LESSON 9



ASM

Strings & Bytes

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ASSEMBLY LANGUAGE

LESSON 9

Strings, Arrays and Input

Byte loops, string length, reading input, converting numbers to text and rep string instruct

String Length

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
```

- A C string is bytes ending with a zero byte. To find its length, walk forward until you …
- This is exactly what the C function strlen does.

01

Changing Bytes: Uppercase

ASSEMBLY LANGUAGE · 9

upper.asm

```
to_upper:                ; rdi = string
.loop:
    mov al, [rdi]
    test al, al
    jz  .done
    cmp al, 'a'
    jb  .skip
    cmp al, 'z'
    ja  .skip
    sub al, 32
    mov [rdi], al
.skip:
    inc rdi
    jmp .loop
.done: ret
```

- Load a byte, test whether it is a lowercase letter between a and z, and if so subtract …
- Then store it back and move to the next byte.

02

Reading Input

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
```

- The read system call, number zero, reads from standard input into a buffer and returns how …
- Reserve the buffer in the bss section, then echo the text back with write.

03

Printing a Number

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
```

- There is no print-number instruction. Divide by ten repeatedly: each remainder is one …
- Digits come out in reverse order, so fill the buffer from the end.

04

rep String Instructions

ASSEMBLY LANGUAGE · 9

memcpy.asm

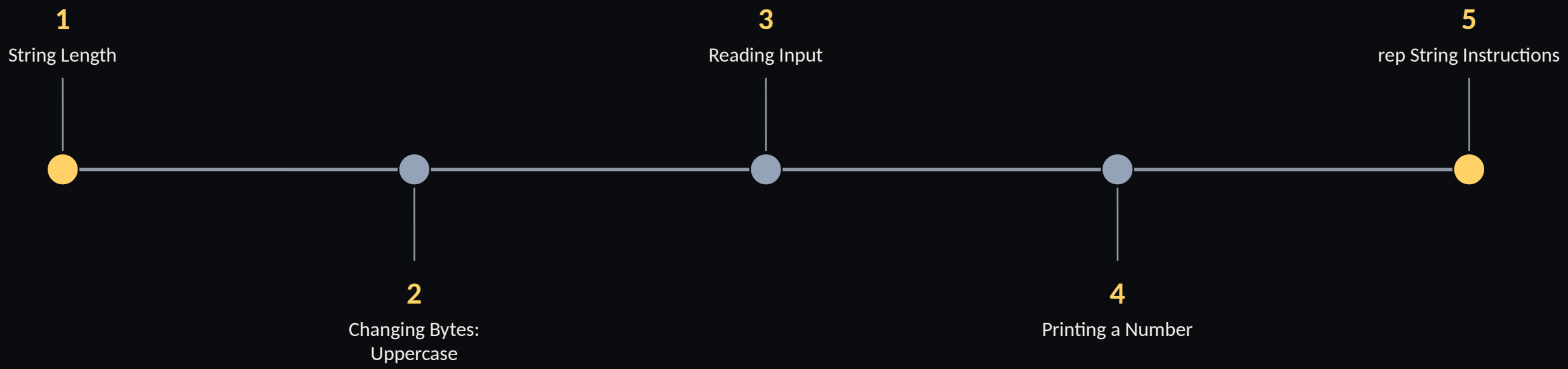
```
; memcpy(dst=rdi, src=rsi, n=rdx)
my_memcpy:
    mov rax, rdi         ; return dst
    mov rcx, rdx
    rep movsb
    ret

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

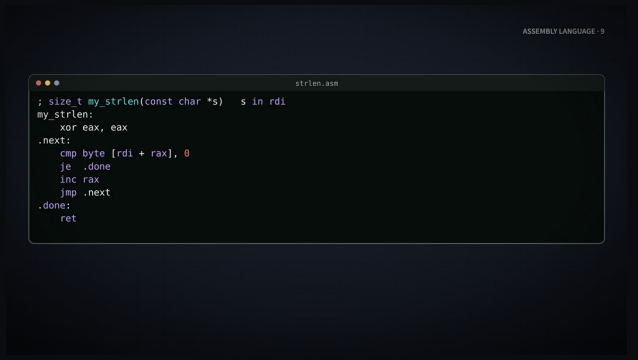
- x86 has special string instructions. rep movsb copies rcx bytes from rsi to rdi, and rep …
- repne scasb searches for a byte. Modern CPUs make rep movsb very fast for copying memory.

05

Lesson Map


1
String Length
2
Changing Bytes:
Uppercase
3
Reading Input
4
Printing a Number
5
rep String Instructions

Sources and Further Reading





Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari