

# Ling. de Máquina e Ling. de Montagem Assembly

Programação Aplicada a Ciência da Computação.

Prof. Dr. Eduardo S. Pereira.

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

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

23 de fevereiro de 2018

## 1 Introdução

## 2 Compiladores

- A linguagem de Máquina
- A linguagem Assembly
- Hello Word - 1
- Hello Word - 2

## 3 Exercícios

## 4 Bibliografia

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Linguagens de Máquina

- Instruções que o processador é capaz de executar.
- São representadas por sequências bits, limitadas pelo número de bits do registrador principal.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Linguagens de Máquina

- Instruções que o processador é capaz de executar.
- São representadas por sequências bits, limitadas pelo número de bits do registrador principal.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Linguagens de Montagem

- Notação Legível para humanos do código de máquina de uma arquitetura específica.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Tradutor

- Converte a linguagem fonte na linguagem alvo

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Montador

- Tradutor que tem como linguagem fonte a linguagem de montagem e a linguagem de máquina é a linguagem alvo.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

**Introdução**

Compiladores

Exercícios

Bibliografia

## Vantagens do Assembly

- Permite o acesso direto a recursos de hardwares.
- Programar em Assembly permite ganhar conhecimento profundo de como computadores funcionam.



## Vantagens do Assembly

- Permite o acesso direto a recursos de hardwares.
- Programar em Assembly permite ganhar conhecimento profundo de como computadores funcionam.

## Construindo um cavalo

### HOW TO BUILD A HORSE WITH PROGRAMMING

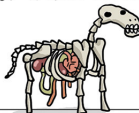
BY  toggl  
Goon Squad

C++

YOU BUILT A HORSE



IT'S UGLY AS HELL AND HAS  
LOTS OF DANGLING PARTS ,  
BUT IT GETS THE JOB DONE



ASSEMBLY

THE HORSE TURNS OUT  
A LITTLE BASIC



BUT BOY CAN IT RUN!



Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Compilador

- Programa que lê um programa escrito numa **linguagem fonte** e o traduz num programa equivalente na **linguagem alvo**.
  - Linguagem de máquina: aquilo que o computador entende
  - Linguagem de programação: pensada para humanos, mais alto o nível, mais próximo da linguagem humana;

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Compilador

- Programa que lê um programa escrito numa **linguagem fonte** e o traduz num programa equivalente na **linguagem alvo**.
- Linguagem de máquina: aquilo que o computador entende
- Linguagem de programação: pensada para humanos, mais alto o nível, mais próximo da linguagem humana;

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Compilador

- Programa que lê um programa escrito numa **linguagem fonte** e o traduz num programa equivalente na **linguagem alvo**.
- Linguagem de máquina: aquilo que o computador entende
- Linguagem de programação: pensada para humanos, mais alto o nível, mais próximo da linguagem humana;

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

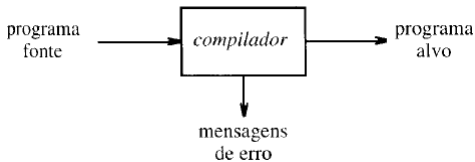
Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Compilador



Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Compiladores

- De uma passagem;
- De passagens múltiplas;
- De carregar e executar;
- Depuradores ou otimizantes;

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Além do compilador

- Tradutores: Traduz codificação em linguagem de alto nível para linguagem de alto nível: C# → Java. Python → C;
- Montadores: baixo Nível para baixo nível (Assembly → .exe);
- Interpretadores: realiza a tradução do código-fonte e também o executa. (Browser → HTML);



Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Geração de Código

- A fase final do compilador é gerar código alvo: código de máquina realocável ou código de montagem;
- São selecionadas as localizações de memória para cada variável.
- Instruções intermediárias são, cada uma, traduzidas numa sequência de instruções de máquina que realiza a mesma tarefa.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Geração de Código

- A fase final do compilador é gerar código alvo: código de máquina realocável ou código de montagem;
- São selecionadas as localizações de memória para cada variável.
- Instruções intermediárias são, cada uma, traduzidas numa sequência de instruções de máquina que realiza a mesma tarefa.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

**Compiladores**

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

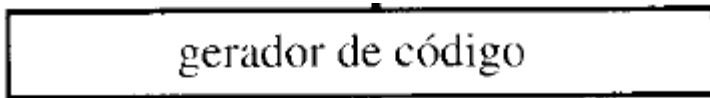
Hello Word - 2

Exercícios

Bibliografia

## Geração de Código

- A fase final do compilador é gerar código alvo: código de máquina realocável ou código de montagem;
- São selecionadas as localizações de memória para cada variável.
- Instruções intermediárias são, cada uma, traduzidas numa sequência de instruções de máquina que realiza a mesma tarefa.



```
MOVF id3, R2
MULF #60.0, R2
MOVF id2, R1
ADDF R2, R1
MOVF R1, id1
```

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem de Máquina

- Um programa a executar é guardado em memória, como os dados.
- Um programa é constituído por uma sequência de instruções codificadas.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem de Máquina

- Um programa a executar é guardado em memória, como os dados.
- Um programa é constituído por uma sequência de instruções codificadas.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem de Máquina

- Cada instrução:
  - pode ocupar uma ou várias posições de memória
  - é constituída pelo código da instrução e informação sobre os possíveis operandos

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem de Máquina

- Cada instrução:
- pode ocupar uma ou várias posições de memória
- é constituída pelo código da instrução e informação sobre os possíveis operandos



Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem de Máquina

- Cada instrução:
- pode ocupar uma ou várias posições de memória
- é constituída pelo código da instrução e informação sobre os possíveis operandos

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções típico

- Transferência de dados, entre registos e memória
- Aritméticas e Lógicas, que correspondem a cálculos efetuados pela ULA
- Transferências de controlo (condicionais), que permitem alterar a sequência normal de execução de instruções
- Cada CPU tem uma linguagem máquina própria que depende da sua arquitetura. e.g. registos, tamanhos de posições de memória, etc.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções típico

- Transferência de dados, entre registos e memória
- Aritméticas e Lógicas, que correspondem a cálculos efetuados pela ULA
- Transferências de controlo (condicionais), que permitem alterar a sequência normal de execução de instruções
- Cada CPU tem uma linguagem máquina própria que depende da sua arquitetura. e.g. registos, tamanhos de posições de memória, etc.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções típico

- Transferência de dados, entre registos e memória
- Aritméticas e Lógicas, que correspondem a cálculos efetuados pela ULA
- Transferências de controlo (condicionais), que permitem alterar a sequência normal de execução de instruções
- Cada CPU tem uma linguagem máquina própria que depende da sua arquitetura. e.g. registos, tamanhos de posições de memória, etc.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções típico

- Transferência de dados, entre registos e memória
- Aritméticas e Lógicas, que correspondem a cálculos efetuados pela ULA
- Transferências de controlo (condicionais), que permitem alterar a sequência normal de execução de instruções
- Cada CPU tem uma linguagem máquina própria que depende da sua arquitetura. e.g. registos, tamanhos de posições de memória, etc.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções

- Por fim, o CPU lê e manipula instruções
- Se a instrução para o um determinado processador for de 8 bits, os primeiro 4 bits diz ao processador o que fazer (opcode) e o segundo conjunto de 4 bits diz ao processador que dado usar (operando)
- Um programa executável é então uma sequência de códigos binários do tipo: 01001000 01100101 01101100 01101100 01101111 00100001

## Conjunto de instruções

- Por fim, o CPU lê e manipula instruções
- Se a instrução para o um determinado processador for de 8 bits, os primeiro 4 bits diz ao processador o que fazer (opcode) e o segundo conjunto de 4 bits diz ao processador que dado usar (operando)
- Um programa executável é então uma sequência de códigos binários do tipo: 01001000 01100101 01101100 01101100 01101111 00100001

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## Conjunto de instruções

- Por fim, o CPU lê e manipula instruções
- Se a instrução para o um determinado processador for de 8 bits, os primeiro 4 bits diz ao processador o que fazer (opcode) e o segundo conjunto de 4 bits diz ao processador que dado usar (operando)
- Um programa executável é então uma sequência de códigos binários do tipo: 01001000 01100101 01101100 01101100 01101111 00100001



## A linguagem Assembly

- É um derivado da linguagem máquina, também chamado de linguagem de montagem e é formada por abreviaturas de letras e números chamados mnemônicos.
- Embora não seja uma linguagem nativa do microprocessador, um programador que use a linguagem Assembly ainda deve compreender as características da arquitetura do processador.

## A linguagem Assembly

- É um derivado da linguagem máquina, também chamado de linguagem de montagem e é formada por abreviaturas de letras e números chamados mnemônicos.
- Embora não seja uma linguagem nativa do microprocessador, um programador que use a linguagem Assembly ainda deve compreender as características da arquitetura do processador.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem Assembly

- Com o aparecimento desta linguagem se criaram os programas tradutores para poder passar os programas escritos em linguagem Assembly para linguagem de máquina.
- Um Assemblador é um programa que traduz um programa em Assembly para linguagem máquina.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

## A linguagem Assembly

- Com o aparecimento desta linguagem se criaram os programas tradutores para poder passar os programas escritos em linguagem Assembly para linguagem de máquina.
- Um Assemblador é um programa que traduz um programa em Assembly para linguagem máquina.

## Assembly

Informação básica do programa	<pre> ; Programa p/ iniciação do porto B e pôr os seus pino a '1' lógico ; Version 1.0  Date: 10.10.1999.  MCU:PIC16F84  Written by: John Smith         </pre>
Directiva	<pre> ; Declaração e configuração do processador PROCESSOR 16F84      ; Tipo de processador #include "p16f84.inc" ; Incluir ficheiro  ; Início do programa org 0x00              ; Vector de reset goto Main             ; Ir p/ o início do programa Main         </pre>
Inclusão de uma macro	<pre> org 0x04              ; Vector de interrupção goto Main             ; Não há rotina de interrupção #include "bank.inc"         </pre>
Comentário	<pre> Main         </pre>
Rótulo	<pre> BANK1 movlw 0x00 movwf TRISB BANK0         </pre>
Instrução	<pre> movlw 0xFF movwf PORTB         </pre>
Operando	<pre> Loop goto Loop end         </pre>

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

Operação	Código	Operando	Significado
LDA	0001	X	Carrega para o AC o conteúdo da posição de memória de endereço X
LDI	0010	X	Carrega (indirectamente) para o AC o conteúdo da posição de memória cujo o endereço é o conteúdo da posição de memória de endereço X
STA	0011	X	Guarda o conteúdo do AC na posição de memória de endereço X
STI	0100	X	Guarda (indirectamente) o conteúdo do AC na posição de memória cujo o endereço é o conteúdo da posição de memória de endereço X
ADD	0101	X	Adiciona o conteúdo da posição de memória de endereço X ao conteúdo do AC
SUB	0110	X	Subtrai o conteúdo da posição de memória de endereço X do conteúdo do AC
JMP	0111	X	Salta para a instrução cujo endereço é X
JMZ	1000	X	Salta para a instrução cujo endereço é X, se conteúdo de AC é 0

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1  
Hello Word - 2

Exercícios

Bibliografia

## Código em C

```
1 #include
2 int main() {
3     int a=5;
4     printf("a=%d, eax=%d\n", a, a+2);
5     return 0;
6 }
```

# Hello Word - 1

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

**Hello Word - 1**

Hello Word - 2

Exercícios

Bibliografia

```
1 extern printf
2 SECTION .data
3     a:    dd    5
4     fmt:   db    "a=%d, eax=%d", 10, 0
5
6
7 SECTION .text
8
9 global main
```

./Codigos/hellowordP0.s



# Hello Word - 1

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1 main:      ; the program label for the entry point
2
3     push    ebp      ; set up stack frame
4     mov     ebp, esp
5
6     mov     eax, [a]   ; put a from store into register
7     add     eax, 2     ; a+2
8     push    eax       ; value of a+2
9     push    dword [a]  ; value of variable a
10    push    dword fmt   ; address of ctrl string
11    call    printf     ; Call C function
12    add     esp, 12     ; pop stack 3 push times 4 bytes
13
14    mov     esp, ebp    ; takedown stack frame
15    pop     ebp        ; same as "leave" op
16
17    mov     eax, 0      ; normal, no error, return value
18    ret          ; return
```

./Codigos/hellowordP1.s

# Hello Word - 1

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1 extern printf
2 SECTION .data
3     a:  dd  5
4     fmt:  db  "a=%d, eax=%d", 10, 0
5 SECTION .text
6 global main
7 main:
8     push    ebp
9     mov     ebp, esp
10    mov     eax, [a]
11    add     eax, 2
12    push    eax
13    push    dword [a]
14    push    dword fmt
15    call    printf
16    add     esp, 12
17    mov     esp, ebp
18    pop     ebp
19    mov     eax, 0
20    ret
```

./Codigos/helloworldComplete.s

# Hello Word - 2

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1 SECTION .data
2     message1: db "Enter the first number: ", 0
3     message2: db "Enter the second number: ", 0
4     formatin: db "%d", 0
5     formatout: db "%d", 10, 0 ; newline, nul terminator
6     integer1: times 4 db 0 ; 32-bits integer = 4 bytes
7     integer2: times 4 db 0 ;
8
9 SECTION .text
10     global main
11     extern scanf
12     extern printf
```

./Codigos/helloworld2P0.s

# Hello Word - 2

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1  main:
2
3      push ebx ; save registers
4      push ecx
5      push message1
6      call printf
7
8      add esp, 4 ; remove parameters
9      push integer1 ; address of integer1 (second parameter)
10     push formatin ; arguments are right to left (first
        parameter)
11     call scanf
12
13     add esp, 8 ; remove parameters
14     push message2
15     call printf
```

./Codigos/helloworld2P1.s

# Hello Word - 2

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1  add esp, 4 ; remove parameters
2  push integer2 ; address of integer2
3  push formatin ; arguments are right to left
4  call scanf
5
6  add esp, 8 ; remove parameters
7
8  mov ebx, dword [integer1]
9  mov ecx, dword [integer2]
10 add ebx, ecx ; add the values
11 push ebx ; the addition
12 push formatout
13 call printf; call printf to display the sum
14 add esp, 8; remove parameters
15 pop ecx
16 pop ebx ; restore registers in reverse order
17 mov eax, 0 ; no error
18 ret
```

./Codigos/helloworld2P2.s

# Hello Word - 1

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

A linguagem de  
Máquina

A linguagem  
Assembly

Hello Word - 1

Hello Word - 2

Exercícios

Bibliografia

```
1
2 SECTION .data          ; data section
3     msg:      db "Hello World",10 ; the string to print, 10=cr
4     len:      equ $-msg          ; "$" means "here"
5                                     ; len is a value, not an address
6
7 SECTION .text          ; code section
8     global main          ; make label available to linker
9 main:                  ; standard gcc entry point
10
11     mov edx,len        ; arg3, length of string to print
12     mov ecx,msg        ; arg2, pointer to string
13     mov ebx,1          ; arg1, where to write, screen
14     mov eax,4          ; write sysout command to int 80 hex
15     int 0x80          ; interrupt 80 hex, call kernel
16
17     mov ebx,0          ; exit code, 0=normal
18     mov eax,1          ; exit command to kernel
19     int 0x80          ; interrupt 80 hex, call kernel
```

./Codigos/helloworld2

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

**Exercícios**

Bibliografia

1) O que é um compilador e qual a sua função?

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

**Exercícios**

Bibliografia

- 2) Qual a diferença entre linguagem de máquina e linguagem de montagem? Como elas se relacionam?



## Livros Texto

- AHO, A; ULLMANN, J; REVI, S. Compiladores: Princípios, técnicas e ferramentas 3 ed. Rio de Janeiro: LTC-Livros
- LOUDEN, Kenneth C; SILVA, Flávio Soares Corrêa. Compiladores : princípios e práticas. 1a ed. São João da Boa Vista: Pioneira - Thomson Learning, 2004.
- PRICE, Ana M. A.; TOSCANI, Simão S.. Implementação de linguagens de programação: compiladores. 3a ed. Porto Alegre: Bookman, 2008.
- SETZER, V.W.. Construção de um Compilador. 1a ed. Rio de Janeiro: Campus - Elsevier, 1983.

Prog. Apl. CC

Dr. E. S.  
Pereira

Sumário

Introdução

Compiladores

Exercícios

Bibliografia

Grato

# MUITO OBRIGADO.