システムプログラム第4回レポート

2024年11月6日

学籍番号:35714121

名前:福富隆大

1. 5-15ページの例5.2を実行し、その結果を報告してください

実行結果

長いので最後に記載しています。

作成したファイル内容、結果について

readelf -h combined.o コマンドを実行すると

combined.o: ファイル形式 elf64-x86-64

と出力され、ファイルはELF形式で、64ビットx86-64 アーキテクチャ用ということがわかった。

readelf -s combined.o コマンドを実行すると

```
Symbol table '.symtab' contains 20 entries:
  番号:
                        サイズ タイプ
                                                     索引名
             値
                                      Bind
                                            Vis
    0: 0000000000000000
                            0 NOTYPE LOCAL
                                             DEFAULT
                                                     UND
                            0 SECTION LOCAL
    1: 0000000000000000
                                             DEFAULT
                                                       1
.note.gnu.property
    2: 00000000000000000
                            0 SECTION LOCAL DEFAULT
                                                       2 .text
    3: 0000000000000000
                            0 SECTION LOCAL
                                             DEFAULT
                                                       4 rodata
    4: 0000000000000000
                            0 SECTION LOCAL
                                            DEFAULT
                                                       5 .eh frame
                                                       7 .data
    5: 0000000000000000
                            0 SECTION LOCAL DEFAULT
    6: 00000000000000000
                            0 SECTION LOCAL DEFAULT
                                                       8 bss
                            0 SECTION LOCAL
    7: 00000000000000000
                                            DEFAULT
                                                       9 .comment
    8: 0000000000000000
                            0 SECTION LOCAL DEFAULT
                                                      10 .note.GNU-stack
    9: 0000000000000000
                            0 FILE
                                     LOCAL DEFAULT ABS e4-3.c
   10: 0000000000000000
                            0 FILE LOCAL DEFAULT ABS e4-3m.c
   11: 00000000000000000
                            0 NOTYPE GLOBAL DEFAULT
                                                     UND printf
   12: 0000000000000000
                            0 NOTYPE GLOBAL DEFAULT
                                                     UND puts
                           24 OBJECT GLOBAL DEFAULT
   13: 00000000000000000
                                                       7 a1
   14: 0000000000000000
                            4 OBJECT GLOBAL DEFAULT
                                                       8 i
                            8 OBJECT GLOBAL DEFAULT
   15: 00000000000000008
                                                       8 va
   4 OBJECT GLOBAL DEFAULT
                                                       8 v
   17: 0000000000000004c
                          157 FUNC
                                      GLOBAL DEFAULT
                                                       2 main
   18: 0000000000000000
                            0 NOTYPE GLOBAL DEFAULT
                                                     UND getchar
   19: 0000000000000000
                           76 FUNC
                                      GLOBAL DEFAULT
                                                       2 assign3
```

2. 5-18ページの例5.3を実行し、その結果を報告してください

実行結果

長いので最後に記載しています。

作成したファイル内容、結果について

objdump -s combined.o コマンドを実行するとセクション内容が表示され、 メモリアドレスとそのアドレスに格納されているバイナリデータがわかった。

objdump -D combined.o コマンドを実行するとオブジェクトファイルの内容の逆アセンブルが表示され、 各命令が格納されているメモリアドレスが表示され、プログラムの実行時の流れがわかった。

講義に対する感想・質問・意見

普段プログラミングをする際などに利用するエミュレータの仕組みについてしれて勉強になった。 また、自分の知らないコマンドが大量にあったので、それらも使いこなせるように頑張っていきたい。

実行結果

### rea	adeli	f –ł	1 C	omb	ined	0.1					
combine	ed.o:	:	-	ファ	イル	形式	t elf6	64-x86-6	54		
セクショ	ン・	tex	t σ)逆)	アセン	ンブ	ル:				
000000	aaaaa	2000	aaa	/30	ccia	n 2 -					
	f3				22 Т	J113-		endbr6	54		
4:		01	10	ıu				push	%rbp		
	48	89	e5						%rsp,%rbp		
	8b				00	00			0x0(%rip),%eax	# e	
<assign< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1</td><td></td><td></td></assign<>									1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1		
_	48							cltq			
10:	48	8d	14	85	00	00	00	lea .	0x0(,%rax,4),%rdx		
17:	00										
18:	48	8d	05	00	00	00	00	lea	0x0(%rip),%rax	# 1f	
<assign< td=""><td>n3+0</td><td><1f></td><td>></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></assign<>	n3+0	<1f>	>								
1f:	48	01	d0					add	%rdx,%rax		
22:	48	89	05	00	00	00	00	mov	%rax,0x0(%rip)	# 29	
<assig< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></assig<>											
29:				00	00	00		mov	0x0(%rip),%eax	# 2f	
<assig< td=""><td></td><td></td><td>></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></assig<>			>								
2f:								cltq			
31:		8d	14	85	00	00	00	lea	0x0(,%rax,4),%rdx		
38:								_			
	48			00	00	00	00	lea	0x0(%rip),%rax	# 40	
<assign< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>(0 1 - 0 1) 0</td><td></td><td></td></assign<>									(0 1 - 0 1) 0		
40:				0.0	0.0	0.0			(%rdx,%rax,1),%eax	# 40	
43:				שש	UU	UU		mov	%eax,0x0(%rip)	# 49	
<assign< td=""><td>n3+0) 90</td><td>(49></td><td>></td><td></td><td></td><td></td><td></td><td>non</td><td></td><td></td><td></td></assign<>	n3+0) 90	(49>	>					non			
491	90							nop			

4a: 4b:	5d c3							pop ret	%rbp		
0000000	naaaa	วดด	<i>1</i> c	∠ m:	a i n s	s •					
4c:	f3 () TII	•		endbr64	1		
50:	55	0 1	10	Tu				push	⁺ %rbp		
50:		20	۵5					mov	%rsp,%rbp		
54:				10				sub	\$0x10,%rsp		
58:	48 8				αα	αα	00		•	# 5f	
<main+0< td=""><td></td><td>ou '</td><td>ده</td><td>00</td><td>00</td><td>00</td><td>00</td><td>lea</td><td>0x0(%rip),%rax</td><td># 31</td><td></td></main+0<>		ou '	ده	00	00	00	00	lea	0x0(%rip),%rax	# 31	
		20	~7						Ourant Ourald		
5f:	48 8			0.0	00			mov	%rax,%rdi		
	e8 (67 <main+0x1b></main+0x1b>		
	e8 (00	00				6c <main+0x20></main+0x20>		
	88 4			٠.				mov	%al,-0x1(%rbp)		
	0f k			††					-0x1(%rbp),%eax		
73:								sub	\$0x30,%eax		
76:	89 (Ø5 I	00	00	00	00		mov	%eax,0x0(%rip)	# 7c	
<main+0< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></main+0<>											
7c:	8b (ð5	00	00	00	00		mov	0x0(%rip),%eax	# 82	
<main+0< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></main+0<>											
82:	85 (с0						test	%eax,%eax		
84:	78 (ðb						js	91 <main+0x45></main+0x45>		
86:	8b (ð5	00	00	00	00		mov	0x0(%rip),%eax	# 8c	
<main+0< td=""><td>)x40></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></main+0<>)x40>										
8c:	83 -	f8	09					cmp	\$0x9,%eax		
8f:	7e (ð7						jle	98 <main+0x4c></main+0x4c>		
91:	b8 (01	00	00	00			mov	\$0x1,%eax		
96:	eb 4	4f						jmp	e7 <main+0x9b></main+0x9b>		
98:	8b (0 5	00	00	00	00		mov	0x0(%rip),%eax	# 9e	
<main+0< td=""><td>)x52></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></main+0<>)x52>										
9e:	89 (c6						mov	%eax,%esi		
a0:	48 8	3d	05	00	00	00	00	lea	0x0(%rip),%rax	# a7	
<main+0< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></main+0<>											
a7:	48 8	39	с7					mov	%rax,%rdi		
aa:	b8 (00	00			mov	\$0x0,%eax		
	e8 (call	b4 <main+0x68></main+0x68>		
	e8 (call	b9 <main+0x6d></main+0x6d>		
b9:	8b (00		mov	0x0(%rip),%ecx	# bf	
<main+0< td=""><td></td><td>.</td><td></td><td></td><td></td><td></td><td></td><td></td><td>σπο (σ. <u>-</u>β / / σσσπ</td><td></td><td></td></main+0<>		.							σπο (σ. <u>-</u> β / / σσσπ		
bf:	48 8	3h	15	00	00	00	00	mov	0x0(%rip),%rdx	# c6	
<main+0< td=""><td></td><td></td><td>10</td><td>55</td><td>00</td><td>00</td><td></td><td> v</td><td>one (or ip /) or an</td><td> 00</td><td></td></main+0<>			10	55	00	00		v	one (or ip /) or an	00	
c6:		25	aa	۵۵	aa	aa		mov	0x0(%rip),%eax	# cc	
<main+0< td=""><td></td><td></td><td>50</td><td>00</td><td>00</td><td>00</td><td></td><td>III V</td><td>ond (or ip /) ocan</td><td><i>"</i> CC</td><td></td></main+0<>			50	00	00	00		III V	ond (or ip /) ocan	<i>"</i> CC	
CC:	89 (6						mov	%eax,%esi		
ce:	48 8		0 5	aa	aa	aa	00	lea	0x0(%rip),%rax	# d5	
<main+0< td=""><td></td><td>Ju</td><td>00</td><td>90</td><td>00</td><td>90</td><td>00</td><td>cca</td><td>ovo (101 Th) Lolax</td><td># UJ</td><td></td></main+0<>		Ju	00	90	00	90	00	cca	ovo (101 Th) Lolax	# UJ	
d5:	48 8	20	c7					mov	eray erdi		
				00	00			mov	%rax,%rdi		
d8:	b8 (mov	\$0x0,%eax		
dd:	e8 (call	e2 <main+0x96></main+0x96>		
e2:	b8 (00	טט	טט	UU			mov	\$0x0,%eax		
e7: e8:	c9							leave			
001	с3							ret			

```
Symbol table '.symtab' contains 20 entries:
                        サイズ タイプ
   番号:
             値
                                     Bind
                                            Vis
                                                      索引名
     0: 0000000000000000
                             0 NOTYPE LOCAL DEFAULT UND
                             0 SECTION LOCAL
     1: 00000000000000000
                                               DEFAULT
.note.gnu.property
     2: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          2 text
     3: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          4 .rodata
     4: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          5 .eh frame
     5: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          7 data
     6: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          8 bss
     7: 00000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                          9 comment
     8: 0000000000000000
                             0 SECTION LOCAL
                                               DEFAULT
                                                         10 .note.GNU-stack
     9: 0000000000000000
                             0 FILE
                                       LOCAL
                                               DEFAULT
                                                       ABS e4-3<sub>c</sub>
    10: 0000000000000000
                             0 FILE
                                       LOCAL
                                                        ABS e4-3m<sub>•</sub>c
                                               DEFAULT
    11: 0000000000000000
                             0 NOTYPE GLOBAL DEFAULT
                                                        UND printf
    12: 0000000000000000
                             0 NOTYPE GLOBAL DEFAULT
                                                        UND puts
    13: 0000000000000000
                            24 OBJECT GLOBAL DEFAULT
                                                          7 a1
                                                          8 i
    14: 00000000000000000
                             4 OBJECT GLOBAL DEFAULT
    15: 00000000000000008
                             8 OBJECT GLOBAL DEFAULT
                                                          8 va
    16: 00000000000000010
                             4 OBJECT GLOBAL DEFAULT
                                                          8 v
    17: 0000000000000004c
                           157 FUNC
                                        GLOBAL DEFAULT
                                                          2 main
    18: 00000000000000000
                             0 NOTYPE GLOBAL DEFAULT UND getchar
    19: 0000000000000000
                            76 FUNC
                                     GLOBAL DEFAULT
                                                          2 assign3
### objdump -s combined.o
combined.o:
                ファイル形式 elf64-x86-64
セクション .note.gnu.property の内容:
 0000 04000000 10000000 05000000 474e5500
                                            0010 020000c0 04000000 03000000 00000000
                                            . . . . . . . . . . . . . . . .
セクション .text の内容:
                                            0000 f30f1efa 554889e5 8b050000 00004898
                                            H......
 0010 488d1485 00000000 488d0500 00000048
 0020 01d04889 05000000 008b0500 00000048
                                            ..H..........
 0030 98488d14 85000000 00488d05 00000000
                                            .H.........
 0040 8b040289 05000000 00905dc3 f30f1efa
                                            . . . . . . . . . . . . . . . . . . .
 0050 554889e5 4883ec10 488d0500 00000048
                                            UH..H...H....H
 0060 89c7e800 000000e8 00000000 8845ff0f
                                            ....E...
 0070 be45ff83 e8308905 00000000 8b050000
                                            .E...0......
 0080 000085c0 780b8b05 00000000 83f8097e
                                            ....×............
 0090 07b80100 0000eb4f 8b050000 000089c6
                                            . . . . . . . 0 . . . . . . .
                                            H. . . . . . H. . . . . . .
 00a0 488d0500 00000048 89c7b800 000000e8
                                            00b0 00000000 e8000000 008b0d00 00000048
 00c0 8b150000 00008b05 00000000 89c6488d
                                            . . . . . . . . . . . . . . . H.
 00d0 05000000 004889c7 b8000000 00e80000
                                            . . . . . H. . . . . . . . . .
 00e0 0000b800 000000c9 c3
                                            . . . . . . . . .
セクション rodata の内容:
 0000 456e7465 72206120 64696769 742e0069
                                            Enter a digit..i
 0010 3d25640a 0061315b 25645d3a 20616464
                                            =%d..a1[%d]: add
 0020 723d2538 70207661 6c3d2564 200a00
                                            r=%8p val=%d ...
セクション _eh_frame の内容:
 0000 14000000 00000000 017a5200 01781001
                                            ....zR..x..
```

```
0010 1b0c0708 90010000 1c000000 1c000000
                                           . . . . . . . . . . . . . . . .
                                           ....L....E....C.
 0020 00000000 4c000000 00450e10 8602430d
 0030 0602430c 07080000 14000000 00000000
                                           ..C........
 0040 017a5200 01781001 1b0c0708 90010000
                                           .zR..x......
 0050 1c000000 1c000000 00000000 9d000000
                                           . . . . . . . . . . . . . . . . .
 0060 00450e10 8602430d 0602940c 07080000
                                           .E....C......
セクション .data の内容:
 0000 01000000 02000000 03000000 04000000
 0010 05000000 06000000
セクション comment の内容:
 0000 00474343 3a202855 62756e74 75203132
                                           .GCC: (Ubuntu 12
 0010 2e332e30 2d317562 756e7475 317e3232
                                           .3.0-1ubuntu1~22
 0020 2e303429 2031322e 332e3000 00474343
                                           .04) 12.3.0..GCC
 0030 3a202855 62756e74 75203132 2e332e30 : (Ubuntu 12.3.0
 0040 2d317562 756e7475 317e3232 2e303429 -1ubuntu1~22.04)
 0050 2031322e 332e3000
                                            12.3.0.
### objdump -D combined.o
combined.o: ファイル形式 elf64-x86-64
セクション .note.gnu.property の逆アセンブル:
0000000000000000 <.note.gnu.property>:
   0:
        04 00
                                add
                                       $0x0,%al
   2:
        00 00
                                add
                                       %al,(%rax)
   4:
        10 00
                                       %al,(%rax)
                                adc
   6:
        00 00
                                add
                                       %al,(%rax)
   8:
        05 00 00 00 47
                                       $0x47000000,%eax
                                add
   d:
       4e 55
                                rex.WRX push %rbp
   f:
        00 02
                                add
                                      %al,(%rdx)
  11:
        00 00
                                add
                                       %al,(%rax)
  13:
        c0 04 00 00
                                rolb
                                       $0x0,(%rax,%rax,1)
  17:
        00 03
                                add
                                       %al,(%rbx)
  19:
        00 00
                                add
                                       %al,(%rax)
                                add
  1b:
        00 00
                                       %al,(%rax)
  1d:
        00 00
                                add
                                    %al,(%rax)
        . . .
セクション text の逆アセンブル:
0000000000000000 <assign3>:
       f3 0f 1e fa
                                endbr64
   0:
   4:
        55
                                push
                                       %rbp
   5:
        48 89 e5
                                mov
                                       %rsp,%rbp
        8b 05 00 00 00 00
                                mov
                                       0x0(%rip),%eax
                                                             # e
<assign3+0xe>
                                clta
  e:
       48 98
  10:
        48 8d 14 85 00 00 00
                                lea
                                       0x0(,%rax,4),%rdx
  17:
        00
  18:
       48 8d 05 00 00 00 00
                                lea
                                       0x0(%rip),%rax
                                                              # 1f
<assign3+0x1f>
  1f:
      48 01 d0
                                add
                                       %rdx,%rax
```

22: 48 89 05	00	00	00	00	mov	%rax,0x0(%rip)	# 29
<assign3+0x29></assign3+0x29>		0.0	0.0			00(0	# 2.f
29: 8b 05 00	00	00	00		mov	0x0(%rip),%eax	# 2f
<assign3+0x2f></assign3+0x2f>							
2f: 48 98					cltq		
31: 48 8d 14	85	00	00	00	lea	0x0(,%rax,4),%rdx	
38: 00					_		
39: 48 8d 05	00	00	00	00	lea	0x0(%rip),%rax	# 40
<assign3+0x40></assign3+0x40>							
40: 8b 04 02					mov	(%rdx,%rax,1),%eax	
43: 89 05 00	00	00	00		mov	%eax,0x0(%rip)	# 49
<assign3+0x49></assign3+0x49>							
49: 90					nop		
4a: 5d					pop	%rbp	
4b: c3					ret		
0000000000000040		ain>	>:			4	
4c: f3 0f 16	ta:				endbr64		
50: 55	_				push	%rbp	
51: 48 89 es					mov	%rsp,%rbp	
54: 48 83 ed					sub	\$0x10,%rsp	
58: 48 8d 05	00	00	00	00	lea	0x0(%rip),%rax	# 5f
<main+0x13></main+0x13>							
5f: 48 89 c7	7				mov	%rax,%rdi	
62: e8 00 00	00	00			call	67 <main+0x1b></main+0x1b>	
67: e8 00 00	00	00			call	6c <main+0x20></main+0x20>	
6c: 88 45 ft	-				mov	%al,-0x1(%rbp)	
6f: 0f be 45	ff				movsbl	-0x1(%rbp),%eax	
73: 83 e8 30)				sub	\$0x30,%eax	
76: 89 05 00	00	00	00		mov	%eax,0x0(%rip)	# 7c
<main+0x30></main+0x30>							
7c: 8b 05 00	00	00	00		mov	0x0(%rip),%eax	# 82
<main+0x36></main+0x36>							
82: 85 c0					test	%eax,%eax	
84: 78 0b					js	91 <main+0x45></main+0x45>	
86: 8b 05 00	00	00	00		mov	0x0(%rip),%eax	# 8c
<main+0x40></main+0x40>						1,77	
8c: 83 f8 09)				cmp	\$0x9,%eax	
8f: 7e 07					jle	98 <main+0x4c></main+0x4c>	
91: b8 01 00	00	00			mov	\$0x1,%eax	
96: eb 4f		55			jmp	e7 <main+0x9b></main+0x9b>	
98: 8b 05 00	00	00	00		mov	0x0(%rip),%eax	# 9e
<main+0x52></main+0x52>		00				ond (or ip/) ocur	30
9e: 89 c6					mov	%eax,%esi	
a0: 48 8d 05	. 00	00	۵۵	00	lea	0x0(%rip),%rax	# a7
<main+0x5b></main+0x5b>	00	00	00	00	tea	0/0(-01 1b), -01 ax	# α <i>1</i>
a7: 48 89 c7	7				mov/	%rav %rdi	
		00			mov	%rax,%rdi	
aa: b8 00 00					mov	\$0x0,%eax	
af: e8 00 00					call	b4 <main+0x68></main+0x68>	
b4: e8 00 00			0.0		call	b9 <main+0x6d></main+0x6d>	# b.f
b9: 8b 0d 00	טט נ	00	00		mov	0x0(%rip),%ecx	# bf
<main+0x73></main+0x73>		00	0.0	0.0		00(0	# -C
bf: 48 8b 15	00	00	00	00	mov	0x0(%rip),%rdx	# c6
<main+0x7a></main+0x7a>							

```
c6:
        8b 05 00 00 00 00
                                  mov
                                          0x0(%rip),%eax
                                                                   # cc
< main + 0 \times 80 >
        89 c6
                                          %eax,%esi
  cc:
                                  mov
        48 8d 05 00 00 00 00
                                   lea
                                          0x0(%rip),%rax
                                                                   # d5
  ce:
< main + 0 \times 89 >
  d5:
        48 89 c7
                                          %rax,%rdi
                                   mov
  d8:
        b8 00 00 00 00
                                  mov
                                          $0x0,%eax
                                          e2 < main + 0 \times 96 >
  dd:
        e8 00 00 00 00
                                   call
  e2:
        b8 00 00 00 00
                                   mov
                                          $0x0,%eax
  e7:
        с9
                                   leave
  e8:
        с3
                                   ret
セクション rodata の逆アセンブル:
0000000000000000000000 <.rodata>:
   0:
        45 6e
                                   rex.RB outsb %ds:(%rsi),(%dx)
   2:
        74 65
                                   jе
                                          69 < main + 0 \times 1d >
        72 20
   4:
                                          26 < rodata+0x26>
                                   ib
   6:
                                   (bad)
        61
   7:
        20 64 69 67
                                   and
                                          %ah, 0x67(%rcx,%rbp, 2)
   b:
        69 74 2e 00 69 3d 25
                                  imul
                                          $0x64253d69,0x0(%rsi,%rbp,1),%esi
  12:
        64
  13:
        0a 00
                                          (%rax),%al
                                   or
  15:
        61
                                   (bad)
        31 5b 25
                                          %ebx,0x25(%rbx)
  16:
                                  xor
  19:
        64 5d
                                   fs pop %rbp
  1b:
        3a 20
                                   cmp
                                          (%rax),%ah
  1d:
        61
                                   (bad)
        64 64 72 3d
                                  fs fs jb 5f <main+0x13>
  1e:
  22:
        25 38 70 20 76
                                  and
                                          $0x76207038,%eax
  27:
        61
                                   (bad)
  28:
        6c
                                   insb
                                          (%dx),%es:(%rdi)
  29:
        3d 25 64 20 0a
                                   cmp
                                          $0xa206425,%eax
        . . .
セクション .eh_frame の逆アセンブル:
0000000000000000 <.eh_frame>:
   0:
        14 00
                                          $0x0,%al
                                   adc
   2:
                                          %al,(%rax)
        00 00
                                   add
   4:
        00 00
                                   add
                                          %al,(%rax)
   6:
        00 00
                                   add
                                          %al,(%rax)
   8:
        01 7a 52
                                   add
                                          %edi,0x52(%rdx)
        00 01
                                          %al,(%rcx)
   b:
                                   add
        78 10
   d:
                                   js
                                          1f <.eh_frame+0x1f>
   f:
        01 1b
                                   add
                                          %ebx,(%rbx)
  11:
        0c 07
                                   or
                                          $0x7,%al
  13:
        08 90 01 00 00 1c
                                          %dl,0x1c000001(%rax)
                                   or
  19:
        00 00
                                   add
                                          %al,(%rax)
  1b:
        00 1c 00
                                   add
                                          %bl,(%rax,%rax,1)
  1e:
        00 00
                                   add
                                          %al,(%rax)
  20:
        00 00
                                   add
                                          %al,(%rax)
  22:
        00 00
                                   add
                                          %al,(%rax)
  24:
        4c 00 00
                                   rex.WR add %r8b,(%rax)
```

```
27: 00 00
                                add %al,(%rax)
  29:
       45 0e
                                rex.RB (bad)
        10 86 02 43 0d 06
                                       %al,0x60d4302(%rsi)
  2b:
                                adc
 31:
        02 43 0c
                                add
                                       0xc(%rbx),%al
 34:
        07
                                (bad)
 35:
        08 00
                                       %al,(%rax)
                                or
 37:
        00 14 00
                                add
                                       %dl,(%rax,%rax,1)
                                       %al,(%rax)
 3a:
        00 00
                                add
 3c:
        00 00
                                add
                                       %al,(%rax)
 3e:
        00 00
                                add
                                       %al,(%rax)
 40:
        01 7a 52
                                add
                                       %edi,0x52(%rdx)
 43:
        00 01
                                add
                                       %al,(%rcx)
 45:
       78 10
                                js
                                       57 < eh_frame+0x57>
 47:
       01 1b
                                add
                                       %ebx,(%rbx)
 49:
                                       $0x7,%al
        0c 07
                                or
 4b:
        08 90 01 00 00 1c
                                or
                                       %dl,0x1c000001(%rax)
 51:
        00 00
                                add
                                       %al,(%rax)
 53:
        00 1c 00
                                add
                                       %bl,(%rax,%rax,1)
 56:
        00 00
                                add
                                       %al,(%rax)
 58:
        00 00
                                add
                                       %al,(%rax)
 5a:
        00 00
                                add
                                       %al,(%rax)
                                popf
 5c:
        9d
 5d:
       00 00
                                add
                                       %al,(%rax)
 5f:
        00 00
                                add
                                       %al,(%rax)
                                rex.RB (bad)
 61:
       45 0e
  63:
       10 86 02 43 0d 06
                                adc
                                       %al,0x60d4302(%rsi)
                                add 0x807(%rsp,%rcx,1),%dl
  69: 02 94 0c 07 08 00 00
セクション .data の逆アセンブル:
000000000000000000000 <a1>:
                                       %eax,(%rax)
  0:
        01 00
                                add
   2:
        00 00
                                add
                                       %al,(%rax)
   4:
        02 00
                                add
                                       (%rax),%al
   6:
        00 00
                                add
                                       %al,(%rax)
   8:
        03 00
                                add
                                       (%rax),%eax
   a:
        00 00
                                add
                                       %al,(%rax)
   C:
        04 00
                                add
                                       $0x0,%al
        00 00
                                add
                                       %al,(%rax)
  e:
  10:
        05 00 00 00 06
                                add
                                       $0x6000000,%eax
  15:
                                add
                                       %al,(%rax)
        00 00
        . . .
セクション .bss の逆アセンブル:
000000000000000000000 <i>:
0000000000000008 <va>:
000000000000010 <v>:
                                       %al,(%rax)
  10:
        00 00
                                add
```

```
セクション .comment の逆アセンブル:
0000000000000000000000 <.comment>:
   0:
        00 47 43
                                   add
                                          %al,0x43(%rdi)
   3:
        43 3a 20
                                   rex.XB cmp (%r8),%spl
        28 55 62
                                          %dl,0x62(%rbp)
   6:
                                   sub
                                          79 < main + 0 \times 2d >
   9:
        75 6e
                                   jne
        74 75
                                          82 < main + 0 \times 36 >
   b:
                                   jе
   d:
        20 31
                                   and
                                          %dh,(%rcx)
        32 2e
   f:
                                  xor
                                          (%rsi),%ch
  11:
        33 2e
                                          (%rsi),%ebp
                                  xor
  13:
        30 2d 31 75 62 75
                                          %ch,0x75627531(%rip)
                                                                         #
                                  xor
7562754a <main+0x756274fe>
  19:
        6e
                                   outsb %ds:(%rsi),(%dx)
  1a:
        74 75
                                   jе
                                          91 < main + 0 \times 45 >
        31 7e 32
                                          %edi,0x32(%rsi)
  1c:
                                  xor
  1f:
        32 2e
                                          (%rsi),%ch
                                   xor
  21:
        30 34 29
                                          %dh,(%rcx,%rbp,1)
                                   xor
  24:
        20 31
                                          %dh,(%rcx)
                                   and
  26:
        32 2e
                                  xor
                                          (%rsi),%ch
        33 2e
  28:
                                          (%rsi),%ebp
                                  xor
  2a:
        30 00
                                  xor
                                          %al,(%rax)
  2c:
        00 47 43
                                   add
                                          %al,0x43(%rdi)
  2f:
        43 3a 20
                                  rex.XB cmp (%r8),%spl
        28 55 62
  32:
                                   sub
                                          %dl,0x62(%rbp)
        75 6e
  35:
                                   jne
                                          a5 < main + 0 \times 59 >
        74 75
  37:
                                          ae < main + 0 \times 62 >
                                   jе
  39:
        20 31
                                          %dh,(%rcx)
                                   and
        32 2e
  3b:
                                  xor
                                          (%rsi),%ch
  3d:
        33 2e
                                   xor
                                          (%rsi),%ebp
        30 2d 31 75 62 75
  3f:
                                  xor
                                          %ch,0x75627531(%rip)
75627576 <main+0x7562752a>
                                   outsb %ds:(%rsi),(%dx)
  45:
        6e
        74 75
                                  jе
  46:
                                          bd < main + 0 \times 71 >
  48:
        31 7e 32
                                          %edi,0x32(%rsi)
                                  xor
        32 2e
  4b:
                                   xor
                                          (%rsi),%ch
  4d:
        30 34 29
                                          %dh,(%rcx,%rbp,1)
                                  xor
        20 31
                                          %dh,(%rcx)
  50:
                                   and
  52:
        32 2e
                                          (%rsi),%ch
                                  xor
  54:
        33 2e
                                          (%rsi),%ebp
                                  xor
        30 00
                                          %al,(%rax)
  56:
                                   xor
```