Tölvutækni og forritun Heimadæmi 6

Ragnar Björn Ingvarsson, rbi3

28. september 2024

1 Segjum að gistið $\%rax$ innihaldi $0x200$ og gistið $\%rcx$ innihaldi $0x5$. Í hverju tilviki reiknið út raunverulegt vistfang.

a) $\$0x18(\%rax)$ Fáum hér $0x18$ sem hliðrun fyrir gildi gistisins $\%rax$ sem er þá $0x18 + 0x200 = 0x218$

b) $(\%rax, \%rcx, 8)$ Hér plúsum við saman gildi $\%rax$ og áttfalt gildi $\%rcx$, svo $0x200 + 8 * 0x5 = 0x200 + 0x28 = 0x228$

c) $\$50(, \%rax, 4)$ Hér er hliðrun um 50 á fjórfalt gildi $\%rax$, svo $50 + 4 * 0x200 = 0x32 + 0x800 = 0x832$

2 Hér eru gagnaflutningsskipanir fyrir mismunandi gagnastærðir. Það vantar hins vegar síðasta bókstafinn í skipanirnar eftir því hvaða gagnastærðir er verið að flytja. Skrifið skipanirnar aftur upp með réttum síðasta bókstaf í stað spurningamerkisins.

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

movq $\$2, \%rdi$

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

movw $\%bx, \%dx$

3 Hér fyrir neðan eru nokkrar einfaldar segðir. Sýnið ódýrar smalamálsskipanir til að reikna þær. Gerið ráð fyrir að breytan x sé í gistinu $\%rax$, en þið megið nota önnur gisti ef þið þurfið.

a) *leaq* $(\%rax, \%rax, 4), \%rax$

b) *leaq* $(\%rax, \%rax, 8), \%rax$
leaq $(\%rax, \%rax, 2), \%rax$

c) *leaq* $(\%rax, \%rax, 8), \%rax$
leaq $(\%rax, \%rax, 4), \%rax$

d) *leaq* $(\%rax, \%rax), \%rdx$
leaq $(\%rax, \%rdx, 5), \%rax$

4 Hér fyrir neðan er smalamálsútgáfa af fallinu *long reikn*(*long x*, *long y*) sem reiknar einfalda segð 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*.

a) Hér er útskýring

reikn :
Hér erum við að margfalda y með 5 og setja það í gistið %rax
— *leaq* (%*rsi*,%*rsi*,4), %*rax*
Hér tökum við 5y og bætum við það x margfaldað með tveimur, og setjum þá 5y + 2x í %rdx
— *leaq* (%*rax*,%*rdi*,2), %*rdx*
Tökum svo 5y + 2x og margföldum það með 8 og setjum í skilagildið %rax
— *leaq* 0(,%*rdx*,8), %*rax*
Drögum svo 5y + 2x einu sinni frá skilagildinu til að fá loks 7(5y + 2x)
— *subq* %*rdx*, %*rax*
Skilum svo 7(5y + 2x)
— *ret*

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

5 Þið fáið gefna Linux keyrsluskránna *haha*. Hún hefur *main*-fall sem tekur inn eina heiltölu á skipanalínunni, kallar á fallið *hvad* og prentar út skilagildi þess. Í þessu dæmi eigið þið að endurskrifa fallið í C kóða og útskýra í orðum hvað það gerir.

Við sjáum að assembly kóðinn fyrir fallið 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
```

Þar sem inntaksgildi fallsins n er í %edi er %rdi bara 64-bita útgáfa af því svo við fyrst

tvöföldum n og setjum í %eax. Síðan bitwise AND-um við þá n og 2n og geymum niðurstöðuna í %eax.

Svo bit shiftum við n um 2 til vinstri og geymum í %edi, og loks bitwise AND-um við %edi, semsagt n«2, við %eax, semsagt n & 2n.

Svo skilum við því gildi, þ.e. %eax.

Þá getum við skrifað C kóðann

```
int hvad(unsigned int n) {
    return n & 2*n & n<<2;
}
```

Þar sem við fyrst AND-um saman n og 2n, og svo niðurstöðuna úr því við n shiftað um 2 til vinstri. Fallið virkar þá þannig að það fær inn tölu og skilar bitastreng sem inniheldur ás allstaðar þar sem þrír ásar koma fram í inntakinu í röð til hægri.