LESSON 10



ASM

Assembly + C

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ASSEMBLY LANGUAGE

LESSON 10

Mixing Assembly and C

Calling printf from assembly, calling assembly from C, linking with gcc, and SIMD

Calling printf

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
```

- Declare printf as extern and define main instead of _start, because the C runtime will call …
- For variadic functions like printf, set al to the number of vector registers used, zero here.

01

Building with gcc

terminal

```
$ nasm -f elf64 hello_c.asm -o hello_c.o
$ gcc hello_c.o -o hello_c
$ ./hello_c
Mariam scored 95
```

- Assemble with NASM as before, but link with gcc. It adds the startup code and the C library.
- The rel and plt keywords make position-independent code, which modern Linux requires by …

02

Calling Assembly from C

main.c

```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
```

- Write a fast function in assembly, declare it in C with its prototype, and link both object …
- Because both follow the System V ABI, C just calls it like any other function.

03

Reading Compiler Output

C

```
long times5(long x) {
    return x * 5;
}
```

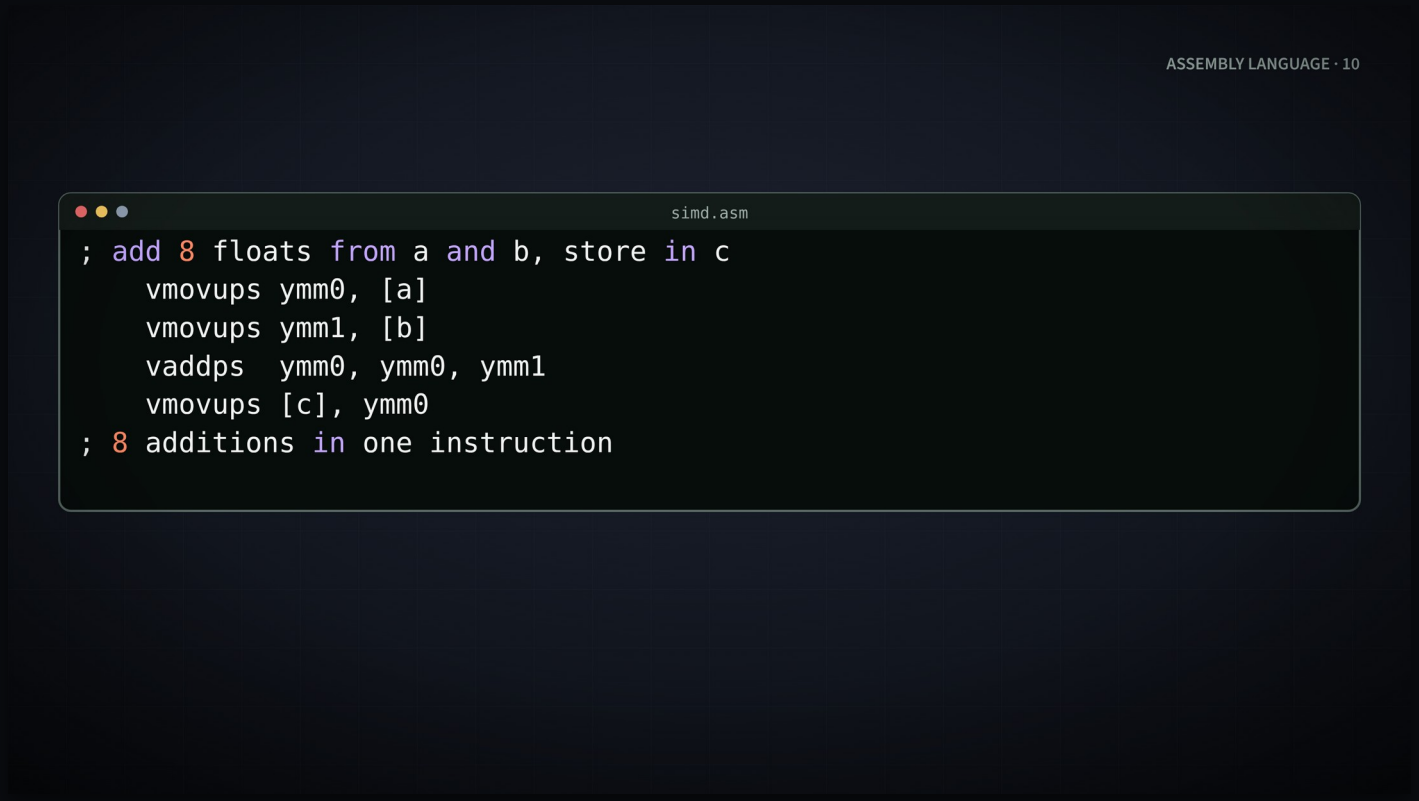
gcc -O2 -S

```
times5:
lea rax, [rdi+rdi*4]
ret
```

- gcc dash S writes the assembly a compiler produces. With dash O2 you see optimizations: …
- The Compiler Explorer website at godbolt dot org shows C, Rust or Go next to the generated …

04

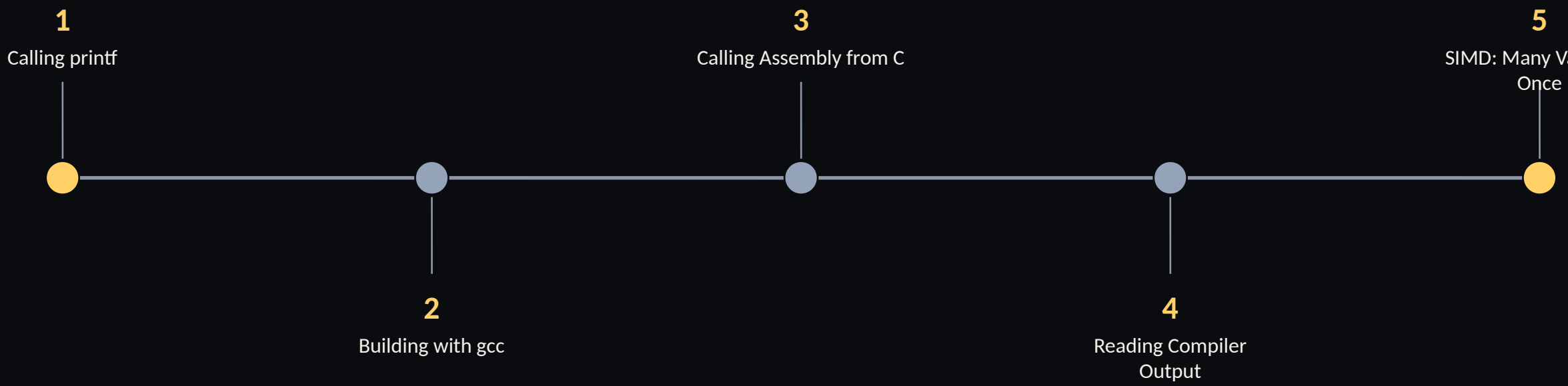
SIMD: Many Values at Once



- SIMD instructions process several values in one step. An AVX register, ymm, holds eight …
- This is where hand-written assembly still beats compilers: video codecs, cryptography and …

05

Lesson Map


1
Calling printf
2
Building with gcc
3
Calling Assembly from C
4
Reading Compiler Output
5
SIMD: Many Values at Once

Sources and Further Reading

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
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari