

Arquitetura e Organização de Computadores

Conjunto de Instruções da
Arquitetura – CompSim



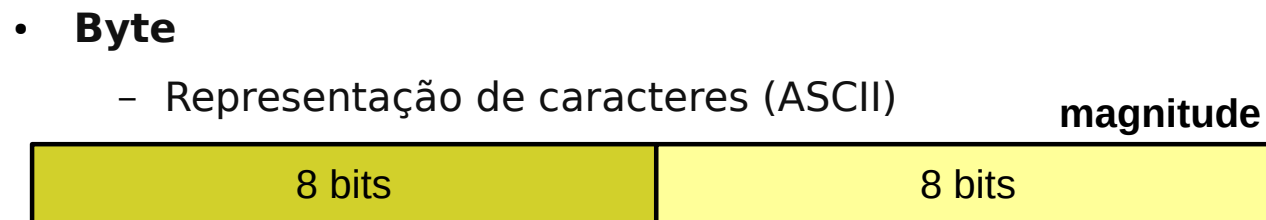
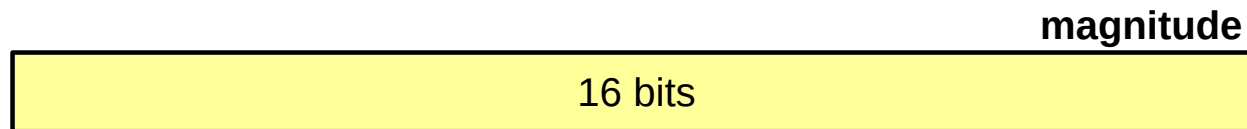
Agenda

- Formato de Instruções e de Operandos
- Tipos de Instruções
- Estrutura de um Programa *Assembly*
- Definindo e Declarando Variáveis



Formatos de Instruções e de Operandos

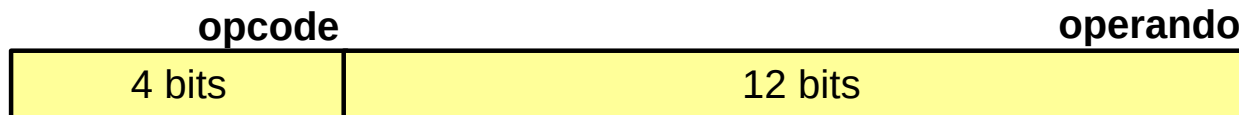
- Palavra de 16 bits, onde:
 - **Operandos:**
 - **Inteiro**
 - Sinalizado;
 - Complemento a 2;
 - Little Endian (mais significativo à esquerda);





Formatos de Instruções e de Operandos

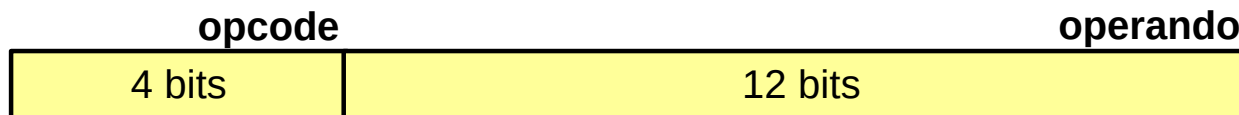
- Palavra de 16 bits, onde:
 - **Instruções**
 - Arquitetura baseada em Acumulador;
 - Operandos: implícitos (Acumulador) e memória.





Formatos de Instruções e de Operandos

- Palavra de 16 bits, onde:
 - **Instruções**
 - Arquitetura baseada em Acumulador;
 - Operandos: implícitos (Acumulador) e memória.



Código	Instrução
0	LDA
1	STA
2	INT
3	CALL
4	ADD
5	SUB
6	SOP
7	MOV

Código	Instrução
8	JMP
9	JN
A	JZ
B	LDI
C	STI
D	NAND
E	SHIFT
F	RET



Tipos de Instruções

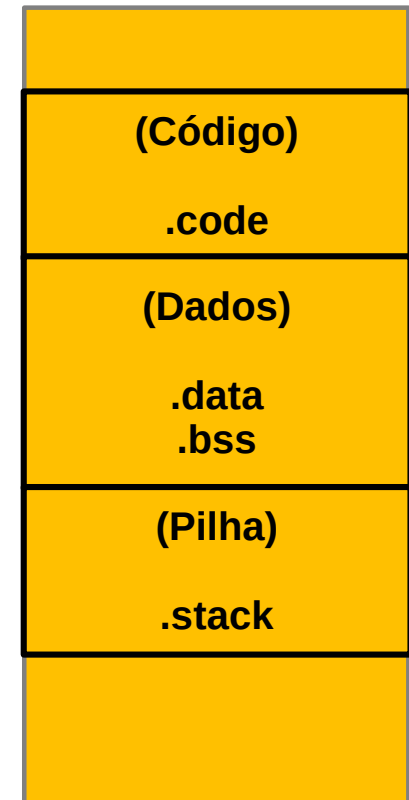
- Pseudo-Instruções do Montador (*Assembler*)
 - Segmento
 - `.code`, `.data`, `.bss`, `.stack`
 - Rótulo ou Nome
 - `:`
 - Delimitador de comentário
 - `;`
 - Definição/Declaração de variáveis
 - `DD`, `DB`, `RESB`, `RESQ`
- Conjunto de Instruções da Arquitetura (ISA)
 - Aritméticas
 - `ADD`, `SUB`
 - Lógicas
 - `NAND`, `SHIFT`
 - Transferência de dados
 - `MOV`, `LDA`, `STA`, `LDI`, `STI`, `SOP`
 - Transferência de controle
 - `JMP`, `JN`, `JZ`, `CALL`, `RET`, `INT`
 - Entrada/Saída
 - `INT`



Estrutura de um Programa *Assembly*

- Seções/Segmentos
 - **.code**
 - instruções
 - **.data**
 - dados inicializados
 - **.bss**
 - *block started by symbol*
 - dados não inicializados
 - segmento opcional
 - **.stack** <size>
 - pilha do programa
 - <size> define o número de palavras

Memória Principal



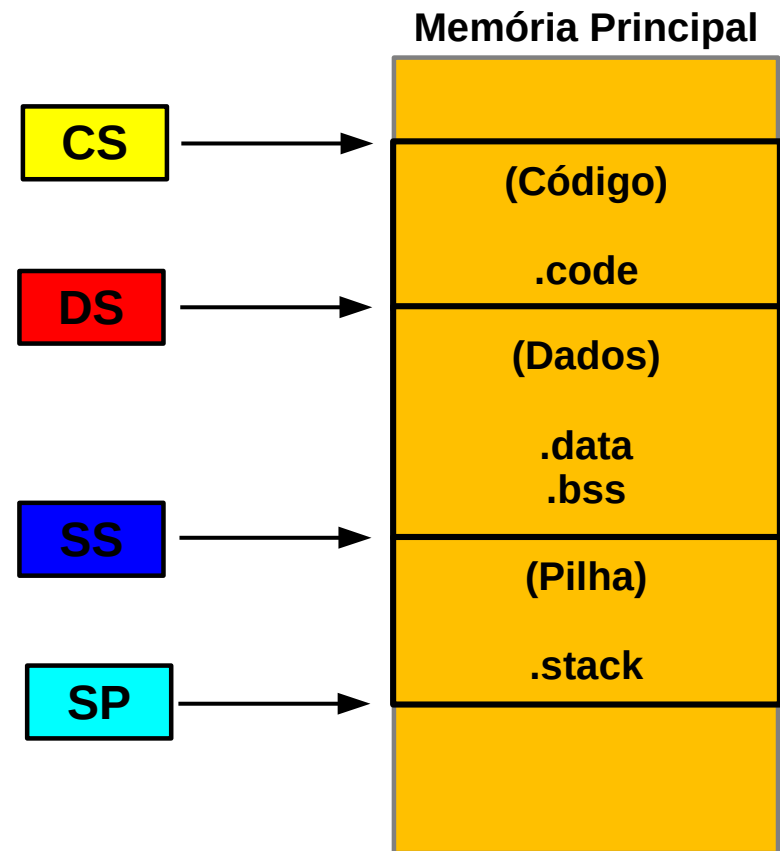
Estrutura de um Programa *Assembly*

- Seções/Segmentos

- Após o cópia do programa para a memória principal, o processador utiliza quatro registradores para gerenciar os limites de cada seção.

- São eles:

- **CS** – *Code Segment*
- **DS** – *Data Segment*
- **SS** – *Stack Segment*
- **SP** – *Stack Pointer*





Definindo e Declarando Variáveis

- **Definição** deve ser realizada no bloco **.data**
 - Utiliza-se:
 - **DD** para números inteiros
 - Sintaxe: <nome>: **DD** <inteiro> | <nome>: **DD** <nome>
 - **DB** para caracteres (*bytes*)
 - Sintaxe: <nome>: **DB** '<string>' | <nome>: **DB** "<string>"
- **Declaração** deve ser realizada no bloco **.bss**
 - Utiliza-se:
 - **RESB** para números inteiros
 - Sintaxe: <nome>: **RESB** <size>
 - **RESB** para caracteres (*bytes*)
 - Sintaxe: <nome>: **RESB** <size>



Definindo e Declarando Variáveis

- Exemplo prático: Declarando Variáveis
- Procedimento:
 - Baixar e extrair o pacote:
 - [1.declaring_variables.zip](#)
 - Menu “File” → “Open”
 - Ou Teclas “Ctrl+o”
 - Arquivo “declaring_vars.asm”

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```



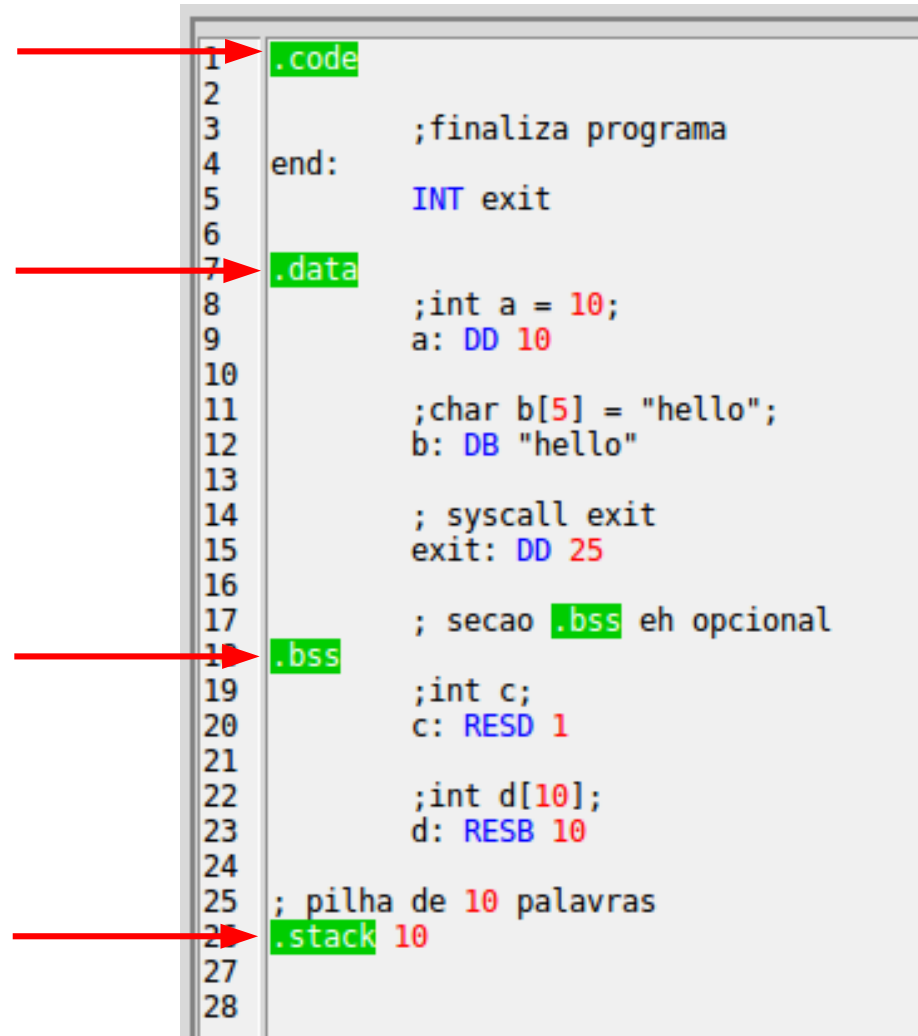
Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```

Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação
 - **Segmentos**



```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```

Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação
 - Segmentos
 - **Instruções**

```
1  .code
2
3      ;finaliza programa
4  end:
5  → INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```

Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação

- Segmentos
- Instruções
- **Comentários**

```
1  .code
2
3  → ;finaliza programa
4  end:
5      INT exit
6
7  .data
8  → ;int a = 10;
9      a: DD 10
10
11 → ;char b[5] = "hello";
12     b: DB "hello"
13
14 → ; syscall exit
15     exit: DD 25
16
17 → ; secao .bss eh opcional
18 .bss
19 → ;int c;
20     c: RESD 1
21
22 → ;int d[10];
23     d: RESB 10
24
25 → ; pilha de 10 palavras
26 .stack 10
27
28
```

Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação
 - Segmentos
 - Instruções
 - Comentários
 - **Definições**

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18     .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26     .stack 10
27
28
```

Definindo e Declarando Variáveis


- Compreendendo a estrutura da aplicação
 - Segmentos
 - Instruções
 - Comentários
 - Definições
 - **Declarações**

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```


Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação
 - Segmentos
 - Instruções
 - Comentários
 - Definições
 - Declarações
 - **Pilha**

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```





Definindo e Declarando Variáveis

- Compreendendo a estrutura da aplicação
 - Segmentos
 - Instruções
 - Comentários
 - Definições
 - Declarações
 - Pilha

```
1  .code
2
3      ;finaliza programa
4  end:
5      INT exit
6
7  .data
8      ;int a = 10;
9      a: DD 10
10
11     ;char b[5] = "hello";
12     b: DB "hello"
13
14     ; syscall exit
15     exit: DD 25
16
17     ; secao .bss eh opcional
18  .bss
19     ;int c;
20     c: RESD 1
21
22     ;int d[10];
23     d: RESB 10
24
25     ; pilha de 10 palavras
26  .stack 10
27
28
```