P2 INF1018

Daniel Stulberg Huf – 1920468 – Turma 3WB

QUESTÃO 3

/\*

DICIONARIO

Reg Variavel Desc

RIP próxima instrução

RSP ponteiro para topo da pilha

RBP ponteiro para base do meu RA

%rdi struct Y \*v

%xmm0 acc

Campos da struct -> c PP PP PP PP PP PP PP d d d d d d d d prox prox prox prox prox prox prox prox

\*/

.text

.globl boofoo

boofoo:

push %rbp /\* Salvando o base pointer da chamadora\*/

movq %rsp, %rbp /\* Criando a base do meu RA \*/

subq $96, %rsp

movq %r12, -8(%rbp) /\* salvando o valor de r12 da função chamadora no RA \*/

movq %rbx, -16(%rbp) /\* salvando o valor de rbx da função chamadora no RA \*/

movq $0, %r10 /\* r10 = 0 \*/

cvtsi2ssq %r10, %xmm0 /\* converte 0 para float e passa para acc \*/

movl $0, %r11d /\* i = 0 \*/

FOR:

cmpl $10, %r11d /\* compara 10 com i \*/

jge WHILE

/\* w[i] = 0.0 \*/

movq $0, %r10 /\* r10 = 0 \*/

cvtsi2ssq %r10, %xmm1 /\* xmm1 = 0.0 \*/

movl %r11d, %r12d /\* r12d = i \*/

imull $8, %r12d /\* r12d = r12d \* 8 \*/

movslq %r12d, %rsi /\* estendo %r12d para 64 bits \*/

addq -96(%rbp), %rsi /\* rsi <- &w[i] \*/

movsd %xmm1, (%rsi) /\* w[i] = 0.0 \*/

incl %r11d /\* i++ \*/

jmp FOR

WHILE:

cmpq $0, %rdi /\* compara NULL com v \*/

je ANTES\_SEGUNDO\_FOR

cmpl $10, (%rdi) /\* compara v->ind com 10 \*/

jge DEPOIS\_IF

/\* w[v->ind] = v-> purp \*/

movsd 8(%rdi), %xmm2 /\* xmm2 = v->purp \*/

movl (%rdi), %ebx /\* ebx = v->ind \*/

imull $8, %ebx /\* (v->ind) = (v->ind)\*8 -> tamanho do double \*/

movslq %ebx, %rsi /\* estendo ebx para 64 bits \*/

addq -96(%rbp), %rsi /\* rsi <- &w[v->ind] \*/

movsd %xmm2, (%rsi) /\* w[v->ind] = v->purp \*/

DEPOIS\_IF:

movq 16(%rdi), %rdi /\* v = v-> prox \*/

jmp WHILE

ANTES\_SEGUNDO\_FOR:

movl $0, %r11d /\* r11d (i) = 0 \*/

SEGUNDO\_FOR:

cmpl $10, %r11d /\* compara i com 10 \*/

jge FIM

/\* acc += (float) w[i] \*/

movl %r11d, %r12d /\* r12d recebe i \*/

imull $8, %r12d /\* r12d(i) = r12d \* 8 (tamanho do double) \*/

movslq %r12d, %rsi /\* estendo r12d para 64 bits \*/

addq -96(%rbp), %rsi /\* rsi <- &w[i] \*/

cvtsi2ssq (%rsi), %xmm3 /\* (float) w[i] \*/

addss %xmm0, %xmm3 /\* acc += (float) w[i] \*/

incl %r11d /\* i++ \*/

jmp SEGUNDO\_FOR

FIM:

movq -8(%rbp), %r12 /\* restaura r12 \*/

movq -16(%rbp), %rbx /\* restaura rbx \*/

leave

ret