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:       値         サイズ タイプ   Bind   Vis      索引名
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 SECTION LOCAL  DEFAULT    1
.note.gnu.property
     2: 0000000000000000     0 SECTION LOCAL  DEFAULT    2 .text
     3: 0000000000000000     0 SECTION LOCAL  DEFAULT    4 .rodata
     4: 0000000000000000     0 SECTION LOCAL  DEFAULT    5 .eh_frame
     5: 0000000000000000     0 SECTION LOCAL  DEFAULT    7 .data
     6: 0000000000000000     0 SECTION LOCAL  DEFAULT    8 .bss
     7: 0000000000000000     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.c
    10: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS e4-3m.c
    11: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND printf
    12: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND puts
    13: 0000000000000000    24 OBJECT  GLOBAL DEFAULT    7 a1
    14: 0000000000000000     4 OBJECT  GLOBAL DEFAULT    8 i
    15: 0000000000000008     8 OBJECT  GLOBAL DEFAULT    8 va
    16: 0000000000000010     4 OBJECT  GLOBAL DEFAULT    8 v
    17: 000000000000004c   157 FUNC    GLOBAL DEFAULT    2 main
    18: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND getchar
    19: 0000000000000000    76 FUNC    GLOBAL DEFAULT    2 assign3
```

と出力され、シンボルテーブルからシンボル名、アドレス、サイズ、型がわかった。

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

```
### readelf -h combined.o
combined.o:     ファイル形式 elf64-x86-64

セクション .text の逆アセンブル:

0000000000000000 <assign3>:
   0:   f3 0f 1e fa             endbr64
   4:   55                      push   %rbp
   5:   48 89 e5                mov    %rsp,%rbp
   8:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # e
<assign3+0xe>
   e:   48 98                   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
<assign3+0x1f>
  1f:   48 01 d0                add    %rdx,%rax
  22:   48 89 05 00 00 00 00    mov    %rax,0x0(%rip)        # 29
<assign3+0x29>
  29:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 2f
<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>
  40:   8b 04 02                mov    (%rdx,%rax,1),%eax
  43:   89 05 00 00 00 00       mov    %eax,0x0(%rip)        # 49
<assign3+0x49>
  49:   90                      nop
```

```
  4a:   5d                      pop    %rbp
  4b:   c3                      ret

000000000000004c <main>:
  4c:   f3 0f 1e fa             endbr64
  50:   55                      push   %rbp
  51:   48 89 e5                mov    %rsp,%rbp
  54:   48 83 ec 10             sub    $0x10,%rsp
  58:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 5f
<main+0x13>
  5f:   48 89 c7                mov    %rax,%rdi
  62:   e8 00 00 00 00          call   67 <main+0x1b>
  67:   e8 00 00 00 00          call   6c <main+0x20>
  6c:   88 45 ff                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>
  7c:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 82
<main+0x36>
  82:   85 c0                   test   %eax,%eax
  84:   78 0b                   js     91 <main+0x45>
  86:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 8c
<main+0x40>
  8c:   83 f8 09                cmp    $0x9,%eax
  8f:   7e 07                   jle    98 <main+0x4c>
  91:   b8 01 00 00 00          mov    $0x1,%eax
  96:   eb 4f                   jmp    e7 <main+0x9b>
  98:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 9e
<main+0x52>
  9e:   89 c6                   mov    %eax,%esi
  a0:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # a7
<main+0x5b>
  a7:   48 89 c7                mov    %rax,%rdi
  aa:   b8 00 00 00 00          mov    $0x0,%eax
  af:   e8 00 00 00 00          call   b4 <main+0x68>
  b4:   e8 00 00 00 00          call   b9 <main+0x6d>
  b9:   8b 0d 00 00 00 00       mov    0x0(%rip),%ecx        # bf
<main+0x73>
  bf:   48 8b 15 00 00 00 00    mov    0x0(%rip),%rdx        # c6
<main+0x7a>
  c6:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # cc
<main+0x80>
  cc:   89 c6                   mov    %eax,%esi
  ce:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # d5
<main+0x89>
  d5:   48 89 c7                mov    %rax,%rdi
  d8:   b8 00 00 00 00          mov    $0x0,%eax
  dd:   e8 00 00 00 00          call   e2 <main+0x96>
  e2:   b8 00 00 00 00          mov    $0x0,%eax
  e7:   c9                      leave
  e8:   c3                      ret

### readelf -s combined.o
```

```
Symbol table '.symtab' contains 20 entries:
   番号:      値         サイズ タイプ  Bind   Vis      索引名
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 SECTION LOCAL  DEFAULT    1
.note.gnu.property
     2: 0000000000000000     0 SECTION LOCAL  DEFAULT    2 .text
     3: 0000000000000000     0 SECTION LOCAL  DEFAULT    4 .rodata
     4: 0000000000000000     0 SECTION LOCAL  DEFAULT    5 .eh_frame
     5: 0000000000000000     0 SECTION LOCAL  DEFAULT    7 .data
     6: 0000000000000000     0 SECTION LOCAL  DEFAULT    8 .bss
     7: 0000000000000000     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.c
    10: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS e4-3m.c
    11: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND printf
    12: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND puts
    13: 0000000000000000    24 OBJECT  GLOBAL DEFAULT    7 a1
    14: 0000000000000000     4 OBJECT  GLOBAL DEFAULT    8 i
    15: 0000000000000008     8 OBJECT  GLOBAL DEFAULT    8 va
    16: 0000000000000010     4 OBJECT  GLOBAL DEFAULT    8 v
    17: 000000000000004c   157 FUNC    GLOBAL DEFAULT    2 main
    18: 0000000000000000     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  ............GNU.
 0010 020000c0 04000000 03000000 00000000  ................
セクション .text の内容:
 0000 f30f1efa 554889e5 8b050000 00004898  ....UH........H.
 0010 488d1485 00000000 488d0500 00000048  H.......H......H
 0020 01d04889 05000000 008b0500 00000048  ..H............H
 0030 98488d14 85000000 00488d05 00000000  .H.......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  ....x..........~
 0090 07b80100 0000eb4f 8b050000 000089c6  .......O........
 00a0 488d0500 00000048 89c7b800 000000e8  H......H........
 00b0 00000000 e8000000 008b0d00 00000048  ...............H
 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  ................
 0020 00000000 4c000000 00450e10 8602430d  ....L....E....C.
 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                   adc    %al,(%rax)
   6:   00 00                   add    %al,(%rax)
   8:   05 00 00 00 47          add    $0x47000000,%eax
   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)
  1b:   00 00                   add    %al,(%rax)
  1d:   00 00                   add    %al,(%rax)
        ...

セクション .text の逆アセンブル:

0000000000000000 <assign3>:
   0:   f3 0f 1e fa             endbr64
   4:   55                      push   %rbp
   5:   48 89 e5                mov    %rsp,%rbp
   8:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # e
<assign3+0xe>
   e:   48 98                   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
<assign3+0x1f>
  1f:   48 01 d0                add    %rdx,%rax
```

```
  22:   48 89 05 00 00 00 00    mov    %rax,0x0(%rip)        # 29
<assign3+0x29>
  29:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 2f
<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>
  40:   8b 04 02                mov    (%rdx,%rax,1),%eax
  43:   89 05 00 00 00 00       mov    %eax,0x0(%rip)        # 49
<assign3+0x49>
  49:   90                      nop
  4a:   5d                      pop    %rbp
  4b:   c3                      ret

000000000000004c <main>:
  4c:   f3 0f 1e fa             endbr64
  50:   55                      push   %rbp
  51:   48 89 e5                mov    %rsp,%rbp
  54:   48 83 ec 10             sub    $0x10,%rsp
  58:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # 5f
<main+0x13>
  5f:   48 89 c7                mov    %rax,%rdi
  62:   e8 00 00 00 00          call   67 <main+0x1b>
  67:   e8 00 00 00 00          call   6c <main+0x20>
  6c:   88 45 ff                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>
  7c:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 82
<main+0x36>
  82:   85 c0                   test   %eax,%eax
  84:   78 0b                   js     91 <main+0x45>
  86:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 8c
<main+0x40>
  8c:   83 f8 09                cmp    $0x9,%eax
  8f:   7e 07                   jle    98 <main+0x4c>
  91:   b8 01 00 00 00          mov    $0x1,%eax
  96:   eb 4f                   jmp    e7 <main+0x9b>
  98:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 9e
<main+0x52>
  9e:   89 c6                   mov    %eax,%esi
  a0:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # a7
<main+0x5b>
  a7:   48 89 c7                mov    %rax,%rdi
  aa:   b8 00 00 00 00          mov    $0x0,%eax
  af:   e8 00 00 00 00          call   b4 <main+0x68>
  b4:   e8 00 00 00 00          call   b9 <main+0x6d>
  b9:   8b 0d 00 00 00 00       mov    0x0(%rip),%ecx        # bf
<main+0x73>
  bf:   48 8b 15 00 00 00 00    mov    0x0(%rip),%rdx        # c6
<main+0x7a>
```

```
  c6:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # cc
<main+0x80>
  cc:   89 c6                   mov    %eax,%esi
  ce:   48 8d 05 00 00 00 00    lea    0x0(%rip),%rax        # d5
<main+0x89>
  d5:   48 89 c7                mov    %rax,%rdi
  d8:   b8 00 00 00 00          mov    $0x0,%eax
  dd:   e8 00 00 00 00          call   e2 <main+0x96>
  e2:   b8 00 00 00 00          mov    $0x0,%eax
  e7:   c9                      leave
  e8:   c3                      ret

セクション .rodata の逆アセンブル:

0000000000000000 <.rodata>:
   0:   45 6e                   rex.RB outsb %ds:(%rsi),(%dx)
   2:   74 65                   je     69 <main+0x1d>
   4:   72 20                   jb     26 <.rodata+0x26>
   6:   61                      (bad)
   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                   or     (%rax),%al
  15:   61                      (bad)
  16:   31 5b 25                xor    %ebx,0x25(%rbx)
  19:   64 5d                   fs pop %rbp
  1b:   3a 20                   cmp    (%rax),%ah
  1d:   61                      (bad)
  1e:   64 64 72 3d             fs fs jb 5f <main+0x13>
  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                   adc    $0x0,%al
   2:   00 00                   add    %al,(%rax)
   4:   00 00                   add    %al,(%rax)
   6:   00 00                   add    %al,(%rax)
   8:   01 7a 52                add    %edi,0x52(%rdx)
   b:   00 01                   add    %al,(%rcx)
   d:   78 10                   js     1f <.eh_frame+0x1f>
   f:   01 1b                   add    %ebx,(%rbx)
  11:   0c 07                   or     $0x7,%al
  13:   08 90 01 00 00 1c       or     %dl,0x1c000001(%rax)
  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)
  2b:   10 86 02 43 0d 06       adc    %al,0x60d4302(%rsi)
  31:   02 43 0c                add    0xc(%rbx),%al
  34:   07                      (bad)
  35:   08 00                   or     %al,(%rax)
  37:   00 14 00                add    %dl,(%rax,%rax,1)
  3a:   00 00                   add    %al,(%rax)
  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:   0c 07                   or     $0x7,%al
  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)
  5c:   9d                      popf
  5d:   00 00                   add    %al,(%rax)
  5f:   00 00                   add    %al,(%rax)
  61:   45 0e                   rex.RB (bad)
  63:   10 86 02 43 0d 06       adc    %al,0x60d4302(%rsi)
  69:   02 94 0c 07 08 00 00    add    0x807(%rsp,%rcx,1),%dl

セクション .data の逆アセンブル:

0000000000000000 <a1>:
   0:   01 00                   add    %eax,(%rax)
   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
   e:   00 00                   add    %al,(%rax)
  10:   05 00 00 00 06          add    $0x6000000,%eax
  15:   00 00                   add    %al,(%rax)
        ...

セクション .bss の逆アセンブル:

0000000000000000 <i>:
        ...

0000000000000008 <va>:
        ...

0000000000000010 <v>:
  10:   00 00                   add    %al,(%rax)
        ...
```

```
セクション .comment の逆アセンブル:

0000000000000000 <.comment>:
   0:   00 47 43                add    %al,0x43(%rdi)
   3:   43 3a 20                rex.XB cmp (%r8),%spl
   6:   28 55 62                sub    %dl,0x62(%rbp)
   9:   75 6e                   jne    79 <main+0x2d>
   b:   74 75                   je     82 <main+0x36>
   d:   20 31                   and    %dh,(%rcx)
   f:   32 2e                   xor    (%rsi),%ch
  11:   33 2e                   xor    (%rsi),%ebp
  13:   30 2d 31 75 62 75       xor    %ch,0x75627531(%rip)        #
7562754a <main+0x756274fe>
  19:   6e                      outsb  %ds:(%rsi),(%dx)
  1a:   74 75                   je     91 <main+0x45>
  1c:   31 7e 32                xor    %edi,0x32(%rsi)
  1f:   32 2e                   xor    (%rsi),%ch
  21:   30 34 29                xor    %dh,(%rcx,%rbp,1)
  24:   20 31                   and    %dh,(%rcx)
  26:   32 2e                   xor    (%rsi),%ch
  28:   33 2e                   xor    (%rsi),%ebp
  2a:   30 00                   xor    %al,(%rax)
  2c:   00 47 43                add    %al,0x43(%rdi)
  2f:   43 3a 20                rex.XB cmp (%r8),%spl
  32:   28 55 62                sub    %dl,0x62(%rbp)
  35:   75 6e                   jne    a5 <main+0x59>
  37:   74 75                   je     ae <main+0x62>
  39:   20 31                   and    %dh,(%rcx)
  3b:   32 2e                   xor    (%rsi),%ch
  3d:   33 2e                   xor    (%rsi),%ebp
  3f:   30 2d 31 75 62 75       xor    %ch,0x75627531(%rip)        #
75627576 <main+0x7562752a>
  45:   6e                      outsb  %ds:(%rsi),(%dx)
  46:   74 75                   je     bd <main+0x71>
  48:   31 7e 32                xor    %edi,0x32(%rsi)
  4b:   32 2e                   xor    (%rsi),%ch
  4d:   30 34 29                xor    %dh,(%rcx,%rbp,1)
  50:   20 31                   and    %dh,(%rcx)
  52:   32 2e                   xor    (%rsi),%ch
  54:   33 2e                   xor    (%rsi),%ebp
  56:   30 00                   xor    %al,(%rax)
```