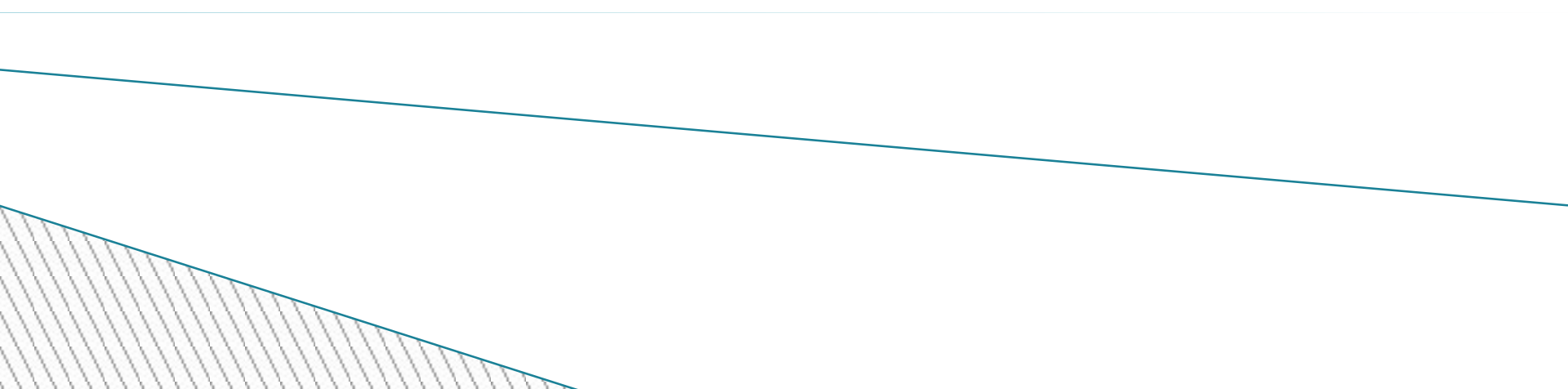


IHS

Aula Prática 1

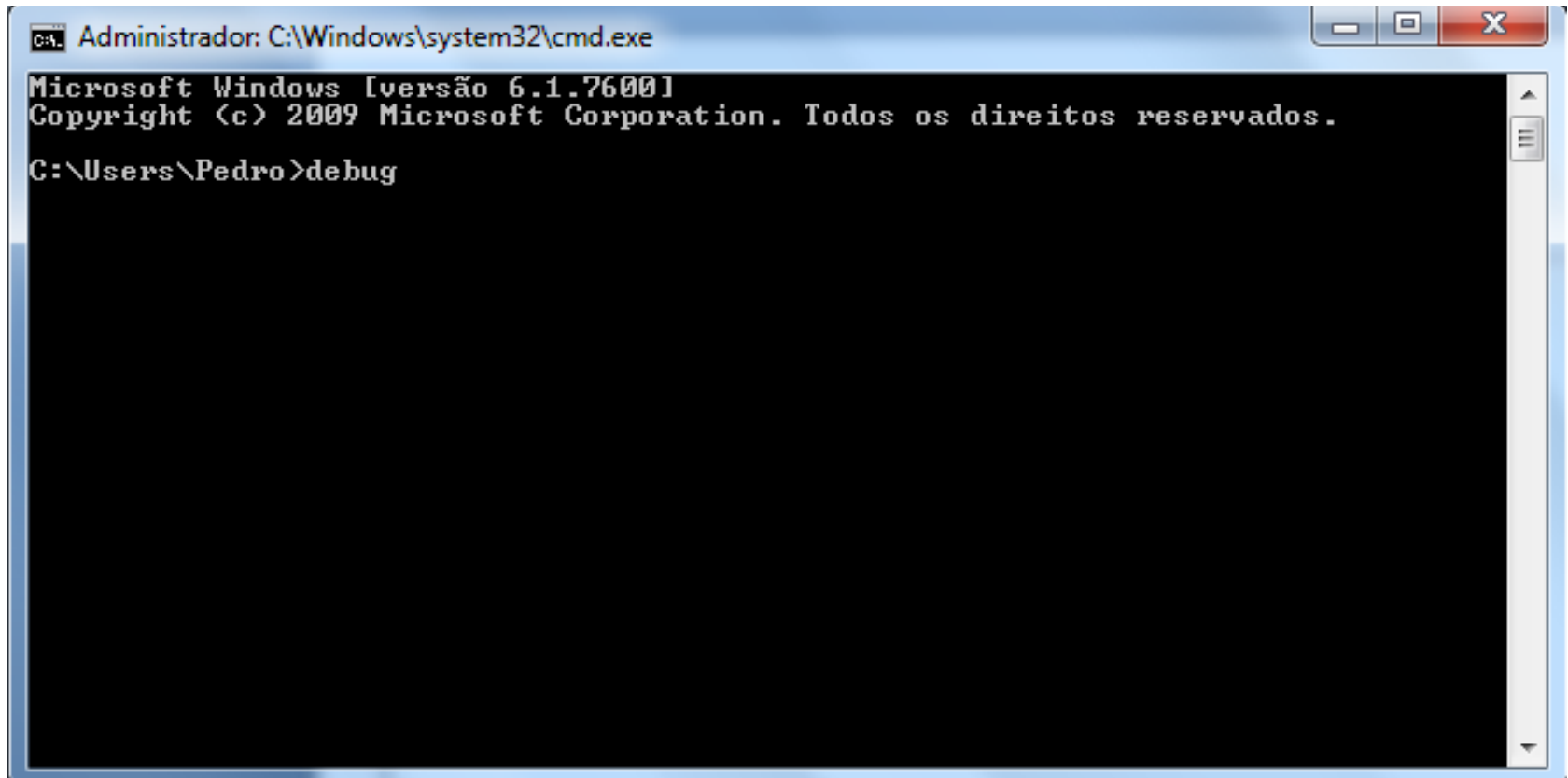
Instruções Assembly e Introdução ao
Debug/GRDB

A decorative teal line curves across the bottom of the slide, and a hatched pattern is visible in the bottom-left corner.

Debug

- ▶ Útil para observar a execução passo a passo das instruções de programa.
- ▶ Disponível em qualquer PC com o Windows/MS-DOS (32-bit)
- ▶ Para iniciá-lo, basta digitar “debug” no prompt de comandos do Windows (antigo MS-DOS).

Debug



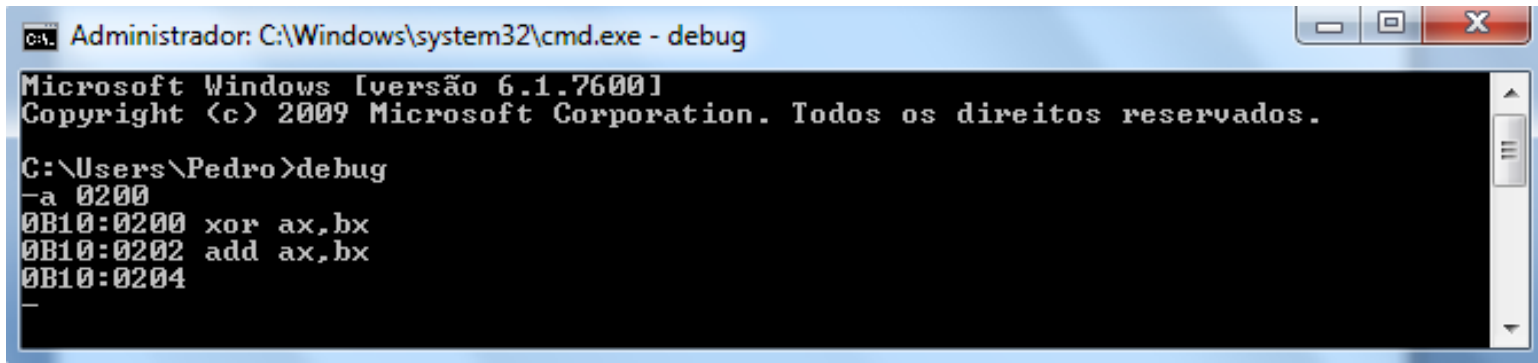
The image shows a screenshot of a Windows command prompt window. The title bar at the top reads "Administrador: C:\Windows\system32\cmd.exe". The main content area of the window displays the following text:

```
Microsoft Windows [versão 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.  
C:\Users\Pedro>debug
```

The command prompt is currently at the C:\Users\Pedro directory, and the 'debug' command has been entered. The rest of the window is black, indicating that the command has not yet been executed or that the output is not visible.

Comandos do Debug

- ▶a – Monta instruções simbólicas em código de máquina

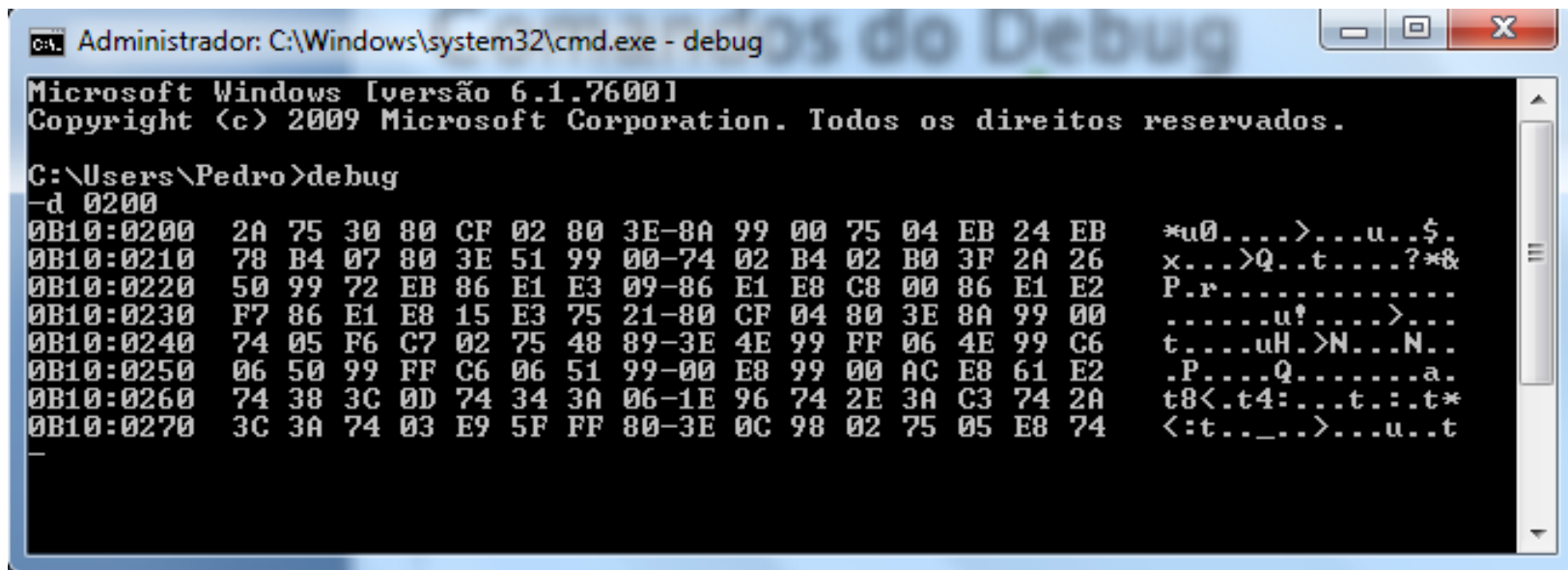


```
C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-a 0200
0B10:0200 xor ax,bx
0B10:0202 add ax,bx
0B10:0204
-
```

Comandos do Debug

- ▶ d – Mostra o conteúdo de uma área da memória

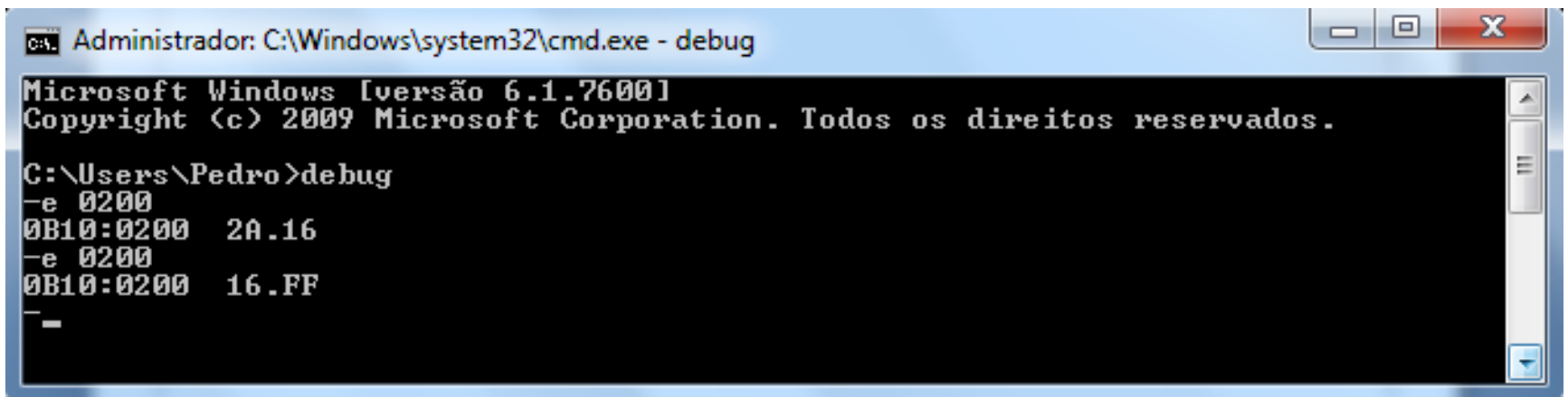


```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-d 0200
0B10:0200  2A 75 30 80 CF 02 80 3E-8A 99 00 75 04 EB 24 EB  *u0....>...u..$.
0B10:0210  78 B4 07 80 3E 51 99 00-74 02 B4 02 B0 3F 2A 26  x...>Q..t....?*&
0B10:0220  50 99 72 EB 86 E1 E3 09-86 E1 E8 C8 00 86 E1 E2  P.r.....
0B10:0230  F7 86 E1 E8 15 E3 75 21-80 CF 04 80 3E 8A 99 00  . . . . .u! . . . . > . .
0B10:0240  74 05 F6 C7 02 75 48 89-3E 4E 99 FF 06 4E 99 C6  t . . . . uH . > N . . N . .
0B10:0250  06 50 99 FF C6 06 51 99-00 E8 99 00 AC E8 61 E2  .P . . . . Q . . . . .a .
0B10:0260  74 38 3C 0D 74 34 3A 06-1E 96 74 2E 3A C3 74 2A  t8<.t4: . . . . t . . : . t *
0B10:0270  3C 3A 74 03 E9 5F FF 80-3E 0C 98 02 75 05 E8 74  <:t . . _ . . > . . . . u . . t
```

Comandos do Debug

- ▶ e – Edita dados na memória, iniciando num endereço específico

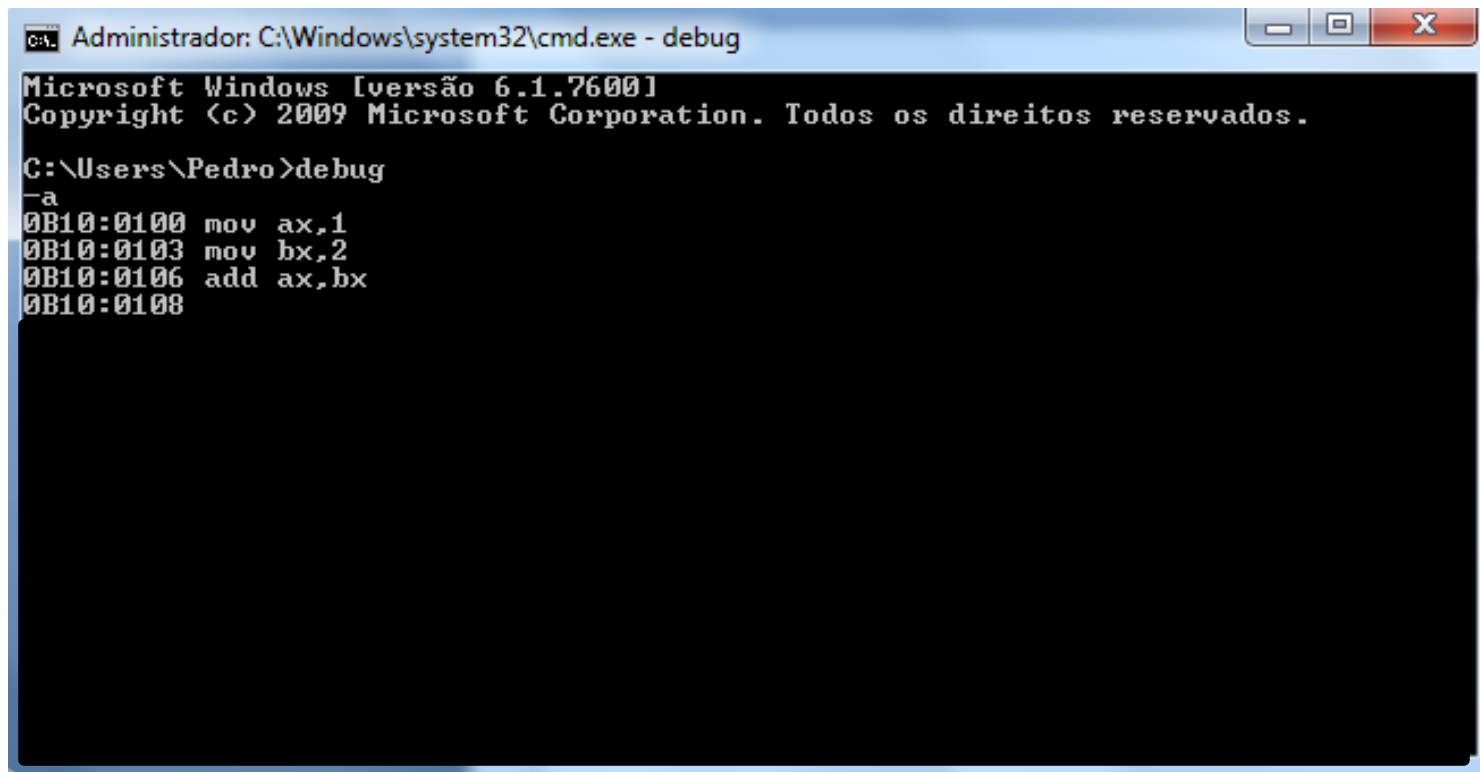


```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-e 0200
0B10:0200  2A.16
-e 0200
0B10:0200  16.FF
-
```

Comandos do Debug

- ▶ t – Executa passo a passo instruções



```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-a
0B10:0100 mov ax,1
0B10:0103 mov bx,2
0B10:0106 add ax,bx
0B10:0108
```

Comandos do Debug

- ▶ t – Executa passo a passo instruções

```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-a
0B10:0100 mov ax,1
0B10:0103 mov bx,2
0B10:0106 add ax,bx
0B10:0108
-t

AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B10 ES=0B10 SS=0B10 CS=0B10 IP=0103  NU UP EI PL NZ NA PO NC
0B10:0103 BB0200          MOVBX,0002
```

Comandos do Debug

- ▶ t – Executa passo a passo instruções

```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-a
0B10:0100 mov ax,1
0B10:0103 mov bx,2
0B10:0106 add ax,bx
0B10:0108
-t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B10 ES=0B10 SS=0B10 CS=0B10 IP=0103  NU UP EI PL NZ NA PO NC
0B10:0103 BB0200          MOV     BX,0002
-t
AX=0001 BX=0002 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B10 ES=0B10 SS=0B10 CS=0B10 IP=0106  NU UP EI PL NZ NA PO NC
0B10:0106 01D8          ADD     AX,BX
```

Comandos do Debug

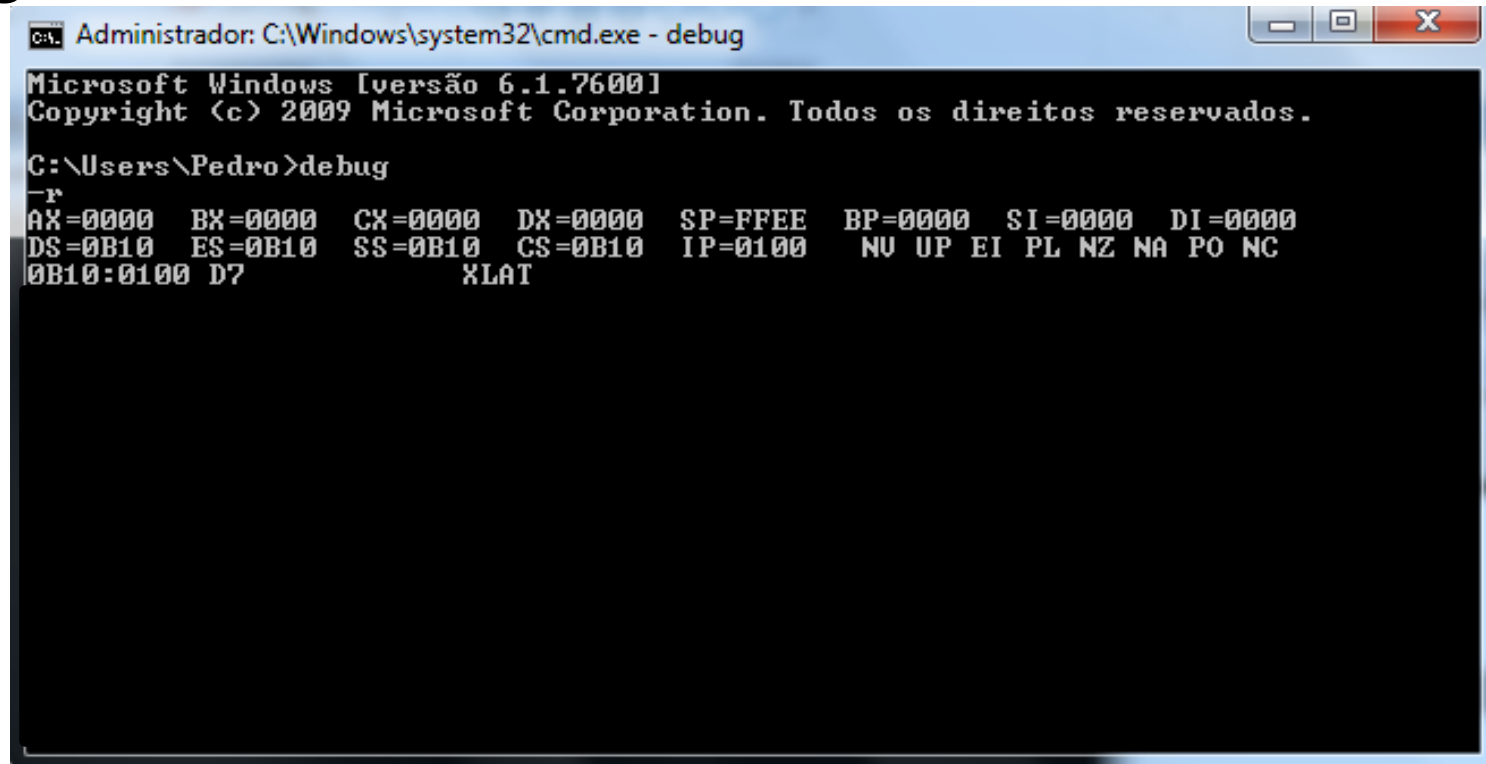
- ▶ t – Executa passo a passo instruções

```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-a
0B10:0100 mov ax,1
0B10:0103 mov bx,2
0B10:0106 add ax,bx
0B10:0108
-t
AX=0001  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0103  NU UP EI PL NZ NA PO NC
0B10:0103 BB0200          MOV     BX,0002
-t
AX=0001  BX=0002  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0106  NU UP EI PL NZ NA PO NC
0B10:0106 01D8          ADD     AX,BX
-t
AX=0003  BX=0002  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0108  NU UP EI PL NZ NA PE NC
0B10:0108 46          INC     SI
-
```

Comandos do Debug

- ▶ r – Mostra o conteúdo de um ou mais registradores

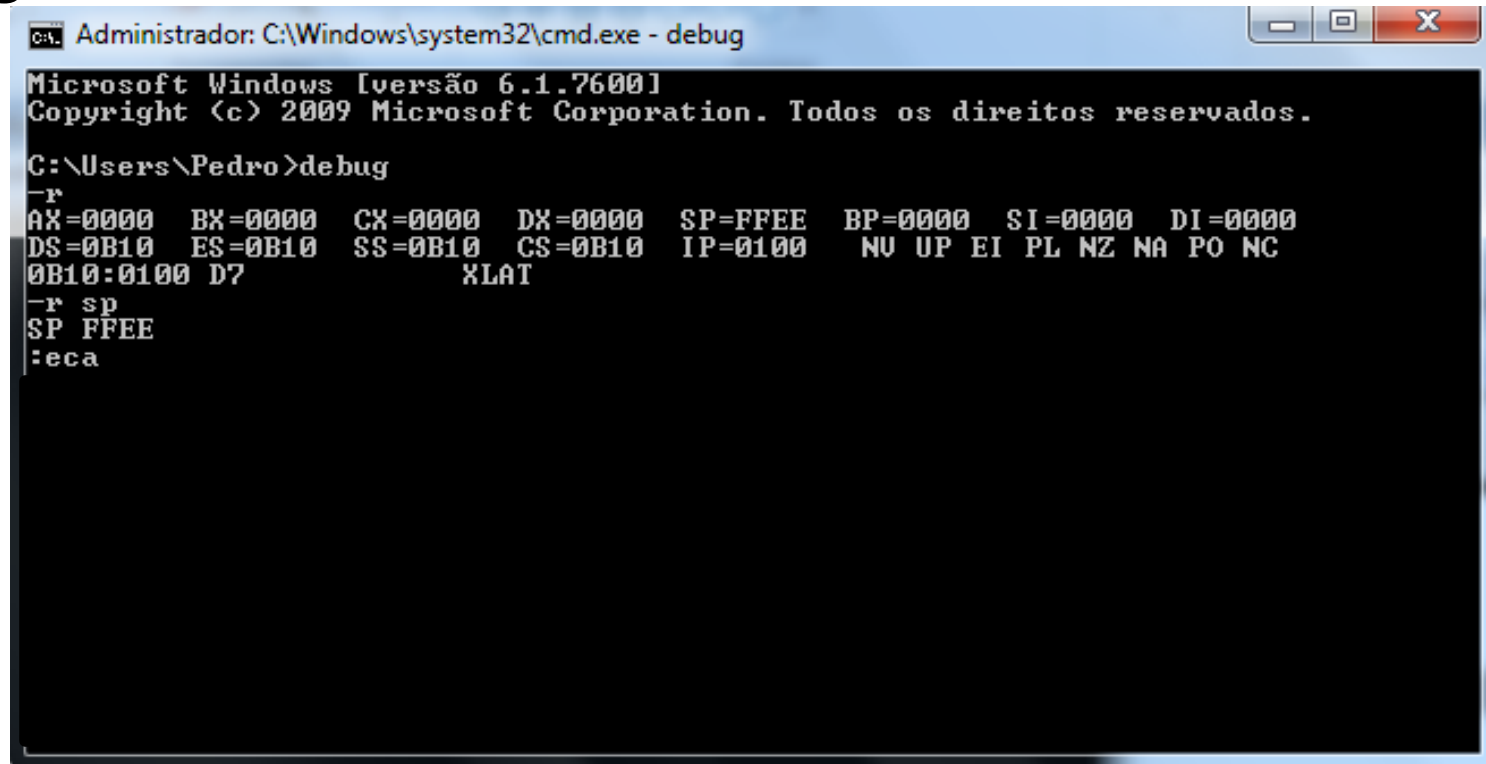


```
Administrator: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0100  NU UP EI PL NZ NA PO NC
0B10:0100 D?          XLAT
```

Comandos do Debug

- ▶ r – Mostra o conteúdo de um ou mais registradores

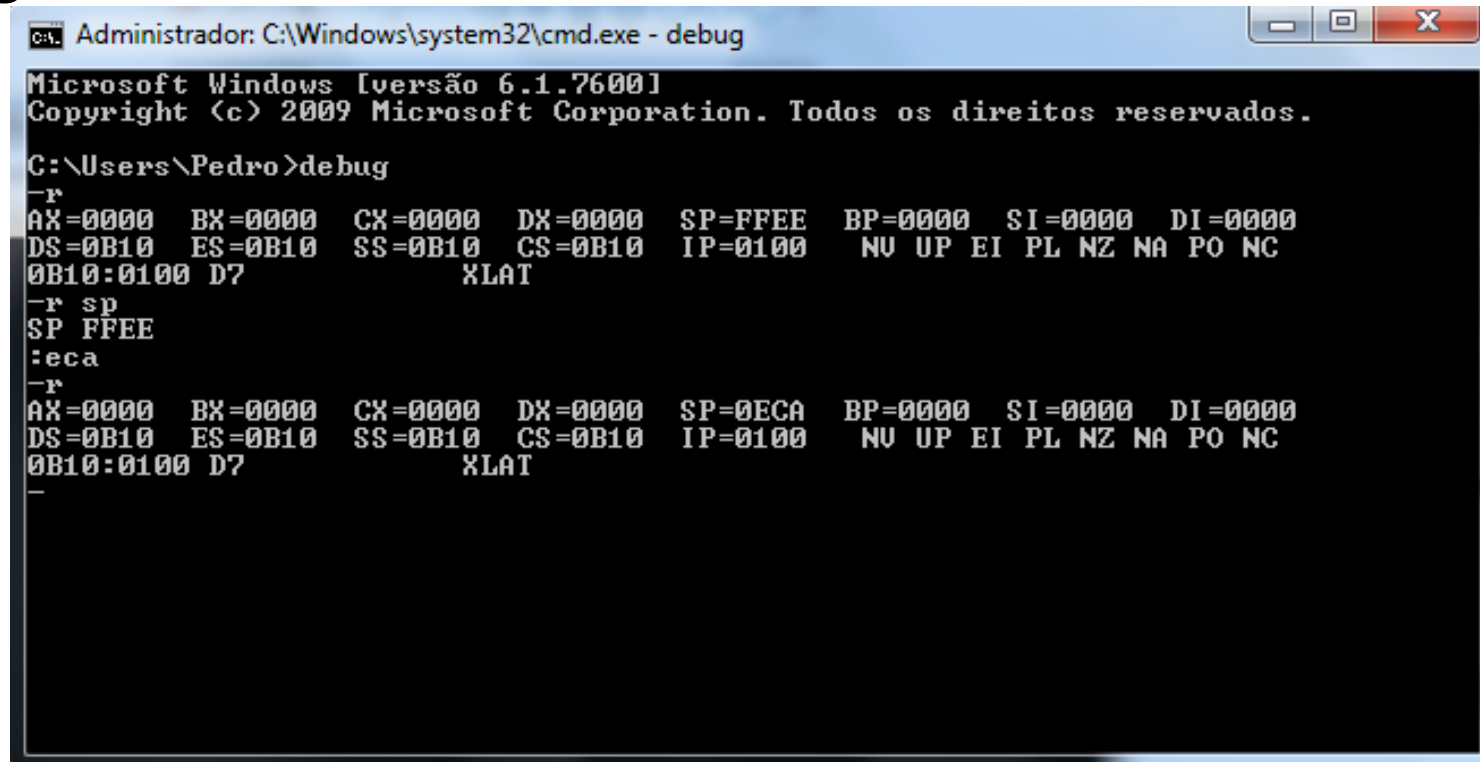


```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0100  NU UP EI PL NZ NA PO NC
0B10:0100 D?          XLAT
-r sp
SP FFEE
:eca
```

Comandos do Debug

- ▶ r – Mostra o conteúdo de um ou mais registradores

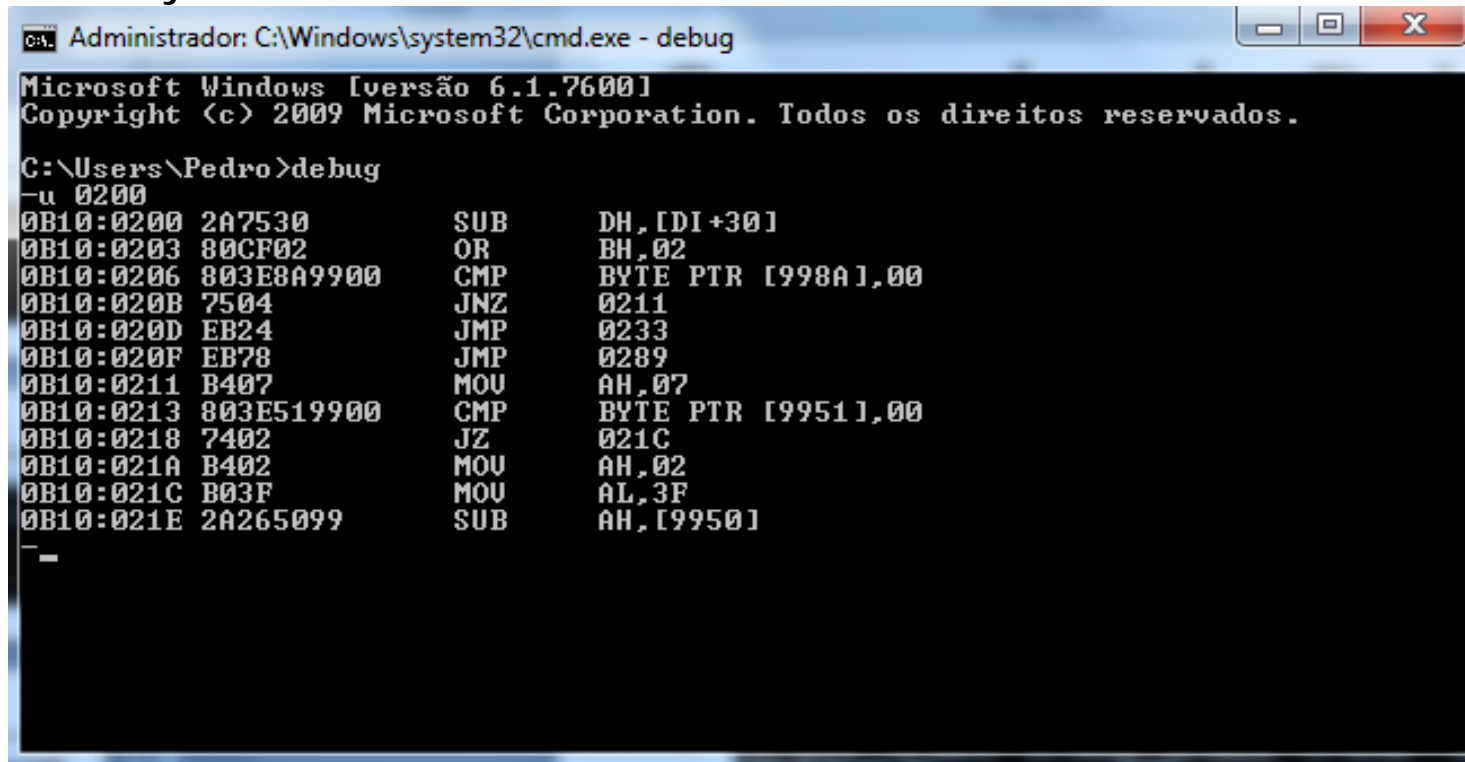


```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0100  NU UP EI PL NZ NA PO NC
0B10:0100 D7          XLAT
-r sp
SP FFEE
:eca
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=0ECA  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0100  NU UP EI PL NZ NA PO NC
0B10:0100 D7          XLAT
-
```

Comandos do Debug

- u – Desmonta o código de máquina em instruções simbólicas

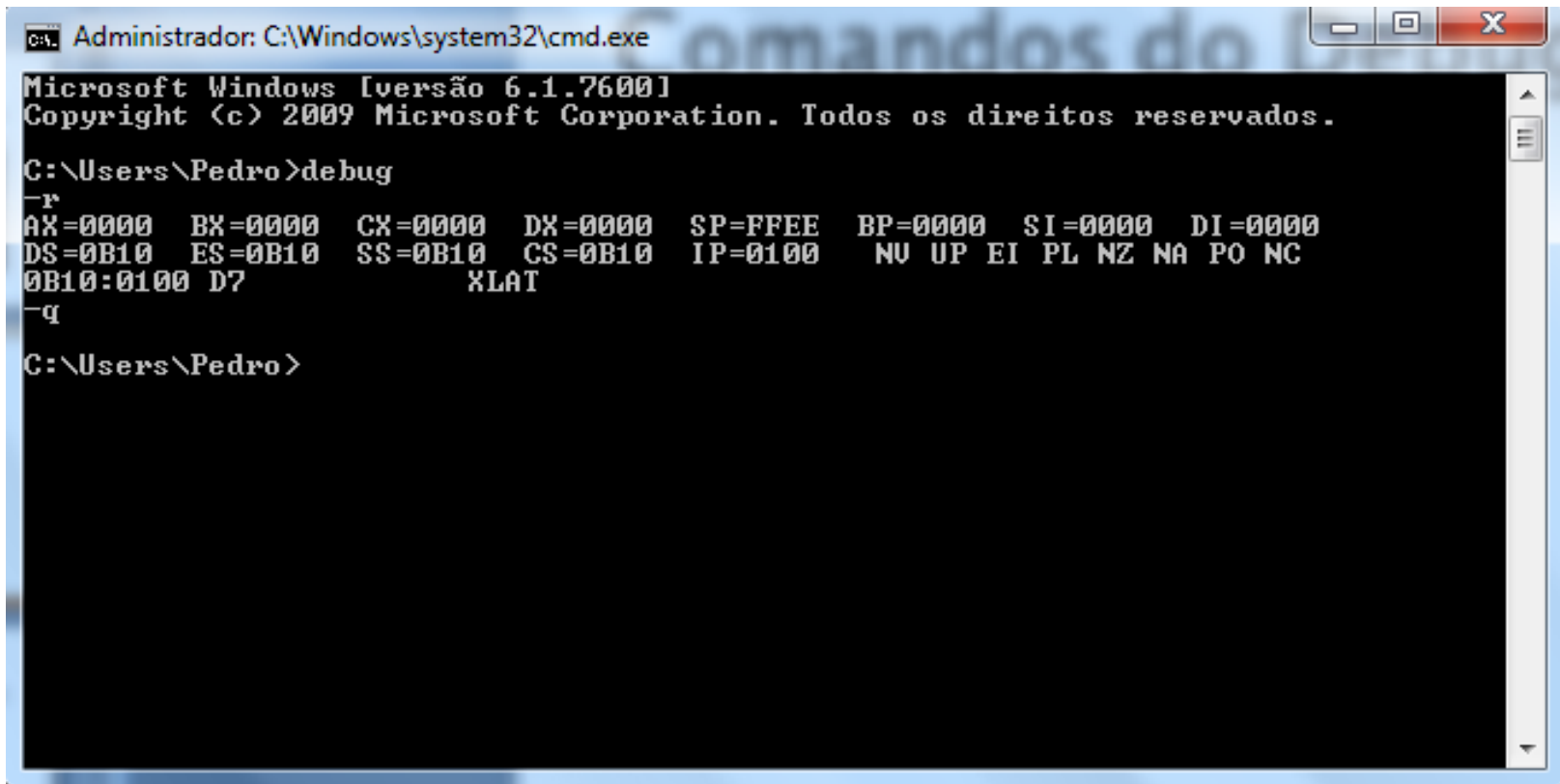


```
Administrador: C:\Windows\system32\cmd.exe - debug
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-u 0200
0B10:0200 2A7530      SUB     DH,[DI+30]
0B10:0203 80CF02      OR     BH,02
0B10:0206 803E8A9900  CMP    BYTE PTR [998A],00
0B10:020B 7504       JNZ    0211
0B10:020D EB24       JMP    0233
0B10:020F EB78       JMP    0289
0B10:0211 B407       MOV    AH,07
0B10:0213 803E519900  CMP    BYTE PTR [9951],00
0B10:0218 7402       JZ     021C
0B10:021A B402       MOV    AH,02
0B10:021C B03F       MOV    AL,3F
0B10:021E 2A265099  SUB    AH,[9950]
-
```

Comandos do Debug

►q – Sai do programa Debug



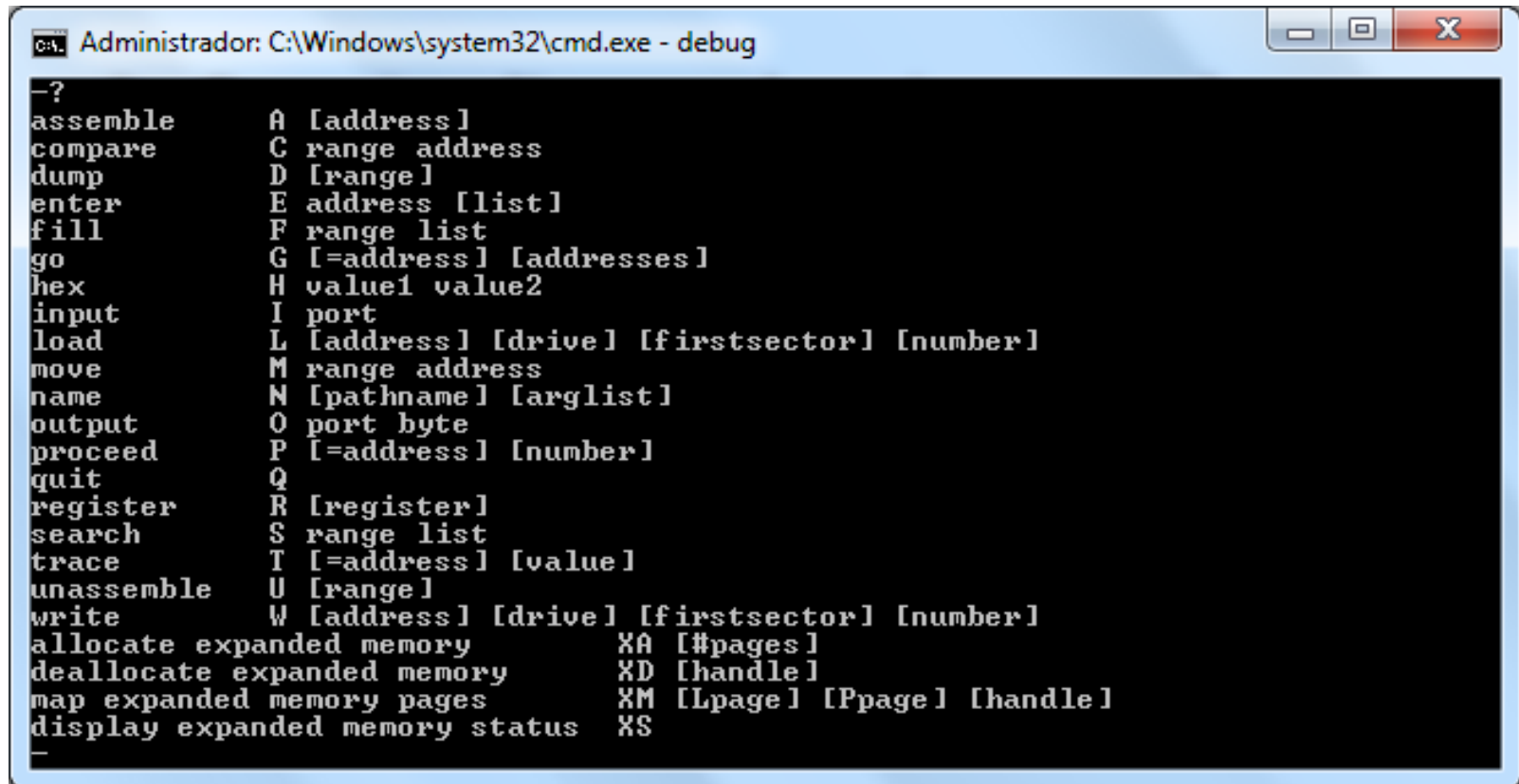
```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [versão 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Pedro>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B10  ES=0B10  SS=0B10  CS=0B10  IP=0100  NU UP EI PL NZ NA PO NC
0B10:0100 D?          XLAT
-q

C:\Users\Pedro>
```

Comandos do Debug

▶? – Para visualizar os demais comandos.



```
Administrador: C:\Windows\system32\cmd.exe - debug
-?
assemble      A [address]
compare       C range address
dump          D [range]
enter         E address [list]
fill          F range list
go            G [=address] [addresses]
hex           H value1 value2
input         I port
load          L [address] [drive] [firstsector] [number]
move          M range address
name          N [pathname] [arglist]
output        O port byte
proceed       P [=address] [number]
quit          Q
register       R [register]
search        S range list
trace         T [=address] [value]
unassemble    U [range]
write         W [address] [drive] [firstsector] [number]
allocate expanded memory  XA [#pages]
deallocate expanded memory XD [handle]
map expanded memory pages XM [Lpage] [Ppage] [handle]
display expanded memory status XS
-
```

Programando em Assembly

Diretivas assembly:

- ▶Diferentemente das **instruções assembly** (mov, add, push, jmp...), que são comandos direcionados à **CPU**, as ***diretivas assembler*** são comandos direcionados ao **assembler** (exemplo: *TASM*). As diretivas assembler são executadas em tempo de “compilação” (montagem). Já as instruções assembly são executadas em tempo de execução.
- ▶Diretivas assembly podem ser comandos para reservar áreas de memória, definir procedimentos, definir constantes, entre outros

Programando em Assembly

Diretivas assembly:

► .model – define o modelo de memória do programa.

Modelo	Segmento de dados	Segmento de código
*TINY	UM	UM
SMALL	UM	UM
MEDIUM	UM	VÁRIOS
COMPACT	VÁRIOS	UM
LARGE	VÁRIOS	VÁRIOS
HUGE	VÁRIOS	VÁRIOS
FLAT	UM	UM

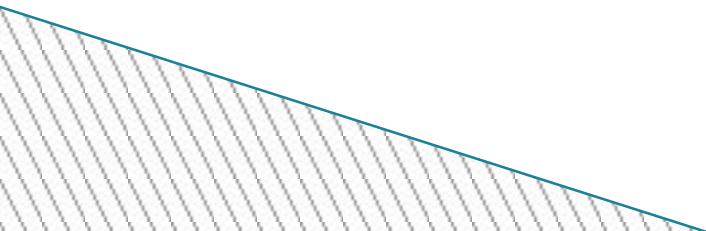
*Obs.: No modelo tiny, o segmento de dados e o de código compartilham a mesma região de memória.

Programando em Assembly

Diretivas assembly:

- ▶ `.stack` – reserva espaço de memória para instruções da pilha.

- ▶ `.code` – define o início de um programa.



Programando em Assembly

Diretivas assembly:

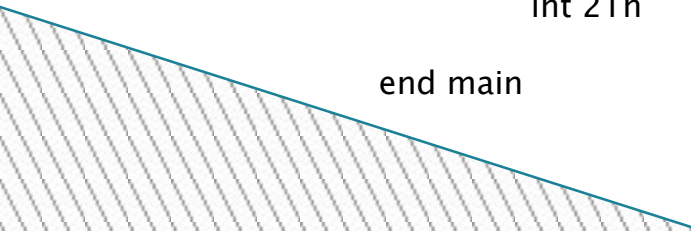
▶.data – define as variáveis e constantes de um programa.

<i>Directive</i>	<i>Name</i>	<i>Minimum Bytes Allocated</i>	<i>Typical Use</i>
db	Define byte	1	Bytes, strings
dw	Define word	2	Integers
dd	Define doubleword	4	Long integers
dp	Define pointer	6	32-bit pointer
df	Define far pointer	6	48-bit pointer
dq	Define quadword	8	Real numbers
dt	Define ten bytes	10	BCD numbers

Programando em Assembly

Exemplo:

```
.model small  
  
.stack 10  
  
.data  
  
    x dw 4h  
    y dw 3h  
  
.code  
  
    main:  
  
    mov ax, @data  
    mov ds, ax  
  
    mov ax, x  
    mov bx, y  
  
    add ax, bx  
  
    mov ax, 4c00h  
    int 21h  
  
end main
```



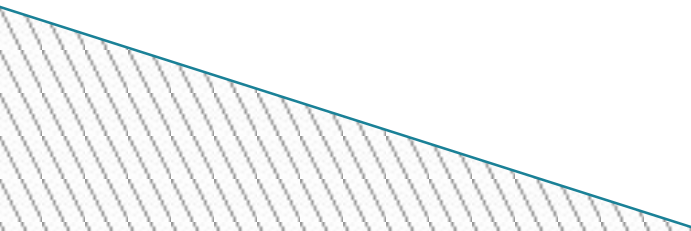
Compilando

Ferramentas necessárias:

- ▶ **Montador:** gera “*.obj” a partir de um “.asm”.
 - TASM
 - NASM
- ▶ **Linker:** transforma os arquivos “.obj” em executáveis.
 - Tlink

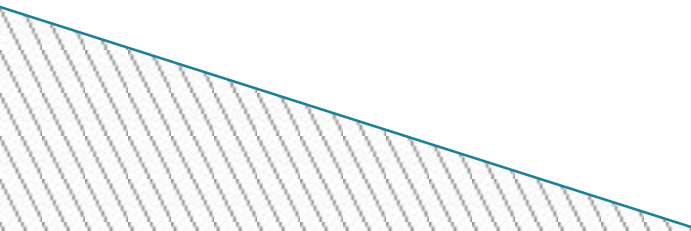
Compilando

Passo a passo:

- ↻ Abra o prompt de comando;
 - ↻ Digite: “tasm nomedoarquivo.asm”
 - ↻ Digite: “tlink nomedoarquivo.obj”
 - ↻ Digite: “nomedoarquivo.exe” para executar
- 

Exercícios

- ▶1 – Escreva um programa que calcula a média aritmética entre dois números e guarde o quociente no registrador AL.

 - ▶2 – Escreva um programa que lê dois números da memória e verifica se eles são divisíveis. Caso sejam, armazene 1 em DX, senão guarde 0.
- 

Referências

- ▶ The Pentium Microprocessor, James Antonakos.
- ▶ [Tabela de Instruções Intel x86](#)