

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Лабораторна робота №5

з дисципліни: «Архітектура комп'ютера»

УПРАВЛІННЯ ХОДОМ ВИКОНАННЯ ПРОГРАМИ НА АСЕМБЛЕРІ АРХІТЕКТУРИ IA-32 (X86) У REAL ADDRESS MODE

СПЕЦІАЛЬНОСТІ

121 – Інженерія програмного забезпечення

Виконав: Степаненко А.Ю., Салимоненко В.О., Дурдинець О.Т.

Підпис: _____

Кількість балів: _____ Оцінка _____

Група: IT-01

Викладач: Бердник Ю. М.

Підпис: _____

Київ 2020

Мета лабораторної роботи:

Вивчити додаткові команди Асемблера, що дають можливість керувати ходом програми. Вивчити додаткові переривання DOS і BIOS для відображення інформації на консолі.


Завдання для ЛР 5

1. Користуючись результатами роботи 1, 2, заповнити внутрі масиву array2Db ділянку розміром 8x8 числами дня, місяця, року народження студенту, що є елементами масиву.
2. Кординати початку ділянки (i, j) верхнього лівого кута мають відповідати варіанту.
3. Створити процедуру Асемблера, що робить сортування масиву для парного варіанту за зростанням, для непарного варіанту за зменшенням. arr_dup2D

За бажанням студента написати фрагмент коду, що здійснює вивід масиву на консоль з використанням функцій DOS або BIOS.

Реалізація:

Для спрощення був використаний .bat файл який дозволяє запускати турбо дебагер

```
C: > TASM >  vatnikTd.bat  
1  @echo off  
2  
3  set arg=%lab5  
4  tasm %arg%.asm  
5  tlink %arg%.obj  
6  td %arg%.exe
```

Код для програми:

```
C: > TASM > lab5.asm
1 .model small
2 .stack 256h
3 ;макрос для виходу з програми
4
5 .data
6 Array dw 1253h, 5739h, 1537h, 8674h, 3752h, 1964h, 2145h, 6831h, 2363h, 2988h, 5482h, 8762h, 1321h, 4895h, 2641h, 0913h ;1
7 dw 5437h, 5627h, 2894h, 3613h, 7865h, 3178h, 8937h, 2164h, 8970h, 5210h, 2615h, 1542h, 7847h, 8263h, 5895h, 4785h ;2
8 dw 4587h, 2361h, 1423h, 1485h, 1369h, 7855h, 1452h, 4685h, 1217h, 9868h, 5232h, 7874h, 5854h, 4578h, 5362h, 2585h ;3
9 dw 1454h, 1530h, 7454h, 8566h, 3226h, 4754h, 1563h, 2573h, 1984h, 8763h, 4129h, 6214h, 8742h, 8962h, 8753h, 8513h ;4
10 dw 2325h, 5784h, 1451h, 6986h, 5743h, 5842h, 8521h, 8969h, 3214h, 4789h, 5263h, 5478h, 5412h, 9534h, 7513h, 2196h ;5
11 dw 6587h, 6542h, 5236h, 5745h, 4523h, 2341h, 7812h, 9648h, 2684h, 3572h, 6255h, 3458h, 1452h, 1457h, 3256h, 8547h ;6
12 dw 1478h, 2486h, 1268h, 3589h, 2452h, 5896h, 6587h, 2359h, 1452h, 6556h, 2547h, 5221h, 2189h, 3278h, 9641h, 5438h ;7
13 dw 4785h, 5244h, 4787h, 5226h, 3247h, 1263h, 1593h, 6547h, 2581h, 3278h, 4258h, 9587h, 4257h, 4215h, 6521h, 9854h ;8
14 dw 2561h, 3274h, 3025h, 5846h, 5810h, 6309h, 8574h, 2653h, 1239h, 8956h, 8751h, 8563h, 8451h, 8712h, 2548h, 3024h
15 dw 2781h, 8756h, 8965h, 5412h, 7845h, 3264h, 9856h, 2187h, 3201h, 4858h, 9526h, 6325h, 3478h, 3218h, 2056h, 3057h
16 dw 0478h, 3205h, 9853h, 2103h, 2076h, 6254h, 7846h, 3558h, 2556h, 4851h, 3259h, 2563h, 1478h, 4512h, 3258h, 3259h
17 dw 1542h, 7845h, 0256h, 2482h, 6842h, 3621h, 3259h, 7412h, 3698h, 9874h, 5721h, 5624h, 3552h, 2541h, 2984h, 2563h
18 dw 2565h, 4152h, 2585h, 3658h, 3256h, 1278h, 5326h, 3025h, 9856h, 2644h, 3265h, 2158h, 4856h, 5680h, 1475h, 0245h
19 dw 7845h, 1588h, 3256h, 7856h, 6325h, 3256h, 8542h, 4521h, 6235h, 7845h, 8562h, 1205h, 3482h, 4784h, 1542h, 1239h
20 dw 2586h, 6589h, 5523h, 4157h, 3584h, 2015h, 4785h, 2635h, 1451h, 4826h, 1243h, 1148h, 6825h, 4521h, 1536h, 5429h
21 dw 9852h, 5374h, 8945h, 1542h, 8752h, 9635h, 2356h, 2784h, 4587h, 5642h, 7319h, 5493h, 4862h, 2255h, 3278h, 9542h; 1DA0h
22 Len dw 257
23 CODESEG
24 ;-----VI. ПОЧАТОК СЕГМЕНТУ КОДУ
25 Start:
26
27 main proc
28 mov ax, @data ; ax <- @data
29 mov ds, ax ; ds <- ax
30 mov es, ax ; es <- ax
31
32 mov cx, 5h
33 mov si, 108h
34 copy_loop:
35 mov [ds:si], 3431h ; get number from array_array stack & set it to bx as a temp variable
36 add si, 2 ; set value from bx to ds with offset
37 mov [ds:si], 3131h
38 add si, 2 ; si value + 2
39 mov [ds:si], 3032h
40 add si, 2
41 mov [ds:si], 3230h
42 add si, 10
43
44 mov [ds:si], 3231h ; get number from array_array stack & set it to bx as a temp variable
45 add si, 2 ; set value from bx to ds with offset
46 mov [ds:si], 3131h
47 add si, 2 ; si value + 2
48 mov [ds:si], 3032h
49 add si, 2
50 mov [ds:si], 3230h
51 add si, 10
52
53 mov [ds:si], 3732h ; get number from array_array stack & set it to bx as a temp variable
54 add si, 2 ; set value from bx to ds with offset
55 mov [ds:si], 3031h
56 add si, 2 ; si value + 2
57 mov [ds:si], 3032h
58 add si, 2
59 mov [ds:si], 3330h
60 add si, 10
61 loop copy_loop
62 mov [ss:1EB8h], 0320h
63 mov [ss:1EB6h], 1027h
64 mov [ss:1EB4h], 0220h
65 mov [ss:1EB2h], 1112h
66 mov [ss:1EB0h], 0420h
67 mov [ss:1EAEh], 1114h
68
```

C: > TASM > ASM lab5.asm

```
51         add si, 10
52
53         mov [ds:si], 3732h          ; get number from array_array stack & set it to bx as a temp variable
54         add si, 2                  ; set value from bx to ds with offset
55         mov [ds:si], 3031h
56         add si, 2                  ; si value + 2
57         mov [ds:si], 3032h
58         add si, 2
59         mov [ds:si], 3330h
60         add si, 10
61         loop copy_loop
62     mov [ss:1EB8h], 0320h
63     mov [ss:1EB6h], 1027h
64     mov [ss:1EB4h], 0220h
65     mov [ss:1EB2h], 1112h
66     mov [ss:1EB0h], 0420h
67     mov [ss:1EAeh], 1114h
68
69     lea si, Array
70     mov cx, Len
71     call BubbleSort
72     mov ax, 01h
73     mov ax, 01h
74     mov ax, 01h
75     mov ax, 01h
76 endp main
77
78 BubbleSort    proc
79     push ax
80     push bx
81     push cx
82     push dx
83     push si
84     push di
85
86     mov bx, si
87     mov dx, cx
88     dec dx
89     shl dx, 1
90     dec cx
91     mov si, 0
92 ForI:
93     mov di, dx
94 ForJ:
95     mov ax, [bx+di-2]
96     cmp ax, [bx+di]
97     jbe NextJ
98     xchg ax, [bx+di]
99     xchg ax, [bx+di-2]
100    xchg ax, [bx+di]
101 NextJ:
102    sub di, 2
103    cmp di, si
104    ja ForJ
105    add si, 2
106    loop ForI
107
108    pop di
109    pop si
110    pop dx
111    pop cx
112    pop bx
113    pop ax
114    ret
115 BubbleSort    endp
116
117 end Start
```

Наш не відсортований масив у дампі пам'яті:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help READY

[]=Dump 2 []

```

ds:0000 53 12 39 57 37 15 74 86 52 37 64 19 45 21 31 68 S+9W78tãR7dJE!1h
ds:0010 63 23 88 29 82 54 62 87 21 13 95 48 41 26 13 09 c#ê)éTbç!!òH&!!o
ds:0020 37 54 27 56 94 28 13 36 65 78 78 31 37 89 64 21 7T' Uó(!!6exx17éd!
ds:0030 70 89 10 52 15 26 42 15 47 78 63 82 95 58 85 47 pè!R&B&GxcéòXàG
ds:0040 87 45 61 23 23 14 85 14 69 13 55 78 52 14 85 46 gEa##qã!i!!UxR!aF
ds:0050 17 12 68 98 32 52 74 78 54 58 78 45 62 53 85 25 f+hj2RtXTxEbSàz
ds:0060 54 14 30 15 54 74 66 85 26 32 54 47 63 15 73 25 T!08Ttfa&2TGc8s%
ds:0070 84 19 63 87 29 41 14 62 42 87 62 89 53 87 13 85 a!cç)A!bBçbèSç!!à
ds:0080 25 23 84 57 51 14 86 69 43 57 42 58 21 85 69 89 %!ãWQ!ã!iCWbX!à!iè
ds:0090 14 32 89 47 63 52 78 54 12 54 34 95 13 75 96 21 !ZèGçRXT+T4ò!!uü!
ds:00A0 87 65 42 65 36 52 45 57 23 45 41 23 12 78 48 96 çeBe6REW##EA#fXHu
ds:00B0 84 26 72 35 55 62 58 34 52 14 57 14 56 32 47 85 a&r5UbX4R!W!WU2Gà
ds:00C0 78 14 86 24 68 12 89 35 52 24 96 58 87 65 59 23 x!ã$!h!è5R$ûXçeY#
ds:00D0 52 14 56 65 47 25 21 52 89 21 78 32 41 96 38 54 R!UeG%!Rè!x2Aû8T
ds:00E0 85 47 44 52 87 47 26 52 47 32 63 12 93 15 47 65 àGDRçG&RG2c!ò8Gç
ds:00F0 81 25 78 32 58 42 87 95 57 42 15 42 21 65 54 98 üxX2XBçòWB8B!eTj
ds:0100 61 25 74 32 25 30 46 58 10 58 09 63 74 85 53 26 axt2x0FX>XoctàS&
ds:0110 39 12 56 89 51 87 63 85 51 84 12 87 48 25 24 30 9!UeQçcàQã!çH!$0
ds:0120 81 27 56 87 65 89 12 54 45 78 64 32 56 98 87 21 ü' Uçèè!TExd2Ujç!
ds:0130 01 32 58 48 26 95 25 63 78 34 18 32 56 20_57 30 @2XH&òçx412U W0
ds:0140 78 04 05 32 53 98 03 21 76 20 54 62 46 78 58 35 x+ç2Sj!v TbFxX5

```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

FileEditViewRunBreakpointsDataOptionsWindowHelp

READY

[]=Dump2[]

ds:00B084267235556258345214571456324785ä&r5UbX4R9W9U2Gà

ds:00C078148624681289355224965887655923x9lâ\$ht#e5R\$ôXgeY#

ds:00D052145665472521528921783241963854R9W9eGz!Rè!x2Aû8T

ds:00E085474452874726524732631293154765àGDRgG&RG2c†ô\$Ge

ds:00F081257832584287955742154221655498ûz×2XBgðWB\$B!eTj

ds:010061257432253046581058096374855326azt2z×OFX>XoctàS&

ds:0110391256895187638551841287482524309†UeQgcàQä†gHz\$0

ds:012081275687658912544578643256988721ü'Ugeë†TExd2Ujg!

ds:013001325848269525637834183256205730@2XH&ðzcx4†2UW0

ds:014078040532539803217620546246785835x+æ2Sij!†vTbF×X5

ds:015056255148593263257814124558325932UzQHY2c×x9l†EX2YZ

ds:016042154578560282244268213659321274B\$ExU0é\$Bh!6YZ†t

ds:017098367498215724565235412584296325ij6tij!W\$UR5A×ä)c%

ds:018065255241852558365632781226532530ezRAàzX6U2x†&S×0

ds:019056984426653258215648805675144502UjD&e2X!UHÇUu9IE0

ds:01A045788815563256782563563242852145Exê\$U2U×zcU2Bà!E

ds:01B0356245786285051282348447421539125bExbàæ†é4äGB\$9†

ds:01C086258965235557418435152085473526âzëe#UWAä5\$àG5&

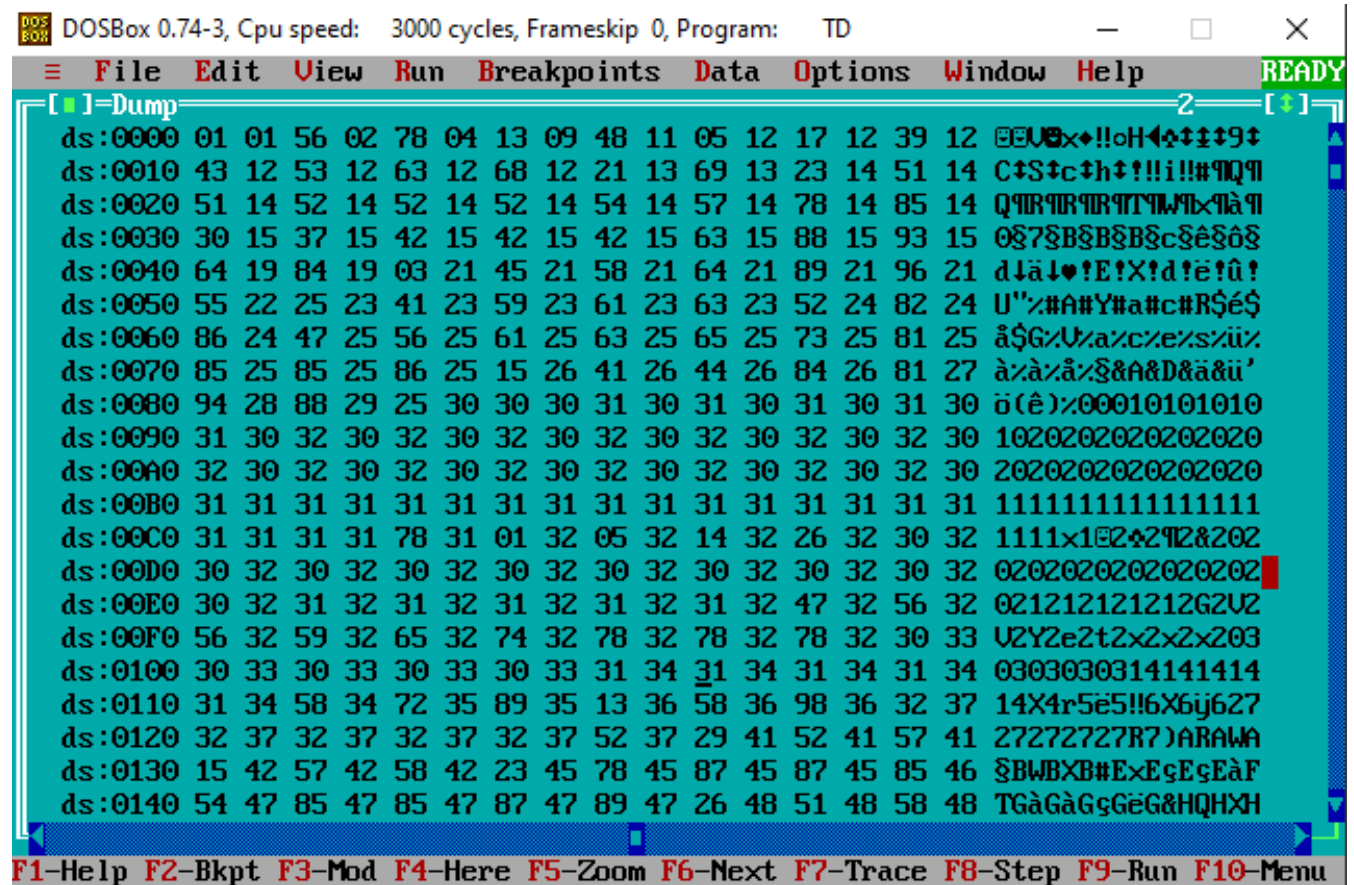
ds:01D051142648431248112568214536152954Q9l&HC#H<z!E6\$)T

ds:01E052987453458942155287359656238427RijtSEeB\$Rg5ûUwä'

ds:01F087454256197393546248552278323030gEBU!sôTbHU"x200

F1-HelpF2-BkptF3-ModF4-HereF5-ZoomF6-NextF7-TraceF8-StepF9-RunF10-Menu

Наш відсортований масив за допомогою Bubble sort:



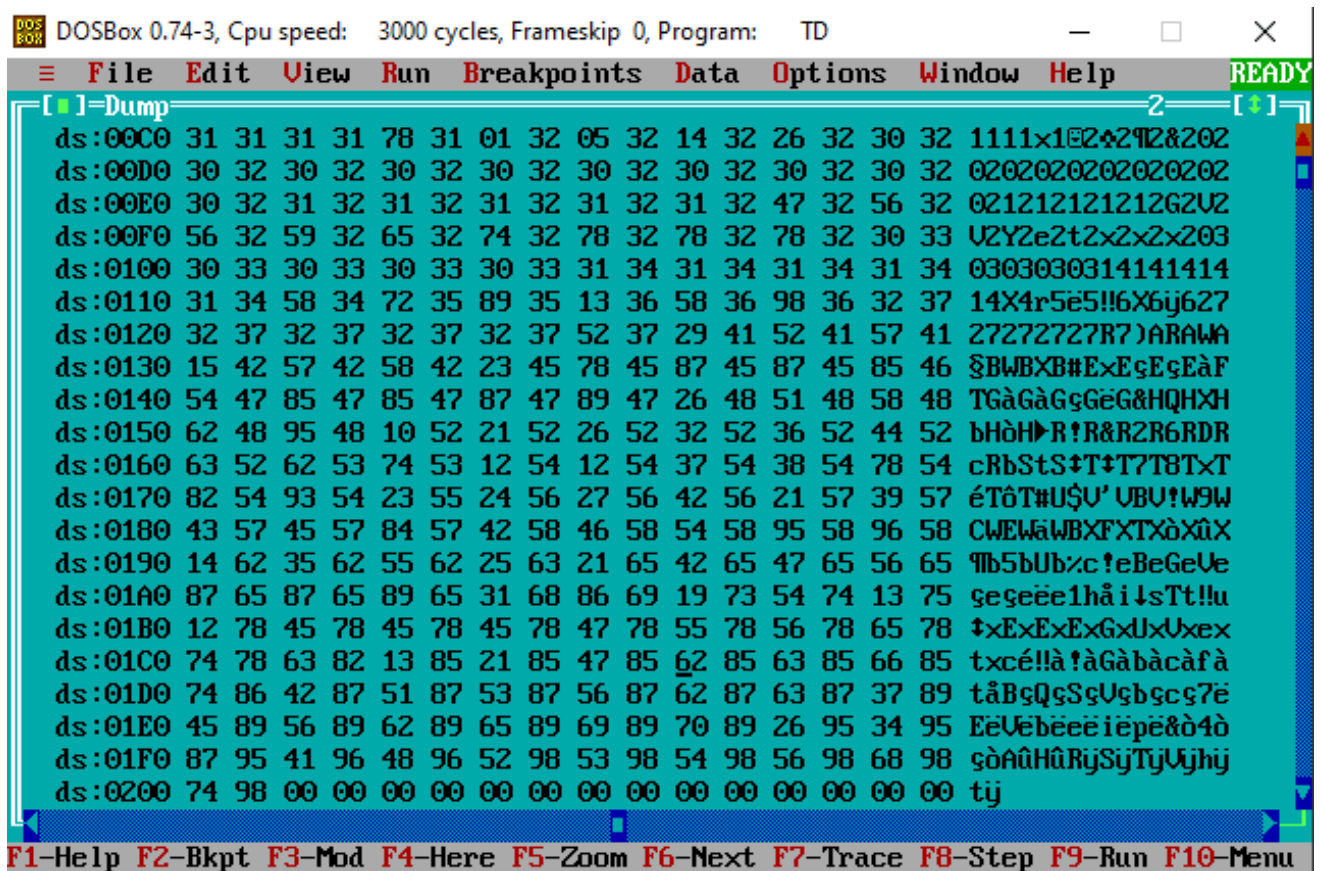
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[]=Dump 2 [+]

```
ds:0000 01 01 56 02 78 04 13 09 48 11 05 12 17 12 39 12 00U0x+!!oH4+!t9+
ds:0010 43 12 53 12 63 12 68 12 21 13 69 13 23 14 51 14 C+S+c+h+!i!!#qQq
ds:0020 51 14 52 14 52 14 52 14 54 14 57 14 78 14 85 14 QqRqRqRqRqRqRqRqRqRq
ds:0030 30 15 37 15 42 15 42 15 42 15 63 15 88 15 93 15 0878B8B8B8c8e8o8
ds:0040 64 19 84 19 03 21 45 21 58 21 64 21 89 21 96 21 d!ä!+!E!X!d!ë!û!
ds:0050 55 22 25 23 41 23 59 23 61 23 63 23 52 24 82 24 U"%#A#Y#a#c#R$é$
ds:0060 86 24 47 25 56 25 61 25 63 25 65 25 73 25 81 25 å$G%U%a%c%e%$%ü%
ds:0070 85 25 85 25 86 25 15 26 41 26 44 26 84 26 81 27 à%à%ä%$&A&D&ä&ü'
ds:0080 94 28 88 29 25 30 30 30 31 30 31 30 31 30 31 30 ö(ê)%00010101010
ds:0090 31 30 32 30 32 30 32 30 32 30 32 30 32 30 32 30 1020202020202020
ds:00A0 32 30 32 30 32 30 32 30 32 30 32 30 32 30 32 30 2020202020202020
ds:00B0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 1111111111111111
ds:00C0 31 31 31 31 78 31 01 32 05 32 14 32 26 32 30 32 1111x1E2+2Y2&202
ds:00D0 30 32 30 32 30 32 30 32 30 32 30 32 30 32 30 32 0202020202020202
ds:00E0 30 32 31 32 31 32 31 32 31 32 31 32 47 32 56 32 021212121212G2U2
ds:00F0 56 32 59 32 65 32 74 32 78 32 78 32 78 32 30 33 U2Y2e2t2x2x2x203
ds:0100 30 33 30 33 30 33 30 33 31 34 31 34 31 34 31 34 0303030314141414
ds:0110 31 34 58 34 72 35 89 35 13 36 58 36 98 36 32 37 14X4r5e5!!6X6ij627
ds:0120 32 37 32 37 32 37 32 37 52 37 29 41 52 41 57 41 27272727R7)ARAWA
ds:0130 15 42 57 42 58 42 23 45 78 45 87 45 87 45 85 46 8BWBXB#ExEgEgEaF
ds:0140 54 47 85 47 85 47 87 47 89 47 26 48 51 48 58 48 TGàGàGgGgG&HQHXH
```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu



Наші дати народження в дампі:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[.] = Dump 2 [.]

ds:00B0	84	26	72	35	55	62	58	34	52	14	57	14	56	32	47	85	ä&r5UbX4R9W9U2Gà
ds:00C0	78	14	86	24	68	12	89	35	52	24	96	58	87	65	59	23	x9lâ\$#e5R\$0XgeY#
ds:00D0	52	14	56	65	47	25	21	52	89	21	78	32	41	96	38	54	R9UeGz!Rè!x2A08T
ds:00E0	85	47	44	52	87	47	26	52	47	32	63	12	93	15	47	65	àGDRcG&RG2c+ô\$Ge
ds:00F0	81	25	78	32	58	42	87	95	57	42	15	42	21	65	54	98	üxZXBçòWB\$B!eTj
ds:0100	61	25	74	32	25	30	46	58	31	34	31	31	32	30	30	32	a\ztZ\0FX14112002
ds:0110	39	12	56	89	51	87	63	85	31	32	31	31	32	30	30	32	9+UeQscà12112002
ds:0120	81	27	56	87	65	89	12	54	32	37	31	30	32	30	30	33	ü'Uçèè+T27102003
ds:0130	01	32	58	48	26	95	25	63	31	34	31	31	32	30	30	32	@ZXH&òxc14112002
ds:0140	78	04	05	32	53	98	03	21	31	32	31	31	32	30	30	32	x+â2Sjü!12112002
ds:0150	56	25	51	48	59	32	63	25	32	37	31	30	32	30	30	33	U\QHY2c\27102003
ds:0160	42	15	45	78	56	02	82	24	31	34	31	31	32	30	30	32	B\$ExU8e\$14112002
ds:0170	98	36	74	98	21	57	24	56	31	32	31	31	32	30	30	32	ij6tj!W\$U12112002
ds:0180	65	25	52	41	85	25	58	36	32	37	31	30	32	30	30	33	e\RAà\X627102003
ds:0190	56	98	44	26	65	32	58	21	31	34	31	31	32	30	30	32	Ujd&e2X!14112002
ds:01A0	45	78	88	15	56	32	56	78	31	32	31	31	32	30	30	32	Exê\$U2Ux12112002
ds:01B0	35	62	45	78	62	85	05	12	32	37	31	30	32	30	30	33	5bExbâ+27102003
ds:01C0	86	25	89	65	23	55	57	41	31	34	31	31	32	30	30	32	â\èe#UWA14112002
ds:01D0	51	14	26	48	43	12	48	11	31	32	31	31	32	30	30	32	Q9lHC#H412112002
ds:01E0	52	98	74	53	45	89	42	15	32	37	31	30	32	30	30	33	RjtSEèB\$27102003
ds:01F0	87	45	42	56	19	73	93	54	62	48	55	22	78	32	30	30	çEBU!sôTbHU"x200

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Github репозиторій: <https://github.com/VadymT95/computer-architecture>

GitHub Анатолія: <https://github.com/Stepanenko-Anatolii>

GitHub Вадима: <https://github.com/VadymT95>

GitHub Олександра: <https://github.com/ParZZ1vaLI>

Висновки:

За допомогою цієї лабораторної роботи ми вивчили додаткові команди Асемблеру, що дають можливість керувати ходом програми. Також дізналися про додаткові переривання DOS і BIOS для відображення інформації на консолі.