Advanced Operating System and Virtualization

Disassember2 and 3

Hiroaki Fukuda

Contents

- Endian
  - Little/Big Engian
- Displacemant
  - DISP in the specification
  - disp used with jmp operation

Endian: Let's try bb1000

```
000e: 01c1          add cx, ax
0010: bb1000        mov bx, 0010
0013: 81fb1400      cmp bx, 0014
```

Immediate to Register

1 0 1 1 w reg	data	data if w = 1
bb	10	00

Little Endian

Low byte is stored fitst

mov bx, 0010

Displacement: Let's try 005589

ADD = Add:

Reg./Memory with Register to Either

0 0 0 0 0 0 d w	mod reg r/m

00000000 01010101 0x89

D = 0 w = 0 mod = 01 reg = 010 r/m = 101

Add r/m, reg

if r/m = 101 then EA = (DI) + DISP

8-Bit (w = 0)	
000	AL
001	CL
010	DL
011	BL
100	AH
101	CH
110	DH
111	BH

Displacement (DISP) is the additional value(1 or 2 bytes) to specify address, which is next to operand

if mod = 01 then DISP = disp-low sign-extended to 16 bits, disp-high is absent

DISP = 0x89

Add [di+89], dl NO!

How to represent minus value?

Assume 1 byte

0 ~ 255 or -128 ~ 127

unsigned signed

What is this value?

11111111 ⟹ 255 or -1

Using 2 complement, we can translate from unsigned to signed

See again!

if mod = 01 then DISP = disp-low sign-extended to 16 bits, disp-high is absent

DISP = 0x89

11111111 10001001 Sign extend!

00000000 01110111 -77

Add [di+89], dl

⬇

Add [di-77], dl

disp in jump: Let's try ebf1

```
0060: 8b9f1600      mov bx, [bx+16]
0064: ffd3          call bx
0066: ebf1          jmp short 0059
0068: e9c900        jmp 0134
```

Direct within Segment-Short

11101011	disp
eb	f1

⬇

Jump short 0059?

f1 = -f

0x0068 – 0xf = 0x0059

Complete your disassembler

- 1.c to 6.c
- nm (/usr/local/core/minix2/usr/bin/nm