درس ۱۰

ASM

Assembly + C

hello.asm

```
extern printf
lea rdi, [rel fmt]
call printf wrt ..plt
```

یک کورس رایگان از MyWebUniversity

زبان Assembly

درس ۱۰

ترکیب Assembly و C



صدا کردن printf از Assembly، صدا کردن Assembly از C، link با gcc، و SIMD

صدا کردن printf

printf را به شکل extern اعلان کنید و به جای _start، main را تعریف کنید، چون runtime C، main ...

برای تابع‌های variadic مثل printf، al را تعداد register های vector استفاده‌شده، اینجا صفر، ...

ASSEMBLY LANGUAGE · 10

hello_c.asm

```
extern printf
section .data
    fmt db "%s scored %d", 10, 0
    who db "Mariam", 0
section .text
global main
main:
    sub rsp, 8            ; align stack
    lea rdi, [rel fmt]
    lea rsi, [rel who]
    mov edx, 95
    xor eax, eax          ; no vector args
    call printf wrt ..plt
    add rsp, 8
    xor eax, eax
    ret
```

01

ساختن با gcc

مثل پیش با NASM، assemble کنید، اما با gcc، link کنید. کود شروع و کتابخانهٔ C را اضافه می‌کند.

کلمه‌های rel و plt کود مستقل از موقعیت می‌سازند، که Linux مدرن به طور پیش‌فرض لازم دارد.



02

صدا کردن Assembly از C

یک تابع سریع را با Assembly بنویسید، آن را در C با prototype آن اعلان کنید، و هر دو فایل ...

چون هر دو از System V ABI پیروی می‌کنند، C آن را مثل هر تابع دیگر صدا می‌کند.



03

خواندن خروجی کمپایلر

gcc -S همان Assembly را که یک کمپایلر تولید می‌کند می‌نویسد. با O2- بهینه‌سازی‌ها را ...

website به نام Compiler Explorer در godbolt.org، کود Rust ،C یا Go را کنار Assembly ...



04

SIMD: چند مقدار یکجا

دستورهای SIMD چند مقدار را در یک قدم پردازش می‌کنند. یک register از نوع AVX، یعنی ymm، هشت ...

اینجا جایی است که Assembly دست‌نویس هنوز از کمپایلرها بهتر است: codec های ویدیو، رمزنگاری و ...

ASSEMBLY LANGUAGE · 10

simd.asm

```
; add 8 floats from a and b, store in c
    vmovups ymm0, [a]
    vmovups ymm1, [b]
    vaddps  ymm0, ymm0, ymm1
    vmovups [c], ymm0
; 8 additions in one instruction
```

05

نقشهٔ درس


1
صدا کردن printf
2
ساختن با gcc
3
صدا کردن Assembly از C
4
خواندن خروجی کمپایلر
5
چند مقدار یکجا SIMD:

منابع و مطالعهٔ بیشتر

ASSEMBLY LANGUAGE · 10

hello_c.asm

```
extern printf
section .data
    fmt db "%s scored %d", 10, 0
    who db "Mariam", 0
section .text
global main
main:
    sub rsp, 8            ; align stack
    lea rdi, [rel fmt]
    lea rsi, [rel who]
    mov edx, 95
    xor eax, eax          ; no vector args
    call printf wrt ..plt
    add rsp, 8
    xor eax, eax
    ret
```

ASSEMBLY LANGUAGE · 10

main.c

```
#include <stdio.h>

long sum_array(const long *a, long n);    // in sum.asm

int main(void) {
    long scores[] = {90, 85, 77, 92};
    printf("%ld\n", sum_array(scores, 4));    // 344
}

// $ nasm -f elf64 sum.asm && gcc main.c sum.o -o app
```

تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity