LESSON 3

ASM

Hello, Syscall

hello.asm

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

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ASSEMBLY LANGUAGE

LESSON 3

Your First Program: Sections and System Calls

section .data and .text, _start, labels, the write and exit system calls

Sections

Section	Holds
.data	initialized variables
.bss	zeroed buffers
.rodata	read-only constants
.text	instructions (code)

- A program is divided into sections. section dot data holds initialized data, dot bss holds …
- The operating system loads each section into memory with the right permissions: code can …

01

_start, Labels and Comments

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
```

- A label is a name followed by a colon. It marks an address in the program. _start is where …
- A semicolon starts a comment. Comment generously: assembly is hard to read later.

02

System Calls

ASSEMBLY LANGUAGE · 3

Call	rax	Arguments
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

- Programs ask the kernel for services with system calls. Put the call number in rax, the …
- write is number one: file descriptor, buffer address and length. exit is number sixty: the …

03

The Instructions

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
```

- mov copies a value into a register. We set rax to one for write, rdi to one for standard …
- Then we exit with code zero. xor rdi with itself is a fast, common way to set a register to …

04

Assemble, Link and Run

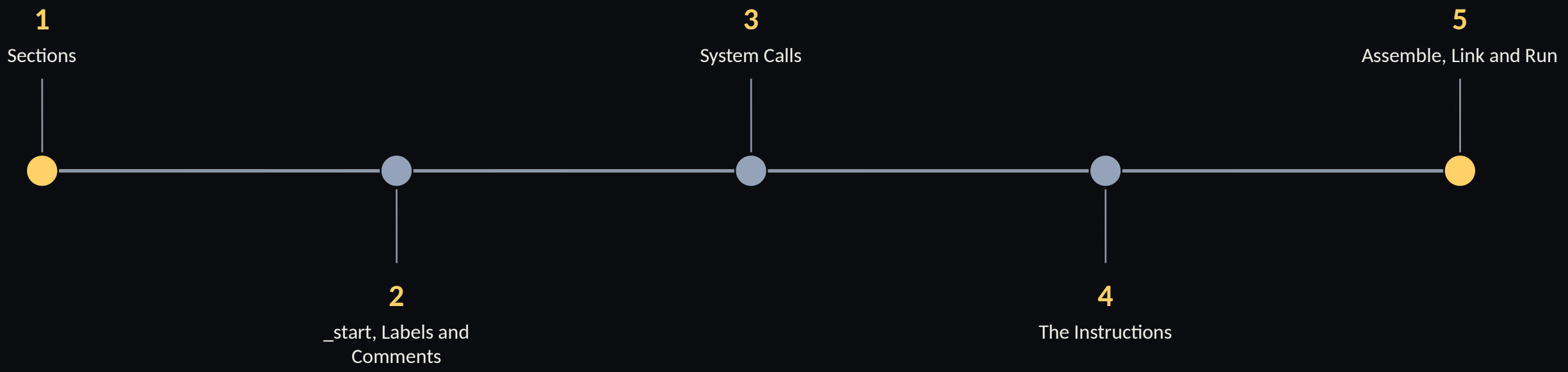
terminal

```
$ nasm -f elf64 hello.asm -o hello.o
$ ld hello.o -o hello
$ ./hello
Salaam, World!
$ echo $?
0
$ ls -l hello      # only about 9 KB
```

- nasm dash f elf64 creates a sixty-four-bit Linux object file. ld links it into an …
- Put these commands in a Makefile so one word rebuilds everything.

05

Lesson Map


1
Sections
2
_start, Labels and Comments
3
System Calls
4
The Instructions
5
Assemble, Link and Run

Sources and Further Reading

Section	Holds
.data	initialized variables
.bss	zeroed buffers
.rodata	read-only constants
.text	instructions (code)

Call	rax	Arguments
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

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari