ARM-assemblyn alkeet

Henrik Lievonen

1. kesäkuuta 2017

Sisällys

Mikä ihmeen assembly

```
int arr[10] = {
    0, 1, 2, 3, 4,
    5, 6, 7, 8, 9
};
int s = 0;
for (int i = 0; i < 10; i++)
    s += arr[i];
```

Mikä ihmeen assembly

```
int arr[10] = {
    0, 1, 2, 3, 4,
    5, 6, 7, 8, 9
};
int s = 0;
for (int i = 0; i < 10; i++)
    s += arr[i];
```

```
    mov r5, #0
    mov r4, #0
    b   .L2
.L3:
    ldr r3, .L5
    ldr r3, [r3, r4, lsl #2]
    add r5, r5, r3
    add r4, r4, #1
.L2:
    cmp r4, #9
    ble .L3

.L5: .word arr
arr: .word 0, 1, 2, 3, 4
     .word 5, 6, 7, 8, 9
```

Mikä ihmeen ARM

- *Advanced RISC Machine* (aiemmin *Acorn RISC Machine*)
- 32-bittinen mikroprosessoriarkkitehtuuri
- 16 yleisrekisteriä (`r0` - `r15`)
- Suosittu pienissä, sulautetuissa järjestelmissä
- Vähävirtainen
- Tarvittaessa hyvinkin yksinkertainen

RISC vs CISC

RISC	CISC
Reduced Instruction Set Computer	*Complex Instruction Set Computer*
Käskyt tekevät yhden asian	Käskyt voivat tehdä monta asiaa
Käskyt tasamittaisia	Käskyt vaihtelevan mittaisia
Suurin osa käskyistä vakioaikaisia	Eri käskyt vievät eri verran aikaa
ARM, *MIPS*, *SPARC*, *Alpha*, *PowerPC*	*x86*, *VAX*, *IBM S/360*, *PDP-11*

RISC vs CISC

ARM:

```
    mov r5, #0
    mov r4, #0
        b   .L2
.L3:
    ldr r3, .L5
    ldr r3, [r3, r4, lsl #2]
    add r5, r5, r3
    add r4, r4, #1
.L2:
    cmp r4, #9
    ble .L3

.L5: .word arr
arr: .word 0, 1, 2, 3, 4
     .word 5, 6, 7, 8, 9
```

x86_64:

```
    movl    $0, %r12d
    movl    $0, %ebx
    jmp .L2
.L3:
    movslq  %ebx, %rax
    movl    arr(,%rax,4), %eax
    addl    %eax, %r12d
    addl    $1, %ebx
.L2:
    cmpl    $9, %ebx
    jle .L3

arr: .word 0, 1, 2, 3, 4
     .word 5, 6, 7, 8, 9
```

Komennot

```
start:
add     r0, r1, r2
subeq   r0, r1, r2
mulseq  r0, r1, r2
eors    r0, r1, #10
rsb     r0, r1, r2, lsl #2
and     r0, r1, r2, asr r3
mov     r0, r1
cmp     r0, r1, ror #15
b       start
bl      start
ldr     r0, [r1, #-4]
ldr     r0, [r1, r2]
ldr     r0, [r1, r2, lsl #2]
```

Lisää ARM-arkkitehtuurista

Ehdolliset komennot	
eq	Yhtäsuuri kuin
ne	Erisuuri kuin
cs / hs	Etumerkitön suurempi tai yhtäsuuri kuin
cc / lo	Etumerkitön pienempi kuin
mi	Negatiivinen
pl	Positiivinen tai nolla
vs	Ylivuoto
vc	Ei ylivuotoa
hi	Etumerkitön suurempi kuin
ls	Etumerkitön pienempi tai yhtäsuuri kuin
ge	Etumerkillinen suurempi tai yhtäsäsuuri kuin
lt	Etumerkillinen pienempi kuin
gt	Etumerkillinen suurempi kuin
le	Etumerkillinen pienempi tai yhtäsuuri kuin
al	Aina

Lisää ARM-arkkitehtuurista

Siirrot	
lsr	Looginen siirto oikealle
lsl	Looginen siirto vasemmalle
asr	Aritmeettinen siirto oikealle
ror	Pyöritys oikealle
rrx	Pyöritys oikealle muistinumerolla

Parametrit

- Vakioiden oltava oikealla välillä ($\approx 0 - \approx 32$)
- Rekistereistä käytetään alinta 8 bittiä

Lisää ARM-arkkitehtuurista

- CPSR
  - *Current Program Status Register*
  - Suorittimen nykyinen tila
  - ALUn tila (ehtoja varten)
  - Onko keskeytykset käytössä

Lisää ARM-arkkitehtuurista

- Rekisterit
  - `r15` program counter `pc`
  - `r14` link register `lr`
  - `r13` stack pointer `sp`
  - `r12` Intra-Procedure-call scratch register
  - `r11` frame pointer `fp`
  - `r4` - `r10` local variables
  - `r0` - `r3` arguments and return values

Esimerkkejä