

# Parte 1 – Exercícios de Software Básico

## 1.1 Módulo 1 – Compiladores

### Questões Teóricas

### Questões Práticas

## 1.2 Módulo 2 – Assembly x86-64

### Questões Teóricas

1.2.1. Os itens abaixo possuem instruções de programas Assembly IA-32 (em modo nativo) que utilizam diversos modos de endereçamento. Classifique cada item como correto ou errado, e justifique o que estiver errado.

- (a) `mov EAX, 10`
- (b) `mov [M], AL`
- (c) `mov AL, [CS + ESI + array]`
- (d) `mov vetor[1], 0`
- (e) `add AX, [X + ECX]`
- (f) `mov ESI, vetor + EBX`
- (g) `inc WORD [inicio + EBX*8 + ESI]`
- (h) `mov [EBX + ESI*4], DWORD 5`
- (i) `dec BYTE [BL]`
- (j) `add [x + 1], AL`
- (k) `mov EAX, [array + ECX*8 + EBX]`
- (l) `mov [EAX*8 + 1], 5`
- (m) `mov BL, AX`
- (n) `cmp [ESI], 10`
- (o) `adc AL, AH`

1.2.2.

### Questões Práticas

1.2.3. O programa abaixo realiza a cópia de um vetor de *double words*, convertendo-o de *little endian* para *big endian*. Complete o programa, indicando as instruções dos espaços em branco (cada espaço deve ser preenchido com uma única instrução).

Listing 1.2.1: código bala

```
1  SIZE EQU 6
2  section .data
3  little dd 42434445h, 45454545h, 4A4B4C4Dh, 414D4E4Fh, 46454948h, 4C474D46h
4
5  section .bss
6  big resd SIZE
7  temp resd 1
8
9  section .start
10 global _start
11 _start:
12 mov ecx, SIZE
13 mov eax, little
14 mov esi, big
15 laco1: mov ebx, esi
16 -----
17 laco2: mov dl, [eax]
18 mov [ebx], dl
19 dec ebx
20 inc eax
21 -----
22 jae laco2
23 add esi, 4
24 -----
25 cmp ecx, 0
26 -----
27 done: mov eax, 1
28 mov ebx, 0
29 int 80h
```