

Program uruchomieniowy DEBUG

Program Debug umożliwia

- podgląd pamięci
- wprowadzanie kodu maszynowego lub wprowadzenie zasemblowanego kodu do pamięci
- wprowadzanie danych
- podgląd wykonywanego kodu
- krokową pracę programu

Rozkazy programu DEBUG

- A – asemblacja poleceń symbolicznych na kod maszynowy
- D – wyświetlanie zawartości pamięci dla danego adresu
- G – wykonanie programu
- T – krokowe wykonanie instrukcji
- R – wyświetlenie zawartości rejestrów
- Q- wyjście z programu

Zasady tworzenia rozkazów

- program nie uwzględnia wielkości liter
- użycie dwukropka na określenie segmentu i offsetu dostępu do pamięci
- kod heksadecymalny liczb
- użycie spacji do separacji mnemonika i argumentów operacji

Ekran programu DEBUG

The image shows a screenshot of the Windows DEBUG program. The main window title is "C:\WINDOWS\system32\debug.exe". The main display area shows a memory dump for address range 0100:0100 to 0100:0170. The dump consists of three columns: address, hexadecimal representation, and ASCII code. The hexadecimal values are all 00, and the ASCII codes are all dots. Three green arrows point from labels below to the columns of the dump: "adres" points to the first column, "Reprezentacja heksadecymalna" points to the second column, and "Kod ASCII" points to the third column.

```
C:\WINDOWS\system32\debug.exe
-d 0100:0100
0100:0100  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0110  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0120  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0130  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0140  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0150  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0160  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0100:0170  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

C:\WINDOWS\system32\debug.exe
0100:0160  00
0100:0170  00
-d
0100:0180  00
0100:0190  00
0100:01A0  00
0100:01B0  00
0100:01C0  00
0100:01D0  00
0100:01E0  00
0100:01F0  00
-r
AX=0000 BX=0000
DS=0C39 ES=0C39 SS=0C39 CS=0C39
0C39:0100 C7069D943E2B MOV     WORD PTR [ESI], EDI
-d 0c39:0100
0C39:0100  C7 06 9D 94 3E 2B 2E C7-06 9F 94 3D 3B E8 83 09  >+. .... =;...
0C39:0110  73 13 B8 FF FF 53 26 8B-1D 26 3A 0F 34 00 28 0C  s...S&.&:.4.<.
0C39:0120  00 5B E9 23 01 2E 89 36-D6 93 53 57 55 8D 1E DE  .[#...6..SWU...
0C39:0130  93 2E F6 06 D5 93 20 75-43 AC E8 0F 0A 72 3C E8  ....uC....r<.
0C39:0140  77 09 74 37 E8 A4 09 75-18 2E F6 06 D5 93 20 75  w.t7...u....u
0C39:0150  05 E8 3F 09 EB 26 2E F6-06 D5 93 41 74 1E 4E EB  ..?..&....At.N.
0C39:0160  1B 2E 88 07 3C 3D 75 06-2E 80 0E D5 93 01 43 E8  ....<=u.....C.
0C39:0170  08 0A 73 C5 AC 2E 88 07-43 EB BE 4E 2E 89 36 CB  ..s.....C..N..6.
```

adres Reprezentacja heksadecymalna Kod ASCII

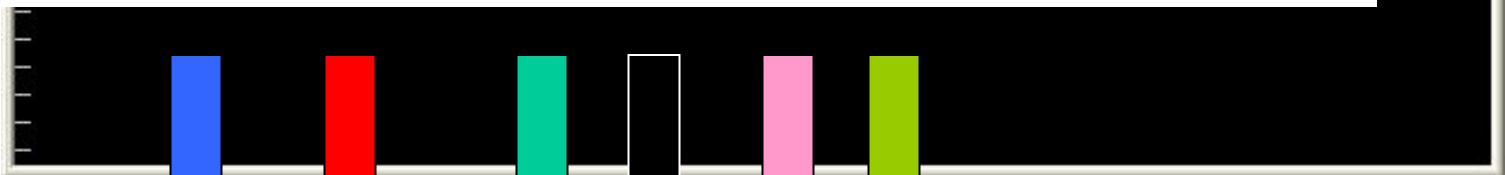
Podgląd lokacji w pamięci umożliwia identyfikację

- Wyposażenie systemowe
- Rozmiar pamięci
- Numer seryjny i informacje o prawach autorskich
- Data utworzenia ROM-BIOS
- Model ID

Lokalizacja słowa statusu sprzętu w obszarze danych BIOS 410H-411H

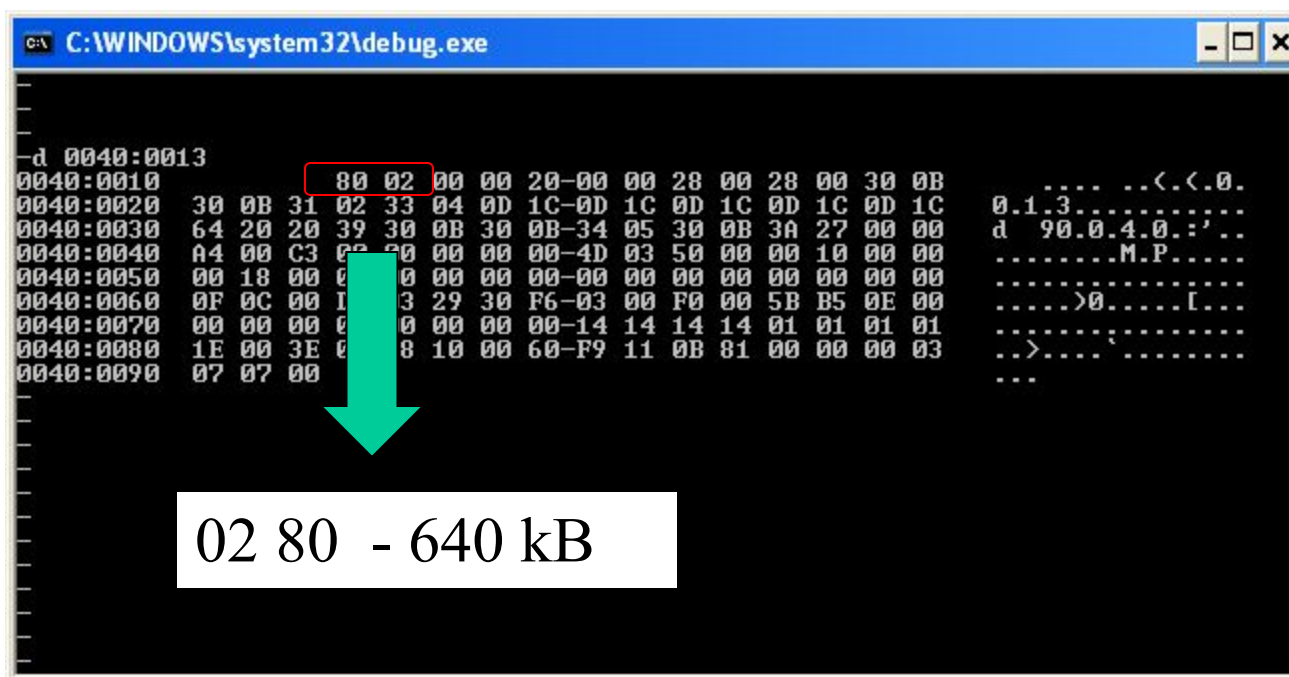
```
C:\WINDOWS\system32\debug.exe
-d 0040:0010
0040:0010 23 C8 00 80 02 00 00 20-00 00 1E 00 1E 00 0D 1C #.....d 90.0.4.
0040:0020 0D 1C 0D 1C 0D 1C 64 20-20 39 30 0B 30 0B 34 05 0.:.'0.0.1.0.....
0040:0030 30 0B 3A 27 30 0B 30 0B-31 02 30 0B 0D 1C 00 00 .....M.P.....
0040:0040 A2 00 C3 00 00 00 00 00-4D 03 50 00 00 10 00 00 .....>0.....11..
0040:0050 00 18 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....:.....
0040:0060 0F 0C 00 D4 03 29 30 F6-03 00 F0 00 5D 5D 0E 00 .....>.....
0040:0070 00 00 00 00 00 00 00 00-14 14 14 01 01 01 01 .....:.....
0040:0080 1E 00 3E 00 18 10 00 60-F9 11 0B 81 00 00 00 03 ..>.....
```

C823 – 1100 1000 0010 0011



Liczba portów równoległych, szeregowych; liczba zainstalowanych napędów elastycznych tryb video
koprocesor matematyczny napęd dysków elastycznych

Rozmiar pamięci konwencjonalnej 413H i 414H



```
C:\WINDOWS\system32\debug.exe

-d 0040:0013
0040:0010      80 02 00 00 20-00 00 28 00 28 00 30 0B      .....<<.
0040:0020 30 0B 31 02 33 04 0D 1C-0D 1C 0D 1C 0D 1C      0.1.3.....
0040:0030 64 20 20 39 30 0B 30 0B-34 05 30 0B 3A 27 00 00      d 90.0.4.0.:'.
0040:0040 A4 00 C3 00 00 00 00-4D 03 50 00 00 10 00 00      .....M.P.....
0040:0050 00 18 00 00 00 00 00-00 00 00 00 00 00 00 00      .....>0.....[...
0040:0060 0F 0C 00 00 00 00 00-03 00 F0 00 5B B5 0E 00      .....>.....[...
0040:0070 00 00 00 00 00 00 00-14 14 14 14 01 01 01 01      .....>.....
0040:0080 1E 00 3E 00 00 00 60-F9 11 0B 81 00 00 00 03      .....>.....
0040:0090 07 07 00
```

02 80 - 640 kB

Numer seryjny i informacje o prawach autorskich FE000H

```
C:\WINDOWS\system32\debug.exe

-d fe00:0000
FE00:0000  41 77 61 72 64 20 AA 01-00 00 00 00 00 00 49 42  Award .....IB
FE00:0010  4D 20 43 4F 4D 50 41 54-49 42 4C 45 20 34 38 36  M COMPATIBLE 486
FE00:0020  20 42 49 4F 53 20 43 4F-50 59 52 49 47 48 54 20  BIOS COPYRIGHT
FE00:0030  41 77 61 72 64 20 53 6F-66 74 77 61 72 65 20 49  Award Software I
FE00:0040  6E 63 2E B0 8F E6 70 E6-EB E4 71 E6 EB 0A C0 E9  nc....p...q.....
FE00:0050  31 14 61 72 64 20 53 6F-00 DA 54 E9 0E 14 20 43  l.ard So..T... C
FE00:0060  1B 41 77 61 72 64 20 4D-6F 64 75 6C 61 72 20 42  .Award Modular B
FE00:0070  49 4F 53 20 76 36 2E 30-30 50 47 00 DF 32 EC 33  IOS v6.00PG..2.3
```

Data ROM-BIOS FFFF5H

```
C:\WINDOWS\system32\debug.exe

-d ffff:0005
FFFF:0000          30 31 2F-30 34 2F 30 36 00 FC 06          01/04/06...
FFFF:0010    34 12 00 00 00 00 00 00-00 00 00 00 00 00 00 00 4.....
FFFF:0020    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
FFFF:0030    70 00 2E 8E 06 30 00 BF-7F 01 B9 02 00 AB 47 47  p...0.....GG
FFFF:0040    E2 FB CB 56 50 51 52 57-55 1E 06 53 8B EC 8B 76  ...UPQRWU..S..v
FFFF:0050    12 2E 8E 1E 30 00 8B 44-02 A2 22 00 88 26 08 01  ...0..D..".&..
FFFF:0060    8B 34 C4 1E 18 00 26 8A-47 01 26 8A 67 0D 26 8B  .4...&.G.&.g.&.
FFFF:0070    4F 12 26 8B 57 14 97 26-8A 47 02 2E 3A 04 73 2C  O.&.W..&.G...:s,
FFFF:0080    98 D1 E0 03 F0  ....
```

• Computer Model ID FFFF:E

- F8 PS/2 models 70 and 80
- FA PS/2 model 30
- FB PC-XT (1986)
- FC PC-AT (1984), PC-XT model 286,
- PS/2 models 50 and 60
- FE PC-XT (1982), portable (1982)
- FF Original IBM PC

```
C:\WINDOWS\system32\debug.exe

-d ffff:000e
ffff:0000          FC 06
ffff:0010  34 12 00 00 00 00 00 00 00-00 00 00 00 00 00 00  4.....
ffff:0020  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
ffff:0030  70 00 2E 8E 06 30 00 BF-7F 01 B9 02 00 AB 47 47  p...0.....GG
ffff:0040  E2 FB CB 56 50 51 52 57-55 1E 06 53 8B EC 8B 76  ...UPQRWU..S...v
ffff:0050  12 2E 8E 1E 30 00 8B 44-02 A2 22 00 88 26 08 01  ...0..D..".&..
ffff:0060  8B 34 C4 1E 18 00 26 8A-47 01 26 8A 67 0D 26 8B  .4...&.G.&.g.&.
ffff:0070  4F 12 26 8B 57 14 97 26-8A 47 02 2E 3A 04 73 2C  O.&.W..&.G...s,
ffff:0080  98 D1 E0 03 F0 97 26 C4-7F 0E FC 2E FF 54  .....&.....T
```

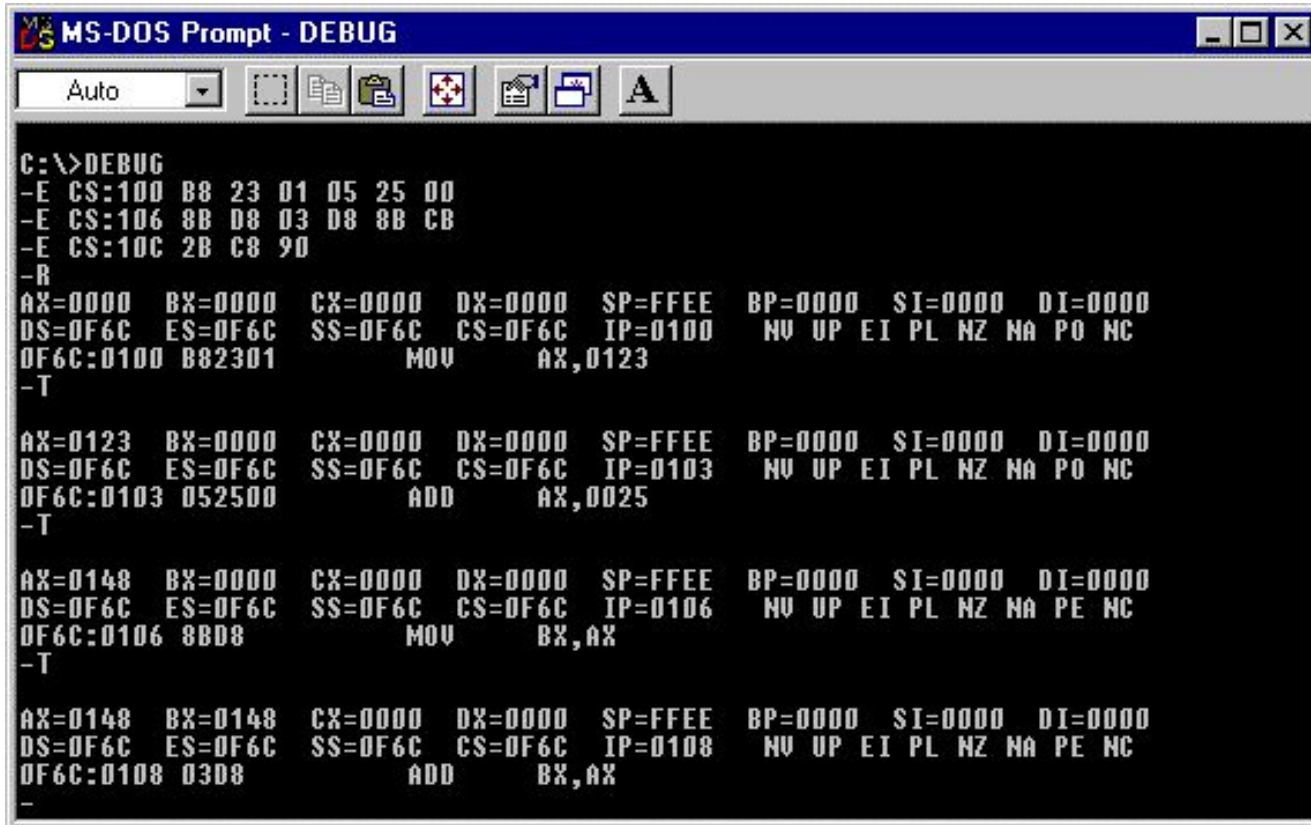
Język maszynowy i język symboliczny

- Wprowadzanie instrukcji programu
- Wykonywanie programu
- Zapisanie programu

Wprowadzanie instrukcji programu

- Kod maszynowy kod symboliczny
- B82301 MOV AX,0123
- 052500 ADD AX,0025
- 8BD8 MOV BX,AX
- 03D8 ADD BX,AX
- 8BCB MOV CX,BX
- 2BC8 SUB CX,AX
- 90 NOP

Bezpośrednie operowanie kodem maszynowym z wykorzystaniem procesu E



```
MS-DOS Prompt - DEBUG
Auto
C:\>DEBUG
-E CS:100 B8 23 01 05 25 00
-E CS:106 8B D8 03 D8 8B CB
-E CS:10C 2B C8 90
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 B82301      MOV     AX,0123
-T
AX=0123 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0103  NV UP EI PL NZ NA PO NC
0F6C:0103 052500      ADD     AX,0025
-T
AX=0148 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0106  NV UP EI PL NZ NA PE NC
0F6C:0106 8BD8      MOV     BX,AX
-T
AX=0148 BX=0148 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0108  NV UP EI PL NZ NA PE NC
0F6C:0108 03D8      ADD     BX,AX
-
```

Asemblacja i dezasemblacja

```
C:\WINDOWS\system32\debug.exe
DS=0C39 ES=0C39 SS=0C39 CS=0C39 IP=011E  NU UP EI PL ZR NA PE NC
0C39:011E 280C          SUB     [SI],CL          DS:0000=CD
-a 011e
0C39:011E mov ax,0200
0C39:0121 mov ax,[0200]
0C39:0124 add ax,bx
0C39:0126 add ax,[0200]
0C39:012A mov [0204],ax
0C39:012D add [0204],ax
0C39:0131 nop
0C39:0132
-u
0C39:011E B80002          MOV     AX,0200
0C39:0121 A10002          MOV     AX,[0200]
0C39:0124 01D8          ADD     AX,BX
0C39:0126 03060002       ADD     AX,[0200]
0C39:012A A30402          MOV     [0204],AX
0C39:012D 01060402       ADD     [0204],AX
0C39:0131 90             NOP
0C39:0132 F606D59320     TEST    BYTE PTR [93D5],20
0C39:0137 7543          JNZ
0C39:0139 AC             LODSB
0C39:013A E80F0A          CALL   0B4C
0C39:013D 723C          JB     017B
```

Język procesu asemblacji

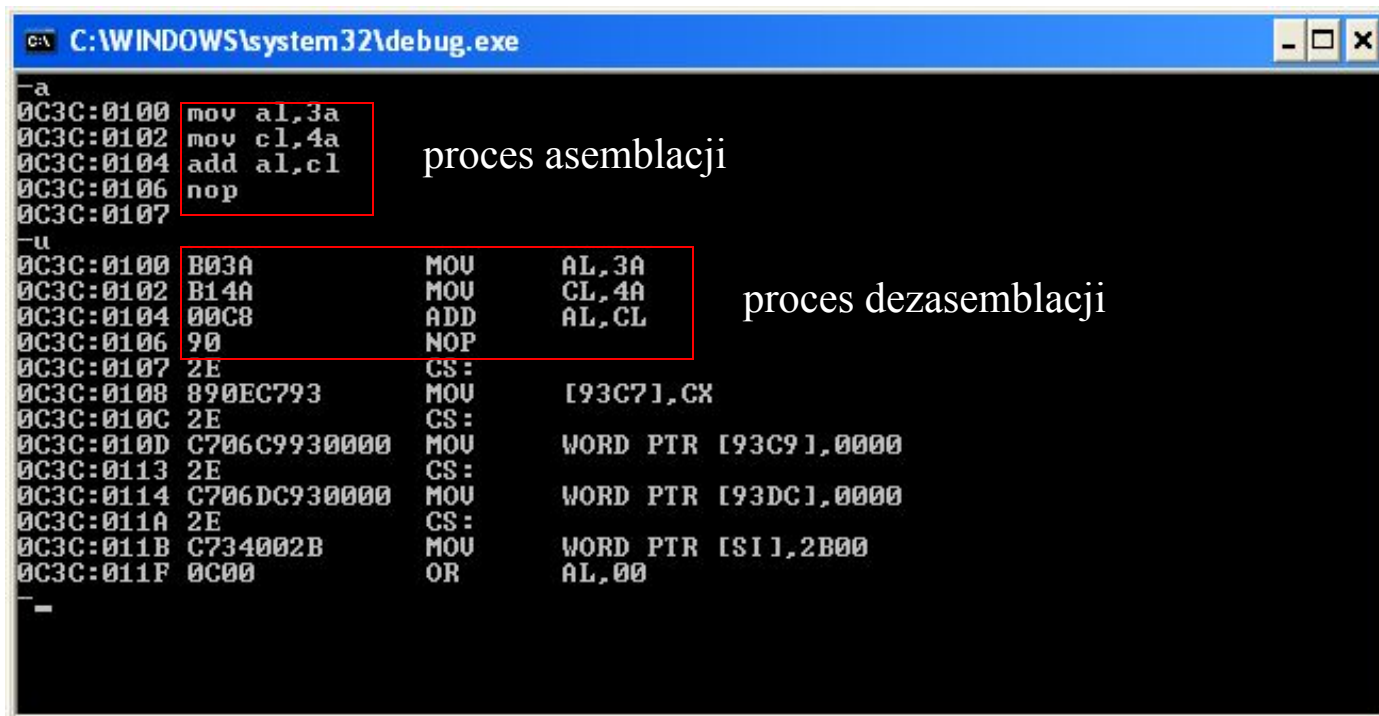
Proces asemblacji – wprowadzenie poleceń języka symbolicznego do pamięci operacyjnej, dekodowanie zapisu symbolicznego na kod maszynowy (zero-jedynkowy) w miejscu pamięci operacyjnej o adresie znajdującym się w rejestrach CS:IP

Rozkaz procesu asemblacji (A)

```
MOV CL, 42  
MOV DL, 2A  
ADD CL, DL  
NOP
```

Rozkaz procesu dezaseblacji (U) – przedstawienie rozkazów języka symbolicznego w kodzie maszynowym

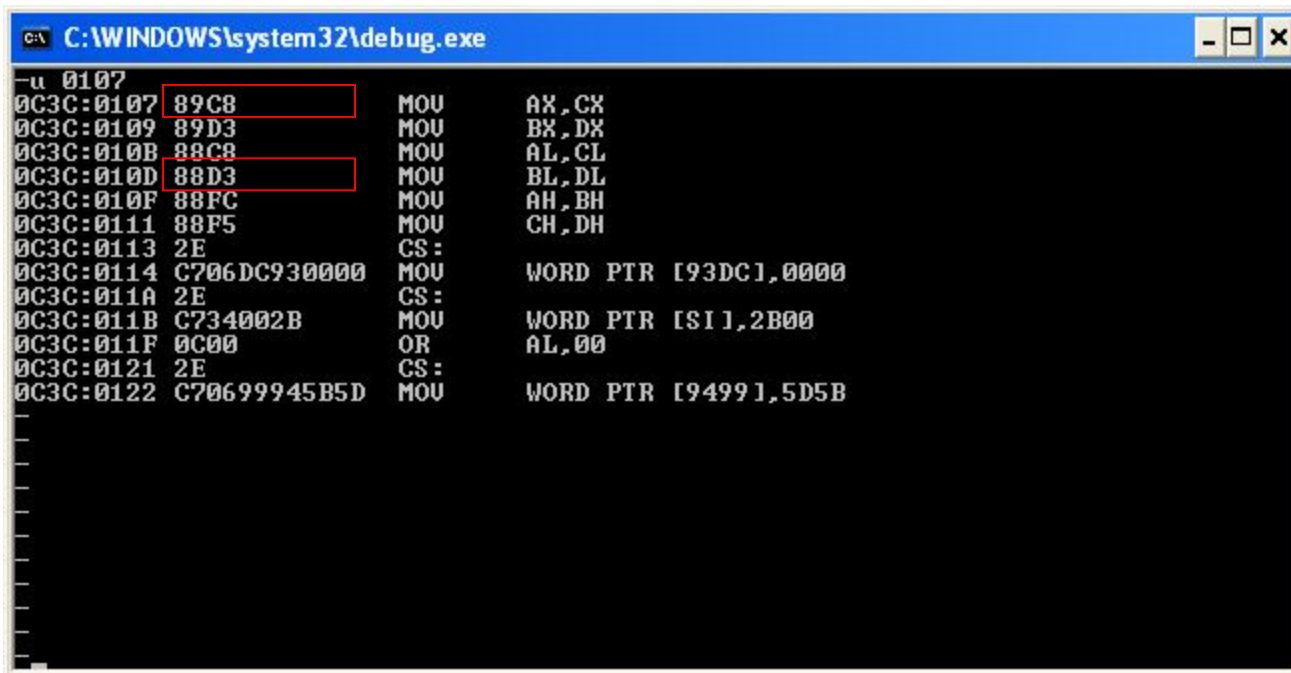
Przykład zastosowania procesu dezasemblacji



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\debug.exe". The window contains two sections of code, each highlighted with a red box. The first section, labeled "proces asemblacji", shows assembly instructions: "mov al,3a", "mov cl,4a", "add al,cl", and "nop". The second section, labeled "proces dezasemblacji", shows the corresponding disassembly instructions: "B03A MOU AL,3A", "B14A MOU CL,4A", "00C8 ADD AL,CL", and "90 NOP".

```
C:\WINDOWS\system32\debug.exe
-a
0C3C:0100 mov al,3a
0C3C:0102 mov cl,4a
0C3C:0104 add al,cl
0C3C:0106 nop
0C3C:0107
proces asemblacji
-u
0C3C:0100 B03A MOU AL,3A
0C3C:0102 B14A MOU CL,4A
0C3C:0104 00C8 ADD AL,CL
0C3C:0106 90 NOP
0C3C:0107 2E CS:
0C3C:0108 890EC793 MOU [93C71],CX
0C3C:010C 2E CS:
0C3C:010D C706C9930000 MOU WORD PTR [93C91],0000
0C3C:0113 2E CS:
0C3C:0114 C706DC930000 MOU WORD PTR [93DC1],0000
0C3C:011A 2E CS:
0C3C:011B C734002B MOU WORD PTR [SI],2B00
0C3C:011F 0C00 OR AL,00
-
```

Analiza kodu maszynowego operacji (1)



```
C:\WINDOWS\system32\debug.exe
-u 0107
0C3C:0107 89C8      MOU    AX, CX
0C3C:0109 89D3      MOU    BX, DX
0C3C:010B 88C8      MOU    AL, CL
0C3C:010D 88D3      MOU    BL, DL
0C3C:010F 88FC      MOU    AH, BH
0C3C:0111 88F5      MOU    CH, DH
0C3C:0113 2E        CS:
0C3C:0114 C706DC930000 MOU    WORD PTR [93DC1,0000]
0C3C:011A 2E        CS:
0C3C:011B C734002B MOU    WORD PTR [SI],2B00
0C3C:011F 0C00     OR     AL,00
0C3C:0121 2E        CS:
0C3C:0122 C70699945B5D MOU    WORD PTR [9499],5D5B
```

89C8 1000 1001 1100 1000

88D3 1000 1000 1101 0001

Analiza kodu maszynowego operacji (2)

mov ax,cx

89C8 1000 1001 11 00 1000

mov bl,dl

88D3 1000 1000 11 01 0011

1000 100 kod operacji przesłania informacji pomiędzy rejestrami

1 dla rejestru 16-bitowego
0 dla podrejestru 8-bitowego

11 kod operacji typu rejestr - rejestr

001 kod rejestru (podrejestru)

Analiza kodu maszynowego operacji (3)

89C8 1000 1001 1100 1000

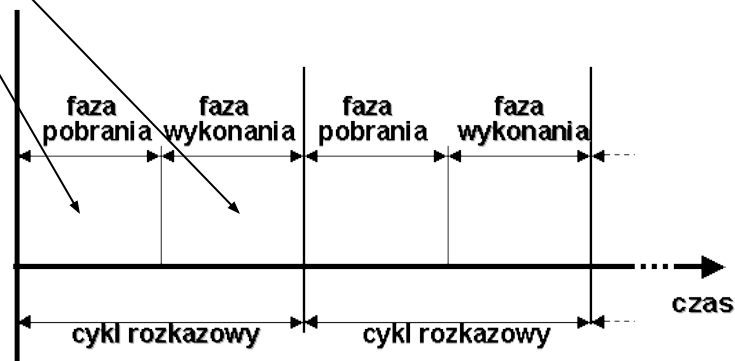
1000 1001

kod operacji ładowany do rejestru rozkazów

1100 1000

przesłanie zawartości rejestrów
(kopiowanie) dla rejestrów o danej wartości
kodowej

Fazy pracy procesora



Analiza kodu maszynowego operacji (4)

1. W rejestrze IP (licznik rozkazów PC, wskaźnik instrukcji) zawarty jest adres kodu wykonania operacji;
2. Zawartość komórki pamięci o adresie wskazywanym przez IP jest przesyłana do rejestrów rozkazów (IR);
3. Rejestr rozkazów współpracuje z dekodерem rozkazów, który „tłumaczy” kod na operacje;
4. Kolejny Bajt kodu zawiera argumenty wykonywanej operacji lub kody dodatkowe wykonywanej operacji

Przerwania – INTerrupts (1)

Przerwania są działaniami, za pomocą których układy zewnętrzne w odniesieniu do jednostki centralnej (CPU) sygnalizują zajście określonego zdarzenia i żądają określonego działania

Przerwania programowe

Przerwania programowe generowane są poprzez programy, które pozwalają na wywołanie określonego programu obsługi przerwania znajdującego się w pamięci ROM lub RAM

Przerwania programowe są częścią systemu BIOS (pamięć ROM-BIOS) oraz systemu operacyjnego (np..DOS)

oprogramowanie

DOS

Przerwania programowe systemu operacyjnego

Dostępny przez INT 21H

oprogramowanie systemowe

BIOS

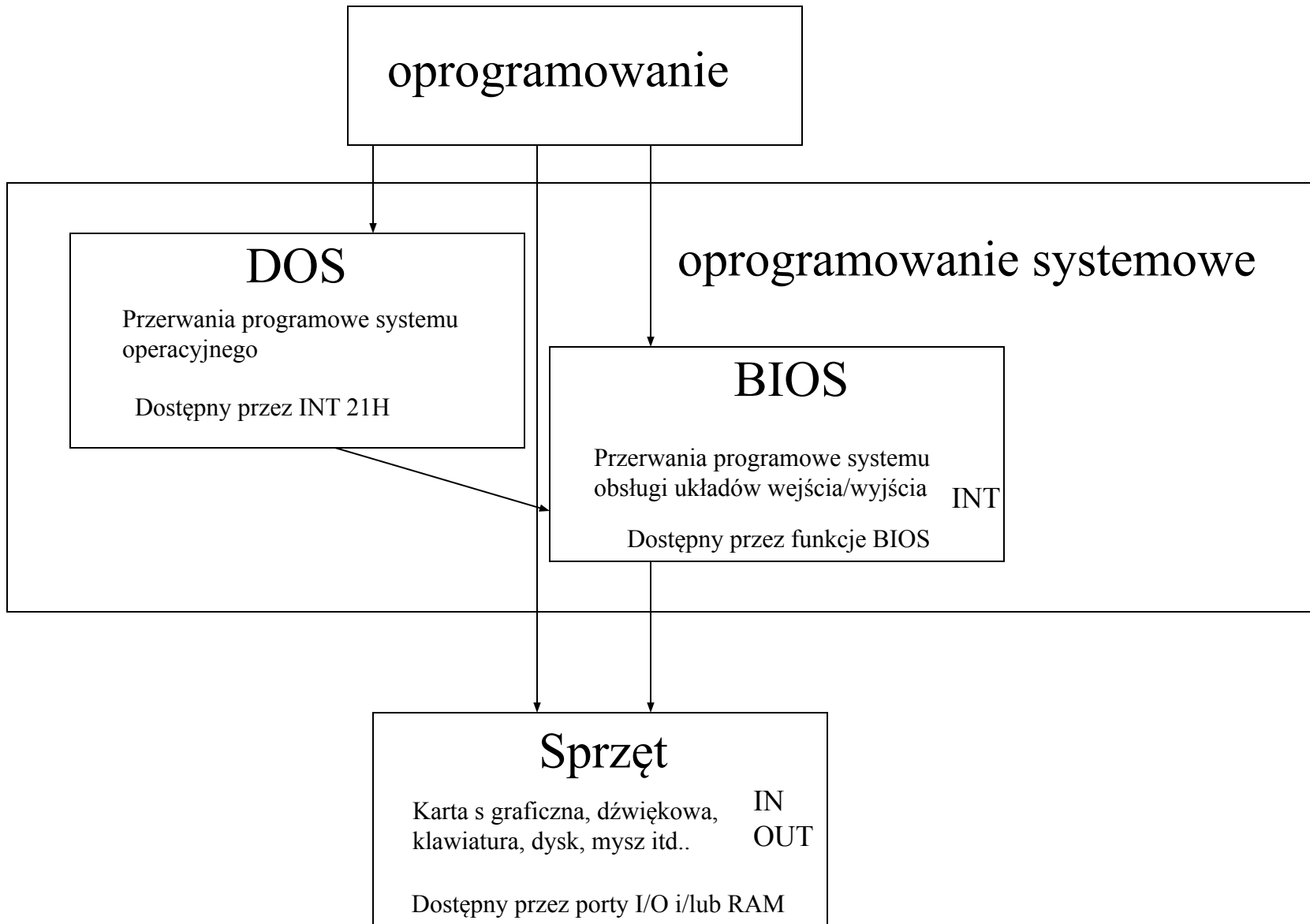
Przerwania programowe systemu obsługi układów wejścia/wyjścia INT

Dostępny przez funkcje BIOS

Sprzęt

Karta s graficzna, dźwiękowa, IN
klawiatura, dysk, mysz itd.. OUT

Dostępny przez porty I/O i/lub RAM



Przerwania – INTerrupts (2)

Program obsługi przerwania systemu operacyjnego jest lokowany w pamięci operacyjnej z HDD (HardDiscDrive), natomiast obsługi przerwania systemu BIOS z pamięci ROM-BIOS

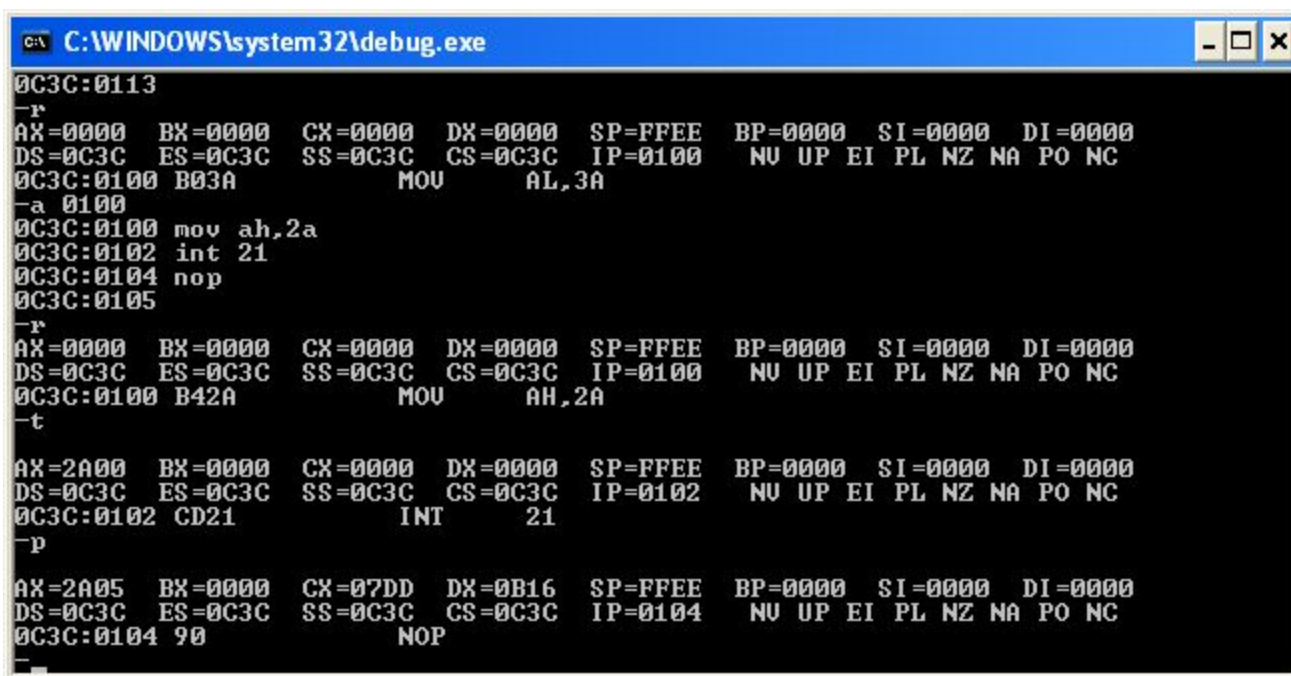
Wywołanie funkcji przerwania:

1. Wywołanie numeru funkcji przerwania poleceniem INT [numer_przerwania]
2. Odwołanie do tablicy wektorów przerwania, w której znajduje się adres funkcji przerwania
3. Wykonanie programu obsługi przerwania w pamięci operacyjnej, której miejsce wskazuje wektor przerwania.

Tablica wektorów przerwania znajduje się w początkowym 1KB obszarze pamięci operacyjnej I zawiera 256 wektorów przerwania, o długości wektora przerwania 4 B.

Przerwania – INTerrupts (3)

INT 21H, AH,2A – obecna data



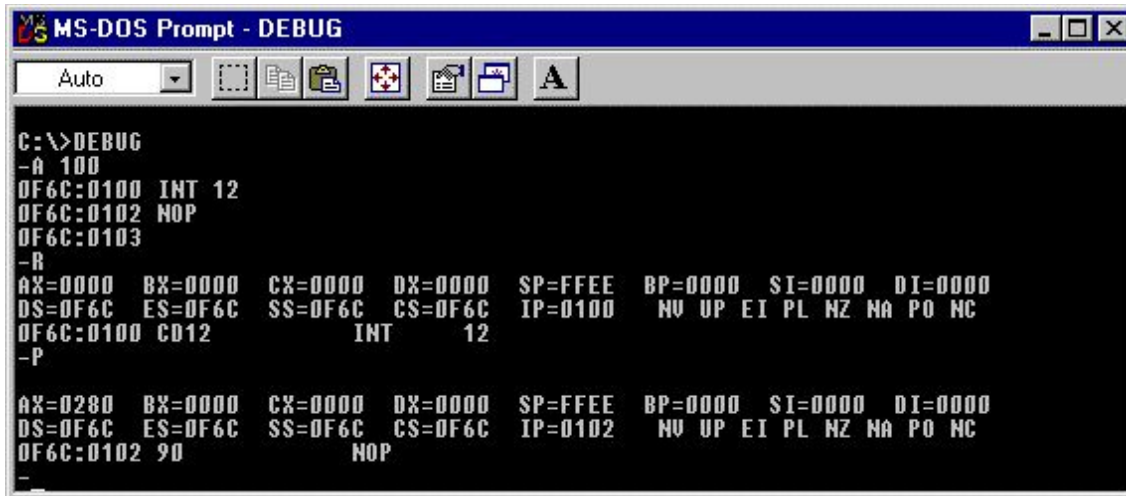
```
C:\WINDOWS\system32\debug.exe
0C3C:0113
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0100  NU UP EI PL NZ NA PO NC
0C3C:0100 B03A      MOV     AL,3A
-a 0100
0C3C:0100 mov ah,2a
0C3C:0102 int 21
0C3C:0104 nop
0C3C:0105
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0100  NU UP EI PL NZ NA PO NC
0C3C:0100 B42A      MOV     AH,2A
-t
AX=2A00 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0102  NU UP EI PL NZ NA PO NC
0C3C:0102 CD21      INT     21
-p
AX=2A05 BX=0000 CX=07DD DX=0B16 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0104  NU UP EI PL NZ NA PO NC
0C3C:0104 90          NOP
```

CX = year (1980-2099)

DH = month

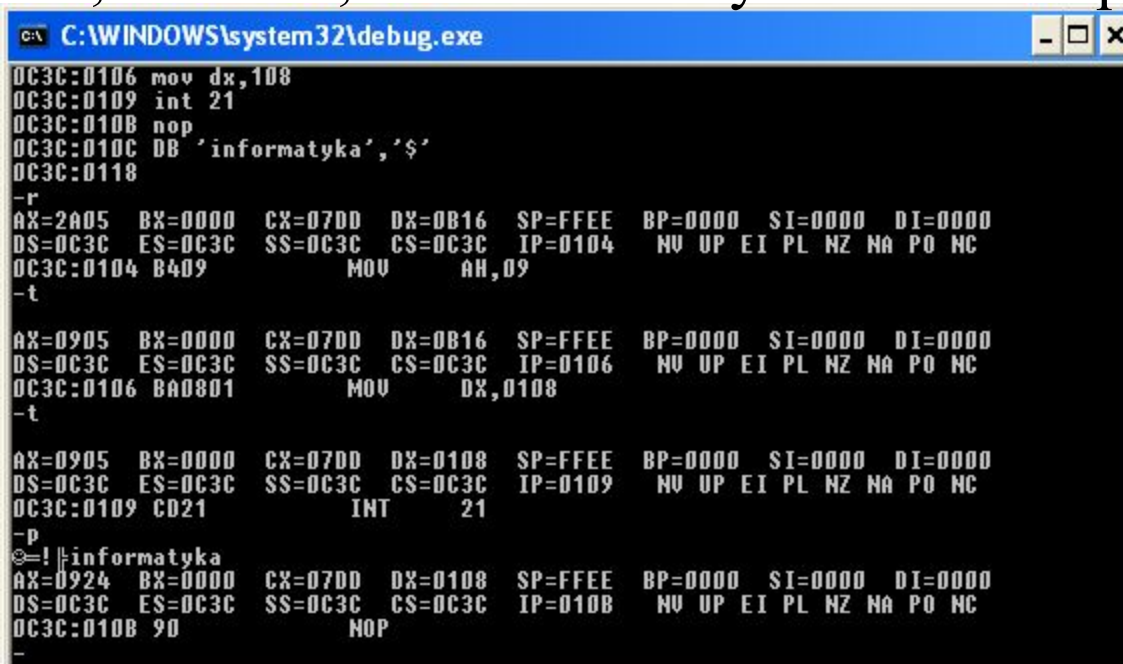
DL = day

INT 12H - rozmiar pamięci podstawowej



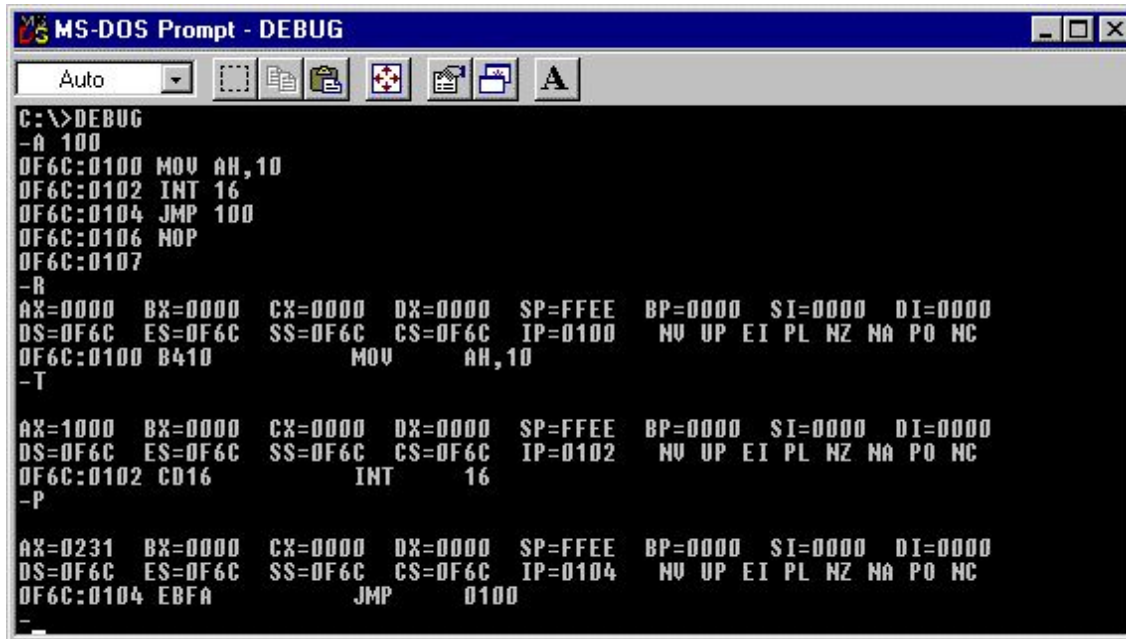
```
MS-DOS Prompt - DEBUG
Auto
C:\>DEBUG
-A 100
DF6C:0100 INT 12
DF6C:0102 NOP
DF6C:0103
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=DF6C ES=DF6C SS=DF6C CS=DF6C IP=0100  NV UP EI PL NZ NA PO NC
DF6C:0100 C012          INT     12
-P
AX=0280 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=DF6C ES=DF6C SS=DF6C CS=DF6C IP=0102  NV UP EI PL NZ NA PO NC
DF6C:0102 90          NOP
-
```

INT 21H, AH 09H, DX 108H – wyświetlenie napisu



```
C:\WINDOWS\system32\debug.exe
DC3C:0106 mov dx,108
DC3C:0109 int 21
DC3C:010B nop
DC3C:010C DB 'informatyka','$'
DC3C:0118
-r
AX=2A05 BX=0000 CX=07DD DX=0B16 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0104  NV UP EI PL NZ NA PO NC
DC3C:0104 B409          MOV     AH,09
-t
AX=0905 BX=0000 CX=07DD DX=0B16 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0106  NV UP EI PL NZ NA PO NC
DC3C:0106 BA0801      MOV     DX,0108
-t
AX=0905 BX=0000 CX=07DD DX=0108 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=0109  NV UP EI PL NZ NA PO NC
DC3C:0109 C021          INT     21
-p
©=!|informatyka
AX=0924 BX=0000 CX=07DD DX=0108 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C3C ES=0C3C SS=0C3C CS=0C3C IP=010B  NV UP EI PL NZ NA PO NC
DC3C:010B 90          NOP
-
```

INT 16 AH 10H –oczekiwanie na naciśnięcie dowolnego klawisza, aby przejść do następnej linijki programu



```
MS-DOS Prompt - DEBUG
Auto
C:\>DEBUG
-A 100
0F6C:0100 MOV AH,10
0F6C:0102 INT 16
0F6C:0104 JMP 100
0F6C:0106 NOP
0F6C:0107
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NU UP EI PL NZ NA PO NC
0F6C:0100 B410          MOV     AH,10
-T
AX=1000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102  NU UP EI PL NZ NA PO NC
0F6C:0102 C016          INT     16
-P
AX=0231 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0104  NU UP EI PL NZ NA PO NC
0F6C:0104 EBFA          JMP     0100
-
```

Stworzenie i zachowanie programu

- wprowadzenie kodu źródłowego przy użyciu procesu: A lub E
- wprowadzenie nazwy programu przy użyciu procesu: N [nazwa_pliku]
- wyzerowanie rejestru BX przy użyciu procesu RBX
- zapisanie lub zachowanie pliku na dysku przy użyciu procesu: W

Modyfikacja istniejącego programu

- wprowadzenie w ścieżce DOS polecenia: DEBUG [nazwa pliku]
- użycie procesu: E dla dokonania poprawek
- użycie procesu: W dla zapisania pliku na dysku

Pamięć operacyjna trybu rzeczywistego

0x0000:0x0000 - tablica wektorów przerwań

0x0000:0x7C00 - tu zostaje załadowany boot-sector przez BIOS

0x1000:0x0000-0x9000:0xFFFF - pamięć użytkownika (najlepiej używać z tego przedziału)

0xA000:0x0000 - pamięć video karty VGA (tylko dla trybu graficznego)

0xB000:0x0000 - pamięć video karty Hercules Monochrome

0xB800:0x0000 - pamięć trybu tekstowego karty VGA

0xC000:0x0000-0xF000:0xFFFF - pamięć BIOSu i inne

Pamięć PC podzielona jest na 4 logiczne obszary:

I. Conventional Memory - jest to pierwsze 640kb pamięci systemowej. Jest to obszar dostępny dla programisty. Adres : 0000h - 9FFFFh.

II. Upper Memory Area (UMA) - jest to wyższe 384 bajty pierwszego megabajta pamięci, bezpośrednio ponad pamięcią konwencjonalną. Jest ona zarezerwowana do użytku przez urządzenia systemowe oraz ROM razem ze sterownikami. Adres : A0000h - FFFFFh.

III. High Memory Area (HMA) - pierwsze 64kb drugiego megabajta pamięci, czyli pierwsze 64kb pamięci rozszerzonej, która może być dostępna w trybie rzeczywistym. Adres: 100000h - 10FFEFh.

IV. Extended Memory (pamięć rozszerzona) - jest to cała pamięć powyżej pierwszego megabajta. Dostęp do niej mamy jedynie w trybie chronionym. Adres 10FFF0h do końca pamięci operacyjnej.