Tölvutækni og forritun Heimadæmi 6

brj46

September 2024

1

Segjum að gistið $\%rax$ innihaldi $0x200$ og gistið $\%rcx$ innihaldi $0x5$. Hér fyrir neðan eru minnistilvísanir. Í hverju tilviki reiknið út raunverulegt vistfang

a)

$$\$0x18(\%rax)$$

Þessi minnistilvísun er samsett af forskoti$0x18$ grunnskrá $\%rax$

Þar sem rax inniheldur $0x200$ og við forskeytum $0x18$.

þá er raunverulegt vistfang: $0x218$ l

b)

$$\%rax, \%rcx, 8$$

Þessi minnistilvísun inniheldur grunnskrá $\%rax$ og vísiskrá $\%rcx$ margfaldað með skalanum 8

- innihald $\%rax$: $0x200$
- innihald $\%rcx$: $0x5$
- Skali: 8

  Þá er raunverulegt vistfang: $0x200 + (0x5 \times 8) = 0x228$

c)

$$\$50(, \%rax, 4)$$

Þessi minnistilvísun er með forskot 50 og vísiskrá $\$\%rax$ margfaldaða með skala 4.

- Forskot: 50
- innihald $\%rax$: $0x200$
- Skali: 4

Svo raunverulegt vistfang er $50 + (0x200 \times 4) = 50 + 0x800 = 0x832$

2

Hér fyrir neðan eru gagnaflutningsskipanir fyrir mismunandi gagnastærðir. Það vantar hins vegar síðasta bókstafinn í skipanirnar **(b, w, l eða q)**, eftir því hvaða gagnastærðir er verið að flytja.

Skrifum skipanirnar aftur upp með réttum síðasta bókstaf í stað spurningamerkis.

1. *mov*?$(\%rbx, \%rax, 2), \%edx$

   Þar sem gistið $\%edx$ er 32-bita skrá þá er réttur bókstafur "l"

   Svo skipunin verður:

   $$movl(\%rbx, \%rax, 2), \%edx$$

2. *mov*? $2, $\%rdi$

   Þar sem gistið $\%rdi$ er 64-bita skrá þá er rétti bókstafurinn "q"

   Svo skipunin verður:

   $$movq \ \$2, \%rdi$$

3. *mov*? $\%ah, (\%rcx)$

   Gistið $\%ah$ er hluti af $\%ax$ sem er 16 bita skrá en $\%ah$ er einungis helmingurinn af því svo það er aðeins 8-bitar

   Svo réttur bókstafur er b:

   $$movb \ \%ah, (\%rcx)$$

4. *mov*? $\%bx, \%dx$

   þar sem gistin $\%bx$ og $\%dx$ eru 16-bita skrár þá er réttur bókstafur "w"

   Svo skipunin verður:

   $$movw \ \%bx, \%dx$$

3

Skipunin imul er frekar dýr í x86-64 og því er reynt að nota ódýrar skipanir eins og leaq, add og sal til að framkvæma margföldun með litlum fasta. Hér fyrir neðan eru nokkrar einfaldar segðir. Sýnum ódýrar smalamálsskipanir til að reikna þær.

a)

$$x = 5 \times x$$

við getum brotið $5 \times x$ í $(4 \times x) + x$

notum þá lea skipunina:

```
leaq (%rax, %rax, 4), %rax # x + (4 * x) = x * 5
```

b)

$$x = 27 \times x$$

Við getum brotið þetta niður sem $((x \times 3) \times 8) + x \times 3$

reiknum $x \times 3$ og geymum það í rbx með skipuninni:

```
leaq (\%rax,%rax,2), %rbx  #x + x * 2 = x * 3
```

hliðrum núna rax með 3 til að margfalda með 8

```
salq $3, %rax # x * 2^3
```

núna bætum við rbx við þrefalt *%rax* og fáum þá út $x \times 27$:

```
leaq (%rbx, %rax,3) # x * 3 + ((x * 8)*3) = x * (24 + 3 ) = x * 27
```

c)

$$x = 45 \times x$$

Við getum brotið þetta niður sem: $45 \times x = (36 \times x) + (9 \times x)$ eða:

$4 * (x + 8x) + (x + 8x)$

Við notum leaq skipunina til að reikna bæði summu og margfeldi:

```
leaq (%rax,%rax,8), %rax # x + 8*x eða 9 * x
leaq (%rax,%rax,4), %rax # 9 * x + 4 * 9 * x eða 45 * x
```

d)

$$x = 11 \times x$$

Við brjótum 11 niður sem $8 + 2 + 1$

þá getum við notað skipanirnar:

```
leaq (%rax, %rax, 1), %rdx # rdx er þá með X * 2
leaq (%rdx, %rdx, 4), %rdx # rdx er þá með (x*2) + ((x*2)*4) = x * (2 + 8)
addq %rax, %rdx # bætir x við rdx og þá höfum við x + (x*10) = x * 11
```

4

Hér fyrir neðan er smalamálsútgáfa af fallinu long reikn(long x, long y) sem reiknar einfalda segð (expression) út frá viðföngunum x og y. Athugið að viðfangið x kemur í gistið *%rdi*, viðfangið y kemur í *%rsi* og skilagildi fallsins fer í gistið *%rax*.

```
reikn:
    leaq (%rsi,%rsi,4), %rax
    leaq (%rax,%rdi,2), %rdx
    leaq 0(,%rdx,8), %rax
    subq %rdx, %rax
    ret
```

a)

Setjum athugasemd fyrir aftan hverja skipun þar sem tilgangur hennar í útreikningnum er útskýrður.

1. $leaq(\%rsi, \%rsi, 4),\ \%rax$

   Hér erum við með lea skipunina sem reiknar $\%rsi + (4 \times \%rsi)$ eða $y + (4 \times y) = 5 \times y$ og er geymir niðurstöðuna í %rax

2. $leaq(\%rax, \%rdi, 2), \%rdx$

   Hér erum við aftur með lea skipunina sem reiknar$\%rax + (\%rdi \times 2)$ eða $5 \times y + (2 \times x)$ eða $5 \times y + 2 \times x$ og er þessi niðurstaða geymd í $\%rdx$

3. $leaq\ 0(, \%rdx, 8), \%rax$

   Hér er aftur lea skipunin sem gerir $0 + (8 \times \%rdx)$ og geymir það í $\%rax$

   svo það sem fer í $\%rax$ verður $8 \times (5 \times y + 2 \times x$

4. $subq\ \%rdx, \%rax$

   Hér er skipunin sub sem dregur $\%rdx$ frá $\%rax$ sem við reiknum þá með

   $8 \times (5 \times y + 2 \times x) - (5 \times y + 2 \times x)$

   $\%rax$ inniheldur núna $7 \times (5 \times y + 2 \times x)$

5. ret

   Skilum niðurstöðunni sem er nú í $\%rax$

b)

Fyrir C kóða getum við bara skrifað þessa segð beint í return skipun:

```
long reikn(long x, long y) {
    return 7 * (5 * y + 2 * x);
}
```

5

Við fáum gefna Linux keyrsluskránna haha.

finnum smalamáls kóðan <hvad> í haha:

eftir að nota objdump -d haha > haha.od og skoða hana síðan með cat þá sjáum við að smalamálskóðinn lítur svona út:

```
0000000000001169 <hvad>:
1169: f3 0f 1e fa           endbr64
116d: 8d 04 3f              lea    (%rdi,%rdi,1),%eax
1170: 21 f8                 and    %edi,%eax
1172: c1 e7 02              shl    $0x2,%edi
1175: 21 f8                 and    %edi,%eax
1177: c3                    ret
```

Túlkum þennan kóða og skrifum hann síðan í C-kóða:

1. `116d: 8d 04 3f  lea  (%rdi,%rdi,1),%eax`

   Hér höfum við lea skipun sem reiknar $2 * \%rdi$ og geymir það í $\%eax$ við gætum skrifað það sem $eax = 2 * n$ í C

2. `1170: 21 f8  and  %edi,%eax`

   þetta er AND skiðun á milli edi og eax sem inniheldur $2 * n$

3. `1172: c1 e7 02  shl  $0x2,%edi`

   þetta er Shift left skipun sem margfaldar n með 2^2 og setur niðurstöðuna í edi

4. `1175: 21 f8  and  %edi,%eax`

   Hér er önnur AND skipun sem er á milli edi og eax og setur niðurstöðuna í eax

5. `1177: c3  ret`

   Skilar niðurstöðunni í $\%eax$

Skrifum nú fallið í C:

```c
int hvad(unsigned int n) {
    int res = (n * 2) & n;
    res = (n * 4) & res;
    return res;
}
```