درس ۵

ASM

حساب و flag

hello.asm

```
add rax, rbx
imul rax, 10
shl rax, 3
```

یک کورس رایگان از MyWebUniversity

زبان Assembly

درس ۵

حساب، منطق و flag ها

add، sub، inc، dec، mul، imul، div، and، or، xor، shift ها و register به نام RFLAGS



add، sub، inc، dec، neg

add و sub نتیجه را در عملوند اول می‌گذارند. inc و dec یک اضافه یا کم می‌کنند، و neg علامت ...

هر دستور حسابی flag ها را هم به‌روز می‌کند، که در درس بعدی برای تصمیم‌ها از آن‌ها استفاده ...

ASSEMBLY LANGUAGE · 5

math.asm

```
mov rax, 10
add rax, 5          ; rax = 15
sub rax, 3          ; rax = 12
inc rax             ; rax = 13
dec rax             ; rax = 12
neg rax             ; rax = -12
add [total], rax    ; memory += rax
```

01

ضرب و تقسیم

imul با دو عملوند اعداد علامت‌دار را به سادگی ضرب می‌کند. شکل‌های یک‌عملوندی mul و imul از ...

div و idiv، rdx:rax را بر عملوند تقسیم می‌کنند: خارج قسمت به rax و باقی‌مانده به rdx ...

ASSEMBLY LANGUAGE · 5

divide.asm

```
imul rax, rbx, 10    ; rax = rbx * 10

mov rax, 17
xor rdx, rdx         ; clear high half
mov rcx, 5
div rcx              ; rax = 3 (quotient)
                     ; rdx = 2 (remainder)

mov rax, -17
cqo                  ; sign-extend rax into rdx
idiv rcx             ; rax = -3, rdx = -2
```

02

منطق بیتی

and بیت‌هایی را که در هر دو عملوند یک هستند نگه می‌دارد، or بیت‌ها را روشن می‌کند، xor ...

این‌ها برای mask ها و flag ها استفاده می‌شوند: and با پانزده چهار بیت پایین را نگه می‌دارد؛ ...

ASSEMBLY LANGUAGE · 5

دستور	al پیش	al بعد
`and al, 0x0F`	`0100 0001`	`0000 0001`
`or al, 0x20`	`0100 0001 'A'`	`0110 0001 'a'`
`xor al, 0xFF`	`0100 0001`	`1011 1110`
`not al`	`0100 0001`	`1011 1110`

03

shift ها و rotate ها

shl بیت‌ها را به چپ shift می‌کند، که برای هر موقعیت ضرب در دو است. shr به راست shift ...

کمپایلرها ضرب در توان‌های دو را با shift عوض می‌کنند چون سریع‌تر هستند.

ASSEMBLY LANGUAGE · 5

shift.asm

```
mov rax, 5
shl rax, 3        ; 5 * 8 = 40
shr rax, 1        ; 40 / 2 = 20
mov rbx, -16
sar rbx, 2        ; -16 / 4 = -4 (sign kept)
rol al, 4         ; swap the two hex digits of al
```

04

register flag ها

بعد از بیشتر عملیات حسابی، CPU flag ها را در RFLAGS تعیین می‌کند. ZF یعنی نتیجه صفر بود، SF ...

jump های شرطی در درس بعدی همین flag ها را می‌خوانند.

ASSEMBLY LANGUAGE · 5

flag	روشن می‌شود وقتی	مثال
ZF	نتیجه صفر است	`sub rax, rax`
SF	نتیجه منفی است	`sub 3, 5`
CF	carry بی‌علامت	`add al, 200 (al=100)`
OF	overflow علامت‌دار	`add al, 100 (al=100)`

05

نقشهٔ درس


1
add، sub، inc، dec، neg
2
ضرب و تقسیم
3
منطق بیتی
4
shift ها و rotate ها
5
register flag ها

منابع و مطالعهٔ بیشتر

ASSEMBLY LANGUAGE · 5

math.asm

```
mov rax, 10
add rax, 5          ; rax = 15
sub rax, 3          ; rax = 12
inc rax             ; rax = 13
dec rax             ; rax = 12
neg rax             ; rax = -12
add [total], rax    ; memory += rax
```

ASSEMBLY LANGUAGE · 5

دستور	al پیش	al بعد
and al, 0x0F	0100 0001	0000 0001
or al, 0x20	0100 0001 'A'	0110 0001 'a'
xor al, 0xFF	0100 0001	1011 1110
not al	0100 0001	1011 1110

تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity