

Text Compare

Produced: 2021-02-12 05:26:20 AM

Mode: All, Ignoring Unimportant

Left file: A:\GNU-ld-for-MicrosoftCOFF-to-LinuxELF2\ldBugImageBase\program_a.elf.dis

Right file: A:\GNU-ld-for-MicrosoftCOFF-to-LinuxELF2\ldBugImageBase\program_a.exe.dis

		++ Microsoft (R) COFF/PE Dumper Version 14.28.29336.0 Copyright (C) Microsoft Corporation. All rights reserved.	
		=	
program.elf: file format elf64-x86-64		<> Dump of file program.exe	
		=	
Disassembly of section .text\$mn:		<> File Type: EXECUTABLE IMAGE	
		=	
000000000401000 <main>:		<>	
401000:	48 89 5c 24 08 mov QWORD PTR [rsp+0x8],rbx	0000000140001000:	48 89 5c 24 08 mov qword ptr [rsp+8],rbx
401005:	48 89 6c 24 10 mov QWORD PTR [rsp+0x10],rbp	0000000140001005:	48 89 6c 24 10 mov qword ptr [rsp+10h],rbp
40100a:	48 89 74 24 20 mov QWORD PTR [rsp+0x20],rsi	000000014000100A:	48 89 74 24 20 mov qword ptr [rsp+20h],rsi
40100f:	57 push rdi	000000014000100F:	57 push rdi
401010:	48 83 ec 20 sub rsp,0x20	0000000140001010:	48 83 EC 20 sub rsp,20h
401014:	cc int3	0000000140001014:	90 nop
401015:	8b 05 1d 20 00 00 mov eax,DWORD PTR [rip+0x201d] # 403038 <deadloopvar>	0000000140001015:	8B 05 E5 1F 00 00 mov eax,dword ptr [0000000140003000h]
40101b:	83 f8 01 cmp eax,0x1	000000014000101B:	83 F8 01 cmp eax,1
40101e:	74 f5 je 401015 <main+0x15>	000000014000101E:	74 F5 je 0000000140001015
401020:	0f 31 rdtsc	0000000140001020:	0F 31 rdtsc
401022:	48 c1 e2 20 shl rdx,0x20	0000000140001022:	48 C1 E2 20 shl rdx,20h
401026:	48 0b c2 or rax,rdx	0000000140001026:	48 0B C2 or rax,rdx
401029:	74 5d je 401088 <main+0x88>	0000000140001029:	74 5D je 0000000140001088
40102b:	33 ff xor edi,edi	000000014000102B:	33 FF xor edi,edi
40102d:	48 8d 2d cc ef ff ff lea rbp,[rip+0xffffffffffffefcc] # 400000 <__ImageBase>	000000014000102D:	48 8D 2D CC EF FF lea rbp,[0000000140000000h]
		FF	
401034:	33 db xor ebx,ebx	0000000140001034:	33 DB xor ebx,ebx
401036:	48 8d 35 ff 1f 00 00 lea rsi,[rip+0x1fff] # 40303c <deadloopvar+0x4>	0000000140001036:	48 8D 35 C7 1F 00 lea rsi,[0000000140003004h]
		00	
40103d:	48 8b 8c 2b 68 30 40 mov rcx,QWORD PTR [rbx+rbp*1+0x403068]	000000014000103D:	48 8B 8C 2B 18 30 mov rcx,qword ptr [rbx+rbp+000000000003018h]
401044:	00	000000014000103D:	00 00
401045:	44 8a 01 mov r8b,BYTE PTR [rcx]	0000000140001045:	44 8A 01 mov r8b,byte ptr [rcx]
401048:	45 84 c0 test r8b,r8b	0000000140001048:	45 84 C0 test r8b,r8b
40104b:	74 28 je 401075 <main+0x75>	000000014000104B:	74 28 je 0000000140001075
40104d:	b8 05 00 00 00 mov eax,0x5	000000014000104D:	B8 05 00 00 00 mov eax,5
401052:	2b 84 2b 58 30 40 00 sub eax,DWORD PTR [rbx+rbp*1+0x403058]	0000000140001052:	2B 84 2B 10 30 00 sub eax,dword ptr [rbx+rbp+000000000003010h]
		00	
401059:	99 cdq	0000000140001059:	99 cdq
40105a:	2b c2 sub eax,edx	000000014000105A:	2B C2 sub eax,edx
40105c:	d1 f8 sar eax,1	000000014000105C:	D1 F8 sar eax,1
40105e:	48 63 d0 movsxd rdx,eax	000000014000105E:	48 63 D0 movsxd rdx,eax
401061:	48 03 d6 add rdx,rsi	0000000140001061:	48 03 D6 add rdx,rsi
401064:	48 ff c1 inc rcx	0000000140001064:	48 FF C1 inc rcx
401067:	44 88 02 mov BYTE PTR [rdx],r8b	0000000140001067:	44 88 02 mov byte ptr [rdx],r8b
40106a:	48 ff c2 inc rdx	000000014000106A:	48 FF C2 inc rdx
40106d:	44 8a 01 mov r8b,BYTE PTR [rcx]	000000014000106D:	44 8A 01 mov r8b,byte ptr [rcx]
401070:	45 84 c0 test r8b,r8b	0000000140001070:	45 84 C0 test r8b,r8b
401073:	75 ef jne 401064 <main+0x64>	0000000140001073:	75 EF jne 0000000140001064
401075:	48 8b ce mov rcx,rsi	0000000140001075:	48 8B CE mov rcx,rsi
401078:	e8 2f 00 00 00 call 4010ac <xstring>	0000000140001078:	E8 2F 00 00 00 call 00000001400010AC
40107d:	ff c7 inc edi	000000014000107D:	FF C7 inc edi
40107f:	48 83 c3 08 add rbx,0x8	000000014000107F:	48 83 C3 08 add rbx,8
401083:	83 ff 01 cmp edi,0x1	0000000140001083:	83 FF 01 cmp edi,1
401086:	72 b5 jb 40103d <main+0x3d>	0000000140001086:	72 B5 jb 000000014000103D
401088:	b1 aa mov cl,0xaa	0000000140001088:	B1 AA mov cl,0AAh
40108a:	e8 19 00 00 00 call 4010a8 <xfunc>	000000014000108A:	E8 19 00 00 00 call 00000001400010A8
40108f:	48 8b 5c 24 30 mov rbx,QWORD PTR [rsp+0x30]	000000014000108F:	48 8B 5C 24 30 mov rbx,qword ptr [rsp+30h]
401094:	33 c0 xor eax,eax	0000000140001094:	33 C0 xor eax,eax
401096:	48 8b 6c 24 38 mov rbp,QWORD PTR [rsp+0x38]	0000000140001096:	48 8B 6C 24 38 mov rbp,qword ptr [rsp+38h]
40109b:	48 8b 74 24 48 mov rsi,QWORD PTR [rsp+0x48]	000000014000109B:	48 8B 74 24 48 mov rsi,qword ptr [rsp+48h]
4010a0:	48 83 c4 20 add rsp,0x20	00000001400010A0:	48 83 C4 20 add rsp,20h
4010a4:	5f pop rdi	00000001400010A4:	5F pop rdi
4010a5:	c3 ret	00000001400010A5:	C3 ret
4010a6:	66 90 xchg ax,ax	00000001400010A6:	CC int 3
		=	
0000000004010a8 <xfunc>:		<>	
4010a8:	66 90 xchg ax,ax	00000001400010A7:	CC int 3
4010aa:	cc int3	00000001400010A8:	66 90 xchg ax,ax
		00000001400010AA:	CC int 3

4010ab: c3 ret	=	
00000000004010ac <xstring>:	<>	00000001400010AB: C3 ret
4010ac: 40 53 rex push rbx		00000001400010AC: 40 53 push rbx
4010ae: 48 83 ec 20 sub rsp,0x20		00000001400010AE: 48 83 EC 20 sub rsp,20h
4010b2: 8a 01 mov al,BYTE PTR [rcx]		00000001400010B2: 8A 01 mov al,byte ptr [rcx]
4010b4: 48 8b d9 mov rbx,rcx		00000001400010B4: 48 8B D9 mov rbx,rcx
4010b7: eb 0c jmp 4010c5 <xstring+0x19>		00000001400010B7: EB 0C jmp 00000001400010C5
4010b9: 8a c8 mov cl,al		00000001400010B9: 8A C8 mov cl,al
4010bb: e8 e8 ff ff ff call 4010a8 <xfunc>		00000001400010BB: E8 E8 FF FF FF call 00000001400010A8
4010c0: 48 ff c3 inc rbx		00000001400010C0: 48 FF C3 inc rbx
4010c3: 8a 03 mov al,BYTE PTR [rbx]		00000001400010C3: 8A 03 mov al,byte ptr [rbx]
4010c5: 84 c0 test al,al		00000001400010C5: 84 C0 test al,al
4010c7: 75 f0 jne 4010b9 <xstring+0xd>		00000001400010C7: 75 F0 jne 00000001400010B9
4010c9: 48 83 c4 20 add rsp,0x20		00000001400010C9: 48 83 C4 20 add rsp,20h
4010cd: 5b pop rbx		00000001400010CD: 5B pop rbx
4010ce: c3 ret		00000001400010CE: C3 ret
	=	
	+	Summary
	=	
	+	1000 .data
		1000 .pdata
		1000 .rdata
		1000 .reloc
		1000 .text