

ELF Binary Comparison

generated: 2021-08-11 23:18:34

Overview

- Binaries
- Resource Consumption
 - Program Storage
 - Static RAM
 - Legend
- Symbols
 - Persisting Symbols
 - Symbols Disappeared
 - New Symbols
 - Similar Symbols
- Details
 - Persisting Symbols
 - Symbols Disappeared
 - New Symbols
 - Similar Symbols

Binaries

old: /usr/lib/gcc/x86_64-linux-gnu/4.8/libstdc++.a

new: /usr/lib/gcc/x86_64-linux-gnu/5/libstdc++.a

Resource Consumption

Program Storage

	Old/bytes	New/bytes	Delta/bytes
overall	1476	1559	+83
text	1460	1543	+83
data	16	16	+0

Static RAM

	Old/bytes	New/bytes	Delta/bytes
overall	16	16	+0
data	16	16	+0
bss	0	0	+0

Legend

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

Symbols

159 symbols found in /usr/lib/gcc/x86_64-linux-gnu/4.8/libstdc++.a

163 symbols found in /usr/lib/gcc/x86_64-linux-gnu/5/libstdc++.a

Persisting Symbols

158 symbols persisted but potentially changed size (+1077 bytes of program memory)

Symbol ↕	Type ↕	Old Size/bytes ↕	New Size/bytes ↕	Delta/bytes ↕
std::string::assign(std::string const&)	W	159	299	+140
std::string::_M_mutate(unsigned long, unsigned long, unsigned long)	W	403	504	+101
std::string::replace(unsigned long, unsigned long, char const*, unsigned long)	W	428	515	+87
std::string::reserve(unsigned long)	W	109	194	+85
std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const	W	146	201	+55
std::string::insert(unsigned long, char const*, unsigned long)	W	375	422	+47
std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const	W	114	160	+46
std::string::_Rep::_M_dispose(std::allocator<char> const&)	W	16	59	+43
std::string::compare(unsigned long, unsigned long, std::string const&) const	W	122	161	+39
std::string::compare(unsigned long, unsigned long, char const*) const	W	140	176	+36
std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)	W	120	141	+21
std::string::assign(std::string const&, unsigned long, unsigned long)	W	49	67	+18
std::string::insert(unsigned long, std::string const&, unsigned long, unsigned long)	W	49	67	+18
std::string::replace(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long)	W	49	67	+18
std::string::_M_check(unsigned long, char const*) const	W	22	38	+16
std::string::find_last_not_of(char const*, unsigned long, unsigned long) const	W	130	146	+16
std::string::ifind(char const*, unsigned long, unsigned long) const	W	130	146	+16
std::string::_S_construct(unsigned long, char, std::allocator<char> const&)	W	110	125	+15
std::string::append(char const*, unsigned long)	W	253	268	+15
std::string::erase(unsigned long, unsigned long)	W	53	68	+15
std::string::insert(unsigned long, unsigned long, char)	W	36	51	+15

std::string::replace(unsigned long, unsigned long, unsigned long, char)	W	47	62	+15
std::string::substr(unsigned long, unsigned long) const	W	35	50	+15
std::string::_M_leak_hard()	W	50	64	+14
std::string::copy(char*, unsigned long, unsigned long) const	W	87	99	+12
std::string::pop_back()	W	41	53	+12
std::string::append(std::string const&, unsigned long, unsigned long)	W	197	208	+11
std::string::append(std::string const&)	W	161	170	+9
std::string::compare(char const*) const	W	114	123	+9
std::string::compare(std::string const&) const	W	86	95	+9
std::string::end()	W	28	37	+9
std::string::_M_assign(char*, unsigned long, char)	W	27	35	+8
std::string::_M_copy(char*, char const*, unsigned long)	W	22	30	+8
std::string::_M_move(char*, char const*, unsigned long)	W	22	30	+8
std::string::_S_copy_chars(char*, __gnu_cxx::__normal_iterator<char const*, std::string>, __gnu_cxx::__normal_iterator<char const*, std::string>)	W	22	30	+8
std::string::_S_copy_chars(char*, __gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>)	W	22	30	+8
std::string::_S_copy_chars(char*, char const*, char const*)	W	22	30	+8
std::string::_S_copy_chars(char*, char*, char*)	W	22	30	+8
std::string::find_last_not_of(char, unsigned long) const	W	72	80	+8
std::string::find_last_of(char const*, unsigned long, unsigned long) const	W	119	127	+8
std::string::_Rep::_M_refcopy()	W	40	46	+6
std::string::at(unsigned long)	W	58	64	+6
std::string::operator+=(char)	W	97	103	+6
std::string::push_back(char)	W	94	100	+6
std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)	W	59	64	+5
std::string::_Rep::_M_set_length_and_sharable(unsigned long)	W	27	32	+5
std::string::assign(char const*, unsigned long)	W	232	237	+5
std::string::at(unsigned long) const	W	26	31	+5
std::string::append(unsigned long, char)	W	179	183	+4
std::string::back()	W	45	46	+1
std::string::begin()	W	24	25	+1
std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>)	W	61	62	+1
std::string::front()	W	24	25	+1
std::string::operator[](unsigned long)	W	40	41	+1
std::string::replace(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>, char const*)	W	66	67	+1
std::string::rfind(char, unsigned long) const	W	61	62	+1
std::string::find(char, unsigned long) const	W	83	82	-1
std::string::find(char const*, unsigned long, unsigned long) const	W	192	188	-4
std::string::find_first_not_of(char const*, unsigned long, unsigned long) const	W	112	105	-7
std::string::find_first_of(char const*, unsigned long, unsigned long) const	W	108	101	-7
std::string::resize(unsigned long, char)	W	80	72	-8

Symbols Disappeared

1 symbols disappeared (-44 bytes of program memory)

Symbol ↕	Type ↕	Size/bytes ↕
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]	t	44

New Symbols

5 symbols were newly introduced (+228 bytes of program memory)

Symbol ↕	Type ↕	Size/bytes ↕
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]	t	50
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]	t	50
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]	t	50
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]	t	39
std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]	t	39

Similar Symbols

5 pairs of similar symbols have been found.

Similarities may be due to symbols having been renamed or moved between classes and namespaces. Please note that the listed pairs are not necessarily related. The table tries to aid finding actually related symbol pairs by listing those pairs that exhibit similarities in terms of namespaces, class names, function/method names or parameter signatures.

ID ↕	Symbols ↕	Types ↕	Sizes/bytes ↕	Delta/bytes ↕	Sym. Sim./% ↕	Instr. Sim./% ↕
1	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12] std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]	t	44	-5	99.3	86.8
		t	39			
2	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12] std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]	t	44	-5	98.7	80.8
		t	39			
3	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12] std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]	t	44	+6	98.7	55.5
		t	50			
4	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12] std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]	t	44	+6	98.7	53.8
		t	50			

5	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12] std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]	t t	44 50	<div>+6</div>	98.7	52.2

Symbol Details

Persisting Symbols

std::string::_M_assign(char*, unsigned long, char) (size: 27 -> 35 bytes (+8 bytes))

Old			New		
f	1	cmp \$0x1,%rsi	f	1	cmp \$0x1,%rsi
t	2	mov <div>-%rsi,%rax</div>	t	2	je <div>20</div> <std::string::_M_assign(char*, unsigned long, char)+0x20>
				3	test %rsi,%rsi
	3	je <div>18</div> <std::string::_M_assign(char*, unsigned long, char)+0x18>		4	jne <div>10</div> <std::string::_M_assign(char*, unsigned long, char)+0x10>
				5	repz retq
				6	nopl (%rax)
	4	movsbl %dl,%esi		7	movsbl %dl,%eax
	5	mov %rax,%rdx		8	mov %rsi,%rdx
				9	mov %eax,%esi
	6	jmpq <div>14</div> <std::string::_M_assign(char*, unsigned long, char)+0x14> <div>10</div> : R_X86_64_PLT32 memset-0x4		10	jmpq <div>1d</div> <std::string::_M_assign(char*, unsigned long, char)+0x1d> <div>19</div> : R_X86_64_PLT32
	7	nopl <div>0x0</div> (%rax)		11	nopl (%rax)
	8	mov %dl,(%rdi)		12	mov %dl,(%rdi)
	9	retq		13	retq
	10	Disassembly of section .text._ZN5s13_5_copy_charsEPcN9__gnu_cxx17__normal_iteratorI5_SsEES2_:		14	Disassembly of section .text._ZN5s13_5_copy_charsEPcN9__gnu_cxx17__normal_iteratorI5_SsE

std::string::_M_check(unsigned long, char const*) const (size: 22 -> 38 bytes (+16 bytes))

Old			New		
n	1	mov (%rdi),%rax	n	1	mov (%rdi),%rcx
	2	cmp <div>-0x18</div> (%rax),%rsi			
	3	ja <div>d</div> <std::string::_M_check(unsigned long, char const*) const+0xd>			
	4	mov %rsi,%rax		2	mov %rsi,%rax
t	5	retq	t	3	mov <div>-0x18</div> (%rcx),%rcx
				4	cmp %rsi,%rcx
				5	jb <div>11</div> <std::string::_M_check(unsigned long, char const*) const+0x11>
				6	repz retq
				7	lea <div>0x0</div> (%rip),%rdi <div># 18</div> <std::string::_M_check(unsigned long, char const*) const+0x18> <div>14</div> : R_
	6	push %rax		8	push %rsi
	7	mov %rdx,%rdi		9	mov %rdx,%rsi
				10	mov %rax,%rdx
				11	xor %eax,%eax
	8	callq <div>16</div> <.LC1+0x5> <div>12</div> : R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4		12	callq <div>26</div> <.LC73+0x13> <div>22</div> : R_X86_64_PLT32 std::__throw_out_of_range_fmt(char const*, ...)-0x4
	9	Disassembly of section .text._ZNKSs15_M_check_lengthEmmPKc:		13	Disassembly of section .text._ZNKSs15_M_check_lengthEmmPKc:

std::string::_M_check_length(unsigned long, unsigned long, char const*) const (size unchanged)

Old			New		
n	1	mov (%rdi),%r8	n	1	mov (%rdi),%rdi
	2	movabs \$0x3fffffffffffffff9,%rax		2	movabs \$0x3fffffffffffffff9,%rax
n	3	sub -0x18(%r8),%rax	n	3	sub -0x18(%rdi),%rax
	4	add %rsi,%rax		4	add %rax,%rsi
	5	cmp %rdx,%rax		5	cmp %rdx,%rsi
	6	jb <div>1b</div> <std::string::_M_check_length(unsigned long, unsigned long, char const*) const+0x1b>		6	jb <div>1b</div> <std::string::_M_check_length(unsigned long, unsigned long, char const*) const+0x1b>
	7	repz retq		7	repz retq
	8	push %rax		8	push %rax
	9	mov %rcx,%rdi		9	mov %rcx,%rdi
t	10	callq <div>24</div> <.LC2> <div>20</div> : R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4	t	10	callq <div>24</div> <.LC73+0x11> <div>20</div> : R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4
	11	Disassembly of section .text._ZNKSs8_M_limitEmm:		11	Disassembly of section .text._ZNKSs8_M_limitEmm:

std::string::_M_copy(char*, char const*, unsigned long) (size: 22 -> 30 bytes (+8 bytes))

Old			New		
f	1	cmp \$0x1,%rdx	f	1	cmp \$0x1,%rdx
t			t	2	je <div>18</div> <std::string::_M_copy(char*, char const*, unsigned long)+0x18>
				3	test %rdx,%rdx
	2	je <div>10</div> <std::string::_M_copy(char*, char const*, unsigned long)+0x10>		4	jne <div>10</div> <std::string::_M_copy(char*, char const*, unsigned long)+0x10>
				5	repz retq
				6	nopl (%rax)
	3	jmpq <div>b</div> <std::string::_M_copy(char*, char const*, unsigned long)+0xb> <div>7</div> : R_X86_64_PLT32 memcpy-0x4		7	jmpq <div>15</div> <std::string::_M_copy(char*, char const*, unsigned long)+0x15> <div>11</div> : R_X86_64_PL
	4	nopl <div>0x0</div> (%rax,%rax,1)		8	nopl (%rax)
	5	movzbl (%rsi),%eax		9	movzbl (%rsi),%eax
	6	mov %al,(%rdi)		10	mov %al,(%rdi)
	7	retq		11	retq
	8	Disassembly of section .text._ZN5s7_M_moveEPcPKcm:		12	Disassembly of section .text._ZN5s7_M_moveEPcPKcm:

std::string::_M_disjunct(char const*) const (size unchanged)

Old	New
-----	-----

f	1	mov	(%rdi),%rdx	f	1	mov	(%rdi),%rdx
	2	mov	\$0x1,%eax		2	mov	\$0x1,%eax
t	3	cmp	%rdx,%rsi	t	3	cmp	%rsi,%rdx
	4	jb	17 <std::string::_M_disjunct(char const*) const+0x17>		4	ja	17 <std::string::_M_disjunct(char const*) const+0x17>
	5	add	-0x18(%rdx),%rdx		5	add	-0x18(%rdx),%rdx
	6	cmp	%rdx,%rsi		6	cmp	%rdx,%rsi
	7	seta	%al		7	seta	%al
	8	repz	retq		8	repz	retq
	9	Disassembly of section .text._ZN5s7_M_copyEPcPKcm:			9	Disassembly of section .text._ZN5s7_M_copyEPcPKcm:	

std::string::_M_leak_hard() (size: 50 -> 64 bytes (+14 bytes))

Old							
f	1	mov	(%rdi),%rax	f	1	mov	(%rdi),%rax
n	2	push	%rbx	n			
	3	mov	%rdi,%rbx				
	4	lea	-0x18(%rax),%rdx		2	lea	-0x18(%rax),%rdx
n	5	cmp	0x0(%rip),%rdx # 12 <std::string::_M_leak_hard()+0x12> 8: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	n	3	cmp	0x0(%rip),%rdx # 8 <std::string::_M
	6	je	30 <std::string::_M_leak_hard()+0x30>		4	je	31 <std::string::_M_leak_hard()+0x31>
	7	mov	-0x8(%rax),%edx		5	mov	-0x8(%rax),%edx
	8	test	%edx,%edx		6	test	%edx,%edx
n	9	jle	29 <std::string::_M_leak_hard()+0x29>	n	7	jle	38 <std::string::_M_leak_hard()+0x38>
					8	push	%rbx
	10	xor	%ecx,%ecx		9	xor	%ecx,%ecx
n				n	10	mov	%rdi,%rbx
	11	xor	%edx,%edx		11	xor	%edx,%edx
	12	xor	%esi,%esi		12	xor	%esi,%esi
	13	callq	26 <std::string::_M_leak_hard()+0x26> 22: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4		13	callq	26 <std::string::_M_leak_hard()+0x26> 22: f
	14	mov	(%rbx),%rax		14	mov	(%rbx),%rax
	15	movl	\$0xffffffff,-0x8(%rax)		15	movl	\$0xffffffff,-0x8(%rax)
	16	pop	%rbx		16	pop	%rbx
t				t	17	repz	retq
					18	nopl	0x0(%rax,%rax,1)
					19	movl	\$0xffffffff,-0x8(%rax)
	17	retq			20	retq	
	18	Disassembly of section .text._ZN5s7_M_leakEv:			21	Disassembly of section .text._ZN5s7_M_leakEv:	

std::string::_M_move(char*, char const*, unsigned long) (size: 22 -> 30 bytes (+8 bytes))

Old				New			
f	1	cmp	\$0x1,%rdx	f	1	cmp	\$0x1,%rdx
t				t	2	je	18 <std::string::_M_move(char*, char const*, unsigned long)+0x18>
					3	test	%rdx,%rdx
	2	je	10 <std::string::_M_move(char*, char const*, unsigned long)+0x10>		4	jne	10 <std::string::_M_move(char*, char const*, unsigned long)+0x10>
					5	repz	retq
					6	nopl	(%rax)
	3	jmpq	b <std::string::_M_move(char*, char const*, unsigned long)+0xb> 7: R_X86_64_PLT32 memmove-0x4		7	jmpq	15 <std::string::_M_move(char*, char const*, unsigned long)+0x15> 11: R_X86_64_f
	4	nopl	0x0(%rax,%rax,1)		8	nopl	(%rax)
	5	movzbl	(%rsi),%eax		9	movzbl	(%rsi),%eax
	6	mov	%al,(%rdi)		10	mov	%al,(%rdi)
	7	retq			11	retq	
	8	Disassembly of section .text._ZN5s9_M_assignEPcmc:			12	Disassembly of section .text._ZN5s9_M_assignEPcmc:	

std::string::_M_mutate(unsigned long, unsigned long, unsigned long) (size: 403 -> 504 bytes (+101 bytes))

Old			
f	1	push	%r15
	2	push	%r14
n	3	mov	%rcx,%r14
	4	sub	%rdx,%r14
	5	push	%r13
n	6	mov	%rdi,%r13
	7	push	%r12
n	8	mov	%rdx,%r12
	9	push	%rbp
n	10	mov	%rsi,%rbp
	11	push	%rbx
n			
	12	sub	\$0x28,%rsp
n			
	13	mov	(%rdi),%rax
n	14	mov	-0x18(%rax),%rbx

[illegible]

t	96	mov %rax,%r8
	97	jmpq 80 <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0x80>
	98	nopl (%rax)
	99	mov %rbx,%rdx
	100	mov %r8,0x8(%rsp)
	101	callq 15d <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0x15d> 159: R_X86_64_PLT32 memcpy-0x4
	102	mov 0x8(%rsp),%r8
	103	jmpq 85 <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0x85>
	104	movl \$0x0,-0x8(%rax)
	105	mov %r14,-0x18(%rax)
	106	movb \$0x0,(%rax,%r14,1)
	107	jmpq b1 <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0xb1>
	108	mov %r15,%rsi
	109	mov %r8,0x8(%rsp)
	110	callq 189 <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0x189> 185: R_X86_64_PC32 .text.unlikely._ZN5s4_Rep10_M_disposeERKSaIcE.part.12-0x4
	111	mov 0x8(%rsp),%r8
	112	jmpq 9d <std::string::_M_mutate(unsigned long, unsigned long, unsigned long)+0x9d>
	113	Disassembly of section .text._ZN5s12_M_leak_hardEv:

std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char) (size unchanged)

		Old		
f	1	push %r13	f	1
n			n	2
	2	movabs \$0x3fffffffffffff9,%rax		3
n	3	push %r12	n	
	4	push %rbp		4
n	5	mov %rdi,%rbp	n	
	6	push %rbx		5
	7	mov %rcx,%rbx		6
	8	sub \$0x8,%rsp		7
	9	mov (%rdi),%rcx		8
	10	sub -0x18(%rcx),%rax		9
	11	add %rdx,%rax		10
	12	cmp %rax,%rbx		11
	13	ja 79 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x79>		12
	14	mov %rbx,%rcx		13
n			n	14
	15	mov %rsi,%r12		15
	16	mov %r8d,%r13d		16
	17	callq 37 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x37> 33: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4		17
	18	test %rbx,%rbx		18
	19	je 55 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x55>		19
	20	mov %r12,%rdi		20
	21	add 0x0(%rbp),%rdi		21
	22	cmp \$0x1,%rbx		22
	23	je 68 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x68>		23
	24	movsbl %r13b,%esi		24
	25	mov %rbx,%rdx		25
	26	callq 55 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x55> 51: R_X86_64_PLT32 memset-0x4		26
	27	add \$0x8,%rsp		27
	28	mov %rbp,%rax		28
	29	pop %rbx		29
	30	pop %rbp		30
	31	pop %r12		31
	32	pop %r13		32
	33	retq		33
	34	nopl 0x0(%rax,%rax,1)		34
	35	mov %r13b,(%rdi)		35
	36	add \$0x8,%rsp		36
	37	mov %rbp,%rax		37
	38	pop %rbx		38
	39	pop %rbp		39
	40	pop %r12		40
	41	pop %r13		41
	42	retq		42
t	43	lea 0x0(%rip),%rdi # 80 <std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)+0x80> 7c: R_X86_64_PC32 .LC5-0x4	t	43
	44	callq 85 <.LC6+0x2> 81: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4		44
	45	Disassembly of section .text._ZN5s6assignEmc:		45

std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long) (size unchanged)

		Old
f	1	push %r13
n		

	2	mov %rcx,%r13	
n			
	3	mov %r8,%rcx	
n	4	push %r12	
	5	mov %rsi,%r12	
t	6	push %rbp	
	7	mov %rdi,%rbp	
	8	push %rbx	
	9	mov %r8,%rbx	
	10	sub \$0x8,%rsp	
	11	callq 1e <std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)+0x1e> 1a: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x	
	12	test %rbx,%rbx	
	13	je 3b <std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)+0x3b>	
	14	mov %r12,%rdi	
	15	add 0x0(%rbp),%rdi	
	16	cmp \$0x1,%rbx	
	17	je 50 <std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)+0x50>	
	18	mov %rbx,%rdx	
	19	mov %r13,%rsi	
	20	callq 3b <std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)+0x3b> 37: R_X86_64_PLT32 memcpy-0x4	
	21	add \$0x8,%rsp	
	22	mov %rbp,%rax	
	23	pop %rbx	
	24	pop %rbp	
	25	pop %r12	
	26	pop %r13	
	27	retq	
	28	nopl 0x0(%rax)	
	29	movzbl 0x0(%r13),%eax	
	30	mov %al,(%rdi)	
	31	add \$0x8,%rsp	
	32	mov %rbp,%rax	
	33	pop %rbx	
	34	pop %rbp	
	35	pop %r12	
	36	pop %r13	
	37	retq	
	38	Disassembly of section .text._ZN5s6assignEPKcm:	

std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long) (size: 120 -> 141 bytes (+21 bytes))

Old			
f	1	push %rbp	
	2	push %rbx	
	3	mov %rdi,%rbx	
	4	mov %rdx,%rdi	
	5	mov %rsi,%rdx	
	6	sub \$0x8,%rsp	
	7	add (%rbx),%rdi	
	8	mov 0x8(%rbx),%rsi	
	9	callq 1b <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x1b> 17: R_X86_64_PLT32 std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&	
	>	x4	
	10	mov (%rbx),%rdx	
	11	mov %rax,%rbp	
	12	lea 0x18(%rax),%rax	
	13	test %rdx,%rdx	
n	14	je 43 <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x43>	
	15	cmp \$0x1,%rdx	
	16	lea 0x18(%rbp),%rcx	
n	17	je 58 <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x58>	
	18	lea 0x18(%rbx),%rsi	
	19	mov %rcx,%rdi	
	20	callq 40 <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x40> 3c: R_X86_64_PLT32 memcpy-0x4	
	21	mov (%rbx),%rdx	
n			
	22	cmp 0x0(%rip),%rbp # 4a <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x4a> 46: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	
	23	jne 67 <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x67>	
	24	add \$0x8,%rsp	
	25	pop %rbx	
	26	pop %rbp	
	27	retq	
	28	nopl 0x0(%rax,%rax,1)	
	29	movzbl 0x18(%rbx),%eax	

n		
30	mov	%al,0x18(%rbp)
31	mov	%rcx,%rax
32	mov	(%rbx),%rdx
n	33	jmp 43 <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x43>
34	movl	\$0x0,0x10(%rbp)
35	mov	%rdx,0x0(%rbp)
t	36	movb \$0x0,(%rax,%rdx,1)
37	jmp	4c <std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)+0x4c>
38		Disassembly of section .text._ZN5s7reserveEm:

std::string::_Rep::_M_destroy(std::allocator<char> const&) (size unchanged)

Old			New		
f	1	jmpq 5 <std::string::_Rep::_M_destroy(std::allocator<char> const&)+0x5> 1: R_X86_64_PLT32 operator delete(void*)-0x4	f	1	jmpq 5 <std::string::_Rep::_M_destroy(std::allocator<char> const&)+0x5>
t	2	Disassembly of section .text.unlikely._ZN5s4_Rep10_M_disposeERKSaIcE.part.12:	t	2	Disassembly of section .text._ZN5s4_Rep10_M_disposeERKSaIcE.part.13:

std::string::_Rep::_M_dispose(std::allocator<char> const&) (size: 16 -> 59 bytes (+43 bytes))

Old					
f	1	cmp 0x0(%rip),%rdi # 7 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x7> 3: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	f	1	cmp 0x0(%rip),%rdi
n	2	jne b <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0xb>	n	2	jne 10 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0xb>
	3	repz retq		3	repz retq
t	4	jmpq 10 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x10> c: R_X86_64_PC32 .text.unlikely._ZN5s4_Rep10_M_disposeERKSaIcE.part.12-0x4	t	4	nopl 0x0(%rax,%rax)
				5	cmpq \$0x0,0x0(%rip)
				6	je 30 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x30>
				7	mov \$0xffffffff,%eax
				8	lock xadd %eax,0x10(%rip)
				9	test %eax,%eax
				10	jg 9 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x9>
				11	jmpq 2d <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x2d>
				12	nopl (%rax)
				13	mov 0x10(%rdi),%eax
				14	lea -0x1(%rax),%eax
				15	mov %edx,0x10(%rdi)
				16	jmp 24 <std::string::_Rep::_M_dispose(std::allocator<char> const&)+0x24>
	5	Disassembly of section .text._ZN5sD2Ev:		17	Disassembly of section .text._ZN5sD2Ev:

std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&) (size: 59 -> 64 bytes (+5 bytes))

Old					
f	1	mov 0x10(%rdi),%eax			
	2	test %eax,%eax			
	3	js 18 <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x18>			
	4	cmp 0x0(%rip),%rdi # e <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0xe> a: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4			
n	5	jne 1f <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x1f>			
	6	lea 0x18(%rdi),%rax			
	7	retq			
	8	nopl (%rax)			
	9	xor %edx,%edx			
	10	jmpq 1f <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x1f> 1b: R_X86_64_PLT32 std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)-0:			
n					
	11	cmpq \$0x0,0x0(%rip) # 27 <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x27> 22: R_X86_64_GOTPCREL __pthread_key_create-0x5			
	12	lea 0x10(%rdi),%rdx			
	13	je 33 <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x33>			
	14	lock addl \$0x1,(%rdx)			
	15	jmp 10 <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x10>			
t					
	16	add \$0x1,%eax			
	17	mov %eax,0x10(%rdi)			
	18	jmp 10 <std::string::_Rep::_M_grab(std::allocator<char> const&, std::allocator<char> const&)+0x10>			
	19	Disassembly of section .text._ZN5sC2ERKSs:			

std::string::_Rep::_M_refcopy() (size: 40 -> 46 bytes (+6 bytes))

Old					
f	1	cmp 0x0(%rip),%rdi # 7 <std::string::_Rep::_M_refcopy()+0x7> 3: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	f	1	cmp 0x0(%rip),%rdi # 7 <std::string::_Rep::_M_refcopy()+0x7>
n	2	jne e <std::string::_Rep::_M_refcopy()+0xe>	n	2	jne 10 <std::string::_Rep::_M_refcopy()+0xe>
	3	lea 0x18(%rdi),%rax		3	lea 0x18(%rdi),%rax
	4	retq		4	retq
n			n	5	xchg %ax,%ax
	5	cmpq \$0x0,0x0(%rip) # 16 <std::string::_Rep::_M_refcopy()+0x16> 11: R_X86_64_GOTPCREL __pthread_key_create-0x5		6	cmpq \$0x0,0x0(%rip) # 18 <std::string::_Rep::_M_refcopy()+0x18>
	6	lea 0x10(%rdi),%rax			
	7	je 22 <std::string::_Rep::_M_refcopy()+0x22>		7	je 28 <std::string::_Rep::_M_refcopy()+0x28>
	8	lock addl \$0x1,(%rax)		8	lock addl \$0x1,0x10(%rdi)

9	jmp 9 <std::string::_Rep::_M_refcopy()+0x9>	9	jmp 9 <std::string::_Rep::_M_refcopy()+0x9>
t		t	10 nopl 0x0(%rax)
10	addl \$0x1,0x10(%rdi)	11	addl \$0x1,0x10(%rdi)
11	jmp 9 <std::string::_Rep::_M_refcopy()+0x9>	12	jmp 9 <std::string::_Rep::_M_refcopy()+0x9>
12	Disassembly of section .text._ZN5s4Rep8_M_cloneERKSaIcEm:	13	Disassembly of section .text._ZN5s4Rep8_M_cloneERKSaIcEm:

```
std::string::_Rep::_M_set_length_and_sharable(unsigned long) (size: 27 -> 32 bytes (+5 bytes) )
```

Old									
f	1	cmp 0x0(%rip),%rdi # 7 <std::string::_Rep::_M_set_length_and_sharable(unsigned long)+0x7> 3: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4			f	1	cmp 0x0(%rip),%		
n	2	jne b <std::string::_Rep::_M_set_length_and_sharable(unsigned long)+0xb>			n	2	jne 10 <std::st		
	3	repz retq				3	repz retq		
t					t	4	nopl 0x0(%rax,%r		
	4	movl \$0x0,0x10(%rdi)				5	movl \$0x0,0x10(%		
	5	mov %rsi,(%rdi)				6	mov %rsi,(%rdi)		
	6	movb \$0x0,0x18(%rdi,%rsi,1)				7	movb \$0x0,0x18(%		
	7	retq				8	retq		
	8	Disassembly of section .text._ZNSs4_Rep10_M_refdataEv:				9	Disassembly of sect		

```
std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&) (size unchanged)
```

Old		New	
f	1 movabs \$0x3ffffffffffff9,%rax	f	1 movabs \$0x3ffffffffffff9,%rax
	2 push %rbx		2 push %rbx
n	3 mov %rdi,%rbx	n	
	4 cmp %rax,%rdi		3 cmp %rax,%rdi
	5 ja 66 <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x66>		4 ja 66 <std::string::_Rep::_
n		n	5 mov %rdi,%rbx
			6 lea 0x19(%rdi),%rdi
	6 cmp %rsi,%rbx		7 cmp %rsi,%rbx
n	7 lea 0x19(%rdi),%rdi	n	
	8 jbe 54 <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x54>		8 jbe 54 <std::string::_Rep::_
	9 lea (%rsi,%rsi,1),%rdx		9 lea (%rsi,%rsi,1),%rdx
	10 cmp %rdx,%rbx		10 cmp %rdx,%rbx
	11 cmovb %rdx,%rbx		11 cmovb %rdx,%rbx
	12 lea 0x39(%rbx),%rdx		12 lea 0x39(%rbx),%rdx
	13 cmp \$0x1000,%rdx		13 cmp \$0x1000,%rdx
	14 jbe 50 <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x50>		14 jbe 50 <std::string::_Rep::_
	15 cmp %rbx,%rsi		15 cmp %rbx,%rsi
	16 jae 50 <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x50>		16 jae 50 <std::string::_Rep::_
	17 add \$0x1000,%rbx		17 add \$0x1000,%rbx
	18 and \$0xfff,%edx		18 and \$0xfff,%edx
	19 sub %rdx,%rbx		19 sub %rdx,%rbx
	20 cmp %rax,%rbx		20 cmp %rax,%rbx
	21 cmova %rax,%rbx		21 cmova %rax,%rbx
	22 lea 0x19(%rbx),%rdi		22 lea 0x19(%rbx),%rdi
	23 callq 59 <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x59> 55: R_X86_64_PLT32 operator new(unsigned long)-0x4		23 callq 59 <std::string::_Rep::_
	24 mov %rbx,0x8(%rax)		24 mov %rbx,0x8(%rax)
	25 movl \$0x0,0x10(%rax)		25 movl \$0x0,0x10(%rax)
	26 pop %rbx		26 pop %rbx
	27 retq		27 retq
t	28 lea 0x0(%rip),%rdi # 6d <std::string::_Rep::_S_create(unsigned long, unsigned long, std::allocator<char> const&)+0x6d> 69: R_X86_64_PC32 .LC3-0x4	t	28 lea 0x0(%rip),%rdi #
	29 callq 72 <.LC5+0xc> 6e: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4		29 callq 72 <.LC118> 6e: R_X86_6
	30 Disassembly of section .text._ZNSt12_S_constructEmRKSalcE:		30 Disassembly of section .text._Z

```
std::string::_S_construct(unsigned long, char, std::allocator<char> const&) (size: 110 -> 125 bytes (+15 bytes) )
```

Old		
n		
1	push	%r12
n 2	test	%rdi,%rdi
3	push	%rbp
n		
4	push	%rbx
n		
5	mov	%rdi,%rbx
n 6	jne	20 <std::string::_S_construct(unsigned long, char, std::allocator<char> const&)+0x20>
7	mov	0x0(%rip),%rax # 13 <std::string::_S_construct(unsigned long, char, std::allocator<char> const&)+0x13> f: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4
8	lea	0x18(%rax),%rcx
n		

[illegible]

7	mov %al,(%rdi)	10	mov %al,(%rdi)
8	retq	11	retq
9	Disassembly of section .text._ZN5s13_S_copy_charsEPcN9__gnu_cxx17__normal_iteratorIPKcSsEE54_:	12	Disassembly of section .text._ZN5s13_S_copy_charsEPcN9__gnu_cxx17__normal_iteratorIPKcSsEE54_:

std::string::_S_copy_chars(char*, char const*, char const*) (size: 22 -> 30 bytes (+8 bytes))

Old				New			
f	1	sub %rsi,%rdx		f	1	sub %rsi,%rdx	
	2	cmp \$0x1,%rdx			2	cmp \$0x1,%rdx	
t				t	3	je 18 <std::string::_S_copy_chars(char*, char const*, char const*)+0x18>	
					4	test %rdx,%rdx	
	3	je 10 <std::string::_S_copy_chars(char*, char const*, char const*)+0x10>			5	jne 10 <std::string::_S_copy_chars(char*, char const*, char const*)+0x10>	
					6	repz retq	
	4	jmpq e <std::string::_S_copy_chars(char*, char const*, char const*)+0xe> a: R_X86_64_PLT32 memcpy-0x4			7	jmpq 15 <std::string::_S_copy_chars(char*, char const*, char const*)+0x15> 11: R_X86_64_PLT32 memcpy-0x4	
	5	xchg %ax,%ax			8	nopl (%rax)	
	6	movzbl (%rsi),%eax			9	movzbl (%rsi),%eax	
	7	mov %al,(%rdi)			10	mov %al,(%rdi)	
	8	retq			11	retq	
	9	Disassembly of section .text._ZN5s10_S_compareEmm:			12	Disassembly of section .text._ZN5s10_S_compareEmm:	

std::string::_S_copy_chars(char*, char*, char*) (size: 22 -> 30 bytes (+8 bytes))

Old				New			
f	1	sub %rsi,%rdx		f	1	sub %rsi,%rdx	
	2	cmp \$0x1,%rdx			2	cmp \$0x1,%rdx	
t				t	3	je 18 <std::string::_S_copy_chars(char*, char*, char*)+0x18>	
					4	test %rdx,%rdx	
	3	je 10 <std::string::_S_copy_chars(char*, char*, char*)+0x10>			5	jne 10 <std::string::_S_copy_chars(char*, char*, char*)+0x10>	
					6	repz retq	
	4	jmpq e <std::string::_S_copy_chars(char*, char*, char*)+0xe> a: R_X86_64_PLT32 memcpy-0x4			7	jmpq 15 <std::string::_S_copy_chars(char*, char*, char*)+0x15> 11: R_X86_64_PLT32 memcpy-0x4	
	5	xchg %ax,%ax			8	nopl (%rax)	
	6	movzbl (%rsi),%eax			9	movzbl (%rsi),%eax	
	7	mov %al,(%rdi)			10	mov %al,(%rdi)	
	8	retq			11	retq	
	9	Disassembly of section .text._ZN5s13_S_copy_charsEPcPKcS1_:			12	Disassembly of section .text._ZN5s13_S_copy_charsEPcPKcS1_:	

std::string::append(char const*) (size unchanged)

Old				New			
f	1	push %rbp		f	1	push %rbp	
n				n	2	push %rbx	
	2	mov %rdi,%rbp			3	mov %rdi,%rbp	
	3	mov %rsi,%rdi			4	mov %rsi,%rdi	
n	4	push %rbx		n			
	5	mov %rsi,%rbx			5	mov %rsi,%rbx	
	6	sub \$0x8,%rsp			6	sub \$0x8,%rsp	
	7	callq 14 <std::string::append(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4			7	callq 14 <std::string::append(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4	
	8	add \$0x8,%rsp			8	add \$0x8,%rsp	
	9	mov %rbx,%rsi			9	mov %rbx,%rsi	
	10	mov %rbp,%rdi			10	mov %rbp,%rdi	
	11	pop %rbx			11	pop %rbx	
	12	pop %rbp			12	pop %rbp	
	13	mov %rax,%rdx			13	mov %rax,%rdx	
t	14	jmpq 28 <.LC2+0x4> 24: R_X86_64_PLT32 std::string::append(char const*, unsigned long)-0x4		t	14	jmpq 28 <.LC73+0x15> 24: R_X86_64_PLT32 std::string::append(char const*, unsigned long)-0x4	
	15	Disassembly of section .text._ZN5s6appendESt16initializer_listIcE:			15	Disassembly of section .text._ZN5s6appendESt16initializer_listIcE:	

std::string::append(char const*, unsigned long) (size: 253 -> 268 bytes (+15 bytes))

Old							
f	1	push %r12		f	1	push %r12	
	2	push %rbp			2	push %rbp	
	3	push %rbx			3	push %rbx	
	4	mov %rdi,%rbx			4	mov %rdi,%rbx	
	5	sub \$0x10,%rsp			5	sub \$0x10,%rsp	
	6	test %rdx,%rdx			6	test %rdx,%rdx	
n	7	je 7a <std::string::append(char const*, unsigned long)+0x7a>		n	7	je 7b <std::string::append(char const*, unsigned long)+0x7b>	
	8	mov (%rdi),%rax			8	mov (%rdi),%rdi	
	9	movabs \$0x3fffffffffffffff9,%rcx			9	movabs \$0x3fffffffffffffff9,%rcx	
n	10	mov -0x18(%rax),%rdi		n	10	mov -0x18(%rdi),%rax	
	11	sub %rdi,%rcx			11	sub %rax,%rcx	
	12	cmp %rcx,%rdx			12	cmp %rcx,%rdx	
n	13	ja f1 <std::string::append(char const*, unsigned long)+0xf1>		n	13	ja 100 <std::string::append(char const*, unsigned long)+0x100>	
	14	lea (%rdx,%rdi,1),%r12			14	lea (%rdx,%rax,1),%r12	
	15	cmp -0x10(%rax),%r12			15	cmp -0x10(%rdi),%r12	
	16	mov %rsi,%rbp			16	mov %rsi,%rbp	
	17	jbe 90 <std::string::append(char const*, unsigned long)+0x90>			17	jbe 90 <std::string::append(char const*, unsigned long)+0x90>	
n	18	cmp %rax,%rbp		n	18	cmp %rdi,%rbp	
	19	jae b0 <std::string::append(char const*, unsigned long)+0xb0>			19	jae d0 <std::string::append(char const*, unsigned long)+0xd0>	

`std::string::append(std::initializer_list<char>)` (size unchanged)

Old	New

f	1	jmpq	5	<std::string::append(std::initializer_list<char>)+0x5>	1: R_X86_64_PLT32	std::string::append(char const*, unsigned long)-0x4	f	1	jmpq	5	<std::string::append(std::initializer_list<char>)+0x5>
t	2	Disassembly of section .text._ZN5spLEPKc:					t	2	Disassembly of section .text._ZN5spLEPKc:		

std::string::append(std::string const&) (size: 161 -> 170 bytes (+9 bytes))

Old														
f	1	push	%r13				f	1	push	%r13				
	2	push	%r12					2	push	%r12				
	3	mov	%rsi,%r12					3	mov	%rsi,%r12				
	4	push	%rbp					4	push	%rbp				
	5	push	%rbx					5	push	%rbx				
	6	mov	%rdi,%rbx					6	mov	%rdi,%rbx				
	7	sub	\$0x8,%rsp					7	sub	\$0x8,%rsp				
	8	mov	(%rsi),%rsi					8	mov	(%rsi),%rsi				
	9	mov	-0x18(%rsi),%rbp					9	mov	-0x18(%rsi),%rbp				
	10	test	%rbp,%rbp					10	test	%rbp,%rbp				
n	11	je	64	<std::string::append(std::string const&)+0x64>			n	11	je	5e	<std::string::append(std::string const&)+0x5e>			
	12	mov	(%rdi),%rax					12	mov	(%rdi),%rdi				
	13	mov	-0x18(%rax),%rdi					13	mov	-0x18(%rdi),%rax				
	14	lea	(%rdi,%rbp,1),%r13					14	lea	(%rax,%rbp,1),%r13				
	15	cmp	-0x10(%rax),%r13					15	cmp	-0x10(%rdi),%r13				
	16	jbe	78	<std::string::append(std::string const&)+0x78>				16	jbe	70	<std::string::append(std::string const&)+0x70>			
	17	mov	%r13,%rsi					17	mov	%r13,%rsi				
	18	mov	%rbx,%rdi					18	mov	%rbx,%rdi				
	19	callq	38	<std::string::append(std::string const&)+0x38>	34: R_X86_64_PLT32	std::string::reserve(unsigned long)-0x4		19	callq	38	<std::string::append(std::string const&)+0x38>			
n	20	mov	(%rbx),%rax				n	20	mov	(%rbx),%rdi				
	21	mov	(%r12),%rsi					21	mov	(%r12),%rsi				
n	22	mov	-0x18(%rax),%rdi				n	22	add	-0x18(%rdi),%rdi				
	23	add	%rax,%rdi					23	cmp	\$0x1,%rbp				
	24	cmp	\$0x1,%rbp					24	jne	80	<std::string::append(std::string const&)+0x80>			
n	25	je	88	<std::string::append(std::string const&)+0x88>			n	25	movzbl	(%rsi),%eax				
	26	mov	%rbp,%rdx					26	mov	%al,(%rdi)				
	27	callq	54	<std::string::append(std::string const&)+0x54>	50: R_X86_64_PLT32	memcpy-0x4		27	mov	(%rbx),%rax				
	28	mov	(%rbx),%rax					28	lea	-0x18(%rax),%rdx				
	29	lea	-0x18(%rax),%rdx					29	cmp	0x0(%rip),%rdx	# 5c	<std::string::append(std::string const&)+0x5c>		
n	30	cmp	0x0(%rip),%rdx	# 62	<std::string::append(std::string const&)+0x62>	5e: R_X86_64_REX_GOTPCRELX	n	30	jne	98	<std::string::append(std::string const&)+0x98>			
	31	jne	8f	<std::string::append(std::string const&)+0x8f>				31	add	\$0x8,%rsp				
	32	add	\$0x8,%rsp					32	mov	%rbx,%rax				
	33	mov	%rbx,%rax					33	pop	%rbx				
	34	pop	%rbx					34	pop	%rbp				
	35	pop	%rbp					35	pop	%r12				
	36	pop	%r12					36	pop	%r13				
	37	pop	%r13					37	retq					
	38	retq						38	nopl	0x0(%rax)				
n	39	nopl	0x0(%rax,%rax,1)				n	39	mov	-0x8(%rdi),%edx				
	40	mov	-0x8(%rax),%edx					40	test	%edx,%edx				
	41	test	%edx,%edx					41	jg	2d	<std::string::append(std::string const&)+0x2d>			
	42	jg	2d	<std::string::append(std::string const&)+0x2d>				42	add	%rax,%rdi				
n							n	43	cmp	\$0x1,%rbp				
								44	je	49	<std::string::append(std::string const&)+0x49>			
	43	jmp	43	<std::string::append(std::string const&)+0x43>				45	mov	%rbp,%rdx				
	44	nopl	0x0(%rax)					46	callq	88	<std::string::append(std::string const&)+0x88>			
	45	movzbl	(%rsi),%eax					47	mov	(%rbx),%rax				
	46	mov	%al,(%rdi)					48	lea	-0x18(%rax),%rdx				
								49	cmp	0x0(%rip),%rdx	# 96	<std::string::append(std::string const&)+0x96>		
								50	je	5e	<std::string::append(std::string const&)+0x5e>			
	47	jmp	54	<std::string::append(std::string const&)+0x54>				51	movl	\$0x0,-0x8(%rax)				
	48	movl	\$0x0,-0x8(%rax)					52	mov	%r13,-0x18(%rax)				
	49	mov	%r13,-0x18(%rax)					53	movb	\$0x0,(%rax,%r13,1)				
	50	movb	\$0x0,(%rax,%r13,1)											
t	51	jmp	64	<std::string::append(std::string const&)+0x64>			t	54	jmp	5e	<std::string::append(std::string const&)+0x5e>			
	52	Disassembly of section .text._ZN5spLEPKSs:						55	Disassembly of section .text._ZN5spLEPKSs:					

std::string::append(std::string const&, unsigned long, unsigned long) (size: 197 -> 208 bytes (+11 bytes))

Old							
f	1	push	%r14	f	1	push	%r14
n	2	mov	%rsi,%r14	n			
	3	push	%r13		2	push	%r13
n				n	3	mov	%r13,%r14
	4	push	%r12		4	push	%r12
	5	push	%rbp		5	push	%rbp
	6	mov	%rdx,%rbp		6	mov	%rdx,%rbp
	7	push	%rbx		7	push	%rbx
n	8	mov	(%rsi),%rdx	n	8	mov	(%rsi),%rdx
	9	mov	-0x18(%rdx),%r12		9	mov	-0x18(%rdx),%r12

```
std::string::append(unsigned long, char) (size: 179 -> 183 bytes (+4 bytes) )
```

f	1	push %r13		f	1	push %r13
	2	push %r12			2	push %r12
	3	push %rbp			3	push %rbp
n	4	mov %rsi,%rbp		n		
	5	push %rbx			4	push %rbx
n	6	mov %rdi,%rbx		n	5	mov %rdi,%rbp
	7	sub \$0x8,%rsp			6	sub \$0x8,%rsp
	8	test %rsi,%rsi			7	test %rsi,%rsi
n	9	je 72 <std::string::append(unsigned long, char)+0x72>		n	8	je 9d <std::string::append(unsigned long, char)+0x9d>
	10	mov (%rdi),%rax			9	mov (%rdi),%rdi
	11	movabs \$0xfffffffffffffff9,%rcx			10	movabs \$0xfffffffffffffff9,%rcx
n				n	11	mov %rsi,%rbx
	12	mov -0x18(%rax),%rdi			12	mov -0x18(%rdi),%rax
	13	sub %rdi,%rcx			13	sub %rax,%rcx
	14	cmp %rcx,%rsi			14	cmp %rcx,%rsi
n	15	ja a7 <std::string::append(unsigned long, char)+0xa7>		n	15	ja ab <std::string::append(unsigned long, char)+0xab>
	16	lea (%rsi,%rdi,1),%r12			16	lea (%rsi,%rax,1),%r12
	17	cmp -0x10(%rax),%r12			17	cmp -0x10(%rdi),%r12
	18	mov %edx,%r13d			18	mov %edx,%r13d
n	19	jbe 80 <std::string::append(unsigned long, char)+0x80>		n	19	jbe 60 <std::string::append(unsigned long, char)+0x60>
	20	mov %rbx,%rdi			20	mov %rbp,%rdi
	21	mov %r12,%rsi			21	mov %r12,%rsi
n	22	callq 46 <std::string::append(unsigned long, char)+0x46> 42: R_X86_64_PLT32 std::string::reserve(unsigned long)-0x4		n	22	callq 4a <std::string::append(unsigned long, char)+0x4a> 38: R_X86_64_PLT32 std::string::reserve(unsigned long)-0x4
	23	mov (%rbx),%rax			23	mov 0x0(%rbp),%rdi
	24	mov -0x18(%rax),%rdi			24	add -0x18(%rdi),%rdi
					25	cmp \$0x1,%rbx
					26	jne 70 <std::string::append(unsigned long, char)+0x70>
					27	mov %r13b,%rdi
					28	jmp 7c <std::string::append(unsigned long, char)+0x7c>
					29	nopl (%rax)
					30	mov -0x8(%rdi),%edx
					31	test %edx,%edx
					32	jg 3f <std::string::append(unsigned long, char)+0x3f>
	25	add %rax,%rdi			33	add %rax,%rdi
n	26	cmp \$0x1,%rbp		n	34	cmp \$0x1,%rsi
	27	je 90 <std::string::append(unsigned long, char)+0x90>			35	je 58 <std::string::append(unsigned long, char)+0x58>
	28	movsbl %r13b,%esi			36	movsbl %r13b,%esi
n	29	mov %rbp,%rdx		n	37	mov %rbx,%rdx
	30	callq 62 <std::string::append(unsigned long, char)+0x62> 5e: R_X86_64_PLT32 memset-0x4			38	callq 7c <std::string::append(unsigned long, char)+0x7c> 38: R_X86_64_PLT32 memset-0x4
	31	mov (%rbx),%rax			39	mov 0x0(%rbp),%rax
	32	lea -0x18(%rax),%rdx			40	lea -0x18(%rax),%rdx
n	33	cmp 0x0(%rip),%rdx # 70 <std::string::append(unsigned long, char)+0x70> 6c: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4		n	41	cmp 0x0(%rip),%rdx # 8b <std::string::append(unsigned long, char)+0x8b>
	34	jne 95 <std::string::append(unsigned long, char)+0x95>			42	je 9d <std::string::append(unsigned long, char)+0x9d>
					43	movl \$0x0,-0x8(%rax)
					44	mov %r12,-0x18(%rax)
					45	movb \$0x0,(%rax,%r12,1)
	35	add \$0x8,%rsp			46	add \$0x8,%rsp
n	36	mov %rbx,%rax		n	47	mov %rbp,%rax
	37	pop %rbx			48	pop %rbx
	38	pop %rbp			49	pop %rbp
	39	pop %r12			50	pop %r12
	40	pop %r13			51	pop %r13
	41	retq			52	retq
t	42	mov -0x8(%rax),%edx		t		
	43	test %edx,%edx				
	44	jg 3b <std::string::append(unsigned long, char)+0x3b>				
	45	jmp 4d <std::string::append(unsigned long, char)+0x4d>				
	46	nopl 0x0(%rax)				
	47	mov %r13b,%rdi				
	48	jmp 62 <std::string::append(unsigned long, char)+0x62>				
	49	movl \$0x0,-0x8(%rax)				
	50	mov %r12,-0x18(%rax)				
	51	movb \$0x0,(%rax,%r12,1)				
	52	jmp 72 <std::string::append(unsigned long, char)+0x72>				
	53	lea 0x0(%rip),%rdi # ae <std::string::append(unsigned long, char)+0xae> aa: R_X86_64_PC32 .LC9-0x4			53	lea 0x0(%rip),%rdi # b2 <std::string::append(unsigned long, char)+0xb2>
	54	callq b3 <.LC8+0x5> af: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4			54	callq b7 <.LC145+0x5> b3: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4
	55	Disassembly of section .text._ZN5s6resizeEmc:			55	Disassembly of section .text._ZN5s6resizeEmc:

std::string::assign(char const*) (size unchanged)

Old			New		
f	1	push %rbp	f	1	push %rbp
n			n	2	push %rbx
	2	mov %rdi,%rbp		3	mov %rdi,%rbp

3	mov %rsi,%rdi	4	mov %rsi,%rdi
n 4	push %rbx	n	
5	mov %rsi,%rbx	5	mov %rsi,%rbx
6	sub \$0x8,%rsp	6	sub \$0x8,%rsp
7	callq 14 <std::string::assign(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4	7	callq 14 <std::string::assign(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4
8	add \$0x8,%rsp	8	add \$0x8,%rsp
9	mov %rbx,%rsi	9	mov %rbx,%rsi
10	mov %rbp,%rdi	10	mov %rbp,%rdi
11	pop %rbx	11	pop %rbx
12	pop %rbp	12	pop %rbp
13	mov %rax,%rdx	13	mov %rax,%rdx
t 14	jmpq 28 <.LC2+0x4> 24: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4	t 14	jmpq 28 <.LC73+0x15> 24: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4
15	Disassembly of section .text._ZNSt16initializer_listIcE:	15	Disassembly of section .text._ZNSt16initializer_listIcE:

std::string::assign(char const*, unsigned long) (size: 232 -> 237 bytes (+5 bytes))

Old				
n			n 1	push %rbp
1	push %rbx		2	push %rbx
n 2	mov %rdi,%rbx		n 3	mov %rdi
3	mov %rdx,%r8		4	mov %rdx
4	movabs \$0x3fffffffffffff9,%rax		5	movabs \$0x3f
n 5	sub \$0x10,%rsp		n 6	sub \$0x8
6	mov (%rdi),%rdi		7	mov (%rdi)
n 7	cmp %rax,%r8		n 8	cmp %rax
8	mov -0x18(%rdi),%rdx		9	mov -0x18(%rdi)
n 9	ja ca <std::string::assign(char const*, unsigned long)+0xca>		n 10	ja e1 <std::string::assign(char const*, unsigned long)+0xe1>
10	cmp %rdi,%rsi		11	cmp %rsi
11	jb 40 <std::string::assign(char const*, unsigned long)+0x40>		12	ja 34 <std::string::assign(char const*, unsigned long)+0x34>
12	lea (%rdi,%rdx,1),%rax		13	lea (%rdi,%rdx,1),%rax
13	cmp %rax,%rsi		14	cmp %rax
n 14	ja 40 <std::string::assign(char const*, unsigned long)+0x40>		n	
15	mov -0x8(%rdi),%r9d			
16	test %r9d,%r9d			
17	jle 58 <std::string::assign(char const*, unsigned long)+0x58>		15	jbe 50 <std::string::assign(char const*, unsigned long)+0x50>
18	nopl 0x0(%rax)			
19	add \$0x10,%rsp		16	add \$0x8
			17	mov %rbx
20	mov %rbx,%rdi		18	mov %rbp
			19	pop %rbx
			20	pop %rbp
21	mov %rsi,%rcx		21	mov %rsi
n 22	pop %rbx		n	
23	xor %esi,%esi		22	xor %esi
n 24	jmpq 52 <std::string::assign(char const*, unsigned long)+0x52> 4e: R_X86_64_PLT32 std::string::M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)-0x4		n 23	jmpq 4a <std::string::M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)-0x4a>
25	nopw 0x0(%rax,%rax,1)		24	nopw 0x0(%rax,%rax,1)
n			n 25	mov -0x8
			26	test %eax
			27	jg 34 <std::string::assign(char const*, unsigned long)+0x34>
26	mov %rsi,%rax		28	mov %rsi
27	sub %rdi,%rax		29	sub %rdi
n 28	cmp %rax,%r8		n 30	cmp %rax
29	ja 98 <std::string::assign(char const*, unsigned long)+0x98>		31	ja 88 <std::string::assign(char const*, unsigned long)+0x88>
30	cmp \$0x1,%r8		32	cmp \$0x1
31	je c0 <std::string::assign(char const*, unsigned long)+0xc0>		33	je a8 <std::string::assign(char const*, unsigned long)+0xa8>
32	mov %r8,%rdx		34	test %rbx
33	mov %r8,0x8(%rsp)		35	jne b8 <std::string::assign(char const*, unsigned long)+0xb8>
34	callq 76 <std::string::assign(char const*, unsigned long)+0x76> 72: R_X86_64_PLT32 memcpy-0x4			
35	mov (%rbx),%rdi			
36	mov 0x8(%rsp),%r8			
37	lea -0x18(%rdi),%rax		36	lea -0x18(%rdi),%rax
n 38	cmp 0x0(%rip),%rax # 89 <std::string::assign(char const*, unsigned long)+0x89> 85: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4		n 37	cmp 0x0(%rip),%rax # 89 <std::string::assign(char const*, unsigned long)+0x89> 85: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4
39	jne d6 <std::string::assign(char const*, unsigned long)+0xd6>		38	jne d0 <std::string::assign(char const*, unsigned long)+0xd0>
40	add \$0x10,%rsp		39	add \$0x8
41	mov %rbx,%rax		40	mov %rbp
42	pop %rbx		41	pop %rbx
n			n 42	pop %rbp
43	retq		43	retq
44	nopl 0x0(%rax)		44	nopl 0x0(%rax)
45	test %rax,%rax		45	test %rax
n 46	je 7e <std::string::assign(char const*, unsigned long)+0x7e>		n 46	je 6d <std::string::assign(char const*, unsigned long)+0x6d>
47	cmp \$0x1,%r8		47	cmp \$0x1
48	je c0 <std::string::assign(char const*, unsigned long)+0xc0>		48	je a8 <std::string::assign(char const*, unsigned long)+0xa8>
49	mov %r8,%rdx		49	mov %rbx

50	mov %r8,0x0(%rsp)				
51	callq b0 <std::string::assign(char const*, unsigned long)+0xb0> ac: R_X86_64_PLT32 memmove-0x4		50	callq 9b <std::string::assign(char const*, unsigned long)+0xb0>	
52	mov (%rbx),%rdi		51	mov 0x0(%rip),%r8	
53	mov 0x8(%rsp),%r8				
54	jmp 7e <std::string::assign(char const*, unsigned long)+0x7e>		52	jmp 6d <std::string::assign(char const*, unsigned long)+0x7e>	
55	nopw 0x0(%rax,%rax,1)		53	nopl 0x0(%rip),%r8	
56	movzbl (%rsi),%eax		54	movzbl (%rsi),%eax	
57	mov %al,(%rdi)		55	mov %al,(%rdi)	
n 58	mov (%rbx),%rdi		n 56	mov 0x0(%rip),%r8	
59	jmp 7e <std::string::assign(char const*, unsigned long)+0x7e>		57	jmp 6d <std::string::assign(char const*, unsigned long)+0x7e>	
60	lea 0x0(%rip),%rdi # d1 <std::string::assign(char const*, unsigned long)+0xd1> cd: R_X86_64_PC32 .LC8-0x4		58	nopl 0x0(%rip),%r8	
			59	mov %rbx,%r8	
61	callq d6 <std::string::assign(char const*, unsigned long)+0xd6> d2: R_X86_64_PLT32 std::_throw_length_error(char const*)-0x4		60	callq c0 <std::string::assign(char const*, unsigned long)+0xc0>	
			61	mov 0x0(%rip),%r8	
			62	jmp 6d <std::string::assign(char const*, unsigned long)+0x7e>	
			63	nopw %cs:(%rax,%rax,1)	
62	movl \$0x0,-0x8(%rdi)		64	movl \$0x0,-0x8(%rdi)	
t 63	mov %r8,-0x18(%rdi)		t 65	mov %rbx,%r8	
64	movb \$0x0,(%rdi,%r8,1)		66	movb \$0x0,(%rdi,%r8,1)	
65	jmp 8b <std::string::assign(char const*, unsigned long)+0x8b>		67	jmp 7a <std::string::assign(char const*, unsigned long)+0x7a>	
			68	lea 0x0(%rip),%r8	
			69	callq ed <std::string::assign(char const*, unsigned long)+0xed>	
66	Disassembly of section .text._ZN5saSEt16initializer_listIcE:		70	Disassembly of section .text._ZN5saSEt16initializer_listIcE:	

std::string::assign(std::initializer_list<char>) (size unchanged)

	Old		
f 1	jmpq 5 <std::string::assign(std::initializer_list<char>)+0x5> 1: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4	f 1	jmpq 5 <std::string::assign(std::initializer_list<char>)+0x5> 1: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4
t 2	Disassembly of section .text._ZN5saSEPKc:	t 2	Disassembly of section .text._ZN5saSEt16initializer_listIcE:

std::string::assign(std::string const&) (size: 159 -> 299 bytes (+140 bytes))

	Old		
n 1	push %rbp	n	
2	push %rbx	1	push %rbx
3	mov %rdi,%rbx	2	mov %rdi,%rbx
n 4	sub \$0x18,%rsp	n 3	sub \$0x20,%rsp
5	mov (%rdi),%rax	4	mov (%rdi),%r8
6	lea -0x18(%rax),%rdx	5	mov %fs:0x28,%rax
		6	mov %rax,0x18(%rsp)
		7	xor %eax,%eax
7	mov (%rsi),%rax	8	mov (%rsi),%rax
n		n 9	lea -0x18(%r8),%rdx
8	lea -0x18(%rax),%rdi	10	lea -0x18(%rax),%rdi
9	cmp %rdi,%rdx	11	cmp %rdi,%rdx
n 10	je 3d <std::string::assign(std::string const&)+0x3d>	n 12	je 4c <std::string::assign(std::string const&)+0x4c>
11	mov -0x8(%rax),%ecx	13	mov -0x8(%rax),%r9d
12	test %ecx,%ecx	14	test %r9d,%r9d
13	js 50 <std::string::assign(std::string const&)+0x50>	15	js 70 <std::string::assign(std::string const&)+0x70>
14	mov 0x0(%rip),%rcx # 2a <std::string::assign(std::string const&)+0x2a> 26: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	16	mov 0x0(%rip),%rsi # 3b <std::string::assign(std::string const&)+0x3b>
15	cmp %rcx,%rdi	17	cmp %rsi,%rdi
16	jne 7b <std::string::assign(std::string const&)+0x7b>	18	jne d0 <std::string::assign(std::string const&)+0xd0>
17	mov %rdx,%rdi		
18	mov %rax,%rbp		
19	cmp %rcx,%rdi	19	cmp %rsi,%rdx
20	jne 6f <std::string::assign(std::string const&)+0x6f>	20	jne 90 <std::string::assign(std::string const&)+0x90>
21	mov %rbp,(%rbx)	21	mov %rax,(%rbx)
22	add \$0x18,%rsp	22	mov 0x18(%rsp),%rcx
		23	xor %fs:0x28,%rcx
23	mov %rbx,%rax	24	mov %rbx,%rax
n		n 25	jne 126 <std::string::assign(std::string const&)+0x126>
		26	add \$0x20,%rsp
24	pop %rbx	27	pop %rbx
n 25	pop %rbp	n	
26	retq	28	retq
n		n 29	nopl 0x0(%rax)
		30	lea 0x17(%rsp),%rsi
		31	xor %edx,%edx
		32	callq 7c <std::string::assign(std::string const&)+0x7c>
		33	mov (%rbx),%r8
		34	mov 0x0(%rip),%rsi # 86 <std::string::assign(std::string const&)+0x86>
		35	lea -0x18(%r8),%rdx
		36	jmp 44 <std::string::assign(std::string const&)+0x44>
		37	nopl 0x0(%rax)
		38	cmpq \$0x0,0x0(%rip) # 98 <std::string::assign(std::string const&)+0x98>
		39	je f8 <std::string::assign(std::string const&)+0xf8>

				40	lea 0x10(%rdx),%rsi
				41	mov %rdx,%rcx
				42	mov \$0xffffffff,%edx
				43	lock xadd %edx,(%rsi)
				44	test %edx,%edx
				45	jg 49 <std::string::assign(std::
				46	lea 0x17(%rsp),%rsi
				47	mov %rcx,%rdi
				48	mov %rax,0x8(%rsp)
				49	callq c0 <std::string::assign(std::
				50	mov 0x8(%rsp),%rax
				51	jmp 49 <std::string::assign(std::
	27	nopw 0x0(%rax,%rax,1)		52	nopw 0x0(%rax,%rax,1)
n	28	lea 0xf(%rsp),%rsi	n	53	cmpq \$0x0,0x0(%rip) # d8 <
	29	xor %edx,%edx		54	je 110 <std::string::assign(std
	30	callq 5c <std::string::assign(std::string const&+0x5c> 58: R_X86_64_PLT32 std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)-0x4		55	lock addl \$0x1,-0x8(%rax)
	31	mov (%rbx),%rcx		56	mov (%rbx),%rcx
t	32	mov %rax,%rbp	t	57	sub \$0x18,%rcx
	33	lea -0x18(%rcx),%rdi		58	cmp %rsi,%rcx
	34	mov 0x0(%rip),%rcx # 6d <std::string::assign(std::string const&+0x6d> 69: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4			
	35	jmp 35 <std::string::assign(std::string const&+0x35>			
	36	lea 0xf(%rsp),%rsi			
	37	callq 79 <std::string::assign(std::string const&+0x79> 75: R_X86_64_PC32 .text.unlikely.__ZN5s4_Rep10_M_disposeERKSaIcE.part.12-0x4			
	38	jmp 3a <std::string::assign(std::string const&+0x3a>			
	39	add \$0x10,%rdi			
	40	cmpq \$0x0,0x0(%rip) # 87 <std::string::assign(std::string const&+0x87> 82: R_X86_64_GOTPCREL __pthread_key_create-0x5			
	41	je 96 <std::string::assign(std::string const&+0x96>	59	je 49 <std::string::assign(std:	
	42	lock addl \$0x1,(%rdi)			
	43	mov (%rbx),%rdx			
	44	lea -0x18(%rdx),%rdi	60	lea 0x10(%rcx),%rsi	
	45	jmp 32 <std::string::assign(std::string const&+0x32>	61	jmp a1 <std::string::assign(std:	
	46	addl \$0x1,-0x8(%rax)	62	nopl (%rax)	
	47	mov %rdx,%rdi	63	mov %rdx,%rcx	
			64	mov -0x8(%r8),%edx	
			65	lea -0x1(%rdx),%esi	
			66	mov %esi,-0x8(%r8)	
48	jmp 32 <std::string::assign(std::string const&+0x32>		67	jmp aa <std::string::assign(std:	
			68	nopl 0x0(%rax,%rax,1)	
			69	add \$0x1,%r9d	
			70	cmp %rsi,%rdx	
			71	mov %rdx,%rcx	
			72	mov %r9d,-0x8(%rax)	
			73	je 49 <std::string::assign(std:	
			74	jmp fb <std::string::assign(std:	
			75	callq 12b <.LC198+0xa> 127: R_X86_64	
49	Disassembly of section .text._ZN5saSERK5s:			76	Disassembly of section .text._ZN5sa

std::string::assign(std::string const&, unsigned long, unsigned long) (size: 49 -> 67 bytes (+18 bytes))

Old					
f	1	mov (%rsi),%rsi	f	1	mov (%rsi),%rsi
	2	mov %rdx,%rax		2	mov %rdx,%rax
n	3	mov -0x18(%rsi),%r8	n	3	mov -0x18(%rsi),%r9
				4	mov %r9,%r8
				5	sub %rdx,%r8
	4	mov %r8,%rdx		6	mov %rcx,%rdx
	5	sub %rax,%rdx			
	6	cmp %rcx,%rdx		7	cmp %rcx,%r8
	7	cmova %rcx,%rdx		8	cmovbe %r8,%rdx
	8	cmp %r8,%rax		9	cmp %r9,%rax
	9	ja 24 <std::string::assign(std::string const&, unsigned long, unsigned long)+0x24>		10	ja 27 <std::string::assign
	10	add %rax,%rsi		11	add %rax,%rsi
t	11	jmpq 24 <std::string::assign(std::string const&, unsigned long, unsigned long)+0x24> 20: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4	t	12	jmpq 27 <std::string::assign
	12	lea 0x0(%rip),%rdi # 2b <std::string::assign(std::string const&, unsigned long, unsigned long)+0x2b> 27: R_X86_64_PC32 .LC8-0x4		13	lea 0x0(%rip),%rsi
	13	push %rax		14	push %rdx
				15	lea 0x0(%rip),%rdi
				16	mov %rax,%rdx
				17	mov %r9,%rcx
				18	xor %eax,%eax
	14	callq 31 <.LC2+0xd> 2d: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4		19	callq 43 <.LC110+0x2> 3f: R
	15	Disassembly of section .text._ZN5s6assignEPKc:		20	Disassembly of section .text.

std::string::assign(std::string&&) (size unchanged)

Old			New		
f	1	push %rbx	f	1	push %rbx

2	mov %rdi,%rbx	2	mov %rdi,%rbx
3	callq 9 <std::string::assign(std::string&&)+0x9> 5: R_X86_64_PLT32 std::string::swap(std::string&)-0x4	3	callq 9 <std::string::assign(std::string&&)+0x9> 5: R_X86_64_PLT32 std::string::swap
4	mov %rbx,%rax	4	mov %rbx,%rax
5	pop %rbx	5	pop %rbx
6	retq	6	retq
t 7	Disassembly of section .text._ZNK5s13get_allocatorEv:	t 7	Disassembly of section .text._ZNK5s5c_strEv:

std::string::assign(unsigned long, char) (size unchanged)

Old		New	
f 1	mov (%rdi),%rax	f 1	mov (%rdi),%rax
2	mov %rsi,%rcx	2	mov %rsi,%rcx
3	movsbl %dl,%r8d	3	movsbl %dl,%r8d
4	xor %esi,%esi	4	xor %esi,%esi
5	mov -0x18(%rax),%rax	5	mov -0x18(%rax),%rax
6	mov %rax,%rdx	6	mov %rax,%rdx
t 7	jmpq 18 <.LC1+0x7> 14: R_X86_64_PLT32 std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4	t 7	jmpq 18 <.LC73+0x5> 14: R_X86_64_PLT32 std::string::_M_replac
8	Disassembly of section .text._ZNSaSEc:	8	Disassembly of section .text._ZNSaSEc:

std::string::at(unsigned long) (size: 58 -> 64 bytes (+6 bytes))

Old		New	
f 1	push %rbp	f 1	push %rbp
n 2	mov %rsi,%rbp	n	
3	push %rbx	2	push %rbx
n 4	mov %rdi,%rbx	n 3	mov %rsi,%rbx
5	sub \$0x8,%rsp	4	sub \$0x8,%rsp
6	mov (%rdi),%rax	5	mov (%rdi),%rax
n 7	cmp -0x18(%rax),%rsi	n 6	mov -0x18(%rax),%rdx
		7	cmp %rdx,%rsi
8	jae 2e <std::string::at(unsigned long)+0x2e>	8	jae 32 <std::string::at(unsigned long)+0x32>
9	mov -0x8(%rax),%edx	9	mov -0x8(%rax),%edx
10	test %edx,%edx	10	test %edx,%edx
n 11	js 24 <std::string::at(unsigned long)+0x24>	n 11	js 28 <std::string::at(unsigned long)+0x28>
		12	mov %rdi,%rbp
12	callq 21 <std::string::at(unsigned long)+0x21> 1d: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4	13	callq 24 <std::string::at(unsigned long)+0x24> 20: R_X86_64_PLT32 std::string::_M_lea
13	mov (%rbx),%rax	14	mov 0x0(%rbp),%rax
14	add \$0x8,%rsp	15	add \$0x8,%rsp
n 15	add %rbp,%rax	n 16	add %rbx,%rax
16	pop %rbx	17	pop %rbx
17	pop %rbp	18	pop %rbp
18	retq	19	retq
t 19	lea 0x0(%rip),%rdi # 35 <std::string::at(unsigned long)+0x35> 31: R_X86_64_PC32 .LC0-0x4	t 20	lea 0x0(%rip),%rdi # 39 <std::string::at(unsigned long)+0x39> 35: R_X86_64_PC3
		21	xor %eax,%eax
20	callq 3a <.LC3> 36: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	22	callq 40 <.LC36+0x8> 3c: R_X86_64_PLT32 std::__throw_out_of_range_fmt(char const*,
21	Disassembly of section .text._ZN5s5rbeginEv:	23	Disassembly of section .text._ZN5s5clearEv:

std::string::at(unsigned long) const (size: 26 -> 31 bytes (+5 bytes))

Old		New	
f 1	mov (%rdi),%rax	f 1	mov (%rdi),%rax
n 2	cmp -0x18(%rax),%rsi	n 2	mov -0x18(%rax),%rdx
		3	cmp %rdx,%rsi
3	jae d <std::string::at(unsigned long) const+0xd>	4	jae 10 <std::string::at(unsigned long) const+0x10>
4	add %rsi,%rax	5	add %rsi,%rax
5	retq	6	retq
n 6	lea 0x0(%rip),%rdi # 14 <std::string::at(unsigned long) const+0x14> 10: R_X86_64_PC32 .LC0-0x4	n 7	lea 0x0(%rip),%rdi # 17 <std::string::at(unsigned long) const+0x17> 13: R_X
7	push %rax	8	push %rax
t		t 9	xor %eax,%eax
8	callq 1a <.LC1+0x9> 16: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	10	callq 1f <.LC73+0xc> 1b: R_X86_64_PLT32 std::__throw_out_of_range_fmt(char const
9	Disassembly of section .text._ZNK5s5frontEv:	11	Disassembly of section .text._ZNK5s5frontEv:

std::string::back() (size: 45 -> 46 bytes (+1 bytes))

Old		New	
f 1	push %rbp	f 1	push %rbp
2	push %rbx	2	push %rbx
n 3	mov %rdi,%rbx	n	
4	sub \$0x8,%rsp	3	sub \$0x8,%rsp
5	mov (%rdi),%rax	4	mov (%rdi),%rax
6	mov -0x18(%rax),%rdx	5	mov -0x18(%rax),%rdx
n 7	lea -0x1(%rdx),%rbp	n 6	lea -0x1(%rdx),%rbp
8	mov -0x8(%rax),%edx	7	mov -0x8(%rax),%edx
9	test %edx,%edx	8	test %edx,%edx
n 10	js 23 <std::string::back()+0x23>	n 9	js 24 <std::string::back()+0x24>
		10	mov %rdi,%rbp
11	callq 20 <std::string::back()+0x20> 1c: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4	11	callq 20 <std::string::back()+0x20> 1c: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4
n 12	mov (%rbx),%rax	n 12	mov 0x0(%rbp),%rax

	13	add	\$0x8,%rsp		13	add	\$0x8,%rsp
t	14	add	%rbp,%rax	t	14	add	%rbx,%rax
	15	pop	%rbx		15	pop	%rbx
	16	pop	%rbp		16	pop	%rbp
	17	retq			17	retq	
	18	Disassembly of section .text._ZN5s2atEm:					
	18	Disassembly of section .text._ZN5s2atEm:					

std::string::begin() (size: 24 -> 25 bytes (+1 bytes))

Old				New			
f	1	mov	(%rdi),%rax	f	1	mov	(%rdi),%rax
n				n	2	mov	-0x8(%rax),%edx
					3	test	%edx,%edx
					4	js	17 <std::string::begin()+0x17>
	2	push	%rbx		5	push	%rbx
	3	mov	%rdi,%rbx		6	mov	%rdi,%rbx
n	4	mov	-0x8(%rax),%edx	n			
	5	test	%edx,%edx				
	6	js	16 <std::string::begin()+0x16>				
	7	callq	13 <std::string::begin()+0x13> f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4	7	callq	13 <std::string::begin()+0x13> f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4	
	8	mov	(%rbx),%rax	8	mov	(%rbx),%rax	
	9	pop	%rbx	9	pop	%rbx	
t	10	retq		t	10	repz retq	
	11	Disassembly of section .text._ZN5s4rendEv:		11	Disassembly of section .text._ZN5s4rendEv:		

std::string::c_str() const (size unchanged)

Old			New		
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
	2	retq		2	retq
t	3	Disassembly of section .text._ZN5s4rendEv:	t	3	Disassembly of section .text._ZN5s4rendEv:

std::string::compare(char const*) const (size: 114 -> 123 bytes (+9 bytes))

Old				New			
f	1	push %r13		f	1	push %r13	
n				n	2	push %r12	
	2	mov %rsi,%r13			3	mov %rsi,%r13	
n	3	push %r12		n			
	4	push %rbp			4	push %rbp	
	5	push %rbx			5	push %rbx	
	6	sub \$0x8,%rsp			6	sub \$0x8,%rsp	
	7	mov (%rdi),%r12			7	mov (%rdi),%r12	
	8	mov %rsi,%rdi			8	mov %rsi,%rdi	
	9	callq 18 <std::string::compare(char const*) const+0x18> 14: R_X86_64_PLT32 strlen-0x4			9	callq 18 <std::string::compare(char const*) const+0x18> 14: R_X86_64_PLT32 strlen-0x4	
n	10	mov %r13,%rsi		n			
	11	mov %rax,%rbp			10	mov %rax,%rbp	
	12	mov -0x18(%r12),%rbx			11	mov -0x18(%r12),%rbx	
n	13	mov %r12,%rdi		n			
	14	cmp %rbx,%rax			12	cmp %rbx,%rax	
	15	mov %rbx,%rdx			13	mov %rbx,%rdx	
	16	cmovbe %rax,%rdx			14	cmovbe %rax,%rdx	
n				n	15	test %rdx,%rdx	
					16	je 50 <std::string::compare(char const*) const+0x50>	
					17	mov %r13,%rsi	
					18	mov %r12,%rdi	
	17	callq 35 <std::string::compare(char const*) const+0x35> 31: R_X86_64_PLT32 memcmp-0x4			19	callq 3a <std::string::compare(char const*) const+0x3a> 36: R_X86_64_PLT32 memcmp-0x4	
	18	test %eax,%eax			20	test %eax,%eax	
n	19	jne 4a <std::string::compare(char const*) const+0x4a>		n			
	20	sub %rbp,%rbx					
	21	mov \$0x7fffffff,%eax					
	22	cmp \$0x7fffffff,%rbx					
	23	jle 58 <std::string::compare(char const*) const+0x58>			21	je 50 <std::string::compare(char const*) const+0x50>	
	24	add \$0x8,%rsp			22	add \$0x8,%rsp	
	25	pop %rbx			23	pop %rbx	
	26	pop %rbp			24	pop %rbp	
	27	pop %r12			25	pop %r12	
	28	pop %r13			26	pop %r13	
	29	retq			27	retq	
t	30	nopl (%rax)		t	28	nopl 0x0(%rax)	
					29	sub %rbp,%rbx	
					30	mov \$0x7fffffff,%eax	
					31	cmp \$0x7fffffff,%rbx	
					32	jg 3e <std::string::compare(char const*) const+0x3e>	
	31	cmp \$0xffffffff80000000,%rbx			33	cmp \$0xffffffff80000000,%rbx	
	32	mov \$0x80000000,%eax			34	mov \$0x80000000,%eax	

33	cmovge %ebx,%eax	35	cmovge %ebx,%eax
34	add \$0x8,%rsp	36	add \$0x8,%rsp
35	pop %rbx	37	pop %rbx
36	pop %rbp	38	pop %rbp
37	pop %r12	39	pop %r12
38	pop %r13	40	pop %r13
39	retq	41	retq
40	Disassembly of section .text._ZNKSs7compareEmmPKc:	42	Disassembly of section .text._ZNKSs7compareEmmPKc:

std::string::compare(std::string const&) const (size: 86 -> 95 bytes (+9 bytes))

		Old			New
f	1	push %rbp	f	1	push %rbp
	2	push %rbx		2	push %rbx
	3	sub \$0x8,%rsp		3	sub \$0x8,%rsp
	4	mov (%rdi),%rdi		4	mov (%rdi),%rdi
	5	mov (%rsi),%rsi		5	mov (%rsi),%rsi
n	6	mov -0x18(%rdi),%rbp	n	6	mov -0x18(%rdi),%rbx
	7	mov -0x18(%rsi),%rbx		7	mov -0x18(%rsi),%rbp
	8	mov %rbp,%rdx		8	mov %rbx,%rdx
	9	cmp %rbp,%rbx		9	cmp %rbx,%rbp
	10	cmovbe %rbx,%rdx		10	cmovbe %rbp,%rdx
				11	test %rdx,%rdx
				12	je 38 <std::string::compare(std::string const&) const+0x38>
	11	callq 23 <std::string::compare(std::string const&) const+0x23> 1f: R_X86_64_PLT32 memcmp-0x4		13	callq 28 <std::string::compare(std::string const&) const+0x28> 24: R_X86_64_PLT32 memcr
	12	test %eax,%eax		14	test %eax,%eax
n	13	jne 38 <std::string::compare(std::string const&) const+0x38>	n	15	je 38 <std::string::compare(std::string const&) const+0x38>
	14	sub %rbx,%rbp			
	15	mov \$0xffffffff,%eax			
	16	cmp \$0xffffffff,%rbp			
	17	jle 40 <std::string::compare(std::string const&) const+0x40>			
	18	add \$0x8,%rsp		16	add \$0x8,%rsp
	19	pop %rbx		17	pop %rbx
	20	pop %rbp		18	pop %rbp
	21	retq		19	retq
n	22	nop	n	20	nopl 0x0(%rax,%rax,1)
				21	sub %rbp,%rbx
				22	mov \$0xffffffff,%eax
				23	cmp \$0xffffffff,%rbx
				24	jg 2c <std::string::compare(std::string const&) const+0x2c>
	23	cmp \$0xffffffff80000000,%rbp		25	cmp \$0xffffffff80000000,%rbx
	24	mov \$0x80000000,%eax		26	mov \$0x80000000,%eax
t	25	cmovge %ebp,%eax	t	27	cmovge %ebx,%eax
	26	add \$0x8,%rsp		28	add \$0x8,%rsp
	27	pop %rbx		29	pop %rbx
	28	pop %rbp		30	pop %rbp
	29	retq		31	retq
	30	Disassembly of section .text._ZNKSs7compareEmmRKs:		32	Disassembly of section .text._ZNKSs7compareEmmRKs:

std::string::compare(unsigned long, unsigned long, char const*) const (size: 140 -> 176 bytes (+36 bytes))

		Old			
f	1	push %r14	f	1	push %r14
	2	push %r13		2	push %r13
n			n	3	mov %rcx,%r14
	3	push %r12		4	push %r12
	4	push %rbp		5	push %rbp
n			n	6	mov %rsi,%rbp
	5	push %rbx		7	push %rbx
	6	mov (%rdi),%r13		8	mov (%rdi),%r13
n	7	mov %rsi,%rbx	n		
	8	mov -0x18(%r13),%rbp		9	mov -0x18(%r13),%rcx
	9	cmp %rbp,%rsi		10	cmp %rcx,%rsi
	10	ja 80 <std::string::compare(unsigned long, unsigned long, char const*) const+0x80>		11	ja 98 <std::string::compare(unsigned long, unsi
	11	sub %rsi,%rbp		12	sub %rsi,%rcx
	12	mov %rcx,%rdi		13	mov %r14,%rdi
	13	mov %rcx,%r14			
	14	cmp %rdx,%rbp		14	cmp %rdx,%rcx
				15	mov %rcx,%rbx
	15	cmova %rdx,%rbp		16	cmova %rdx,%rbx
	16	callq 2c <std::string::compare(unsigned long, unsigned long, char const*) const+0x2c> 28: R_X86_64_PLT32 strlen-0x4		17	callq 2f <std::string::compare(unsigned long, unsi
	17	mov %rbp,%rdx		18	mov %rbx,%rdx
	18	cmp %rbp,%rax		19	cmp %rbx,%rax
	19	lea 0x0(%r13,%rbx,1),%rdi		20	mov %rax,%r12
	20	cmovbe %rax,%rdx		21	cmovbe %rax,%rdx
n	21	mov %r14,%rsi	n		

22	mov %rax,%r12			
23	callq 46 <std::string::compare(unsigned long, unsigned long, char const*) const+0x46> 42: R_X86_64_PLT32 memcmp-0x4			
24	test %eax,%eax	22	test %rdx,%rdx	
25	jne 5b <std::string::compare(unsigned long, unsigned long, char const*) const+0x5b>	23	jne 60 <std::string::compare(unsigned long, unsigned long, char const*) const+0x60>	
26	sub %r12,%rbp	24	sub %r12,%rbx	
27	mov \$0x7fffffff,%eax	25	mov \$0x7fffffff,%eax	
n 28	cmp \$0x7fffffff,%rbp	n 26	cmp \$0x7fffffff,%rbx	
29	jle 68 <std::string::compare(unsigned long, unsigned long, char const*) const+0x68>	27	jle 80 <std::string::compare(unsigned long, unsigned long, char const*) const+0x80>	
30	pop %rbx	28	pop %rbx	
31	pop %rbp	29	pop %rbp	
32	pop %r12	30	pop %r12	
33	pop %r13	31	pop %r13	
34	pop %r14	32	pop %r14	
35	retq	33	retq	
n 36	nopl 0x0(%rax)	n 34	nopl 0x0(%rax,%rax,1)	
		35	lea 0x0(%r13,%rbp,1),%rdi	
		36	mov %r14,%rsi	
		37	callq 6d <std::string::compare(unsigned long, unsigned long, char const*) const+0x6d> 62: R_X86_64_PLT32 memcmp-0x4	
		38	test %eax,%eax	
		39	je 41 <std::string::compare(unsigned long, unsigned long, char const*) const+0x41>	
37	pop %rbx	40	pop %rbx	
n 38	cmp \$0xffffffff80000000,%rbp	n		
39	mov \$0x80000000,%eax			
40	cmovge %ebp,%eax			
41	pop %rbp	41	pop %rbp	
42	pop %r12	42	pop %r12	
43	pop %r13	43	pop %r13	
44	pop %r14	44	pop %r14	
45	retq	45	retq	
t		t 46	nopw 0x0(%rax,%rax,1)	
		47	cmp \$0xffffffff80000000,%rbx	
		48	mov \$0x80000000,%eax	
		49	cmovge %ebx,%eax	
		50	pop %rbx	
		51	pop %rbp	
		52	pop %r12	
		53	pop %r13	
		54	pop %r14	
		55	retq	
		56	mov %rsi,%rdx	
46	lea 0x0(%rip),%rdi # 87 <std::string::compare(unsigned long, unsigned long, char const*) const+0x87> 83: R_X86_64_PC32 .LC2-0x4	57	lea 0x0(%rip),%rdi # a2 <std::string::compare(unsigned long, unsigned long, char const*) const+0xa2> 99: R_X86_64_PC32 .LC3-0x4	
		58	lea 0x0(%rip),%rsi # a9 <std::string::compare(unsigned long, unsigned long, char const*) const+0xa9>	
		59	xor %eax,%eax	
47	callq 8c <.LC6+0x9> 88: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	60	callq b0 <.LC199> ac: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	
48	Disassembly of section .text._ZNKSs7compareEmmPKcm:	61	Disassembly of section .text._ZNKSs7compareEmmPKcm:	

std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const (size: 114 -> 160 bytes (+46 bytes))

Old			
n 1	push %rbp	n	
2	push %rbx	1	push %rbx
n 3	sub \$0x8,%rsp	n 2	sub \$0x10,%rsp
4	mov (%rdi),%rdi	3	mov (%rdi),%rdi
n 5	mov -0x18(%rdi),%rbp	n 4	mov -0x18(%rdi),%rbx
6	cmp %rbp,%rsi	5	cmp %rbx,%rsi
7	ja 66 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x66>	6	ja 85 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x85>
8	sub %rsi,%rbp	7	sub %rsi,%rbx
9	mov %r8,%rbx		
10	cmp %rdx,%rbp	8	cmp %rdx,%rbx
11	cmova %rdx,%rbp	9	cmova %rdx,%rbx
		10	cmp %rbx,%r8
12	mov %r8,%rdx	11	mov %rbx,%rdx
13	cmp %r8,%rbp		
14	cmovbe %rbp,%rdx	12	cmovbe %r8,%rdx
		13	test %rdx,%rdx
		14	jne 48 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x48>
		15	sub %r8,%rbx
		16	mov \$0x7fffffff,%eax
		17	cmp \$0x7fffffff,%rbx
		18	jle 70 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x70>
		19	add \$0x10,%rsp
		20	pop %rbx
		21	retq
		22	nopl 0x0(%rax)

	15	add	%rsi,%rdi		23	add	%rsi,%rdi
	16	mov	%rcx,%rsi		24	mov	%rcx,%rsi
n				n	25	mov	%r8,0x8(%rsp)
	17	callq	34 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x34> 30: R_X86_64_PLT32 memcmp-0x4		26	callq	58 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x58> 30: R_X86_64_PLT32 memcmp-0x4
	18	test	%eax,%eax		27	test	%eax,%eax
n				n	28	mov	0x8(%rsp),%r8
	19	jne	49 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x49>		29	je	2a <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x2a>
	20	sub	%rbx,%rbp				
	21	mov	\$0x7fffffff,%eax				
	22	cmp	\$0x7fffffff,%rbp				
	23	jle	50 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x50>				
	24	add	\$0x8,%rsp		30	add	\$0x10,%rsp
	25	pop	%rbx		31	pop	%rbx
n	26	pop	%rbp	n			
	27	retq			32	retq	
n				n	33	nopw	0x0(%rax,%rax,1)
	28	cmp	\$0xffffffff80000000,%rbp		34	cmp	\$0xffffffff80000000,%rbx
	29	mov	\$0x80000000,%eax		35	mov	\$0x80000000,%eax
n	30	cmovge	%ebp,%eax	n	36	cmovge	%ebx,%eax
	31	add	\$0x8,%rsp		37	add	\$0x10,%rsp
	32	pop	%rbx		38	pop	%rbx
n	33	pop	%rbp	n			
	34	retq			39	retq	
t				t	40	mov	%rsi,%rdx
					41	lea	0x0(%rip),%rdi # 8f <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x8f> 69: R_X86_64_PC32 .LC2-0x4
	35	lea	0x0(%rip),%rdi # 6d <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x6d> 69: R_X86_64_PC32 .LC2-0x4		42	lea	0x0(%rip),%rsi # 96 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x96> 69: R_X86_64_PC32 .LC2-0x4
					43	mov	%rbx,%rcx
					44	xor	%eax,%eax
	36	callq	72 <.LC5+0xc> 6e: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4		45	callq	a0 <.LC126+0x3> 9c: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4
	37	Disassembly of section .text._ZNSt12_Alloc_hiderC2EPcRKSAIcE:			46	Disassembly of section .text._ZNSt12_Alloc_hiderC2EPcRKSAIcE:	

std::string::compare(unsigned long, unsigned long, std::string const&) const (size: 122 -> 161 bytes (+39 bytes))

Old								
f	1	push	%rbp		f	1	push	%rbp
n	2	mov	%rsi,%rax		n			
	3	push	%rbx		2	push	%rbx	
	4	sub	\$0x8,%rsp		3	sub	\$0x8,%rsp	
	5	mov	(%rdi),%rdi		4	mov	(%rdi),%rdi	
n	6	mov	-0x18(%rdi),%rbp		n	5	mov	-0x18(%rdi),%rbx
	7	cmp	%rbp,%rsi		6	cmp	%rbx,%rsi	
	8	ja	6e <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x6e>		7	ja	86 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x86>	
					8	mov	(%rcx),%rax	
	9	sub	%rsi,%rbp		9	sub	%rsi,%rbx	
	10	mov	(%rcx),%rsi					
	11	cmp	%rdx,%rbp		10	cmp	%rdx,%rbx	
	12	cmova	%rdx,%rbp		11	cmova	%rdx,%rbx	
	13	mov	-0x18(%rsi),%rbx		12	mov	-0x18(%rax),%rbp	
					13	mov	%rbx,%rdx	
	14	cmp	%rbx,%rbp		14	cmp	%rbx,%rbp	
n	15	mov	%rbx,%rdx		n			
	16	cmovbe	%rbp,%rdx		15	cmovbe	%rbp,%rdx	
n	17	add	%rax,%rdi		n			
	18	callq	38 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x38> 34: R_X86_64_PLT32 memcmp-0x4					
	19	test	%eax,%eax		16	test	%rdx,%rdx	
	20	jne	4d <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x4d>		17	jne	50 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x50>	
	21	sub	%rbx,%rbp		18	sub	%rbp,%rbx	
	22	mov	\$0x7fffffff,%eax		19	mov	\$0x7fffffff,%eax	
n	23	cmp	\$0x7fffffff,%rbp		n	20	cmp	\$0x7fffffff,%rbx
	24	jle	58 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x58>		21	jle	70 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x70>	
	25	add	\$0x8,%rsp		22	add	\$0x8,%rsp	
	26	pop	%rbx		23	pop	%rbx	
	27	pop	%rbp		24	pop	%rbp	
	28	retq			25	retq		
n	29	nopl	0x0(%rax)		n	26	nopw 0x0(%rax,%rax,1)	
	30	cmp	\$0xffffffff80000000,%rbp		27	add	%rsi,%rdi	
	31	mov	\$0x80000000,%eax		28	mov	%rax,%rsi	
	32	cmovge	%ebp,%eax		29	callq	5b <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x5b> 30: R_X86_64_PLT32 memcmp-0x4	
					30	test	%eax,%eax	
					31	je	32 <std::string::compare(unsigned long, unsigned long, char const*, unsigned long) const+0x32> 30: R_X86_64_PLT32 memcmp-0x4	
	33	add	\$0x8,%rsp		32	add	\$0x8,%rsp	
	34	pop	%rbx		33	pop	%rbx	
	35	pop	%rbp		34	pop	%rbp	
	36	retq			35	retq		
t					t	36	nopw %cs:0x0(%rax,%rax,1)	

				37	cmp	\$0xffffffff80000000,%rbx
				38	mov	\$0x80000000,%eax
				39	cmovge	%ebx,%eax
				40	add	\$0x8,%rsp
				41	pop	%rbx
				42	pop	%rbp
				43	retq	
				44	mov	%rsi,%rdx
				45	lea	0x0(%rip),%rdi # 90 <std::st
37	lea	0x0(%rip),%rdi # 75 <std::string::compare(unsigned long, unsigned long, std::string const&) const+0x75>	71: R_X86_64_PC32 .LC2-0x4	46	lea	0x0(%rip),%rsi # 97 <std::st
				47	mov	%rbx,%rcx
				48	xor	%eax,%eax
38	callq	7a <.LC5+0x14>	76: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	49	callq	a1 <.LC126+0x4> 9d: R_X86_64_PLT32
39	Disassembly of section .text._ZNKs7compareEmmRK5smm:			50	Disassembly of section .text._ZNKs7compareEmmRK5smm:	

std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const (size: 146 -> 201 bytes (+55 bytes))

Old						
f	1	push	%rbp	f	1	push %rbp
n	2	mov	%rsi,%rax	n	2	push %rbx
	3	push	%rbx		2	push %rbx
	4	sub	\$0x8,%rsp		3	sub \$0x8,%rsp
	5	mov	(%rdi),%rdi		4	mov (%rdi),%rdi
n	6	mov	-0x18(%rdi),%rbp	n	5	mov -0x18(%rdi),%rbp
	7	cmp	%rbp,%rsi		6	cmp %rbp,%rsi
					7	ja ae <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x86>
					8	mov (%rcx),%rsi
					9	mov -0x18(%rsi),%rbx
					10	cmp %rbx,%r8
					11	ja 86 <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x86>
					12	sub %rax,%rbp
					13	cmp %rdx,%rbp
					14	cmova %rdx,%rbp
					15	sub %r8,%rbx
					16	cmp %r9,%rbx
					17	cmova %r9,%rbx
					18	cmp %rbx,%rbp
					19	mov %rbx,%rdx
					20	cmovbe %rbp,%rdx
					21	add %r8,%rsi
					22	add %rax,%rdi
					23	callq 4a <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x4a> 46: R_X86_64_PLT32 memcmp-0x4
					24	test %eax,%eax
					25	jne 5f <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x5f>
					26	sub %rbx,%rbp
					27	mov \$0x7fffffff,%eax
n	29	cmp	\$0x7fffffff,%rbp	n	26	cmp \$0x7fffffff,%rbp
					27	jle 70 <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x70>
					28	add \$0x8,%rsp
					29	pop %rbx
					30	pop %rbp
					31	retq
n	35	nopw	%cs:0x0(%rax,%rax,1)	n	32	add %rax,%rsi
					33	add %rsi,%rsi
					34	mov %r8,%rcx
					35	callq 6e <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x6e>
					36	test %eax,%eax
					37	je 48 <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x48>
					38	add \$0x8,%rsp
					39	pop %rbx
					40	pop %rbp
					41	retq
t				t	42	nopl 0x0(%rip),%rax
					43	cmp \$0xffffffff,%rbx
					44	mov \$0x80000000,%eax
					45	cmovge %ebx,%eax
					46	add \$0x8,%rsp
					47	pop %rbx
					48	pop %rbp

				49	retq
				50	lea 0x0(%rip),%rdi
				51	lea 0x0(%rip),%rdi
				52	mov %r8,%rax
				53	xor %eax,%eax
				54	callq ae <.LC6+0xf>
				55	mov %rsi,%r8
43	lea 0x0(%rip),%rdi	# 8d <std::string::compare(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) const+0x8d>	89: R_X86_64_PC32 .LC2-0x4	56	lea 0x0(%rip),%rdi
				57	lea 0x0(%rip),%rdi
				58	mov %rbx,%rax
				59	xor %eax,%eax
44	callq 92 <.LC6+0xf>	8e: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4		60	callq c9 <.LC7+0x1>
45	Disassembly of section .text._ZNKs7compareEPKc:			61	Disassembly of section .text._ZNKs7compareEPKc:

std::string::copy(char*, unsigned long, unsigned long) const (size: 87 -> 99 bytes (+12 bytes))

Old							
f	1	push %rbx		f	1	push %rbx	
n				n	2	mov %rsi,%r8	
	2	mov (%rdi),%rax			3	mov (%rdi),%rsi	
					4	mov %rcx,%rax	
	3	mov -0x18(%rax),%rbx			5	mov -0x18(%rsi),%rcx	
	4	cmp %rbx,%rcx			6	cmp %rcx,%rax	
	5	ja 4b <std::string::copy(char*, unsigned long, unsigned long) const+0x4b>			7	ja 4b <std::string::copy(char*, unsigned long, unsigned long) const+0x4b>	
n	6	sub %rcx,%rbx		n	8	sub %rax,%rcx	
	7	cmp %rdx,%rbx			9	cmp %rdx,%rcx	
					10	mov %rcx,%rbx	
	8	cmova %rdx,%rbx			11	cmova %rdx,%rbx	
	9	test %rbx,%rbx			12	test %rbx,%rbx	
n	10	je 34 <std::string::copy(char*, unsigned long, unsigned long) const+0x34>		n	13	je 39 <std::string::copy(char*, unsigned long, unsigned long) const+0x39>	
					14	add %rax,%rsi	
	11	cmp \$0x1,%rbx			15	cmp \$0x1,%rbx	
n	12	mov %rsi,%r8		n			
	13	lea (%rax,%rcx,1),%rsi					
	14	je 40 <std::string::copy(char*, unsigned long, unsigned long) const+0x40>			16	je 40 <std::string::copy(char*, unsigned long, unsigned long) const+0x40>	
	15	mov %rbx,%rdx			17	mov %rbx,%rdx	
	16	mov %r8,%rdi			18	mov %r8,%rdi	
n	17	callq 34 <std::string::copy(char*, unsigned long, unsigned long) const+0x34> 30: R_X86_64_PLT32 memcpy-0x4		n	19	callq 39 <std::string::copy(char*, unsigned long, unsigned long) const+0x39> 30: R_X86_64_PLT32 memcpy-0x4	
	18	mov %rbx,%rax			20	mov %rbx,%rax	
	19	pop %rbx			21	pop %rbx	
	20	retq			22	retq	
n	21	nopl 0x0(%rax)		n	23	xchg %ax,%ax	
	22	movzbl (%rsi),%eax			24	movzbl (%rsi),%eax	
	23	mov %al,(%r8)			25	mov %al,(%r8)	
	24	mov %rbx,%rax			26	mov %rbx,%rax	
	25	pop %rbx			27	pop %rbx	
	26	retq			28	retq	
t	27	lea 0x0(%rip),%rdi # 52 <std::string::copy(char*, unsigned long, unsigned long) const+0x52> 4e: R_X86_64_PC32 .LC1-0x4		t	29	lea 0x0(%rip),%rsi # 52 <std::string::copy(char*, unsigned long, unsigned long) const+0x52> 4e: R_X86_64_PC32 .LC1-0x4	
					30	lea 0x0(%rip),%rdi # 59 <std::string::copy(char*, unsigned long, unsigned long) const+0x59> 4e: R_X86_64_PC32 .LC1-0x4	
					31	mov %rax,%rdx	
					32	xor %eax,%eax	
					33	callq 63 <.LC14+0xe> 5f: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	
	28	callq 57 <.LC4+0x5> 53: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4					
	29	Disassembly of section .text._ZN5s4swapERSs:			34	Disassembly of section .text._ZN5s4swapERSs:	

std::string::data() const (size unchanged)

Old			New		
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
	2	retq		2	retq
t	3	Disassembly of section .text._ZNKS8lengthEv:	t	3	Disassembly of section .text._ZNKS13get_allocatorEv:

std::string::end() (size: 28 -> 37 bytes (+9 bytes))

Old				New			
f	1	mov (%rdi),%rax		f	1	mov (%rdi),%rax	
n				n	2	mov -0x8(%rax),%edx	
					3	test %edx,%edx	
					4	js 20 <std::string::end()+0x20>	
	2	push %rbx			5	push %rbx	
	3	mov %rdi,%rbx			6	mov %rdi,%rbx	
n	4	mov -0x8(%rax),%edx		n			
	5	test %edx,%edx					
	6	js 16 <std::string::end()+0x16>					
	7	callq 13 <std::string::end()+0x13>	f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4		7	callq 13 <std::string::end()+0x13>	f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4
	8	mov (%rbx),%rax			8	mov (%rbx),%rax	
n				n	9	pop %rbx	
	9	add -0x18(%rax),%rax			10	add -0x18(%rax),%rax	

n	10	pop %rbx	n		
	11	retq		11	retq
t			t	12	nopl 0x0(%rax)
				13	add -0x18(%rax),%rax
				14	retq
	12	Disassembly of section .text._ZN5si4EM:		15	Disassembly of section .text._ZN5si4beginEV:

std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>) (size unchanged)

Old								
f	1	push %rbp				f	1	push %r
n	2	xor %ecx,%ecx				n		
	3	mov %rdi,%rbp						
	4	mov \$0x1,%edx						
	5	push %rbx					2	push %r
	6	mov %rsi,%rbx					3	mov %r:
t						t	4	mov %r
							5	xor %e
							6	mov \$0:
	7	sub \$0x8,%rsp					7	sub \$0:
	8	sub (%rdi),%rbx					8	sub (%:
	9	mov %rbx,%rsi					9	mov %r
	10	callq 1e <std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>)+0x1e> 1a: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4					10	callq 1e
	11	mov 0x0(%rbp),%rax					11	mov 0x(
	12	movl \$0xffffffff,-0x8(%rax)					12	movl \$0:
	13	add \$0x8,%rsp					13	add \$0:
	14	add %rbx,%rax					14	add %r
	15	pop %rbx					15	pop %r
	16	pop %rbp					16	pop %r
	17	retq					17	retq
	18	Disassembly of section .text._ZN5s5eraseEN9__gnu_cxx17__normal_iteratorIPc5sEEES2_:					18	Disassemb

std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>) (size: 61 -> 62 bytes (+1 bytes))

Old								
f	1	sub %rsi,%rdx						
	2	mov %rsi,%rax						
	3	jne 10 <std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>)+0x10>						
	4	retq						
	5	nopl 0x0(%rax)						
	6	push %rbp						
n								
	7	xor %ecx,%ecx						
n	8	push %rbx						
	9	mov %rdi,%rbx						
	10	sub \$0x8,%rsp						
	11	sub (%rdi),%rax						
	12	mov %rax,%rsi						
n	13	mov %rax,%rbp						
	14	callq 29 <std::string::erase(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>)+0x29> 25: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4						
n	15	mov (%rbx),%rax						
	16	movl \$0xffffffff,-0x8(%rax)						
	17	add \$0x8,%rsp						
t	18	add %rbp,%rax						
	19	pop %rbx						
	20	pop %rbp						
	21	retq						
	22	Disassembly of section .text._ZN5s14_M_replace_auxEmmmc:						

std::string::erase(unsigned long, unsigned long) (size: 53 -> 68 bytes (+15 bytes))

Old								
f	1	push %rbx				f	1	push %rbx
	2	mov (%rdi),%rax					2	mov (%rdi),%rax
	3	mov %rdi,%rbx					3	mov %rdi,%rbx
n	4	mov -0x18(%rax),%rax				n	4	mov -0x18(%rax),%r
	5	mov %rax,%rcx					5	mov %rcx,%rdi
	6	sub %rsi,%rcx					6	sub %rsi,%rdi
	7	cmp %rdx,%rcx					7	cmp %rdx,%rdi
	8	cmovbe %rcx,%rdx					8	cmovbe %rdi,%rdx
	9	cmp %rax,%rsi					9	cmp %rcx,%rsi
	10	ja 29 <std::string::erase(unsigned long, unsigned long)+0x29>					10	ja 2c <std::string::erase(u
							11	mov %rbx,%rdi
	11	xor %ecx,%ecx					12	xor %ecx,%ecx
n	12	callq 24 <std::string::erase(unsigned long, unsigned long)+0x24> 20: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4				n	13	callq 27 <std::string::erase(u

13	mov %rbx,%rax	14	mov %rbx,%rax
14	pop %rbx	15	pop %rbx
15	retq	16	retq
t		t	mov %rsi,%rdx
16	lea 0x0(%rip),%rdi # 30 <std::string::erase(unsigned long, unsigned long)+0x30> 2c: R_X86_64_PC32 .LC4-0x4	18	lea 0x0(%rip),%rdi #
		19	lea 0x0(%rip),%rsi #
		20	xor %eax,%eax
17	callq 35 <.LC2+0x11> 31: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	21	callq 44 <.LC110+0x3> 40: R_X86_64_PLT32
18	Disassembly of section .text._ZN5s5eraseEN9__gnu_cxx17__normal_iteratorIPCs5EE:	22	Disassembly of section .text._ZN5s5eraseEN9__gnu_cxx17__normal_iteratorIPCs5EE:

std::string::find(char const*, unsigned long) const (size unchanged)

Old		New	
f 1	push %r12	f 1	push %r12
n		n 2	push %rbp
2	mov %rdx,%r12	3	mov %rdx,%r12
n 3	push %rbp	n 4	push %rbp
4	mov %rdi,%rbp	5	mov %rdi,%rbp
n		n 6	mov %rsi,%rbx
5	mov %rsi,%rdi	7	mov %rsi,%rdi
n 6	push %rbx	n	
7	mov %rsi,%rbx		
8	callq 15 <std::string::find(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4	8	callq 15 <std::string::find(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32
n		n 9	mov %r12,%rdx
9	mov %rbx,%rsi	10	mov %rbx,%rsi
10	mov %rbp,%rdi	11	mov %rbp,%rdi
n 11	mov %r12,%rdx	n	
12	pop %rbx	12	pop %rbx
13	pop %rbp	13	pop %rbp
14	pop %r12	14	pop %r12
15	mov %rax,%rcx	15	mov %rax,%rcx
t 16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::find(char const*, unsigned long, unsigned long) const-0x4	t 16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::find(char const*, unsigned long, unsigned long) const-0x4
17	Disassembly of section .text._ZNKs4findEcm:	17	Disassembly of section .text._ZNKs4findEcm:

std::string::find(char const*, unsigned long, unsigned long) const (size: 192 -> 188 bytes (-4 bytes) *)

Old		New	
f 1	push %r15	f 1	push %r15
n 2	mov \$0xffffffffffffffff,%rax	n	
3	push %r14	2	push %r14
4	push %r13	3	push %r13
5	push %r12	4	push %r12
6	push %rbp	5	push %rbp
7	push %rbx	6	push %rbx
8	sub \$0x18,%rsp	7	sub \$0x18,%rsp
9	mov (%rdi),%rbp	8	mov (%rdi),%rbp
10	test %rcx,%rcx	9	test %rcx,%rcx
n 11	mov -0x18(%rbp),%rdi	n 10	mov -0x18(%rbp),%r13
12	jne 40 <std::string::find(char const*, unsigned long, unsigned long) const+0x40>	11	jne 40 <std::string::find(char const*, unsigned long, unsigned long) const+0x40>
n 13	cmp %rdi,%rdx	n 12	cmp %rdx,%r13
		13	mov \$0xffffffffffffffff,%r15
14	cmovbe %rdx,%rax	14	cmovae %rdx,%r15
15	add \$0x18,%rsp	15	add \$0x18,%rsp
n		n 16	mov %r15,%rax
16	pop %rbx	17	pop %rbx
17	pop %rbp	18	pop %rbp
18	pop %r12	19	pop %r12
19	pop %r13	20	pop %r13
20	pop %r14	21	pop %r14
21	pop %r15	22	pop %r15
22	retq	23	retq
23	nopw 0x0(%rax,%rax,1)	24	nopw 0x0(%rax,%rax,1)
n 24	cmp %rdi,%rcx	n 25	cmp %rcx,%r13
		26	mov \$0xffffffffffffffff,%r15
25	ja 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>	27	jb 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>
26	sub %rcx,%rdi	28	sub %rcx,%r13
27	cmp %rdi,%rdx	29	cmp %r13,%rdx
28	mov %rdi,%r12		
29	ja 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>	30	ja 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>
30	lea 0x1(%rdx),%rbx	31	lea 0x1(%rdx),%rbx
n 31	lea (%rcx,%rax,1),%r14	n	
32	lea 0x1(%rsi),%rax	32	lea 0x1(%rsi),%rax
n 33	movzbl (%rsi),%r15d	n 33	movzbl (%rsi),%r14d
		34	lea (%rcx,%r15,1),%r12
34	add %rbx,%rbp	35	add %rbx,%rbp
35	mov %rax,0x8(%rsp)	36	mov %rax,0x8(%rsp)

	36	jmp 7d <std::string::find(char const*, unsigned long, unsigned long) const+0x7d>		37	jmp 7d <std::string::find(char const*, unsigned long, unsigned long) co
n	37	nopw 0x0(%rax,%rax,1)		n	38 xchg %ax,%ax
	38	add \$0x1,%rbx		39	add \$0x1,%rbx
	39	add \$0x1,%rbp		40	add \$0x1,%rbp
n	40	cmp %rdi,%r12		n	41 cmp %r13,%rax
	41	jb a8 <std::string::find(char const*, unsigned long, unsigned long) const+0xa8>		42	ja b0 <std::string::find(char const*, unsigned long, unsigned long) co
	42	cmp %r15b,-0x1(%rbp)		43	cmp -0x1(%rbp),%r14b
	43	lea -0x1(%rbx),%r13		44	lea -0x1(%rbx),%r15
	44	mov %rbx,%rdi		45	mov %rbx,%rax
	45	jne 70 <std::string::find(char const*, unsigned long, unsigned long) const+0x70>		46	jne 70 <std::string::find(char const*, unsigned long, unsigned long) co
n				n	47 test %r12,%r12
				48	je 28 <std::string::find(char const*, unsigned long, unsigned long) co
	46	mov 0x8(%rsp),%rsi		49	mov 0x8(%rsp),%rsi
n	47	mov %r14,%rdx		n	50 mov %r12,%rdx
	48	mov %rbp,%rdi		51	mov %rbp,%rdi
n	49	callq 9a <std::string::find(char const*, unsigned long, unsigned long) const+0x9a> 96: R_X86_64_PLT32 memcmp-0x4		n	52 callq 9f <std::string::find(char const*, unsigned long, unsigned long) co
	50	test %eax,%eax		53	test %eax,%eax
n	51	je b8 <std::string::find(char const*, unsigned long, unsigned long) const+0xb8>		n	54 je 28 <std::string::find(char const*, unsigned long, unsigned long) co
	52	mov %rbx,%rdi		55	mov %rbx,%rax
	53	jmp 70 <std::string::find(char const*, unsigned long, unsigned long) const+0x70>		56	jmp 70 <std::string::find(char const*, unsigned long, unsigned long) co
	54	nopl 0x0(%rax,%rax,1)		57	nopl 0x0(%rax,%rax,1)
t	55	mov \$0xffffffffffffffff,%rax		t	58 mov \$0xffffffffffffffff,%r15
	56	jmpq 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>			
	57	nopl 0x0(%rax)			
	58	mov %r13,%rax			
	59	jmpq 28 <std::string::find(char const*, unsigned long, unsigned long) const+0x28>		59	jmpq 28 <std::string::find(char const*, unsigned long, unsigned long) co
	60	Disassembly of section .text._ZNKSs4findERKSsm:		60	Disassembly of section .text._ZNKSs4findERKSsm:

std::string::find(char, unsigned long) const (size: 83 -> 82 bytes (-1 bytes) *)

Old			New		
f	1	push %rbp	f	1	push %rbp
	2	push %rbx		2	push %rbx
n	3	mov \$0xffffffffffffffff,%rbx	n	3	mov \$0xffffffffffffffff,%rbp
	4	sub \$0x8,%rsp		4	sub \$0x8,%rsp
n	5	mov (%rdi),%rbp	n	5	mov (%rdi),%rbx
	6	mov -0x18(%rbp),%rax		6	mov -0x18(%rbx),%rax
	7	cmp %rax,%rdx		7	cmp %rax,%rdx
	8	jb 28 <std::string::find(char, unsigned long) const+0x28>		8	jb 28 <std::string::find(char, unsigned long) const+0x28>
	9	add \$0x8,%rsp		9	add \$0x8,%rsp
n	10	mov %rbx,%rax	n	10	mov %rbp,%rax
	11	pop %rbx		11	pop %rbx
	12	pop %rbp		12	pop %rbp
	13	retq		13	retq
	14	nopl 0x0(%rax,%rax,1)		14	nopl 0x0(%rax,%rax,1)
	15	sub %rdx,%rax		15	sub %rdx,%rax
n	16	lea 0x0(%rbp,%rdx,1),%rdi	n	16	mov %rdx,%rdi
	17	movsbl %sil,%esi		17	movsbl %sil,%esi
n			n	18	add %rbx,%rdi
	18	mov %rax,%rdx		19	mov %rax,%rdx
n	19	callq 3c <std::string::find(char, unsigned long) const+0x3c> 38: R_X86_64_PLT32 memchr-0x4	n	20	callq 3d <std::string::find(char, unsigned long) const+0x3d> 39: R_X86_64_PLT32 memchr-0x4
	20	mov %rax,%rdx			
	21	sub %rbp,%rdx			
	22	test %rax,%rax		21	test %rax,%rax
n	23	cmovne %rdx,%rbx	n	22	je 19 <std::string::find(char, unsigned long) const+0x19>
				23	sub %rbx,%rax
	24	add \$0x8,%rsp		24	add \$0x8,%rsp
t			t	25	mov %rax,%rbp
	25	mov %rbx,%rax		26	mov %rbp,%rax
	26	pop %rbx		27	pop %rbx
	27	pop %rbp		28	pop %rbp
	28	retq		29	retq
	29	Disassembly of section .text._ZNKSs5rfindEPKcmm:		30	Disassembly of section .text._ZNKSs5rfindEPKcmm:

std::string::find_first_not_of(char const*, unsigned long) const (size unchanged)

Old			New		
f	1	push %r12	f	1	push %r12
n			n	2	push %rbp
	2	mov %rdx,%r12		3	mov %rdx,%r12
n	3	push %rbp	n	4	push %rbx
	4	mov %rdi,%rbp		5	mov %rdi,%rbp
n			n	6	mov %rsi,%rbx
	5	mov %rsi,%rdi		7	mov %rsi,%rdi
n	6	push %rbx	n		

7	mov %rsi,%rbx				
8	callq 15 <std::string::find_first_not_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4		8	callq 15 <std::string::find_first_not_of(char const*, unsigned	
n			n	9 mov %r12,%rdx	
9	mov %rbx,%rsi		10	mov %rbx,%rsi	
10	mov %rbp,%rdi		11	mov %rbp,%rdi	
n	11 mov %r12,%rdx		n		
12	pop %rbx		12	pop %rbx	
13	pop %rbp		13	pop %rbp	
14	pop %r12		14	pop %r12	
15	mov %rax,%rcx		15	mov %rax,%rcx	
t	16 jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x4	t	16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::find_	
17	Disassembly of section .text._ZNKSs17find_first_not_ofEcm:	17	Disassembly of section .text._ZNKSs17find_first_not_ofEcm:		

std::string::find_first_not_of(char const*, unsigned long, unsigned long) const (size: 112 -> 105 bytes (-7 bytes) *)

Old											
f	1	push %r14				f	1	push %r14			
n						n	2	push %r13			
	2	mov \$0xffffffffffffffff,%rax					3	mov \$0xffffffffffffffff,%rax			
n	3	mov %rsi,%r14				n					
	4	push %r13									
	5	mov %rcx,%r13									
	6	push %r12					4	push %r12			
	7	push %rbp					5	push %rbp			
	8	push %rbx					6	push %rbx			
n	9	mov (%rdi),%rbp				n	7	mov (%rdi),%r13			
							8	mov -0x18(%r13),%r12			
							9	cmp %r12,%rdx			
							10	jae 56 <std::string::find_first_not_of(char const*, uns			
							11	test %rcx,%rcx			
							12	je 5f <std::string::find_first_not_of(char const*, uns			
							13	mov %rcx,%rbp			
							14	mov %rsi,%r14			
							15	mov %rdx,%rbx			
	10	mov %rdx,%rbx									
n	11	mov -0x18(%rbp),%r12				n	16	nopl 0x0(%rax)			
							17	movsbl 0x0(%r13,%rbx,1),%esi			
	12	cmp %r12,%rdx					18	mov %rbp,%rdx			
							19	mov %r14,%rdi			
							20	callq 41 <std::string::find_first_not_of(char const*, uns			
							21	test %rax,%rax			
	13	jb 39 <std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x39>					22	je 64 <std::string::find_first_not_of(char const*, uns			
	14	jmp 51 <std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x51>									
	15	nopw %cs:0x0(%rax,%rax,1)									
	16	add \$0x1,%rbx					23	add \$0x1,%rbx			
	17	cmp %r12,%rbx					24	cmp %r12,%rbx			
n	18	jae 60 <std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x60>				n					
	19	movsbl 0x0(%rbp,%rbx,1),%esi									
	20	mov %r13,%rdx									
	21	mov %r14,%rdi									
	22	callq 49 <std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x49> 45: R_X86_64_PLT32 memchr-0x4									
	23	test %rax,%rax									
	24	jne 30 <std::string::find_first_not_of(char const*, unsigned long, unsigned long) const+0x30>					25	jb 30 <std::string::find_first_not_of(char const*, uns			
	25	mov %rbx,%rax					26	mov \$0xffffffffffffffff,%rax			
	26	pop %rbx					27	pop %rbx			
	27	pop %rbp					28	pop %rbp			
	28	pop %r12					29	pop %r12			
	29	pop %r13					30	pop %r13			
	30	pop %r14					31	pop %r14			
	31	retq					32	retq			
t	32	nopw 0x0(%rax,%rax,1)				t	33	mov %rdx,%rax			
	33	pop %rbx					34	jmp 56 <std::string::find_first_not_of(char const*, uns			
	34	pop %rbp					35	mov %rbx,%rax			
	35	pop %r12					36	jmp 56 <std::string::find_first_not_of(char const*, uns			
	36	pop %r13									
	37	mov \$0xffffffffffffffff,%rax									
	38	pop %r14									
	39	retq									
	40	Disassembly of section .text._ZNKSs17find_first_not_ofERKSsm:					37	Disassembly of section .text._ZNKSs17find_first_not_ofERKS:			

std::string::find_first_not_of(char, unsigned long) const (size unchanged)

Old			New		
f	1	mov (%rdi),%rcx	f	1	mov (%rdi),%rcx
n			n	2	mov %esi,%r8d
	2	mov \$0xffffffffffffffff,%rax		3	mov \$0xffffffffffffffff,%rax
	3	mov -0x18(%rcx),%rdi		4	mov -0x18(%rcx),%rdi

4	cmp %rdi,%rdx	5	cmp %rdi,%rdx
5	jae 43 <std::string::find_first_not_of(char, unsigned long) const+0x43>	6	jae 43 <std::string::find_first_not_of(char, unsigned long) const+0x43>
n 6	movzbl (%rcx,%rdx,1),%eax	n 7	cmp (%rcx,%rdx,1),%sil
7	cmp %sil,%al		
8	je 25 <std::string::find_first_not_of(char, unsigned long) const+0x25>	8	je 26 <std::string::find_first_not_of(char, unsigned long) const+0x26>
9	jmp 40 <std::string::find_first_not_of(char, unsigned long) const+0x40>	9	jmp 40 <std::string::find_first_not_of(char, unsigned long) const+0x40>
10	xchg %ax,%ax	10	xchg %ax,%ax
n 11	cmp %al,(%rcx,%rdx,1)	n 11	cmp %r8b,(%rcx,%rdx,1)
12	jne 40 <std::string::find_first_not_of(char, unsigned long) const+0x40>	12	jne 40 <std::string::find_first_not_of(char, unsigned long) const+0x40>
13	add \$0x1,%rdx	13	add \$0x1,%rdx
14	cmp %rdi,%rdx	14	cmp %rdi,%rdx
15	jb 20 <std::string::find_first_not_of(char, unsigned long) const+0x20>	15	jb 20 <std::string::find_first_not_of(char, unsigned long) const+0x20>
16	mov \$0xfffffffffffffff,%rax	16	mov \$0xfffffffffffffff,%rax
17	retq	17	retq
t 18	nopw %cs:0x0(%rax,%rax,1)	t 18	nopw 0x0(%rax,%rax,1)
19	mov %rdx,%rax	19	mov %rdx,%rax
20	repz retq	20	repz retq
21	Disassembly of section .text._ZNKSt16find_last_not_ofEPKcmm:	21	Disassembly of section .text._ZNKSt16find_last_not_ofEPKcmm:

std::string::find_first_of(char const*, unsigned long) const (size unchanged)

Old			New		
f 1	push %r12		f 1	push %r12	
n			n 2	push %rbp	
2	mov %rdx,%r12		3	mov %rdx,%r12	
n 3	push %rbp		n 4	push %rbx	
4	mov %rdi,%rbp		5	mov %rdi,%rbp	
n			n 6	mov %rsi,%rbx	
5	mov %rsi,%rdi		7	mov %rsi,%rdi	
n 6	push %rbx		n		
7	mov %rsi,%rbx				
8	callq 15 <std::string::find_first_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4		8	callq 15 <std::string::find_first_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4	
n			n 9	mov %r12,%rdx	
9	mov %rbx,%rsi		10	mov %rbx,%rsi	
10	mov %rbp,%rdi		11	mov %rbp,%rdi	
n 11	mov %r12,%rdx		n		
12	pop %rbx		12	pop %rbx	
13	pop %rbp		13	pop %rbp	
14	pop %r12		14	pop %r12	
15	mov %rax,%rcx		15	mov %rax,%rcx	
t 16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::find_first_of(char const*, unsigned long, unsigned long) const-0x4		t 16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::find_first_of(char const*, unsigned long, unsigned long) const-0x4	
17	Disassembly of section .text._ZNKSt13find_first_ofEcm:		17	Disassembly of section .text._ZNKSt13find_first_ofEcm:	

std::string::find_first_of(char const*, unsigned long, unsigned long) const (size: 108 -> 101 bytes (-7 bytes) *)

Old			New		
n 1	push %r14		n		
2	test %rcx,%rcx		1	test %rcx,%rcx	
3	mov \$0xfffffffffffffff,%rax		2	mov \$0xfffffffffffffff,%rax	
n			n 3	je 5d <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x5d>	
			4	push %r14	
4	push %r13		5	push %r13	
n			n 6	mov %rsi,%r13	
5	push %r12		7	push %r12	
n 6	mov %rcx,%r12		n		
7	push %rbp		8	push %rbp	
n			n 9	mov %rcx,%rbp	
8	push %rbx		10	push %rbx	
n 9	je 56 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x56>		n		
10	mov (%rdi),%r13		11	mov (%rdi),%r12	
11	mov %rsi,%r14				
12	mov %rdx,%rbx		12	mov %rdx,%rbx	
n 13	mov -0x18(%r13),%rbp		n 13	mov -0x18(%r12),%r14	
14	cmp %rdx,%rbp		14	cmp %r14,%rdx	
15	jbe 56 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x56>		15	jbe 55 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x55>	
16	nop 0x0(%rax)		16	nop 0x0(%rax,%rax,1)	
17	movsbl 0x0(%r13,%rbx,1),%esi		17	movsbl (%r12,%rbx,1),%esi	
18	mov %r12,%rdx		18	mov %rbp,%rdx	
19	mov %r14,%rdi		19	mov %r13,%rdi	
20	callq 41 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x41> 3d: R_X86_64_PLT32 memchr-0x4		20	callq 40 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x40> 3d: R_X86_64_PLT32 memchr-0x4	
21	test %rax,%rax		21	test %rax,%rax	
22	jne 60 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x60>		22	jne 60 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x60>	
23	add \$0x1,%rbx		23	add \$0x1,%rbx	
n 24	cmp %rbx,%rbp		n 24	cmp %r14,%rbx	
25	ja 30 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x30>		25	jb 30 <std::string::find_first_of(char const*, unsigned long, unsigned long) const+0x30>	
26	mov \$0xfffffffffffffff,%rax		26	mov \$0xfffffffffffffff,%rax	

27	pop %rbx	27	pop %rbx
28	pop %rbp	28	pop %rbp
29	pop %r12	29	pop %r12
30	pop %r13	30	pop %r13
31	pop %r14	31	pop %r14
n 32	retq	n 32	repz retq
33	nop	33	nop
34	mov %rbx,%rax	34	mov %rbx,%rax
t 35	pop %rbx	t 35	jmp 55 <std::string::find_first_of(char const*, unsigned long)
36	pop %rbp		
37	pop %r12		
38	pop %r13		
39	pop %r14		
40	retq		
41	Disassembly of section .text._ZNKSs13find_first_ofERKSsm:	36	Disassembly of section .text._ZNKSs13find_first_ofERKSsm:

std::string::find_last_not_of(char const*, unsigned long) const (size unchanged)

Old			New		
f 1	push %r12		f 1	push %r12	
n			n 2	push %rbp	
2	mov %rdx,%r12		3	mov %rdx,%r12	
n 3	push %rbp		n 4	push %rbx	
4	mov %rdi,%rbp		5	mov %rdi,%rbp	
n			n 6	mov %rsi,%rbx	
5	mov %rsi,%rdi		7	mov %rsi,%rdi	
n 6	push %rbx		n		
7	mov %rsi,%rbx				
8	callq 15 <std::string::find_last_not_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4		8	callq 15 <std::string::find_last_not_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4	
n			n 9	mov %r12,%rdx	
9	mov %rbx,%rsi		10	mov %rbx,%rsi	
10	mov %rbp,%rdi		11	mov %rbp,%rdi	
n 11	mov %r12,%rdx		n		
12	pop %rbx		12	pop %rbx	
13	pop %rbp		13	pop %rbp	
14	pop %r12		14	pop %r12	
15	mov %rax,%rcx		15	mov %rax,%rcx	
t 16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x4		t 16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x4	
17	Disassembly of section .text._ZNKSs16find_last_not_ofEcm:		17	Disassembly of section .text._ZNKSs16find_last_not_ofEcm:	

std::string::find_last_not_of(char const*, unsigned long, unsigned long) const (size: 130 -> 146 bytes (+16 bytes))

Old			New		
f 1	push %r13		f 1	push %r13	
n 2	mov %rcx,%r13		n 2	push %r12	
3	mov \$0xfffffffffffffff,%rax		3	mov \$0xfffffffffffffff,%rax	
n 4	push %r12		n		
5	push %rbp		4	push %rbp	
6	push %rbx		5	push %rbx	
7	sub \$0x8,%rsp		6	sub \$0x8,%rsp	
n 8	mov (%rdi),%rbp		n 7	mov (%rdi),%r12	
9	mov -0x18(%rbp),%rcx		8	mov -0x18(%r12),%rdi	
10	test %rcx,%rcx		9	test %rdi,%rdi	
11	jne 30 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x30>		10	jne 30 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x30>	
12	add \$0x8,%rsp		11	add \$0x8,%rsp	
13	pop %rbx		12	pop %rbx	
14	pop %rbp		13	pop %rbp	
15	pop %r12		14	pop %r12	
16	pop %r13		15	pop %r13	
17	retq		16	retq	
n 18	nopl 0x0(%rax,%rax,1)		n 17	nopl 0x0(%rax)	
19	lea (%rcx,%rax,1),%rbx		18	add %rax,%rdi	
			19	mov %rcx,%rbp	
20	mov %rsi,%r12		20	mov %rsi,%r13	
21	cmp %rbx,%rdx		21	cmp %rdx,%rdi	
			22	cmovbe %rdi,%rdx	
22	cmovbe %rdx,%rbx		23	mov %rdx,%rbx	
23	jmp 4c <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x4c>		24	jmp 69 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x4c>	
			25	nopl (%rax)	
			26	movsbl (%r12,%rbx,1),%esi	
			27	mov %rbp,%rdx	
			28	mov %r13,%rdi	
			29	callq 58 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x58> 58: R_X86_64_PLT32 strlen-0x4	
			30	test %rax,%rax	
			31	je 6e <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x6e> 6e: R_X86_64_PLT32 strlen-0x4	

	24	test %rbx,%rbx		32	test %rbx,%rbx
	25	lea -0x1(%rbx),%rax		33	lea -0x1(%rbx),%rax
n	26	je 70 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x70>	n	34	je 80 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x80>
	27	mov %rax,%rbx		35	mov %rax,%rbx
n	28	movsbl 0x0(%rbp,%rbx,1),%esi	n		
	29	mov %r13,%rdx			
	30	mov %r12,%rdi			
	31	callq 5c <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x5c> 58: R_X86_64_PLT32 memchr-0x4			
	32	test %rax,%rax		36	test %rbp,%rbp
	33	jne 40 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x40>		37	jne 48 <std::string::find_last_not_of(char const*, unsigned long, unsigned long) const+0x48>
	34	add \$0x8,%rsp		38	add \$0x8,%rsp
	35	mov %rbx,%rax		39	mov %rbx,%rax
	36	pop %rbx		40	pop %rbx
	37	pop %rbp		41	pop %rbp
	38	pop %r12		42	pop %r12
	39	pop %r13		43	pop %r13
	40	retq		44	retq
t	41	nop	t	45	nopl 0x0(%rax)
	42	add \$0x8,%rsp		46	add \$0x8,%rsp
	43	mov \$0xffffffffffffffff,%rax		47	mov \$0xffffffffffffffff,%rax
	44	pop %rbx		48	pop %rbx
	45	pop %rbp		49	pop %rbp
	46	pop %r12		50	pop %r12
	47	pop %r13		51	pop %r13
	48	retq		52	retq
	49	Disassembly of section .text._ZNKSs16find_last_not_ofERKSsm:		53	Disassembly of section .text._ZNKSs16find_last_not_ofERKSsm:

std::string::find_last_not_of(char, unsigned long) const (size: 72 -> 80 bytes (+8 bytes))

Old			New		
f	1	mov (%rdi),%rcx	f	1	mov (%rdi),%rcx
	2	mov \$0xffffffffffffffff,%rax		2	mov \$0xffffffffffffffff,%rax
	3	mov -0x18(%rcx),%rdi		3	mov -0x18(%rcx),%rdi
	4	test %rdi,%rdi		4	test %rdi,%rdi
	5	jne 18 <std::string::find_last_not_of(char, unsigned long) const+0x18>		5	jne 18 <std::string::find_last_not_of(char, unsigned long) const+0x18>
	6	repz retq		6	repz retq
	7	nopl (%rax)		7	nopl (%rax)
n			n	8	add %rax,%rdi
	8	add %rdi,%rax		9	mov %rdx,%rax
	9	cmp %rax,%rdx		10	cmp %rdx,%rdi
	10	cmovbe %rdx,%rax		11	cmovbe %rdi,%rax
	11	jmp 34 <std::string::find_last_not_of(char, unsigned long) const+0x34>		12	jmp 3c <std::string::find_last_not_of(char, unsigned long) const+0x3c>
	12	nopl 0x0(%rax)		13	nopl 0x0(%rax,%rax,1)
	13	test %rax,%rax		14	test %rax,%rax
	14	lea -0x1(%rax),%rdx		15	lea -0x1(%rax),%rdx
n	15	je 40 <std::string::find_last_not_of(char, unsigned long) const+0x40>	n	16	je 48 <std::string::find_last_not_of(char, unsigned long) const+0x48>
	16	mov %rdx,%rax		17	mov %rdx,%rax
	17	cmp %sil,(%rcx,%rax,1)		18	cmp %sil,(%rcx,%rax,1)
t	18	je 28 <std::string::find_last_not_of(char, unsigned long) const+0x28>	t	19	je 30 <std::string::find_last_not_of(char, unsigned long) const+0x30>
	19	jmp 13 <std::string::find_last_not_of(char, unsigned long) const+0x13>		20	jmp 13 <std::string::find_last_not_of(char, unsigned long) const+0x13>
	20	nopl 0x0(%rax)		21	nopl 0x0(%rax)
	21	mov \$0xffffffffffffffff,%rax		22	mov \$0xffffffffffffffff,%rax
	22	retq		23	retq
	23	Disassembly of section .text._ZNKSs7compareERKSs:		24	Disassembly of section .text._ZNKSs7compareERKSs:

std::string::find_last_of(char const*, unsigned long) const (size unchanged)

Old			New		
f	1	push %r12	f	1	push %r12
n			n	2	push %rbp
	2	mov %rdx,%r12		3	mov %rdx,%r12
n	3	push %rbp	n	4	push %rbx
	4	mov %rdi,%rbp		5	mov %rdi,%rbp
n			n	6	mov %rsi,%rbx
	5	mov %rsi,%rdi		7	mov %rsi,%rdi
n	6	push %rbx	n		
	7	mov %rsi,%rbx			
	8	callq 15 <std::string::find_last_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4		8	callq 15 <std::string::find_last_of(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4
n			n	9	mov %r12,%rdx
	9	mov %rbx,%rsi		10	mov %rbx,%rsi
	10	mov %rbp,%rdi		11	mov %rbp,%rdi
n	11	mov %r12,%rdx	n		
	12	pop %rbx		12	pop %rbx
	13	pop %rbp		13	pop %rbp
	14	pop %r12		14	pop %r12
	15	mov %rax,%rcx		15	mov %rax,%rcx

t	16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::find_last_of(char const*, unsigned long, unsigned long) const-0x4	t	16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::find_last_of(char const*, unsigned long, unsigned long) const-0x4
	17	Disassembly of section .text._ZNKSs12find_last_ofEcm:		17	Disassembly of section .text._ZNKSs12find_last_ofEcm:

std::string::find_last_of(char const*, unsigned long, unsigned long) const (size: 119 -> 127 bytes (+8 bytes))

Old			New		
f	1	push %r13	f	1	push %r13
	2	push %r12		2	push %r12
n	3	mov %rcx,%r12	n		
	4	push %rbp		3	push %rbp
	5	push %rbx		4	push %rbx
	6	sub \$0x8,%rsp		5	sub \$0x8,%rsp
n	7	mov (%rdi),%rbp	n	6	mov (%rdi),%r12
				7	mov -0x18(%r12),%r8
				8	test %r8,%r8
	8	test %rcx,%rcx		9	je 1c <std::string::find_last_of(char const*, unsigned long, unsigned long) const-0x30>
n	9	mov -0x18(%rbp),%rbx	n	10	test %rcx,%rcx
	10	jne 30 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x30>		11	jne 30 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x30>
	11	add \$0x8,%rsp		12	add \$0x8,%rsp
	12	mov \$0xffffffffffffffff,%rax		13	mov \$0xffffffffffffffff,%rax
	13	pop %rbx		14	pop %rbx
	14	pop %rbp		15	pop %rbp
	15	pop %r12		16	pop %r12
	16	pop %r13		17	pop %r13
	17	retq		18	retq
n	18	nopl 0x0(%rax,%rax,1)	n	19	xchg %ax,%ax
	19	test %rbx,%rbx			
	20	je 19 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x19>			
	21	sub \$0x1,%rbx		20	sub \$0x1,%r8
				21	mov %rcx,%rbp
	22	mov %rsi,%r13		22	mov %rsi,%r13
n	23	cmp %rbx,%rdx	n	23	cmp %rdx,%r8
				24	cmovbe %r8,%rdx
	24	cmovbe %rdx,%rbx		25	mov %rdx,%rbx
	25	jmp 54 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x54>		26	jmp 5c <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x5c>
	26	nopl (%rax)		27	nopw %cs:0x0(%rax,%rax,1)
	27	test %rbx,%rbx		28	test %rbx,%rbx
	28	lea -0x1(%rbx),%rax		29	lea -0x1(%rbx),%rax
n	29	je 19 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x19>	n	30	je 1c <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x1c>
	30	mov %rax,%rbx		31	mov %rax,%rbx
n	31	movsbl 0x0(%rbp,%rbx,1),%esi	n	32	movsbl %r12,%rbx,1,%esi
	32	mov %r12,%rdx		33	mov %rbp,%rdx
	33	mov %r13,%rdi		34	mov %r13,%rdi
n	34	callq 64 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x64> 60: R_X86_64_PLT32 memchr-0x4	n	35	callq 6c <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x6c> 60: R_X86_64_PLT32 memchr-0x4
	35	test %rax,%rax		36	test %rax,%rax
t	36	je 48 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x48>	t	37	je 50 <std::string::find_last_of(char const*, unsigned long, unsigned long) const+0x50>
	37	add \$0x8,%rsp		38	add \$0x8,%rsp
	38	mov %rbx,%rax		39	mov %rbx,%rax
	39	pop %rbx		40	pop %rbx
	40	pop %rbp		41	pop %rbp
	41	pop %r12		42	pop %r12
	42	pop %r13		43	pop %r13
	43	retq		44	retq
	44	Disassembly of section .text._ZNKSs12find_last_ofERKSsm:		45	Disassembly of section .text._ZNKSs12find_last_ofERKSsm:

std::string::front() (size: 24 -> 25 bytes (+1 bytes))

Old			New		
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
n			n	2	mov -0x8(%rax),%edx
				3	test %edx,%edx
				4	js 17 <std::string::front()+0x17>
	2	push %rbx		5	push %rbx
	3	mov %rdi,%rbx		6	mov %rdi,%rbx
n	4	mov -0x8(%rax),%edx	n		
	5	test %edx,%edx			
	6	js 16 <std::string::front()+0x16>			
	7	callq 13 <std::string::front()+0x13> f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4		7	callq 13 <std::string::front()+0x13> f: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4
	8	mov (%rbx),%rax		8	mov (%rbx),%rax
	9	pop %rbx		9	pop %rbx
t	10	retq	t	10	repz retq
	11	Disassembly of section .text._ZNSS4backEv:		11	Disassembly of section .text._ZNSS4backEv:

std::string::insert(__gnu_cxx::__normal_iterator<char*, std::string>, char) (size unchanged)

Old			New		

f	1	push %rbp	
n	2	movsbl %dl,%r8d	
	3	mov %rdi,%rbp	
	4	mov \$0xi,%ecx	
	5	xor %edx,%edx	
	6	push %rbx	
	7	mov %rsi,%rbx	
n			
	8	sub \$0x8,%rsp	
	9	sub (%rdi),%rbx	
t			
	10	mov %rbx,%rsi	
	11	callq 22 <std::string::insert(__gnu_cxx::__normal_iterator<char*, std::string>, char)+0x22> 1e: R_X86_64_PLT32 std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4	
	12	mov 0x0(%rbp),%rax	
	13	movl \$0xffffffff,-0x8(%rax)	
	14	add \$0x8,%rsp	
	15	add %rbx,%rax	
	16	pop %rbx	
	17	pop %rbp	
	18	retq	
	19	Disassembly of section .text._ZN5s7replaceEmmmc:	

std::string::insert(__gnu_cxx::__normal_iterator<char*, std::string>, unsigned long, char) (size unchanged)

Old						
f	1	sub (%rdi),%rsi		f	1	sub (%rdi),%rsi
	2	movsbl %cl,%r8d			2	movsbl %cl,%r8d
	3	mov %rdx,%rcx			3	mov %rdx,%rcx
	4	xor %edx,%edx			4	xor %edx,%edx
t	5	jmpq 11 <.LC1> d: R_X86_64_PLT32 std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4		t	5	jmpq 11 <std::string::insert(__gnu_cxx::__normal_iterator<char*, std::string>, char)+0x22> 1e: R_X86_64_PLT32 std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4
					>	har)-0x4
	6	Disassembly of section .text._ZN5s15_M_replace_safeEmmPKcm:			6	Disassembly of section .text._ZN5s15_M_replace_safeEmmPKcm:

std::string::insert(unsigned long, char const*) (size unchanged)

Old				New		
f	1	push %r12		f	1	push %r12
n				n	2	push %rbp
	2	mov %rsi,%r12			3	mov %rsi,%r12
n	3	push %rbp		n	4	push %rbx
	4	mov %rdi,%rbp			5	mov %rdi,%rbp
n				n	6	mov %rdx,%rbx
	5	mov %rdx,%rdi			7	mov %rdx,%rdi
n	6	push %rbx		n		
	7	mov %rdx,%rbx				
	8	callq 15 <std::string::insert(unsigned long, char const*)+0x15> 11: R_X86_64_PLT32 strlen-0x4			8	callq 15 <std::string::insert(unsigned long, char const*)+0x15> 11: R_X86_64_PLT32 strlen-0x4
	9	mov %rbx,%rdx			9	mov %rbx,%rdx
n				n	10	mov %r12,%rsi
	10	mov %rbp,%rdi			11	mov %rbp,%rdi
n	11	mov %r12,%rsi		n		
	12	pop %rbx			12	pop %rbx
	13	pop %rbp			13	pop %rbp
	14	pop %r12			14	pop %r12
	15	mov %rax,%rcx			15	mov %rax,%rcx
t	16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::insert(unsigned long, char const*, unsigned long)-0x4		t	16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::insert(unsigned long, char const*, unsigned long)-0x4
	17	Disassembly of section .text._ZN5s6insertEmRK5s:			17	Disassembly of section .text._ZN5s6insertEmRK5s:

std::string::insert(unsigned long, char const*, unsigned long) (size: 375 -> 422 bytes (+47 bytes))

Old			
n			n
	1	push %r12	
n	2	mov %rcx,%r8	n
	3	push %rbp	
n	4	mov %rsi,%rbp	n
	5	push %rbx	
n	6	mov %rdi,%rbx	n
	7	sub \$0x10,%rsp	
	8	mov (%rdi),%r10	
	9	mov -0x18(%r10),%rcx	
	10	cmp %rcx,%rsi	
n	11	ja 15f <std::string::insert(unsigned long, char const*, unsigned long)+0x15f>	n
	12	movabs \$0x3fffffffffffffff9,%r9	
	13	sub %rcx,%r9	
	14	cmp %r9,%r8	

[illegible]

	34	nopl 0x0(%rax,%rax,1)	
n	35	sub %r10,%rdx	n
	36	mov %rbp,%rsi	
	37	mov %r8,%rcx	
	38	mov %rdx,%r12	
	39	xor %edx,%edx	
	40	mov %r8,0x8(%rsp)	
	41	callq 88 <std::string::insert(unsigned long, char const*, unsigned long)+0x88> 84: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4	
	42	mov (%rbx),%rax	
	43	mov 0x8(%rsp),%r8	
	44	lea (%rax,%r12,1),%rsi	
	45	add %rax,%rbp	
	46	lea (%rsi,%r8,1),%rax	
	47	cmp %rax,%rbp	
	48	jb c0 <std::string::insert(unsigned long, char const*, unsigned long)+0xc0>	
	49	cmp \$0x1,%r8	
	50	jne d8 <std::string::insert(unsigned long, char const*, unsigned long)+0xd8>	
	51	movzbl (%rsi),%eax	
n	52	mov %al,0x0(%rbp)	n
	53	add \$0x10,%rsp	
	54	mov %rbx,%rax	
	55	pop %rbx	
	56	pop %rbp	
	57	pop %r12	
	58	retq	
	59	nopl 0x0(%rax,%rax,1)	
n	60	cmp %rbp,%rsi	n
	61	jb e8 <std::string::insert(unsigned long, char const*, unsigned long)+0xe8>	
	62	cmp \$0x1,%r8	
	63	jne 120 <std::string::insert(unsigned long, char const*, unsigned long)+0x120>	
	64	movzbl (%rax),%eax	
	65	mov %al,0x0(%rbp)	
	66	jmp ac <std::string::insert(unsigned long, char const*, unsigned long)+0xac>	
	67	nopl 0x0(%rax,%rax,1)	
	68	mov %r8,%rdx	
	69	mov %rbp,%rdi	
	70	callq e3 <std::string::insert(unsigned long, char const*, unsigned long)+0xe3> df: R_X86_64_PLT32 memcpy-0x4	
	71	jmp ac <std::string::insert(unsigned long, char const*, unsigned long)+0xac>	
	72	nopl (%rax)	
	73	mov %rbp,%r12	
	74	sub %rsi,%r12	
	75	cmp \$0x1,%r12	
	76	jne 148 <std::string::insert(unsigned long, char const*, unsigned long)+0x148>	
	77	movzbl (%rsi),%eax	
	78	mov %al,0x0(%rbp)	
	79	mov %r8,%rdx	
	80	lea 0x0(%rbp,%r8,1),%rsi	
	81	lea 0x0(%rbp,%r12,1),%rdi	
	82	sub %r12,%rdx	
	83	cmp \$0x1,%rdx	
	84	jne 138 <std::string::insert(unsigned long, char const*, unsigned long)+0x138>	
	85	movzbl (%rsi),%eax	
	86	mov %al,(%rdi)	
n	87	jmp a8 <std::string::insert(unsigned long, char const*, unsigned long)+0xa8>	n
	88	nopw 0x0(%rax,%rax,1)	
t	89	mov %r8,%rdx	t
	90	mov %rax,%rsi	
	91	mov %rbp,%rdi	
	92	callq 12e <std::string::insert(unsigned long, char const*, unsigned long)+0x12e> 12a: R_X86_64_PLT32 memcpy-0x4	
	93	jmpq a8 <std::string::insert(unsigned long, char const*, unsigned long)+0xa8>	
	94	nopl 0x0(%rax,%rax,1)	

95	callq 13d <std::string::insert(unsigned long, char const*, unsigned long)+0x13d> 139: R_X86_64_PLT32 memcpy-0x4	
96	nopl (%rax)	
97	jmpq ac <std::string::insert(unsigned long, char const*, unsigned long)+0xac>	
98	nopl (%rax)	
99	mov %r12,%rdx	
100	mov %rbp,%rdi	
101	mov %r8,0x8(%rsp)	
102	callq 158 <std::string::insert(unsigned long, char const*, unsigned long)+0x158> 154: R_X86_64_PLT32 memcpy-0x4	
103	mov 0x8(%rsp),%r8	
104	jmp fa <std::string::insert(unsigned long, char const*, unsigned long)+0xfa>	
105	lea 0x0(%rip),%rdi # 166 <std::string::insert(unsigned long, char const*, unsigned long)+0x166> 162: R_X86_64_PC32 .LC6-0x4	
106	callq 16b <std::string::insert(unsigned long, char const*, unsigned long)+0x16b> 167: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	
107	lea 0x0(%rip),%rdi # 172 <std::string::insert(unsigned long, char const*, unsigned long)+0x172> 16e: R_X86_64_PC32 .LC6-0x4	
108	callq 177 <.LC13+0x6f> 173: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4	
109	Disassembly of section .text._ZNSt6insertEN9_gnu_cxx17__normal_iteratorIPcSsEESt16initializer_listIcE:	

std::string::insert(unsigned long, std::string const&, unsigned long, unsigned long) (size: 49 -> 67 bytes (+18 bytes))

Old				
f	1	mov (%rdx),%rdx	f	1
	2	mov %rcx,%rax		2
n	3	mov -0x18(%rdx),%r9	n	3
				4
				5
4	mov %r9,%rcx			6
5	sub %rax,%rcx			
6	cmp %r8,%rcx			7
7	cmova %r8,%rcx			8
8	cmp %r9,%rax			9
9	ja 24 <std::string::insert(unsigned long, std::string const&, unsigned long, unsigned long)+0x24>			10
10	add %rax,%rdx			11
t	11	jmpq 24 <std::string::insert(unsigned long, std::string const&, unsigned long, unsigned long)+0x24> 20: R_X86_64_PLT32 std::string::insert(unsigned long, char const*, unsigned long)-0x4	t	12
12	lea 0x0(%rip),%rdi # 2b <std::string::insert(unsigned long, std::string const&, unsigned long, unsigned long)+0x2b> 27: R_X86_64_PC32 .LC6-0x4			13
13	push %rax			14
				15
				16
				17
				18
14	callq 31 <.LC2+0xd> 2d: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4			19
15	Disassembly of section .text._ZNSt6insertEmPKc:			20

std::string::insert(unsigned long, unsigned long, char) (size: 36 -> 51 bytes (+15 bytes))

Old				
f	1	mov (%rdi),%rax	f	1
	2	movsbl %cl,%r8d		2
n	3	cmp -0x18(%rax),%rsi	n	3
				4
4	ja 17 <std::string::insert(unsigned long, unsigned long, char)+0x17>			5
5	mov %rdx,%rcx			6
6	xor %edx,%edx			7
n	7	jmpq 17 <std::string::insert(unsigned long, unsigned long, char)+0x17> 13: R_X86_64_PLT32 std::string::M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4	n	8
8	lea 0x0(%rip),%rdi # 1e <std::string::insert(unsigned long, unsigned long, char)+0x1e> 1a: R_X86_64_PC32 .LC6-0x4			9
9	push %rax			10
t			t	11
				12
				13
10	callq 24 <.LC2> 20: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4			14
11	Disassembly of section .text._ZNSt6insertEN9_gnu_cxx17__normal_iteratorIPcSsEEc:			15

std::string::length() const (size unchanged)

Old		New	
f	1	f	1
	2		2
	3		3
t	4	t	4

std::string::operator+=(char const*) (size unchanged)

Old		New	
f	1	f	1
n		n	2
	2		3
3	mov %rsi,%rdi		4
n	4	n	
5	mov %rsi,%rbx		5

6	sub	\$0x8,%rsp	6	sub	\$0x8,%rsp
7	callq	14 <std::string::operator+=(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4	7	callq	14 <std::string::operator+=(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4
8	add	\$0x8,%rsp	8	add	\$0x8,%rsp
9	mov	%rbx,%rsi	9	mov	%rbx,%rsi
10	mov	%rbp,%rdi	10	mov	%rbp,%rdi
11	pop	%rbx	11	pop	%rbx
12	pop	%rbp	12	pop	%rbp
13	mov	%rax,%rdx	13	mov	%rax,%rdx
t 14	jmpq	28 <.LC2+0x4> 24: R_X86_64_PLT32 std::string::append(char const*, unsigned long)-0x4	t 14	jmpq	28 <.LC73+0x15> 24: R_X86_64_PLT32 std::string::append(char const*, unsigned long)-0x4
15	Disassembly of section .text._ZNSt6appendE				
15	Disassembly of section .text._ZNSt6appendE				

std::string::operator+=(char) (size: 97 -> 103 bytes (+6 bytes))

Old					
f 1	push	%r12	f 1	push	%r12
n			n 2	push	%rbp
2	mov	%esi,%r12d	3	mov	%esi,%r12d
n 3	push	%rbp	n		
4	push	%rbx	4	push	%rbx
n 5	mov	(%rdi),%rax	n 5	mov	(%rdi),%rdx
6	mov	%rdi,%rbx	6	mov	%rdi,%rbx
n 7	mov	-0x18(%rax),%rdx	n 7	mov	-0x18(%rdx),%rax
8	lea	0x1(%rdx),%rbp	8	lea	0x1(%rax),%rbp
9	cmp	-0x10(%rax),%rbp	9	cmp	-0x10(%rdx),%rbp
10	ja	22 <std::string::operator+=(char)+0x22>	10	ja	22 <std::string::operator+=(char)+0x22>
n 11	mov	-0x8(%rax),%ecx	n 11	mov	-0x8(%rdx),%ecx
12	test	%ecx,%ecx	12	test	%ecx,%ecx
13	jle	34 <std::string::operator+=(char)+0x34>	13	jle	34 <std::string::operator+=(char)+0x34>
14	mov	%rbp,%rsi	14	mov	%rbp,%rsi
15	mov	%rbx,%rdi	15	mov	%rbx,%rdi
16	callq	2d <std::string::operator+=(char)+0x2d> 29: R_X86_64_PLT32 std::string::reserve(unsigned long)-0x4	16	callq	2d <std::string::operator+=(char)+0x2d>
n 17	mov	(%rbx),%rax	n 17	mov	(%rbx),%rdx
18	mov	-0x18(%rax),%rdx	18	mov	-0x18(%rdx),%rax
19	mov	%r12b,(%rax,%rdx,1)	19	mov	%r12b,(%rdx,%rax,1)
20	mov	(%rbx),%rax	20	mov	(%rbx),%rax
21	lea	-0x18(%rax),%rdx	21	lea	-0x18(%rax),%rdx
22	cmp	0x0(%rip),%rdx # 46 <std::string::operator+=(char)+0x46> 42: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4	22	cmp	0x0(%rip),%rdx # 46 <std::string
23	jne	50 <std::string::operator+=(char)+0x50>	23	jne	50 <std::string::operator+=(char)+0x50>
24	mov	%rbx,%rax	24	mov	%rbx,%rax
25	pop	%rbx	25	pop	%rbx
26	pop	%rbp	26	pop	%rbp
27	pop	%r12	27	pop	%r12
28	retq		28	retq	
29	movl	\$0x0,-0x8(%rax)	29	movl	\$0x0,-0x8(%rax)
30	mov	%rbp,-0x18(%rax)	30	mov	%rbp,-0x18(%rax)
31	movb	\$0x0,(%rax,%rbp,1)	31	movb	\$0x0,(%rax,%rbp,1)
t 32	jmp	48 <std::string::operator+=(char)+0x48>	t 32	mov	%rbx,%rax
			33	pop	%rbx
			34	pop	%rbp
			35	pop	%r12
			36	retq	
33	Disassembly of section .text._ZNSt4_Rep7_M_grabERKSaIcES2_:				
33	Disassembly of section .text._ZNSt4_Rep7_M_grab				

std::string::operator=(char const*) (size unchanged)

Old			New		
f 1	push	%rbp	f 1	push	%rbp
n			n 2	push	%rbx
2	mov	%rdi,%rbp	3	mov	%rdi,%rbp
3	mov	%rsi,%rdi	4	mov	%rsi,%rdi
n 4	push	%rbx	n		
5	mov	%rsi,%rbx	5	mov	%rsi,%rbx
6	sub	\$0x8,%rsp	6	sub	\$0x8,%rsp
7	callq	14 <std::string::operator=(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4	7	callq	14 <std::string::operator=(char const*)+0x14> 10: R_X86_64_PLT32 strlen-0x4
8	add	\$0x8,%rsp	8	add	\$0x8,%rsp
9	mov	%rbx,%rsi	9	mov	%rbx,%rsi
10	mov	%rbp,%rdi	10	mov	%rbp,%rdi
11	pop	%rbx	11	pop	%rbx
12	pop	%rbp	12	pop	%rbp
13	mov	%rax,%rdx	13	mov	%rax,%rdx
t 14	jmpq	28 <.LC2+0x4> 24: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4	t 14	jmpq	28 <.LC73+0x15> 24: R_X86_64_PLT32 std::string::assign(char const*, unsigned long)-0x4
15	Disassembly of section .text._ZNSt6assignE				
15	Disassembly of section .text._ZNSt6assignE				

std::string::operator[](unsigned long) (size: 40 -> 41 bytes (+1 bytes))

Old			New		
f 1	push	%rbp	f 1	push	%rbp

n	2	mov %rsi,%rbp	n		
	3	push %rbx		2	push %rbx
n	4	mov %rdi,%rbx	n	3	mov %rsi,%rbx
	5	sub \$0x8,%rsp		4	sub \$0x8,%rsp
	6	mov (%rdi),%rax		5	mov (%rdi),%rax
	7	mov -0x8(%rax),%edx		6	mov -0x8(%rax),%edx
	8	test %edx,%edx		7	test %edx,%edx
n	9	js 1e <std::string::operator[](unsigned long)+0x1e>	n	8	js 1f <std::string::operator[](unsigned long)+0x1f>
				9	mov %rdi,%rbp
	10	callq 1b <std::string::operator[](unsigned long)+0x1b> 17: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4		10	callq 1b <std::string::operator[](unsigned long)+0x1b> 17: R_X86_64_PLT32 s
n	11	mov (%rbx),%rax	n	11	mov 0x0(%rbp),%rax
	12	add \$0x8,%rsp		12	add \$0x8,%rsp
t	13	add %rbp,%rax	t	13	add %rbx,%rax
	14	pop %rbx		14	pop %rbx
	15	pop %rbp		15	pop %rbp
	16	retq		16	retq
	17	Disassembly of section .text._ZNSt5S5frontEv:		17	Disassembly of section .text._ZNSt5S5frontEv:

std::string::pop_back() (size: 41 -> 53 bytes (+12 bytes))

Old					
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
n	2	mov -0x18(%rax),%rax	n	2	mov -0x18(%rax),%rcx
	3	lea -0x1(%rax),%rsi		3	lea -0x1(%rcx),%rsi
	4	cmp %rax,%rsi		4	cmp %rcx,%rsi
	5	ja 1c <std::string::pop_back()+0x1c>		5	ja 1c <std::string::pop_back()+0x1c>
	6	xor %ecx,%ecx		6	xor %ecx,%ecx
	7	mov \$0x1,%edx		7	mov \$0x1,%edx
	8	jmpq 1c <std::string::pop_back()+0x1c> 18: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4		8	jmpq 1c <std::string::pop_back()+0x1c> 18: R_X86_64
n	9	lea 0x0(%rip),%rdi # 23 <std::string::pop_back()+0x23> 1f: R_X86_64_PC32 .LC4-0x4	n	9	mov %rsi,%rdx
	10	push %rax		10	push %rax
t			t	11	lea 0x0(%rip),%rsi # 27 <std::string::pop_ba
				12	lea 0x0(%rip),%rdi # 2e <std::string::pop_ba
				13	xor %eax,%eax
	11	callq 29 <.LC2+0x5> 25: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4		14	callq 35 <.LC87+0xc> 31: R_X86_64_PLT32 std::__th
	12	Disassembly of section .text._ZNSt4Rep10_M_refcopyEv:		15	Disassembly of section .text._ZNSt4Rep10_M_refcopyEv:

std::string::push_back(char) (size: 94 -> 100 bytes (+6 bytes))

Old					
f	1	push %r12	f	1	push %r12
n			n	2	push %rbp
	2	mov %esi,%r12d		3	mov %esi,%r12d
n	3	push %rbp	n		
	4	push %rbx		4	push %rbx
	5	mov (%rdi),%rax		5	mov (%rdi),%rax
	6	mov %rdi,%rbx		6	mov %rdi,%rbx
	7	mov -0x18(%rax),%rdx		7	mov -0x18(%rax),%rdx
	8	lea 0x1(%rdx),%rbp		8	lea 0x1(%rdx),%rbp
	9	cmp -0x10(%rax),%rbp		9	cmp -0x10(%rax),%rbp
	10	ja 22 <std::string::push_back(char)+0x22>		10	ja 22 <std::string::push_back(char)+0x22>
	11	mov -0x8(%rax),%ecx		11	mov -0x8(%rax),%ecx
	12	test %ecx,%ecx		12	test %ecx,%ecx
	13	jle 34 <std::string::push_back(char)+0x34>		13	jle 34 <std::string::push_back(char)+0x34>
	14	mov %rbp,%rsi		14	mov %rbp,%rsi
	15	mov %rbx,%rdi		15	mov %rbx,%rdi
	16	callq 2d <std::string::push_back(char)+0x2d> 29: R_X86_64_PLT32 std::string::reserve(unsigned long)-0x4		16	callq 2d <std::string::push_back(char)+0x2d>
	17	mov (%rbx),%rax		17	mov (%rbx),%rax
	18	mov -0x18(%rax),%rdx		18	mov -0x18(%rax),%rdx
	19	mov %r12b,(%rax,%rdx,1)		19	mov %r12b,(%rax,%rdx,1)
	20	mov (%rbx),%rax		20	mov (%rbx),%rax
	21	lea -0x18(%rax),%rdx		21	lea -0x18(%rax),%rdx
	22	cmp 0x0(%rip),%rdx # 46 <std::string::push_back(char)+0x46> 42: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4		22	cmp 0x0(%rip),%rdx # 46 <std::string
n	23	jne 4d <std::string::push_back(char)+0x4d>	n	23	jne 50 <std::string::push_back(char)+0x50>
	24	pop %rbx		24	pop %rbx
	25	pop %rbp		25	pop %rbp
	26	pop %r12		26	pop %r12
	27	retq		27	retq
n			n	28	nopl (%rax)
	28	movl \$0x0,-0x8(%rax)		29	movl \$0x0,-0x8(%rax)
	29	mov %rbp,-0x18(%rax)		30	mov %rbp,-0x18(%rax)
	30	movb \$0x0,(%rax,%rbp,1)		31	movb \$0x0,(%rax,%rbp,1)
t	31	jmp 48 <std::string::push_back(char)+0x48>	t	32	pop %rbx
				33	pop %rbp
				34	pop %r12
				35	retq

32	Disassembly of section .text._ZN5splEc:	36	Disassembly of section .text._ZN5splEc:
----	---	----	---

std::string::rbegin() (size unchanged)

Old			New		
f	1	push %rbp	f	1	push %rbp
n			n	2	push %rbx
	2	mov %rdi,%rbp		3	mov %rdi,%rbp
n	3	push %rbx	n		
	4	mov %rsi,%rbx			
	5	sub \$0x8,%rsp		4	sub \$0x8,%rsp
	6	mov (%rsi),%rdx		5	mov (%rsi),%rdx
	7	mov -0x8(%rdx),%eax		6	mov -0x8(%rdx),%eax
	8	test %eax,%eax		7	test %eax,%eax
	9	js 21 <std::string::rbegin()+0x21>		8	js 21 <std::string::rbegin()+0x21>
n			n	9	mov %rsi,%rbx
	10	mov %rsi,%rdi		10	mov %rsi,%rdi
	11	callq 1e <std::string::rbegin()+0x1e> 1a: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4		11	callq 1e <std::string::rbegin()+0x1e> 1a: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4
	12	mov (%rbx),%rdx		12	mov (%rbx),%rdx
	13	add -0x18(%rdx),%rdx		13	add -0x18(%rdx),%rdx
	14	mov %rbp,%rax		14	mov %rbp,%rax
	15	mov %rdx,0x0(%rbp)		15	mov %rdx,0x0(%rbp)
	16	add \$0x8,%rsp		16	add \$0x8,%rsp
	17	pop %rbx		17	pop %rbx
	18	pop %rbp		18	pop %rbp
	19	retq		19	retq
t	20	Disassembly of section .text._ZN5splEc:	t	20	Disassembly of section .text._ZN5splEc:

std::string::rend() (size unchanged)

Old			New		
f	1	push %rbp	f	1	push %rbp
n			n	2	push %rbx
	2	mov %rdi,%rbp		3	mov %rdi,%rbp
n	3	push %rbx	n		
	4	mov %rsi,%rbx			
	5	sub \$0x8,%rsp		4	sub \$0x8,%rsp
	6	mov (%rsi),%rdx		5	mov (%rsi),%rdx
	7	mov -0x8(%rdx),%eax		6	mov -0x8(%rdx),%eax
	8	test %eax,%eax		7	test %eax,%eax
	9	js 21 <std::string::rend()+0x21>		8	js 21 <std::string::rend()+0x21>
t			t	9	mov %rsi,%rbx
	10	mov %rsi,%rdi		10	mov %rsi,%rdi
	11	callq 1e <std::string::rend()+0x1e> 1a: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4		11	callq 1e <std::string::rend()+0x1e> 1a: R_X86_64_PLT32 std::string::_M_leak_hard()-0x4
	12	mov (%rbx),%rdx		12	mov (%rbx),%rdx
	13	mov %rdx,0x0(%rbp)		13	mov %rdx,0x0(%rbp)
	14	add \$0x8,%rsp		14	add \$0x8,%rsp
	15	mov %rbp,%rax		15	mov %rbp,%rax
	16	pop %rbx		16	pop %rbx
	17	pop %rbp		17	pop %rbp
	18	retq		18	retq
	19	Disassembly of section .text._ZN5splEc:		19	Disassembly of section .text._ZN5splEc:

std::string::replace(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>, char const*) (size: 66 -> 67 bytes (+1 bytes))

Old					
f	1	push %r13	f	1	push
n	2	mov %rdx,%r13	n		
	3	push %r12		2	push
n	4	mov %rcx,%r12	n	3	mov
	5	push %rbp		4	push
n	6	mov %rsi,%rbp	n		
	7	push %rbx		5	push
n	8	mov %rdi,%rbx	n	6	mov
	9	mov %rcx,%rdi		7	mov
n			n	8	mov
				9	mov
	10	sub \$0x8,%rsp		10	sub
	11	callq 1e <std::string::replace(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>, char const*)+0x1e> 1a: R_X86_64_PLT32 strlen-0x4		11	callq
n	12	mov %rbp,%rsi	n	12	mov
				13	mov
				14	mov
	13	sub (%rbx),%rsi		15	sub
				16	sub
	14	add \$0x8,%rsp		17	add
n	15	mov %r13,%rdx	n	18	mov
	16	mov %rbx,%rdi		19	mov

17	mov %r12,%rcx			
18	pop %rbx		20	pop
n	sub %rbp,%rdx	n		
20	mov %rax,%r8			
21	pop %rbp		21	pop
22	pop %r12		22	pop
23	pop %r13		23	pop
t	42 <.LC3+0x8> 3e: R_X86_64_PLT32 std::string::replace(unsigned long, unsigned long, char const*, unsigned long)-0x4	t	24	jmpq
25	Disassembly of section .text._ZN5S7replaceEN9_gnu_cxx17__normal_iteratorIPcSsEES2_S1_S1_:		25	Disa

```
std::string::replace(__gnu_cxx::__normal_iterator<char*, std::string>, __gnu_cxx::__normal_iterator<char*, std::string>, std::string const&) (size unchanged)
```

Old		New	
f	1 mov (%rcx),%rcx	f	1 mov (%rcx),%rcx
	2 sub %rsi,%rdx		2 sub %rsi,%rdx
	3 sub (%rdi),%rsi		3 sub (%rdi),%rsi
	4 mov -0x18(%rcx),%r8		4 mov -0x18(%rcx),%r8
t	5 jmpq 12 <.LC1+0x1> e: R_X86_64_PLT32 std::string::replace(unsigned long, unsigned long, char const*, unsigned long)-0x4	t	5 jmpq 12 <std::string::replace(_gnu_cxx::__normal_iterator<char*, s
			> signed long, unsigned long, char const*, unsigned long)-0x4
	6 Disassembly of section .text._ZN5s7replaceEN9__gnu_cxx17__normal_iteratorIPcSsEE52_PKc:		6 Disassembly of section .text._ZN5s7replaceEN9__gnu_cxx17__normal_iter

```
std::string::replace(unsigned long, unsigned long, char const*) (size unchanged)
```

Old			New		
f	1	push %r13	f	1	push %r13
n			n	2	push %r12
	2	mov %rdx,%r13		3	mov %rdx,%r13
n	3	push %r12	n		
	4	mov %rsi,%r12			
	5	push %rbp		4	push %rbp
n			n	5	push %rbx
	6	mov %rdi,%rbp		6	mov %rdi,%rbp
	7	mov %rcx,%rdi		7	mov %rcx,%rdi
n	8	push %rbx	n	8	mov %rsi,%r12
	9	mov %rcx,%rbx		9	mov %rcx,%rbx
	10	sub \$0x8,%rsp		10	sub \$0x8,%rsp
	11	callq 1e <std::string::replace(unsigned long, unsigned long, char const*)+0x1e> 1a: R_X86_64_PLT32 strlen-0x4		11	callq 1e <std::string::replace(unsigned long, unsigned long, ch
	12	add \$0x8,%rsp		12	add \$0x8,%rsp
	13	mov %rbx,%rcx		13	mov %rbx,%rcx
n			n	14	mov %r13,%rdx
				15	pop %rbx
				16	mov %r12,%rsi
	14	mov %rbp,%rdi		17	mov %rbp,%rdi
n	15	pop %rbx	n	18	mov %rax,%r8
	16	pop %rbp		19	pop %rbp
n	17	mov %r12,%rsi	n		
	18	mov %r13,%rdx			
	19	mov %rax,%r8			
	20	pop %r12		20	pop %r12
	21	pop %r13		21	pop %r13
t	22	jmpq 3c <.LC36+0x2> 38: R_X86_64_PLT32 std::string::replace(unsigned long, unsigned long, char const*, unsigned long)-0x4	t	22	jmpq 3c <.LC36+0x4> 38: R_X86_64_PLT32 std::string::replac
	23	Disassembly of section .text._ZNs7replaceEN9_gnu_cxx17__normal_iteratorIPcSsEES2_PKcm:		23	Disassembly of section .text._ZNs7replaceEN9_gnu_cxx17__normal

```
std::string::replace(unsigned long, unsigned long, char const*, unsigned long) (size: 428 -> 515 bytes (+87 bytes) )
```

Old		
n		
	1	push %r13
n		
	2	push %r12
	3	push %rbp
n	4	mov %rdi,%rbp
	5	push %rbx
n	6	mov %rsi,%rbx
	7	mov %rcx,%rsi
n	8	sub \$0x38,%rsp
	9	mov (%rdi),%rax
	10	mov -0x18(%rax),%rcx
	11	cmp %rcx,%rbx

[illegible]

[illegible]

	>	ring(char const*, unsigned long, std::allocator<char> const&)-0x4
94	mov	0x8(%rsp),%r8
95	mov	0x20(%rsp),%rcx
96	mov	%r12,%rdx
97	mov	%rbx,%rsi
n		
98	mov	%rbp,%rdi
t 99	callq	142 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x142> 13e: R_X86_64_PLT32 std::string::_M_replace_safe(unsigned long, unsigned long, char const*, unsigned long)-0x4
>		
100	mov	0x20(%rsp),%rbx
101	lea	-0x18(%rbx),%rdi
102	cmp	0x0(%rip),%rdi # 152 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x152> 14e: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4
103	je	5f <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x5f>
104	mov	%r13,%rsi
105	mov	%rax,0x8(%rsp)
106	callq	165 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x165> 161: R_X86_64_PC32 .text.unlikely._ZNSt4Rep10_M_disposeERKSaIcE.part.12-0x4
107	mov	0x8(%rsp),%rax
108	jmpq	5f <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x5f>
109	lea	0x0(%rip),%rdi # 176 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x176> 172: R_X86_64_PC32 .LC7-0x4
110	callq	17b <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x17b> 177: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4
111	lea	0x0(%rip),%rdi # 182 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x182> 17e: R_X86_64_PC32 .LC7-0x4
112	callq	187 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x187> 183: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4
113	mov	%rax,%rbx
114	mov	0x20(%rsp),%rax
115	lea	-0x18(%rax),%rdi
116	cmp	0x0(%rip),%rdi # 19a <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x19a> 196: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4
117	je	1a4 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x1a4>
118	mov	%r13,%rsi
119	callq	1a4 <std::string::replace(unsigned long, unsigned long, char const*, unsigned long)+0x1a4> 1a0: R_X86_64_PC32 .text.unlikely._ZNSt4Rep10_M_disposeERKSaIcE.part.12-0x4
120	mov	%rbx,%rdi
121	callq	1ac <.LC13+0xa4> 1a8: R_X86_64_PLT32 _Unwind_Resume-0x4
122	Disassembly of section .text._ZNSt7replaceEmmRKSS:	

std::string::replace(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long) (size: 49 -> 67 bytes (+18 bytes))

Old		
f 1	mov	(%rcx),%rcx
2	mov	%r8,%rax
n 3	mov	-0x18(%rcx),%r10
4	mov	%r10,%r8
5	sub	%rax,%r8
6	cmp	%r9,%r8
7	cmova	%r9,%r8
8	cmp	%r10,%rax
9	ja	24 <std::string::replace(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long)+0x24>
10	add	%rax,%rcx
t 11	jmpq	24 <std::string::replace(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long)+0x24> 20: R_X86_64_PLT32 std::string::replace(unsigned long, unsigned long, char const*, unsigned long)-0x4
>		
12	lea	0x0(%rip),%rdi # 20 <std::string::replace(unsigned long, unsigned long, std::string const&, unsigned long, unsigned long)+0x20> 27: R_X86_64_PC32 .LC7-0x4
13	push	%rax
14	callq	31 <.LC2+0xd> 2d: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4
15	Disassembly of section .text._ZNSt7replaceEmmPKc:	

std::string::replace(unsigned long, unsigned long, unsigned long, char) (size: 47 -> 62 bytes (+15 bytes))

Old		
f 1	mov	(%rdi),%rax
n		
2	movsbl	%r8b,%r8d
n 3	mov	-0x18(%rax),%rax

4	mov	%rax,%r9	
5	sub	%rsi,%r9	
6	cmp	%rdx,%r9	
7	cmovbe	%r9,%rdx	
8	cmp	%rax,%rsi	
9	ja	22 <std::string::replace(unsigned long, unsigned long, unsigned long, char)+0x22>	
10	jmpq	22 <std::string::replace(unsigned long, unsigned long, unsigned long, char)+0x22> 1e: R_X86_64_PLT32 std::string::_M_replace_aux(unsigned long, unsigned long, unsigned long, char)-0x4	
n 11	lea	0x0(%rip),%rdi # 29 <std::string::replace(unsigned long, unsigned long, unsigned long, char)+0x29> 25: R_X86_64_PC32 .LC7-0x4	n
12	push	%rax	
t			t
13	callq	2f <.LC2+0xb> 2b: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4	
14	Disassembly of section .text._ZN5s7replaceEN9_gnu_cxx17__normal_iteratorIPcSsEES2_mc:		

std::string::reserve(unsigned long) (size: 109 -> 194 bytes (+85 bytes))

Old					
f	1	push %rbp	f	1	push %rbp
	2	push %rbx		2	push %rbx
	3	mov %rdi,%rbx		3	mov %rdi,%rbx
	4	sub \$0x28,%rsp		4	sub \$0x28,%rsp
n	5	mov (%rdi),%rax	n	5	mov (%rdi),%rdi
				6	mov %fs:0x28,%rax
				7	mov %rax,0x18(%rsp)
				8	xor %eax,%eax
	6	cmp -0x10(%rax),%rsi		9	cmp -0x10(%rdi),%rsi
	7	je 50 <std::string::reserve(unsigned long)+0x50>		10	je 70 <std::string::reserve(unsigned long)+0x70>
	8	mov -0x18(%rax),%rcx		11	mov -0x18(%rdi),%rax
				12	lea 0x17(%rsp),%rbp
				13	cmp %rax,%rsi
				14	cmovb %rax,%rsi
				15	sub \$0x18,%rdi
	9	mov %rsi,%rdx		16	mov %rsi,%rdx
n	10	lea 0x1f(%rsp),%rbp	n		
	11	lea -0x18(%rax),%rdi			
	12	cmp %rcx,%rsi			
	13	mov %rbp,%rsi		17	mov %rbp,%rsi
n	14	cmovb %rcx,%rdx	n		
	15	sub %rcx,%rdx		18	sub %rax,%rdx
	16	callq 34 <std::string::reserve(unsigned long)+0x34> 30: R_X86_64_PLT32 std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)-0x4		19	callq 44 <std::string::reserve(unsigned long)+0x44> 3e: R_X86_64_PLT32 std::string::_Rep::_M_clone(std::allocator<char> const&, unsigned long)-0x4
	17	mov (%rbx),%rcx		20	mov (%rbx),%rdx
	18	lea -0x18(%rcx),%rdi		21	lea -0x18(%rdx),%rdi
	19	cmp 0x0(%rip),%rdi # 42 <std::string::reserve(unsigned long)+0x42> 3e: R_X86_64_REX_GOTPCRELX std::string::_Rep::_S_empty_rep_storage-0x4		22	cmp 0x0(%rip),%rdi # 52 <std::string::reserve(unsigned long)+0x52>
	20	jne 59 <std::string::reserve(unsigned long)+0x59>		23	jne 80 <std::string::reserve(unsigned long)+0x80>
	21	mov %rax,(%rbx)		24	mov %rax,(%rbx)
n			n	25	mov 0x18(%rsp),%rax
				26	xor %fs:0x28,%rax
				27	jne bd <std::string::reserve(unsigned long)+0xbd>
	22	add \$0x28,%rsp		28	add \$0x28,%rsp
	23	pop %rbx		29	pop %rbx
	24	pop %rbp		30	pop %rbp
	25	retq		31	retq
	26	xchg %ax,%ax		32	xchg %ax,%ax
n	27	mov -0x8(%rax),%edx	n	33	mov -0x8(%rdi),%eax
				34	test %eax,%eax
				35	jg 22 <std::string::reserve(unsigned long)+0x22>
				36	jmp 57 <std::string::reserve(unsigned long)+0x57>
				37	nopl 0x0(%rax)
				38	cmpq \$0x0,0x0(%rip) # 88 <std::string::reserve(unsigned long)+0x88>
				39	je b0 <std::string::reserve(unsigned long)+0xb0>
				40	mov \$0xffffffff,%ecx
				41	lock xadd %ecx,-0x8(%rdx)
				42	mov %ecx,%edx
	28	test %edx,%edx		43	test %edx,%edx
n	29	jg 12 <std::string::reserve(unsigned long)+0x12>	n	44	jg 54 <std::string::reserve(unsigned long)+0x54>
	30	jmp 47 <std::string::reserve(unsigned long)+0x47>			
	31	mov %rbp,%rsi		45	mov %rbp,%rsi
	32	mov %rax,0x8(%rsp)		46	mov %rax,0x8(%rsp)
n	33	callq 66 <std::string::reserve(unsigned long)+0x66> 62: R_X86_64_PC32 .text.unlikely._ZN5s4_Rep10_M_disposeERKSaIcE.part.12-0x4	n	47	callq a7 <std::string::reserve(unsigned long)+0xa7> 6e: R_X86_64_PC32 .text.unlikely._ZN5s4_Rep10_M_disposeERKSaIcE.part.12-0x4
	34	mov 0x8(%rsp),%rax		48	mov 0x8(%rsp),%rax
t	35	jmp 44 <std::string::reserve(unsigned long)+0x44>	t	49	jmp 54 <std::string::reserve(unsigned long)+0x54>
				50	xchg %ax,%ax
				51	mov -0x8(%rdx),%ecx

			52	lea -0x1(%rcx),%esi
			53	mov %esi,-0x8(%rdx)
			54	mov %ecx,%edx
			55	jmp 96 <std::string::reserve(unsigned long, char*)>
			56	callq c2 <.LC145+0x10> be: R_X86_64_PLT32
36	Disassembly of section .text._ZNSt13shrink_to_fitEv:		57	Disassembly of section .text._ZNSt13shrink_to_fitEv:

std::string::resize(unsigned long, char) (size: 80 -> 72 bytes (-8 bytes) *)

Old					
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
	2	movabs \$0x3fffffff9,%rcx		2	movabs \$0x3fffffff9,%rcx
	3	cmp %rcx,%rsi		3	cmp %rcx,%rsi
	4	mov -0x18(%rax),%rax		4	mov -0x18(%rax),%rax
n	5	ja 43 <std::string::resize(unsigned long, char)+0x43>	n	5	ja 3b <std::string::resize(unsigned long, char)+0x3b>
	6	cmp %rax,%rsi		6	cmp %rax,%rsi
n	7	ja 38 <std::string::resize(unsigned long, char)+0x38>	n	7	ja 3b <std::string::resize(unsigned long, char)+0x3b>
	8	jb 28 <std::string::resize(unsigned long, char)+0x28>		8	jb 2b <std::string::resize(unsigned long, char)+0x2b>
	9	nopl (%rax)			
	10	repz retq		9	repz retq
n	11	nopw 0x0(%rax,%rax,1)	n	10	nop
	12	sub %rsi,%rax		11	sub %rsi,%rax
	13	xor %ecx,%ecx		12	xor %ecx,%ecx
	14	mov %rax,%rdx		13	mov %rax,%rdx
n	15	jmpq 35 <std::string::resize(unsigned long, char)+0x35> 31: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4	n	14	jmpq 2d <std::string::resize(unsigned long, char)+0x2d> 29: R_X86_64_PLT32 std::string::_M_mutate(unsigned long, unsigned long, unsigned long)-0x4
	16	nopl (%rax)		15	nopl (%rax)
	17	movsbl %dl,%edx		16	movsbl %dl,%edx
	18	sub %rax,%rsi		17	sub %rax,%rsi
n	19	jmpq 43 <std::string::resize(unsigned long, char)+0x43> 3f: R_X86_64_PLT32 std::string::append(unsigned long, char)-0x4	n	18	jmpq 3b <std::string::resize(unsigned long, char)+0x3b> 2f: R_X86_64_PLT32 std::string::append(unsigned long, char)-0x4
	20	lea 0x0(%rip),%rdi # 4a <std::string::resize(unsigned long, char)+0x4a> 46: R_X86_64_PC32 .LC10-0x4		19	lea 0x0(%rip),%rdi # 42 <std::string::resize(unsigned long, char)+0x42> 46: R_X86_64_PC32 .LC10-0x4
	21	push %rax		20	push %rax
t	22	callq 50 <.LC3+0x16> 4c: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4	t	21	callq 48 <.LC10+0x7> 44: R_X86_64_PLT32 std::__throw_length_error(char const*)-0x4
	23	Disassembly of section .text._ZNSt6resizeEm:		22	Disassembly of section .text._ZNSt6resizeEm:

std::string::rfind(char const*, unsigned long) const (size unchanged)

Old			New		
f	1	push %r12	f	1	push %r12
n			n	2	push %rbp
	2	mov %rdx,%r12		3	mov %rdx,%r12
n	3	push %rbp	n	4	push %rbx
	4	mov %rdi,%rbp		5	mov %rdi,%rbp
n			n	6	mov %rsi,%rbx
	5	mov %rsi,%rdi		7	mov %rsi,%rdi
n	6	push %rbx	n		
	7	mov %rsi,%rbx			
	8	callq 15 <std::string::rfind(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4		8	callq 15 <std::string::rfind(char const*, unsigned long) const+0x15> 11: R_X86_64_PLT32 strlen-0x4
n			n	9	mov %r12,%rdx
	9	mov %rbx,%rsi		10	mov %rbx,%rsi
	10	mov %rbp,%rdi		11	mov %rbp,%rdi
n	11	mov %r12,%rdx	n		
	12	pop %rbx		12	pop %rbx
	13	pop %rbp		13	pop %rbp
	14	pop %r12		14	pop %r12
	15	mov %rax,%rcx		15	mov %rax,%rcx
t	16	jmpq 2a <.LC2+0x6> 26: R_X86_64_PLT32 std::string::rfind(char const*, unsigned long, unsigned long) const-0x4	t	16	jmpq 2a <.LC87+0x1> 26: R_X86_64_PLT32 std::string::rfind(char const*, unsigned long, unsigned long) const-0x4
	17	Disassembly of section .text._ZNKSt5rfindEcm:		17	Disassembly of section .text._ZNKSt5rfindEcm:

std::string::rfind(char const*, unsigned long, unsigned long) const (size: 130 -> 146 bytes (+16 bytes))

Old			New		
f	1	push %r13	f	1	push %r13
n			n	2	push %r12
	2	mov \$0xffffffffffffff,%rax		3	mov \$0xffffffffffffff,%rax
n	3	push %r12	n		
	4	push %rbp		4	push %rbp
n	5	mov %rcx,%rbp	n		
	6	push %rbx		5	push %rbx
	7	sub \$0x8,%rsp		6	sub \$0x8,%rsp
	8	mov (%rdi),%r12		7	mov (%rdi),%r12
n	9	mov -0x18(%r12),%rbx	n	8	mov -0x18(%r12),%rdi
	10	cmp %rbx,%rcx		9	cmp %rdi,%rcx
	11	jbe 30 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x30>		10	jbe 30 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x30>
	12	add \$0x8,%rsp		11	add \$0x8,%rsp
	13	pop %rbx		12	pop %rbx
	14	pop %rbp		13	pop %rbp
	15	pop %r12		14	pop %r12

	16	pop %r13		15	pop %r13
	17	retq		16	retq
	18	nopl 0x0(%rax)		17	nopl 0x0(%rax)
n	19	sub %rcx,%rbx	n	18	sub %rcx,%rdi
				19	mov %rcx,%rbp
	20	mov %rsi,%r13		20	mov %rsi,%r13
n	21	cmp %rdx,%rbx	n	21	cmp %rdx,%rdi
				22	cmovbe %rdi,%rdx
	22	cmova %rdx,%rbx		23	mov %rdx,%rbx
	23	jmp 4c <std::string::rfind(char const*, unsigned long, unsigned long) const+0x4c>		24	jmp 63 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x63>
	24	nop		25	nopl (%rax)
				26	mov %rbp,%rdx
				27	mov %r13,%rsi
				28	callq 53 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x53>
				29	test %eax,%eax
				30	je 6c <std::string::rfind(char const*, unsigned long, unsigned long) const+0x6c>
	25	test %rbx,%rbx		31	test %rbx,%rbx
	26	lea -0x1(%rbx),%rax		32	lea -0x1(%rbx),%rax
n	27	je 70 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x70>	n	33	je 80 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x80>
	28	mov %rax,%rbx		34	mov %rax,%rbx
n			n	35	test %rbp,%rbp
	29	lea (%r12,%rbx,1),%rdi		36	lea (%r12,%rbx,1),%rdi
n	30	mov %rbp,%rdx	n		
	31	mov %r13,%rsi			
	32	callq 5b <std::string::rfind(char const*, unsigned long, unsigned long) const+0x5b> 57: R_X86_64_PLT32 memcmp-0x4			
	33	test %eax,%eax			
	34	jne 40 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x40>		37	jne 48 <std::string::rfind(char const*, unsigned long, unsigned long) const+0x48>
	35	add \$0x8,%rsp		38	add \$0x8,%rsp
	36	mov %rbx,%rax		39	mov %rbx,%rax
	37	pop %rbx		40	pop %rbx
	38	pop %rbp		41	pop %rbp
	39	pop %r12		42	pop %r12
	40	pop %r13		43	pop %r13
	41	retq		44	retq
t	42	nopl (%rax)	t	45	nopw 0x0(%rax,%rax,1)
	43	add \$0x8,%rsp		46	add \$0x8,%rsp
	44	mov \$0xffffffffffffffff,%rax		47	mov \$0xffffffffffffffff,%rax
	45	pop %rbx		48	pop %rbx
	46	pop %rbp		49	pop %rbp
	47	pop %r12		50	pop %r12
	48	pop %r13		51	pop %r13
	49	retq		52	retq
	50	Disassembly of section .text._ZNKS5rfindERKSsm:		53	Disassembly of section .text._ZNKS5rfindERKSsm:

std::string::rfind(char, unsigned long) const (size: 61 -> 62 bytes (+1 bytes))

Old			New		
f	1	mov (%rdi),%rcx	f	1	mov (%rdi),%rcx
	2	mov \$0xffffffffffffffff,%rax		2	mov \$0xffffffffffffffff,%rax
	3	mov -0x18(%rcx),%rdi		3	mov -0x18(%rcx),%rdi
	4	test %rdi,%rdi		4	test %rdi,%rdi
	5	jne 18 <std::string::rfind(char, unsigned long) const+0x18>		5	jne 18 <std::string::rfind(char, unsigned long) const+0x18>
	6	repz retq		6	repz retq
	7	nopl (%rax)		7	nopl (%rax)
	8	add %rdi,%rax		8	add %rdi,%rax
n	9	cmp %rax,%rdx	n	9	cmp %rdx,%rax
	10	cmovbe %rdx,%rax		10	cmova %rdx,%rax
	11	jmp 35 <std::string::rfind(char, unsigned long) const+0x35>		11	jmp 36 <std::string::rfind(char, unsigned long) const+0x36>
	12	nopl 0x0(%rax)		12	nopl 0x0(%rax)
n	13	cmp %sil,%rcx,%rax,1	n		
	14	lea -0x1(%rax),%rdx		13	lea -0x1(%rax),%rdx
t			t	14	cmp %sil,0x1(%rcx,%rdx,1)
	15	je 13 <std::string::rfind(char, unsigned long) const+0x13>		15	je 13 <std::string::rfind(char, unsigned long) const+0x13>
	16	mov %rdx,%rax		16	mov %rdx,%rax
	17	cmp \$0xffffffffffffffff,%rax		17	cmp \$0xffffffffffffffff,%rax
	18	jne 28 <std::string::rfind(char, unsigned long) const+0x28>		18	jne 28 <std::string::rfind(char, unsigned long) const+0x28>
	19	repz retq		19	repz retq
	20	Disassembly of section .text._ZNKSs13find_first_ofEPKcmm:		20	Disassembly of section .text._ZNKSs13find_first_ofEPKcmm:

std::string::shrink_to_fit() (size unchanged)

Old			New		
f	1	mov (%rdi),%rax	f	1	mov (%rdi),%rax
n	2	mov -0x18(%rax),%rdx	n	2	mov -0x10(%rax),%rdx
	3	cmp %rdx,-0x10(%rax)		3	cmp %rdx,-0x18(%rax)

4	ja	10	<std::string::shrink_to_fit()+0x10>	4	jbe	10	<std::string::shrink_to_fit()+0x10>
5	retq			5	retq		
6	xchg	%ax,%ax		6	xchg	%ax,%ax	
7	sub	\$0x8,%rsp		7	sub	\$0x8,%rsp	
8	xor	%esi,%esi		8	xor	%esi,%esi	
9	callq	1b	<std::string::shrink_to_fit()+0x1b> 17: R_X86_64_PLT32 std::string::reserve(unsigned long)-0x4	9	callq	1b	<std::string::shrink_to_fit()+0x1b> 17: R_X86_64_PLT32 std::string
10	add	\$0x8,%rsp		10	add	\$0x8,%rsp	
11	retq			11	retq		
12	mov	%rax,%rdi		12	mov	%rax,%rdi	
13	callq	28	<std::string::shrink_to_fit()+0x28> 24: R_X86_64_PLT32 __cxa_begin_catch-0x4	13	callq	28	<std::string::shrink_to_fit()+0x28> 24: R_X86_64_PLT32 __cxa_begin
14	pop	%rax		14	pop	%rax	
t 15	jmpq	2e	<.LC2+0xa> 2a: R_X86_64_PLT32 __cxa_end_catch-0x4	t 15	jmpq	2e	<.LC87+0x5> 2a: R_X86_64_PLT32 __cxa_end_catch-0x4
16	Disassembly of section .text._ZN5s6appendERKSs:			16	Disassembly of section .text._ZN5s6appendERKSs:		

std::string::size() const (size unchanged)

Old			New		
f 1	mov	(%rdi),%rax	f 1	mov	(%rdi),%rax
2	mov	-0x18(%rax),%rax	2	mov	-0x18(%rax),%rax
3	retq		3	retq	
t 4	Disassembly of section .text._ZNKS8max_sizeEv:				
t 4	Disassembly of section .text._ZNKS8lengthEv:				

std::string::substr(unsigned long, unsigned long) const (size: 35 -> 50 bytes (+15 bytes))

Old		
n		
1	mov	(%rsi),%rax
n 2	push	%rbx
3	cmp	-0x18(%rax),%rdx
4	ja	17 <std::string::substr(unsigned long, unsigned long) const+0x17>
5	mov	%rdi,%rbx
n 6	callq	12 <std::string::substr(unsigned long, unsigned long) const+0x12> e: R_X86_64_PLT32 std::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string(std::string const&
>	signed long, unsigned long)-0x4	
7	mov	%rbx,%rax
8	pop	%rbx
9	retq	
t 10	lea	0x0(%rip),%rdi # 1e <std::string::substr(unsigned long, unsigned long) const+0x1e> 1a: R_X86_64_PC32 .LC13-0x4
11	callq	23 <.LC1+0x12> 1f: R_X86_64_PLT32 std::__throw_out_of_range(char const*)-0x4
12	Disassembly of section .text._ZN5sC2ERKS5mmRKSaIcE:	

Symbols Disappeared

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]: 44 bytes

```
cmpq    $0x0,0x0(%rip)          # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8>
lea     0x10(%rdi),%rax
je      19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>
or      $0xffffffff,%edx
lock xadd %edx,(%rax)
mov     %edx,%eax
jmp     22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>
mov     0x10(%rdi),%eax
mov     -0x1(%rax),%edx
mov     %edx,0x10(%rdi)
test    %eax,%eax
jg      2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>
jmpq    2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>
retq
```

Disassembly of section .text._ZN5s4_Rep10_M_disposeERKSaIcE:

3: R_X86_64_GOTPCREL __pthread_key_create-0x5

27: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4

New Symbols

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]: 50 bytes

```
cmpq    $0x0,0x0(%rip)          # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]+0x8>
je      20 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]+0x20>
mov     $0xffffffff,%eax
lock xadd %eax,0x10(%rdi)
test    %eax,%eax
jle     2d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]+0x2d>
repz retq
nopw    0x0(%rax,%rax,1)
mov     0x10(%rdi),%eax
lea     -0x1(%rax),%edx
test    %eax,%eax
mov     %edx,0x10(%rdi)
jg      18 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]+0x18>
jmpq    32 <.LC277+0x15> 2e: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4
```

Disassembly of section .text._ZNKSt7collateIcE10do_compareEPKcS2_S2_:

3: R_X86_64_GOTPCREL __pthread_key_create-0x5

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]: 50 bytes

```
cmpq    $0x0,0x0(%rip)          # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]+0x8>
je      20 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]+0x20>
mov     $0xffffffff,%eax
lock xadd %eax,0x10(%rdi)
test    %eax,%eax
jle     2d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]+0x2d>
repz retq
nopw    0x0(%rax,%rax,1)
mov     0x10(%rdi),%eax
lea     -0x1(%rax),%edx
test    %eax,%eax
mov     %edx,0x10(%rdi)
```

3: R_X86_64_GOTPCREL __pthread_key_create-0x5

```
jg     18 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]+0x18>
jmpq   32 <_.LC87+0x9>      2e: R_X86_64_PLT32      std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4
Disassembly of section .text._ZN5s4_Rep10_M_disposeERK5aIcE:
```

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]: 50 bytes

```
cmpq   $0x0,0x0(%rip)      # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]+0x8>      3: R_X86_64_GOTPCREL      __pthread_key_create-0x5
je     20 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]+0x20>
mov     $0xffffffff,%eax
lock xadd %eax,0x10(%rdi)
test    %eax,%eax
jle    2d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]+0x2d>
repz retq
nopw    0x0(%rax,%rax,1)
mov     0x10(%rdi),%eax
lea     -0x1(%rax),%edx
test    %eax,%eax
mov     %edx,0x10(%rdi)
jg     18 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]+0x18>
jmpq   32 <_.LC277+0x1>      2e: R_X86_64_PLT32      std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4
Disassembly of section .text._ZN5bIwSt11char_traitsIwE5aIwEE4_Rep10_M_disposeERKS1_.part.14:
```

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]: 39 bytes

```
cmpq   $0x0,0x0(%rip)      # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x8>      3: R_X86_64_GOTPCREL      __pthread_key_create-0x5
je     14 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x14>
or      $0xffffffff,%eax
lock xadd %eax,0x10(%rdi)
jmp     1d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x1d>
mov     0x10(%rdi),%eax
lea     -0x1(%rax),%edx
mov     %edx,0x10(%rdi)
test    %eax,%eax
jg     26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>
jmpq   26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>      22: R_X86_64_PLT32      std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4
retq
```

Disassembly of section .text._ZNKSt13__facet_shims12_GLOBAL__N_114money_put_shimIcE6do_putEst19ostreambuf_iteratorIcSt11char_traitsIcEEbRSt8ios_basecRK5s:

std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]: 39 bytes

```
cmpq   $0x0,0x0(%rip)      # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]+0x8>      3: R_X86_64_GOTPCREL      __pthread_key_create-0x5
je     14 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]+0x14>
or      $0xffffffff,%eax
lock xadd %eax,0x10(%rdi)
jmp     1d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]+0x1d>
mov     0x10(%rdi),%eax
lea     -0x1(%rax),%edx
mov     %edx,0x10(%rdi)
test    %eax,%eax
jg     26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]+0x26>
jmpq   26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.29]+0x26>      22: R_X86_64_PLT32      std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4
retq
```

Disassembly of section .text._ZNSt15basic_stringbufIcSt11char_traitsIcE5aIcEED0Ev:

Similar Symbols

Similar pair 1 (size: 44 -> 39 bytes (-5 bytes) *) (sym. sim.: 99.3 %, instr. sim.: 86.8 %)

Old: [std::string::_Rep::_M_dispose\(std::allocator<char> const&\) \[clone .part.12\]](#)
New: [std::string::_Rep::_M_dispose\(std::allocator<char> const&\) \[clone .part.1\]](#)

Old									
n	1	cmpq \$0x0,0x0(%rip) # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8> 3: R_X86_64_GOTPCREL __pthread_key_create-0x5					n	1	cmpq \$0x0,0x0(%rip)
	2	lea 0x10(%rdi),%rax							
	3	je 19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>						2	je 14 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x14>
	4	or \$0xffffffff,%edx						3	or \$0xffffffff,%edx
	5	lock xadd %edx,(%rax)						4	lock xadd %eax,0x10(%rdi)
	6	mov %edx,%eax							
	7	jmp 22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>						5	jmp 1d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x1d>
	8	mov 0x10(%rdi),%eax						6	mov 0x10(%rdi),%eax
	9	lea -0x1(%rax),%edx						7	lea -0x1(%rax),%edx
	10	mov %edx,0x10(%rdi)						8	mov %edx,0x10(%rdi)
	11	test %eax,%eax						9	test %eax,%eax
n	12	jg 2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>					n	10	jg 26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>
	13	jmpq 2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b> 27: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4						11	jmpq 26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>
	14	retq						12	retq
t	15	Disassembly of section .text._ZN5s4_Rep10_M_disposeERK5aIcE:					t	13	Disassembly of section .text._ZN5bIwSt11char_traitsIwE5aIwEE4_Rep10_M_disposeERKS1_.part.14:

Similar pair 2 (size: 44 -> 39 bytes (-5 bytes) *) (sym. sim.: 98.7 %, instr. sim.: 80.8 %)

Old: [std::string::_Rep::_M_dispose\(std::allocator<char> const&\) \[clone .part.12\]](#)
New: [std::string::_Rep::_M_dispose\(std::allocator<char> const&\) \[clone .part.29\]](#)

Old											
n	1	cmpq	\$0x0,0x0(%rip)	# 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8>	3: R_X86_64_GOTPCREL	__pthread_key_create-0x5	n	1	cmpq	\$0x0,0x0(%rip)	
	2	lea	0x10(%rdi),%rax								
	3	je	19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>					2	je	14 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x14>	
	4	or	\$0xffffffff,%edx					3	or	\$0xffffffff,%edx	
	5	lock xadd	%edx,(%rax)					4	lock xadd	%eax,0x10(%rdi)	
	6	mov	%edx,%eax								
	7	jmp	22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>					5	jmp	1d <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x1d>	
	8	mov	0x10(%rdi),%eax					6	mov	0x10(%rdi),%eax	
	9	lea	-0x1(%rax),%edx					7	lea	-0x1(%rax),%edx	
	10	mov	%edx,0x10(%rdi)					8	mov	%edx,0x10(%rdi)	
	11	test	%eax,%eax					9	test	%eax,%eax	
n	12	jg	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>				n	10	jg	26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>	
	13	jmpq	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>	27: R_X86_64_PLT32	std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4			11	jmpq	26 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.1]+0x26>	
	14	retq						12	retq		
t	15	Disassembly of section .text._ZN5s4_Rep10_M_disposeERK5aIcE:					t	13	Disassembly of section .text._ZN5bIwSt11char_traitsIwE5aIwEE4_Rep10_M_disposeERKS1_.part.14:		

Similar pair 3 (size: 44 -> 50 bytes (+6 bytes)) (sym. sim.: 98.7 %, instr. sim.: 55.5 %)

Old: [std::string::_Rep::_M_dispose\(std::allocator<char> const&\) \[clone .part.12\]](#)

New: std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.13]

Old							
n	1	cmpq	\$0x0,0x0(%rip) # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8> 3: R_X86_64_GOTPCREL __pthread_key_create-0x5	n	1	cmpq	\$0x0,0
	2	lea	0x10(%rdi),%rax				
	3	je	19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>		2	je	20 <st
	4	or	\$0xffffffff,%edx		3	mov	\$0xff
	5	lock xadd %edx,(%rax)			4	lock xadd %e	
	6	mov	%edx,%eax		5	test	%eax,%
	7	jmp	22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>		6	jle	2d <st
					7	repz retq	
					8	nopw	0x0(%
	8	mov	0x10(%rdi),%eax		9	mov	0x10(
	9	lea	-0x1(%rax),%edx		10	lea	-0x1(
n				n	11	test	%eax,%
	10	mov	%edx,0x10(%rdi)		12	mov	%edx,(
t	11	test	%eax,%eax	t			
	12	jg	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>		13	jg	18 <st
	13	jmpq	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b> 27: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4		14	jmpq	32 <.l
	14	retq					
	15	Disassembly of section .text._ZN5s4_Rep10_M_disposeERKSaIcE:			15	Disassembly of	

Similar pair 4 (size: 44 -> 50 bytes (+6 bytes)) (sym. sim.: 98.7 %, instr. sim.: 53.8 %)

Old: std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]
New: std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.15]

Old							
n	1	cmpq	\$0x0,0x0(%rip) # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8> 3: R_X86_64_GOTPCREL __pthread_key_create-0x5	n	1	cmpq	\$0x0,0
	2	lea	0x10(%rdi),%rax				
	3	je	19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>		2	je	20 <st
	4	or	\$0xffffffff,%edx		3	mov	\$0xff
	5	lock xadd %edx,(%rax)			4	lock xadd %e	
	6	mov	%edx,%eax		5	test	%eax,%
	7	jmp	22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>		6	jle	2d <st
					7	repz retq	
					8	nopw	0x0(%
	8	mov	0x10(%rdi),%eax		9	mov	0x10(
	9	lea	-0x1(%rax),%edx		10	lea	-0x1(
n				n	11	test	%eax,%
	10	mov	%edx,0x10(%rdi)		12	mov	%edx,(
t	11	test	%eax,%eax	t			
	12	jg	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>		13	jg	18 <st
	13	jmpq	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b> 27: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4		14	jmpq	32 <.l
	14	retq					
	15	Disassembly of section .text._ZN5s4_Rep10_M_disposeERKSaIcE:			15	Disassembly of	

Similar pair 5 (size: 44 -> 50 bytes (+6 bytes)) (sym. sim.: 98.7 %, instr. sim.: 52.2 %)

Old: std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]
New: std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.10]

Old							
n	1	cmpq	\$0x0,0x0(%rip) # 8 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x8> 3: R_X86_64_GOTPCREL __pthread_key_create-0x5	n	1	cmpq	\$0x0,0
	2	lea	0x10(%rdi),%rax				
	3	je	19 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x19>		2	je	20 <st
	4	or	\$0xffffffff,%edx		3	mov	\$0xff
	5	lock xadd %edx,(%rax)			4	lock xadd %e	
	6	mov	%edx,%eax		5	test	%eax,%
	7	jmp	22 <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x22>		6	jle	2d <st
					7	repz retq	
					8	nopw	0x0(%
	8	mov	0x10(%rdi),%eax		9	mov	0x10(
	9	lea	-0x1(%rax),%edx		10	lea	-0x1(
n				n	11	test	%eax,%
	10	mov	%edx,0x10(%rdi)		12	mov	%edx,(
t	11	test	%eax,%eax	t			
	12	jg	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b>		13	jg	18 <st
	13	jmpq	2b <std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.12]+0x2b> 27: R_X86_64_PLT32 std::string::_Rep::_M_destroy(std::allocator<char> const&)-0x4		14	jmpq	32 <.l
	14	retq			15	Disassembly of	
	15	Disassembly of section .text._ZN5s4_Rep10_M_disposeERKSaIcE:					