

10.09.2024

Laboratório 6

Theo Canuto - 2311293

Professor Raúl Renteria

INF1018 - 3WA

2.

```
    movl $0, %ebx /* ebx = 0; */
    movl $0, %r13d /* ecx = 0 -> variavel soma */
    movq $nums, %r12 /* r12 = &nums */

L1:
    cmpl $4, %ebx /* if (ebx == 4) ? */
    je L2 /* goto L2 */

    movl (%r12), %eax /* eax = *r12 */
    addl %eax, %r13d /* %r13d = %r13d + *eax */

    /*****
    /* este trecho imprime o valor de %eax (estraga %eax) - opcional */
    movq $Sf, %rdi /* primeiro parametro (ponteiro)*/
    movl %eax, %esi /* segundo parametro (inteiro) */
    call printf /* chama a funcao da biblioteca */
    *****/

    addl $1, %ebx /* ebx += 1; */
    addq $4, %r12 /* r12 += 4; */

    jmp L1 /* goto L1; */

L2:
    movq $Sf, %rdi /* imprime a soma final */
    movl %r13d, %esi
    call printf /* chama a funcao da biblioteca */
```

3.

```
L1:
    cmpl $4, %ebx /* if (i == 4) ? */
    je L2 /* Se sim, vá para L2 (fim do loop) */

    movl (%r12), %eax /* eax = *p (carrega valor do array nums) */
    testl $1, %eax /* Testa se o bit menos significativo é 1 (impar) */
    jnz Lskip /* Se o número é ímpar, pula a impressão */

    /* Código de impressão (para números pares) */
    movq $Sf, %rdi /* Primeiro argumento para printf (endereco da string de formato) */
    movl %eax, %esi /* Segundo argumento para printf (valor a ser impresso) */
    call printf /* Chama a função printf */

Lskip:
    addl $1, %ebx /* i++ (incrementa o índice) */
    addq $4, %r12 /* p++ (avança o ponteiro para o próximo elemento do array) */
    jmp L1 /* Volta para o início do loop */
```

4.

```
.data
nums: .byte 10, -21, -30, 45
Sf: .string "%d\n" # string de formato para printf

.text
.globl main
main:

/*****
/* mantenha este trecho aqui e nao mexa - prologo !!! */
pushq %rbp
movq %rsp, %rbp
subq $16, %rsp
movq %rbx, -8(%rbp)
movq %r12, -16(%rbp)
*****/

movl $0, %ebx /* ebx = 0; */
movq $nums, %r12 /* r12 = &nums */

L1:
cmpl $4, %ebx /* if (ebx == 4) ? */
je L2 /* goto L2 */

movsbl (%r12), %eax /* Carrega o byte de nums e faz extensao com sinal para 32 bits */

/*****
/* este trecho imprime o valor de %eax (estraga %eax) */
movq $Sf, %rdi /* primeiro parametro (ponteiro) */
movl %eax, %esi /* segundo parametro (inteiro) */
call printf /* chama a funcao da biblioteca */
*****/

addl $1, %ebx /* ebx += 1; */
addq $1, %r12 /* r12 += 1; avanca para o proximo byte */
jmp L1 /* goto L1; */
```