درس ۳

ASM

سلام، syscall

hello.asm

```
mov rax, 1
mov rdi, 1
mov rsi, msg
syscall
```

یک کورس رایگان از MyWebUniversity

زبان Assembly

درس ۳

اولین پروگرام شما: section ها و system call ها

.data section و .text، _start، label ها، system call های write و exit



section ها

یک پروگرام به section ها تقسیم می‌شود.
section .data دادهٔ مقداردهی‌شده را نگه می‌دارد،
...

سیستم عامل هر section را با اجازه‌های درست به حافظه بار می‌کند: کود می‌تواند اجرا شود، اما ...

ASSEMBLY LANGUAGE · 3

section	نگه می‌دارد
.data	متغیرهای مقداردهی‌شده
.bss	buffer های صفرشده
.rodata	ثابت‌های فقط خواندنی
.text	دستورها (کود)

01

label_، start ها و comment ها

یک label یک نام است که بعد از آن یک کالن می‌آید. یک آدرس را در پروگرام نشانی می‌کند. start_ ...

یک سیمی‌کالن یک comment را شروع می‌کند. با سخاوت comment بنویسید: Assembly بعداً سخت خوانده ...

ASSEMBLY LANGUAGE · 3

hello.asm

```
section .data
    msg     db  "Salaam, World!", 10    ; 10 = newline
    msglen  equ $ - msg                  ; length = 15

section .text
    global _start                        ; tell ld where to begin
_start:                                  ; a label
```

02

system call ها

پروگرام‌ها با system call ها از kernel خدمات می‌خواهند. شمارهٔ call را در rax، آرگومان‌ها ...

write شمارهٔ یک است: file descriptor، آدرس buffer و طول. exit شمارهٔ شصت است: کود خروج.

ASSEMBLY LANGUAGE · 3

call	rax	آرگومان‌ها
read	0	rdi=fd, rsi=buf, rdx=count
write	1	rdi=fd, rsi=buf, rdx=count
open	2	rdi=path, rsi=flags
exit	60	rdi=status

03

دستورها

mov یک مقدار را به یک register کاپی می‌کند.
rax را برای write یک، rdi را برای standard ...

بعد با کود صفر خارج می‌شویم. xor کردن rdi با خودش یک راه سریع و رایج برای صفر کردن یک ...

ASSEMBLY LANGUAGE · 3

hello.asm

```
_start:
    mov rax, 1          ; syscall: write
    mov rdi, 1          ; fd 1 = stdout
    mov rsi, msg        ; address of the text
    mov rdx, msglen     ; number of bytes
    syscall

    mov rax, 60         ; syscall: exit
    xor rdi, rdi        ; status 0
    syscall
```

04

assemble، link و اجرا

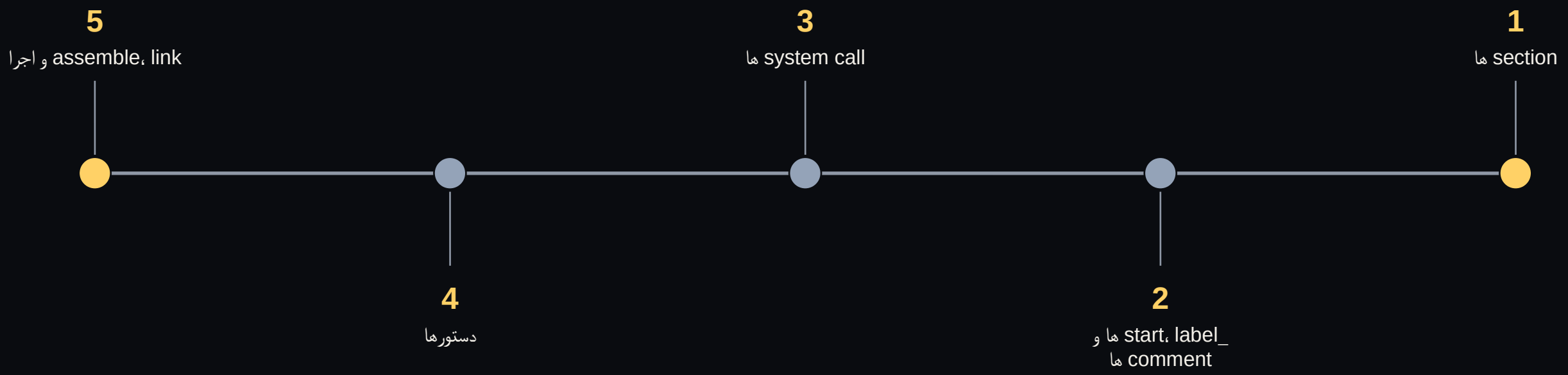
nasm -f elf64 یک فایل object شصت و چهار بیتی Linux می‌سازد. ld آن را به یک فایل اجرایی link ...

این فرمان‌ها را در یک Makefile بگذارید تا یک کلمه همه چیز را دوباره بسازد.



05

نقشهٔ درس


5
assemble، link و اجرا
4
دستورها
3
system call ها
2
start، label_ ها و
comment ها
1
section ها

منابع و مطالعهٔ بیشتر



ASSEMBLY LANGUAGE · 3

section	نگه می‌دارد
.data	متغیرهای مقداردهی‌شده
.bss	buffer های صفرشده
.rodata	ثابت‌های فقط خواندنی
.text	دستورها (کود)



ASSEMBLY LANGUAGE · 3

call	rax	آرگومان‌ها
read	0	rdi=fd, rsi=buf, rdx=count
write	1	rdi=fd, rsi=buf, rdx=count
open	2	rdi=path, rsi=flags
exit	60	rdi=status

تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity