

Programação Estruturada Em Assembly

Programação Aplicada a Ciência da Computação.

Prof. Dr. Eduardo S. Pereira.

[http:](http://eduardopereira.upcursosetreinamentosonline.com/)

[//eduardopereira.upcursosetreinamentosonline.com/](http://eduardopereira.upcursosetreinamentosonline.com/)

7 de maio de 2018

1 Introdução

2 Programação Estruturada

■ Programação Sequencial

3 Estrutura de Controle

■ Controle de Decisão

■ Controle de Iteração

- Enquanto/faça (while)
- Repita/enquanto (do while)
- para (for each)

4 Sub-rotinas como funções

Introdução

- Uma boa prática é projetar um programa em unidades pequenas e bem documentadas
- Documentação deve se referir a conceitos gerais e como usar o programa
- Em linguagens de baixo nível, como Assembly, acaba sendo necessário explicar em mais detalhes, como o código funciona.

Introdução

- Uma boa prática é projetar um programa em unidades pequenas e bem documentadas
- Documentação deve se referir a conceitos gerais e como usar o programa
- Em linguagens de baixo nível, como Assembly, acaba sendo necessário explicar em mais detalhes, como o código funciona.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Sub-rotinas
como funções

Introdução

- Uma boa prática é projetar um programa em unidades pequenas e bem documentadas
- Documentação deve se referir a conceitos gerais e como usar o programa
- Em linguagens de baixo nível, como Assembly, acaba sendo necessário explicar em mais detalhes, como o código funciona.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Programação Estruturada

- **Formaliza a ideia de dividir em blocos**
- Força o programador a saber exatamente o estado do programa antes e depois de cada bloco.
- Faz uso de três estruturas de controle para a construção de um programa: Sequência, Seleção e Repetição.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Programação Estruturada

- Formaliza a ideia de dividir em blocos
- Força o programador a saber exatamente o estado do programa antes e depois de cada bloco.
- Faz uso de três estruturas de controle para a construção de um programa: Sequência, Seleção e Repetição.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Programação Estruturada

- Formaliza a ideia de dividir em blocos
- Força o programador a saber exatamente o estado do programa antes e depois de cada bloco.
- Faz uso de três estruturas de controle para a construção de um programa: Sequência, Seleção e Repetição.

Programação Estruturada

- Foi apresentada no início dos anos de 1970 pelo suíço Niklaus Wirth.
- O domínio da técnica de programação estruturada é a base para aprendizado de outras técnicas de programação, como modular e orientada a objetos.
- Permite que os programas tenham um fluxo de execução de instruções mais claro e que facilita a compreensão lógica do programa,
- Além de permitir que o código do programa seja escrito com uma estrutura facilmente reconhecida por quem fará a manutenção do programa

Programação Estruturada

- Foi apresentada no início dos anos de 1970 pelo suíço Niklaus Wirth.
- O domínio da técnica de programação estruturada é a base para aprendizado de outras técnicas de programação, como modular e orientada a objetos.
- Permite que os programas tenham um fluxo de execução de instruções mais claro e que facilita a compreensão lógica do programa,
- Além de permitir que o código do programa seja escrito com uma estrutura facilmente reconhecida por quem fará a manutenção do programa

Programação Estruturada

- Foi apresentada no início dos anos de 1970 pelo suíço Niklaus Wirth.
- O domínio da técnica de programação estruturada é a base para aprendizado de outras técnicas de programação, como modular e orientada a objetos.
- Permite que os programas tenham um fluxo de execução de instruções mais claro e que facilita a compreensão lógica do programa,
- Além de permitir que o código do programa seja escrito com uma estrutura facilmente reconhecida por quem fará a manutenção do programa

Programação Estruturada

- Foi apresentada no início dos anos de 1970 pelo suíço Niklaus Wirth.
- O domínio da técnica de programação estruturada é a base para aprendizado de outras técnicas de programação, como modular e orientada a objetos.
- Permite que os programas tenham um fluxo de execução de instruções mais claro e que facilita a compreensão lógica do programa,
- Além de permitir que o código do programa seja escrito com uma estrutura facilmente reconhecida por quem fará a manutenção do programa

Programação Estruturada

- Em Assembly, o que usamos efetivamente são comparações e uso de desvios incondicionais;
- Nesse caso, o uso de programação estruturada está mais ligado a prática do programador do que o paradigma da linguagem
- Assembly usa a programação imperativa, no qual a computação é tratada como ações, enunciados e comandos que mudam o estado de um programa.
- Representam ordens que são passadas para o processador.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Programação Estruturada

- Em Assembly, o que usamos efetivamente são comparações e uso de desvios incondicionais;
- Nesse caso, o uso de programação estruturada está mais ligado a prática do programador do que o paradigma da linguagem
- Assembly usa a programação imperativa, no qual a computação é tratada como ações, enunciados e comandos que mudam o estado de um programa.
- Representam ordens que são passadas para o processador.

Programação Estruturada

- Em Assembly, o que usamos efetivamente são comparações e uso de desvios incondicionais;
- Nesse caso, o uso de programação estruturada está mais ligado a prática do programador do que o paradigma da linguagem
- Assembly usa a programação imperativa, no qual a computação é tratada como ações, enunciados e comandos que mudam o estado de um programa.
- Representam ordens que são passadas para o processador.

Programação Estruturada

- Em Assembly, o que usamos efetivamente são comparações e uso de desvios incondicionais;
- Nesse caso, o uso de programação estruturada está mais ligado a prática do programador do que o paradigma da linguagem
- Assembly usa a programação imperativa, no qual a computação é tratada como ações, enunciados e comandos que mudam o estado de um programa.
- Representam ordens que são passadas para o processador.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Estrutura de Sequencia

- Define que as instruções do programa são executadas sequencialmente, de cima para baixo, linha a linha, de forma sequencial.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Estrutura de Seleção

- Permite que o fluxo de execução das instruções seja executado de acordo com uma condição lógica que é avaliada e caso seja verdadeira, permite que uma ou um grupo de instruções seja executado.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

**Programação
Estruturada**

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Estrutura de Repetição

- Permite que uma ou um grupo de instruções seja executado repetidamente, de acordo com uma condição lógica.

Programação Sequencial

- O fluxo de controle de um programa estruturado é feito de bloco em bloco, sequencialmente, em que o bloco pode ser:
 - Uma instrução, um conjunto de instruções,
 - um aninhamento de outros blocos, uma rotina (função), um programa.

Programação Sequencial

- O fluxo de controle de um programa estruturado é feito de bloco em bloco, sequencialmente, em que o bloco pode ser:
- Uma instrução, um conjunto de instruções,
- um aninhamento de outros blocos, uma rotina (função), um programa.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Programação
Sequencial

Estrutura de
Controle

Sub-rotinas
como funções

Programação Sequencial

- O fluxo de controle de um programa estruturado é feito de bloco em bloco, sequencialmente, em que o bloco pode ser:
- Uma instrução, um conjunto de instruções,
- um aninhamento de outros blocos, uma rotina (função), um programa.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Estrutura de Controle

- Na programação estruturada, a estruturação está baseada num pequeno número de estruturas de controle
 - i- Sequencial;
 - ii- Decisão;
 - iii- Seleção múltipla;
 - iv- Iteração do tipo enquanto-faça;
 - v- Iteração do tipo repita-enquanto;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- **Princípio da programação estruturada:**
- Qualquer algoritmo pode ser escrito combinando-se blocos formados pelas estruturas de controle.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

**Estrutura de
Controle**

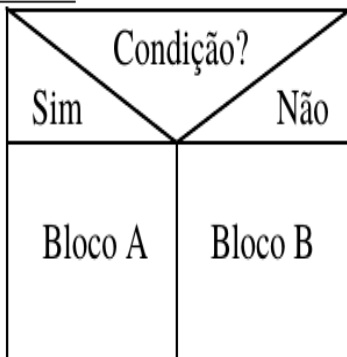
Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Estrutura de Controle

- Princípio da programação estruturada:
- Qualquer algoritmo pode ser escrito combinando-se blocos formados pelas estruturas de controle.

Controle de Decisão



Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

a- JE - Pula se Igual;

b- JNE - Pula se diferente ;

c- JG - Pula se Maior Que;

d- JGE - Pula se Maior ou Igual a;

e- JL - Pula se Menor que;

f- JLE - Pula se Menor ou Igual a;

g- JE - Pula Igual a;

h- JNE - Pula se diferente de;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

a- JE - Pula se Igual;

b- JNE - Pula se diferente ;

c- JG - Pula se Maior Que;

d- JGE - Pula se Maior ou Igual a;

e- JL - Pula se Menor que;

f- JLE - Pula se Menor ou Igual a;

g- JE - Pula Igual a;

h- JNE - Pula se diferente de;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

a- JE - Pula se Igual;

b- JNE - Pula se diferente ;

c- JG - Pula se Maior Que;

d- JGE - Pula se Maior ou Igual a;

e- JL - Pula se Menor que;

f- JLE - Pula se Menor ou Igual a;

g- JE - Pula Igual a;

h- JNE - Pula se diferente de;

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

- a- JE - Pula se Igual;
- b- JNE - Pula se diferente ;
- c- JG - Pula se Maior Que;
- d- JGE - Pula se Maior ou Igual a;
- e- JL - Pula se Menor que;
- f- JLE - Pula se Menor ou Igual a;
- g- JE - Pula Igual a;
- h- JNE - Pula se diferente de;

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

- a- JE - Pula se Igual;
- b- JNE - Pula se diferente ;
- c- JG - Pula se Maior Que;
- d- JGE - Pula se Maior ou Igual a;
- e- JL - Pula se Menor que;
- f- JLE - Pula se Menor ou Igual a;
- g- JE - Pula Igual a;
- h- JNE - Pula se diferente de;

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

- a- JE - Pula se Igual;
- b- JNE - Pula se diferente ;
- c- JG - Pula se Maior Que;
- d- JGE - Pula se Maior ou Igual a;
- e- JL - Pula se Menor que;
- f- JLE - Pula se Menor ou Igual a;
- g- JE - Pula Igual a;
- h- JNE - Pula se diferente de;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

- a- JE - Pula se Igual;
- b- JNE - Pula se diferente ;
- c- JG - Pula se Maior Que;
- d- JGE - Pula se Maior ou Igual a;
- e- JL - Pula se Menor que;
- f- JLE - Pula se Menor ou Igual a;
- g- JE - Pula Igual a;
- h- JNE - Pula se diferente de;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Sub-rotinas
como funções

Saltos condicionais

Alguns Saltos condicionais - (J - Jump)

- a- JE - Pula se Igual;
- b- JNE - Pula se diferente ;
- c- JG - Pula se Maior Que;
- d- JGE - Pula se Maior ou Igual a;
- e- JL - Pula se Menor que;
- f- JLE - Pula se Menor ou Igual a;
- g- JE - Pula Igual a;
- h- JNE - Pula se diferente de;

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

```
%include "io.inc"
SECTION .data
    msg1: db 'Escolha um Numero inteiro entre 0 e 5.',10, 0
    msgCerto: db "Parabens, vc acertou",10,0
    msgErrado: db "Nao foi dessa vez",10,0
    formatin: db "%d", 0
    usrinput: times 4 db 0 ; 32-bits integer = 4 bytes
SECTION .text
    extern scanf
    extern printf
    global CMAIN
```


Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

CMAIN:

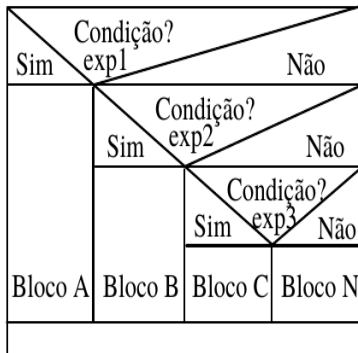
```
; Pedindo que o usuario digite um numero
push msg1
call printf
add esp, 4
; Armazenando o numero digitado na variavel usrinput
push usrinput
push formatin
call scanf
add esp, 8
```

```
;Acessando o valor da variavel, para comp.  
mov ebx, [usrinput]  
;Comparando o valor da variavel com o n. 2  
cmp ebx, 2  
;Se o valor digitado pelo usuário for igual a dois, va  
para o bloco if_block  
je if_block  
;Caso a variavel tenha valor diferente de 2, va para o  
bloco else_block  
jmp else_block
```

```
if_block:
    ; Se o usuario digitar 2, o bloco corrente sera
    ; executado
    ; e a mensagem armazenado em msgCerto serÃ; exibida
    push msgCerto
    call printf
    add esp, 4
    ; Como o bloco else_block nÃo deve ser executado,
    ; fazemos um salto para o end_if
    jmp end_if
else_block:
    ; Caso o valor digitado pelo usuÃrio seja diferente
    ; de 2, o bloco else_block sera executado
    ; E a mensagem contida em msgErrado serÃ; exibida
    push msgErrado
    call printf
    add esp, 4
    ; Ao finalizar esse bloco, fazemos o salto para o
    ; end_if.
    jmp end_if
end_if:
    ret
```

```
%include "io.inc"
SECTION .data
    msg1: db 'Escolha um Numero inteiro entre 0 e 5.',10,0
    msgCerto: db "Parabens, vc acertou",10,0
    msgErrado: db "Nao foi dessa vez",10,0
    formatin: db "%d",0
    usrinput: times 4 db 0 ; 32-bits integer = 4 bytes
SECTION .text
    extern scanf
    extern printf
    global CMAIN
CMAIN:
    push msg1
    call printf
    add esp, 4
    push usrinput
    push formatin
    call scanf
    add esp, 8
    mov ebx, [usrinput]
    cmp ebx, 2
    je if_block
    jmp else_block
if_block:
    push msgCerto
    call printf
    add esp, 4
    jmp end_if
else_block:
    push msgErrado
    call printf
    add esp, 4
    jmp end_if
end_if:
    xor eax, eax
    ret
```

Controle de Decisão - If/ else if / else



Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão
Controle de Iteração

Sub-rotinas
como funções

Exercício

- Construa um programa, em NASM, que permita o acesso a estrutura do tipo if/else if/else.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

Iteração

- Podemos ter basicamente dois tipos de controle de iteração:
 - Enquanto/faça (while);
 - Repita/enquanto (do while);

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

Iteração

- Podemos ter basicamente dois tipos de controle de iteração:
- Enquanto/faça (while);
- Repita/enquanto (do while);

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

Iteração

- Podemos ter basicamente dois tipos de controle de iteração:
- Enquanto/faça (while);
- Repita/enquanto (do while);

Enquanto/faça (while)

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

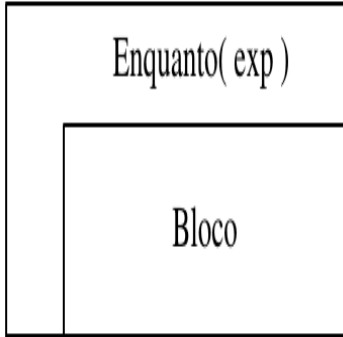
**Enquanto/faça
(while)**

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

Enquanto/faça (while)



while

- Nesse caso, primeiro precisamos verificar se a condição é verdadeira, antes de executar o bloco

Enquanto/faça

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

```
%include "io.inc"
SECTION .data
    msg: db 'ebx = %d.',10, 0
SECTION .text
    extern scanf
    extern printf
    global CMAIN
CMAIN:
    ;Inicializa ebx com 0;
    mov ebx, 0
    while:
        ;Compara o valor de ebx com 20,
        ;Se for igual a 20, va para o fim do while
        cmp ebx, 20
        je end_while

        ;Exibe o valor de ebx na tela
        push ebx
        push msg
        call printf
        add esp,8

        ;Soma ebx com 1 ebx += 1
        add ebx, 1
        ;Volta para o inicio do while
        jmp while
    end_while:
    xor eax, eax
    ret
```

Repita/enquanto (do while)

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

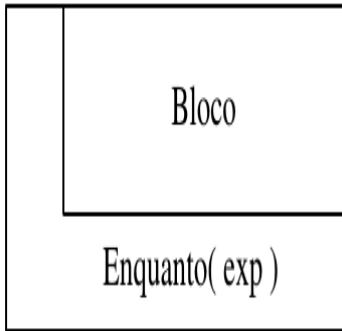
Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

Repita/enquanto (do while)



do while

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

do while

- Nesse caso, o bloco deve ser executado, para então verificar se o condicional é verdadeiro ou não.

Enquanto/faça

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

```
%include "io.inc"
SECTION .data
    msg: db 'ebx = %d.',10, 0
SECTION .text
    extern scanf
    extern printf
    global CMAIN
CMAIN:
    ;Inicializa ebx com 0;
    mov ebx, 0
    dowhile:
        ;Exibe o valor de ebx na tela
        push ebx
        push msg
        call printf
        add esp,8

        ;Soma ebx com 1. ebx += 1
        add ebx, 1

        ;Compara o valor de ebx com 20,
        ;Se for igual a 20, va para o fim do dowhile
        cmp ebx, 20
        je end_dowhile
        ;Volta para o inicio do dowhile
        jmp dowhile
    end_dowhile:
    xor eax, eax
    ret
```

for each

- Outra estrutura comum, em linguagens estruturadas modernas, é a presença da estrutura de valores em (for each).

for each

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Controle de Decisão

Controle de Iteração

Enquanto/faça
(while)

Repita/enquanto (do
while)

para (for each)

Sub-rotinas
como funções

```
%include "io.inc"
SECTION .data
    msg: db 'Valor = %d, Posicao %d',10, 0
    DADOS: dd 100, 320, 400, 500
           dd 600, 700, 800, 900
    DADOSLEN: equ ($-DADOS) / 4
SECTION .text
    extern scanf
    extern printf
    global CMAIN
CMAIN:
    ;Inicializa ebx com 0;
    ;ebx sera o contador for(int ebx=0; ebx < dados len; ebx++);
    mov ebx, 0
    for_each:
        ;Compara o valor de ebx com DADOSLEN,
        ;Se for igual a DADOSLEN, va para o fim do for_each
        cmp ebx, DADOSLEN
        je end_for_each

        ;Pega o valor contido no vetor de dados
        mov eax, [DADOS+4*ebx]

        ;Exibe o valor do dado e da posicao na tela
        push ebx
        push eax
        push msg
        call printf
        add esp,12

        ;Incrementa o ebx
        inc ebx
        ;Volta para o inicio do for each
        jmp for_each
    end_for_each:
    xor eax, eax
    ret
```

Exercício

- Crie um programa, em NASM, no qual uma estrutura de repetição poderá ser interrompida caso uma condição ocorra. Nesse caso, será necessário aninhar blocos de condicional if/else if/else dentro de blocos while ou do while.
- Crie um programa em NASM, que exibe na tela a soma dos valores armazenados num vetor.

Sub-rotinas

- Sub-rotinas são blocos de códigos que podem ser chamados usando a diretiva call.
- A sub-rotina tem como retorno o registrador eax, dessa forma, é preciso cuidado ao se trabalhar com esse registrador.
- Em Assembly temos acesso direto aos dados em memória, dessa forma, não fica muito claro a questão de escopo de variáveis.

Sub-rotinas

- Sub-rotinas são blocos de códigos que podem ser chamados usando a diretiva call.
- A sub-rotina tem como retorno o registrador eax, dessa forma, é preciso cuidado ao se trabalhar com esse registrador.
- Em Assembly temos acesso direto aos dados em memória, dessa forma, não fica muito claro a questão de escopo de variáveis.

Sub-rotinas

- Sub-rotinas são blocos de códigos que podem ser chamados usando a diretiva call.
- A sub-rotina tem como retorno o registrador eax, dessa forma, é preciso cuidado ao se trabalhar com esse registrador.
- Em Assembly temos acesso direto aos dados em memória, dessa forma, não fica muito claro a questão de escopo de variáveis.

```
%include "io.inc"
SECTION .data
    introducao: db 'Programa para calculo de potencia',10,0
    msg1: db 'Digite um valor inteiro',10, 0
    msg2: db 'Digite a potencia inteira',10, 0
    msgresult: db "Valor de %d elevado a %d e %d",10,0
    formatin: db "%d", 0
    valor: times 4 db 0 ; 32-bits integer = 4 bytes
    potc: times 4 db 0
    resultado: times 4 db 0
section .text
global CMAIN
```

Cálculo de Potência

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Sub-rotinas
como funções

```
CMAIN:
    push introducao
    call printf
    add esp,4

    push msg1
    call printf
    add esp,4

    push valor
    push formatin
    call scanf
    add esp, 8

    push msg2
    call printf
    add esp,4

    push potc
    push formatin
    call scanf
    add esp, 8

    call POTENCIA

    mov eax, [resultado]
    mov ebx, [potc]
    mov ecx, [valor]

    push eax
    push ebx
    push ecx
    push msgresult
    call printf
    add esp,16

    xor eax, eax
    ret
```

Cálculo de Potência

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Sub-rotinas
como funções

```
POTENCIA:
    mov ecx, [valor]
    mov eax, [valor]
    mov ebx, 1
    while:
        cmp ebx, [potc]
        je end_while
        mul ecx
        inc ebx
        jmp while
    end_while:

    mov [resultado], eax
    ret
```


Cálculo de Potência

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Sub-rotinas
como funções

```
include "io.inc"
SECTION .data
introducao: db "Programa para calculo de potencia",10,0
msg1: db "Digite um valor inteiro",10, 0
msg2: db "Digite a potencia inteira",10, 0
msgresult: db "Valor de %d elevado a %d e %d",10,0
formatin: db "%d", 0
valor: times 4 db 0 ; 32-bits integer = 4 bytes
potr: times 4 db 0
resultado: times 4 db 0

section .text
global MAIN
CHAIN:
    push introducao
    call printf
    add esp,4

    push msg1
    call printf
    add esp,4

    push valor
    push formatin
    call scanf
    add esp, 8

    push msg2
    call printf
    add esp,4

    push potr
    push formatin
    call scanf
    add esp, 8

    call POTENCIA

    mov eax, [resultado]
    mov ebx, [potr]
    mov ecx, [valor]

    push eax
    push ebx
    push ecx
    push msgresult
    call printf
    add esp,16

    mov eax, eax
    ret

POTENCIA:
    mov ecx, [valor]
    mov eax, [valor]
    mov ebx, 1
    while:
        cmp ebx, [potr]
        je end_while
        mul ecx
        inc ebx
        jmp while
    end_while:
    mov [resultado], eax
    ret
```

Exercício

- Crie um programa, em NASM, usando sub-rotinas, para o cálculo da área de triângulo, retângulo e trapézio.
- O programa deve perguntar primeiramente ao usuário, qual das três figuras ele quer calcular a área.
- Em seguida, dado o tipo de figura, ele deve fazer o pedido das entradas necessárias para o cálculo de área.
- Ao final deverá ser exibida o valor da área e para qual figura essa área foi calculada. O cálculo de cada área deverá ser escrito em sub-rotinas distintas.

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Introdução

Programação
Estruturada

Estrutura de
Controle

Sub-rotinas
como funções

Grato

MUITO OBRIGADO.

Referências

- MARTHA, L. F. Fundamentos de Programação Estruturada em C. Disponível em http://webserver2.tecgraf.puc-rio.br/ftp_pub/lfm/CIV2801ProgramC.pdf. Acessado em 7 de Maio de 2018.
- SIQUEIRA, F. Programação Estruturada. Disponível em <https://sites.google.com/site/unipliprogramacaoestruturada/artigos/o-que-e-programacao-estruturada>. Acessado em 7 de Maio de 2018.

