25 bytes, new size: 25 bytes, delta: 0 bytes, sig. sim.: 97.7 %, instr. sim.: 100.0 %

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

Instructions unchanged

Similar pair 1 : old size: 17 bytes, new size: 17 bytes, delta: 0 bytes, sig. sim.: 97.1 %, instr. sim.: 100.0 %

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

Instructions unchanged

Similar pair 3 : old size: 17 bytes, new size: 25 bytes, delta: 8 bytes, sig. sim.: 61.5 %, instr. sim.: 59.9 %

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

	Old			New	
f	1	push rbp	f	1	push rbp
	2	mov rbp, rsp		2	mov rbp, rsp
n	3	mov dword ptr [rbp - 4], edi	n	3	mov qword ptr [rbp - 8], rdi
	4	mov dword ptr [rbp - 8], esi		4	movss dword ptr [rbp - 0xc], xmm0
				5	movss dword ptr [rbp - 0x10], xmm1
	5	mov eax, 0x2a		6	mov eax, 1
	6	pop rbp		7	pop rbp
	7	ret		8	ret
t	8	nop word ptr cs:[rax + rax]	t	9	nop dword ptr [rax]
	9			10	

Similar pair 4 : old size: 25 bytes, new size: 17 bytes, delta: -8 bytes, sig. sim.: 61.5 %, instr. sim.: 59.9 %

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

	Old			New	
f	1	push rbp	f	1	push rbp
	2	mov rbp, rsp		2	mov rbp, rsp
n	3	mov qword ptr [rbp - 8], rdi	n	3	mov dword ptr [rbp - 4], edi
	4	movss dword ptr [rbp - 0xc], xmm0		4	mov dword ptr [rbp - 8], esi

	Old		New
	5 movss dword ptr [rbp - 0x10], xmm1		
	6 mov eax, 1		5 mov eax, 0x2a
	7 pop rbp		6 pop rbp
	8 ret		7 ret
t	9 nop dword ptr [rax]	t	8 nop word ptr cs:[rax + rax]
	10		9

Similar pair 5 : old size: 14 bytes, new size: 16 bytes, delta: 2 bytes, sig. sim.: 57.1 %, instr. sim.: 87.1 %

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

	Old		New
f	1 push rbp	f	1 push rbp
	2 mov rbp, rsp		2 mov rbp, rsp
n	3 mov dword ptr [rbp - 4], edi	n	3 movsd qword ptr [rbp - 8], xmm0
	4 mov eax, 0x2a		4 mov eax, 0x2a
	5 pop rbp		5 pop rbp
	6 ret		6 ret
t	7 nop	t	7 push rbp
	8		8

Generated 2023-11-28 19:41:33 by elf_diff cc4f547fd5217e77595e4c02ac799461dc62cf8f (https://github.com/noseglasses/elf_diff)

Using sortable tables from kryogenix.org