

Chamada de função externa

;arquivo fonte **test.asm**

; Define printf como uma função externa

extern printf

SECTION .DATA

msg: db "Hello world", 0 ; Zero é o '\0' de uma string C

fmt: db "%s", 10, 0 ; printf format string follow by a newline(10) and a null terminator(0), "\n",'0'

SECTION .TEXT

global main ;quando se usa gcc para linkar usa-se main

main:

push rbp ; push stack, para evitar um erro de programação

; Passa os parâmetros e chama as funções C

mov rdi,fmt ;1o parâmetro

mov rsi,msg ;2o parâmetro

mov rax,0

call printf

pop rbp ; Pop stack

mov rax,0 ; Exit code 0

ret ; Return

;compilar e linkar:

;nasm test.asm -f elf64

;gcc test.o -o test, linka com gcc

;função e parâmetros para 64 bits

;foo(rdi,rsi,rdx,rcx,r8,r9), os demais na stack

Chamada de função externa

```
;arquivo fonte mainAsm.asm
;usa o arquivo addC.c
extern add ;no arquivo add.c com a funcao add(int, int)
extern printf ;chamada da função printf da linguagem C
extern exit ;chamada da função exit da linguagem C
section .data
    format db "%d", 10, 0 ; 10 é o mesmo que "\n" em C
section .text
    global main ; quando usa gcc, melhor ter rotulo main
main:
    push rbp
    mov rdi,6 ;primeiro parâmetro
    mov rsi,2 ;segundo parâmetro
    call add ;add(2, 6)
    mov rdi,format ;Nos parâmetros para C printf, em rdi
passamos o formato
    mov rsi, rax ;depois passamos o que vai imprimir
    call printf ; printf(format, eax)
    pop rbp
    xor rdi, rdi ; zera rdi
    call exit ; exit(0)
```

Função e parâmetros para 64 bits

foo(rdi,rsi,rdx,rcx,r8,r9)

os demais na stack

Passos para compilar e linkar

1) nasm -f elf64 mainAsm.asm, gera mainAsm.o

2) gcc -Wall -c addC.c.

atenção ao -c, que apenas compila,
gerando add.o

3) gcc mainAsm.o addC.o -o nasmMainCallC

Junta mainsAsm e addC, gerando o
executável nasmMainCallC