درس ۹

ASM

رشته و بایت

```asm
hello.asm
cmp byte [rdi+rax], 0
je .done
inc rax
```

یک کورس رایگان از MyWebUniversity

زبان Assembly

درس ۹

رشته‌ها، آرایه‌ها و ورودی



حلقه‌های بایتی، طول رشته، خواندن ورودی، تبدیل عدد به متن و دستورهای رشته‌ای rep

طول رشته

یک رشتهٔ C بایت‌هایی است که با یک بایت صفر تمام می‌شود. برای پیدا کردن طول آن، تا پیدا کردن ...

این دقیقاً همان کاری است که تابع strlen در C انجام می‌دهد.

ASSEMBLY LANGUAGE · 9

strlen.asm

```
; size_t my_strlen(const char *s)   s in rdi
my_strlen:
    xor eax, eax
.next:
    cmp byte [rdi + rax], 0
    je  .done
    inc rax
    jmp .next
.done:
    ret
```

01

تغییر بایت‌ها: حروف بزرگ

یک بایت را بار کنید، بررسی کنید که آیا یک حرف کوچک بین a و z است، و اگر بلی سی و دو کم کنید ...

بعد آن را دوباره ذخیره کنید و به بایت بعدی بروید.



02

خواندن ورودی

system call به نام read، شمارهٔ صفر، از standard input به یک buffer می‌خواند و برمی‌گرداند ...

buffer را در section .bss رزرو کنید، بعد متن را با write پس چاپ کنید.

ASSEMBLY LANGUAGE · 9

echo.asm

```
section .bss
    buf resb 128
section .text
    mov rax, 0          ; read
    mov rdi, 0          ; stdin
    mov rsi, buf
    mov rdx, 128
    syscall             ; rax = bytes read
    mov rdx, rax        ; write that many
    mov rax, 1
    mov rdi, 1
    syscall
```

03

چاپ یک عدد

هیچ دستور چاپ عدد وجود ندارد. پشت سر هم بر ده تقسیم کنید: هر باقی‌مانده یک رقم است. چهل و ...

رقم‌ها به ترتیب برعکس بیرون می‌آیند، پس buffer را از آخر پر کنید.

ASSEMBLY LANGUAGE · 9

itoa.asm

```
; rax = number, rsi = end of buffer
    mov rcx, 10
.digit:
    xor rdx, rdx
    div rcx              ; rax /= 10, rdx = digit
    add dl, '0'          ; 7 → '7'
    dec rsi
    mov [rsi], dl
    test rax, rax
    jnz .digit           ; rsi now points at the text
```

04

دستورهای رشته‌ای rep

x86 دستورهای رشته‌ای ویژه دارد. rep movsb به تعداد rcx بایت از rsi به rdi کاپی می‌کند، و ...

repne scasb یک بایت را جستجو می‌کند. CPU های مدرن rep movsb را برای کاپی حافظه خیلی سریع ...

ASSEMBLY LANGUAGE · 9

memcpy.asm

```
; memcpy(dst=rdi, src=rsi, n=rdx)
my_memcpy:
    mov rax, rdi        ; return dst
    mov rcx, rdx
    rep movsb
    ret

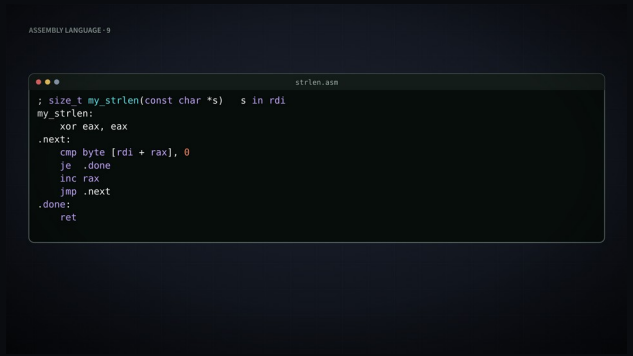
; memset(buf, 0, 64)
    lea rdi, [buf]
    xor eax, eax
    mov ecx, 64
    rep stosb
```

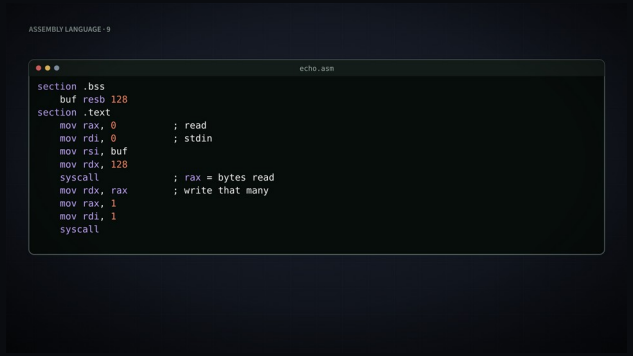
05

نقشهٔ درس


1
طول رشته
2
تغییر بایت‌ها: حروف بزرگ
3
خواندن ورودی
4
چاپ یک عدد
5
دستورهای رشته‌ای rep

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity