Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота № 5

Архітектура комп’ютера – 2. Основи алгоритмізації

“УПРАВЛІННЯ ХОДОМ ВИКОНАННЯ ПРОГРАМИ НА АСЕМБЛЕРІ В АРХІТЕКТУРІ IA-32(X86) У REAL ADRESS MODE”

Виконали
студенти групи ІТ-01:

Бражник О.О.
Девіцький І.Д.
Кривоносюк В.В.

Перевірив:

Дата здачі: 22.04.2021

Захищено з
балом:_________

Бердник Ю.М.

Посилання на github репозиторії учасників:
https://github.com/Critalic/CA_Labs/tree/master/L4
https://github.com/forasgarddd/asm_lab4

Вихідний код:
В ході виконання даної лабораторної роботи було використано процедурний стиль. Було створено 6 процедур: 4 процедури вставки для дат 3 студентів групи та дати за замовчуванням, процедуру для копіювання масиву до сегменту даних а також процедуру сортування даних бульбашкою в порядку зростання.

```
TITLE ЛР_5

;-------------------------------------------------------------------------------
; Комп'ютерна архітектура
; ВУЗ:          НТУУ "КПІ"
; Факультет:    ФІОТ
; Курс:          1
; Група:       IT-01
;-------------------------------------------------------------------------------

IDEAL
MODEL small
STACK 2048
;-------------------------------------------------------------------------------

DATASEG
matrixArray db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
            db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9

LEN DW 256

birthdate1 db "12122002"
birthdate2 db "0512002"
birthdate3 db "22072003"
birthdate0 db "11111111"
;-------------------------------------------------------------------------------
```

```
41  CODESEG
42  Start:
43      mov ax, @data ; data segment init
44      mov ds, ax
45      mov es, ax
46
47      mov cx, [LEN]    ;Cx is counter for OUTERLOOP CX=5
48      dec cx           ; CX = 4
49      call sort
50
51      mov cx, 256                 ; repeats amount
52      call copyArray
53
54      mov cx, 8
55      mov bp, 0162h
56      call set_bdatesd
57
58      mov cx, 8
59      mov bp, 0172h
60      call set_bdatesd
61
62      mov cx, 8
63      mov bp, 0182h
64      call set_bdates1
65
66      mov cx, 8
67      mov bp, 0192h
68      call set_bdates2
69
70      mov cx, 8
71      mov bp, 01A2h
72      call set_bdates3
73
74
75
76      mov cx, 8
77      mov bp, 01B2h
78      call set_bdatesd
79
80      mov cx, 8
81      mov bp, 01C2h
82      call set_bdatesd
```

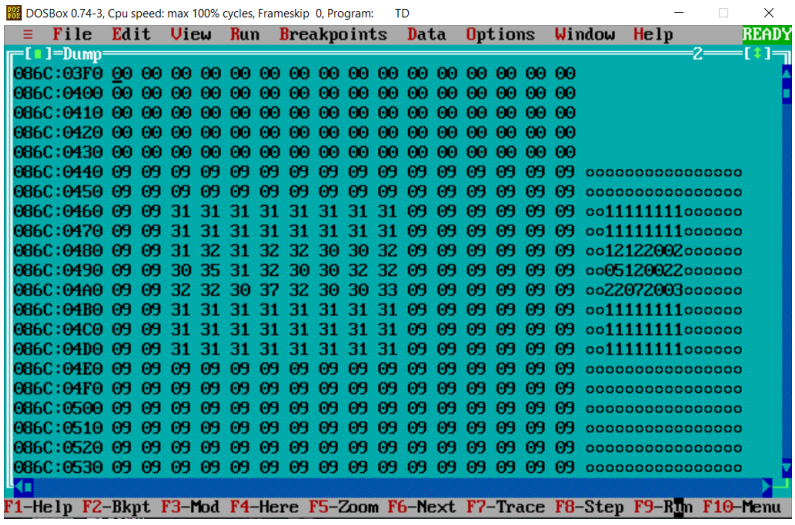
```
83
84      mov cx, 8
85      mov bp, 01D2h
86      call set_bdatesd
87
88
89  ; the end :)
90      mov ah, 4ch
91      int 21h
92
93  ;-----------------------------------------------Copy array ----------------------------------------
94  ; Input: cx - initial size of the array ,
95  ;-------------------------------------------------------------------------------------------------------------
96      PROC copyArray
97          xor si, si                      ; si to 0
98          array_coping_loop:
99              mov bx, [ds:si]             ; get number from matrixArray & set it to bx as a temp variable
100             mov [ds:[si+270h]], bx      ; copy number from bx to ds with offset
101             add si, 2                   ; si+= 2
102             loop array_coping_loop
103
104         ret
105     ENDP
106
107 ;-----------------------------------------Add birthdate to stack-----------------------------------------
108 ; Input: cx - birthday date,
109 ;              bp - offset
110 ;-------------------------------------------------------------------------------------------------------------
111     PROC set_bdates1
112         xor si,si                               ; set si to 0
113         birthdate1_label:
114             mov ah, [birthdate1+si]             ; put value of birthdate1 to ah with offset si
115             mov [bp], ah                        ; add value from ah to stack
116             inc si                              ; si++
117             inc bp                              ; bp++
118             loop birthdate1_label
119
120         ret
121     ENDP
122
```

```
123 ;-----------------------------------------Add birthdate to stack-----------------------------------------
124 ; Input: cx - birthday date,
125 ;              bp - offset
126 ;-------------------------------------------------------------------------------------------------------------
127     PROC set_bdates2
128         xor si,si                               ; set si to 0
129         birthdate2_label:
130             mov ah, [birthdate2+si]             ; put value of birthdate2 to ah with offset si
131             mov [bp], ah                        ; add value from ah to stack
132             inc si                              ; si++
133             inc bp                              ; bp++
134             loop birthdate2_label
135
136         ret
137     ENDP
138
139
140 ;-----------------------------------------Add birthdate to stack-----------------------------------------
141 ; Input: cx - birthday date,
142 ;              bp - offset
143 ;-------------------------------------------------------------------------------------------------------------
144     PROC set_bdates3
145         xor si,si                               ; set si to 0
146         birthdate3_label:
147             mov ah, [birthdate3+si]             ; put value of birthdate3 to ah with offset si
148             mov [bp], ah                        ; add value from ah to stack
149             inc si                              ; si++
150             inc bp                              ; bp++
151             loop birthdate3_label
152
153         ret
154     ENDP
155
```

```
156 ;-----------------------------------------Add birthdate to stack-----------------------------------------
157 ; Input: cx - birthday date,
158 ;              bp - offset
159 ;-------------------------------------------------------------------------------------------------------
160     PROC set_bdatesd
161         xor si,si                               ; set si to 0
162         birthdated_label:
163             mov ah, [birthdate0+si]             ; put value of birthdate0 to ah with offset si
164             mov [bp], ah                        ; add value from ah to stack
165             inc si                              ; si++
166             inc bp                              ; bp++
167             loop birthdated_label
168 
169         ret
170     ENDP
171 
172     PROC sort
173         nextscan:
174             mov bx,cx
175             mov si,0
176 
177         nextcomp:
178 
179             mov al,[matrixArray+si]
180             mov dl,[matrixArray+si+1]
181             cmp al,dl
182 
183             jnc noswap
184 
185             mov [matrixArray+si], dl
186             mov [matrixArray+si+1], al
187 
188         noswap:
189             inc si
190             dec bx
191             jnz nextcomp
192 
193             loop nextscan
194 
195         ret
196     ENDP
197 
198 end Start
```

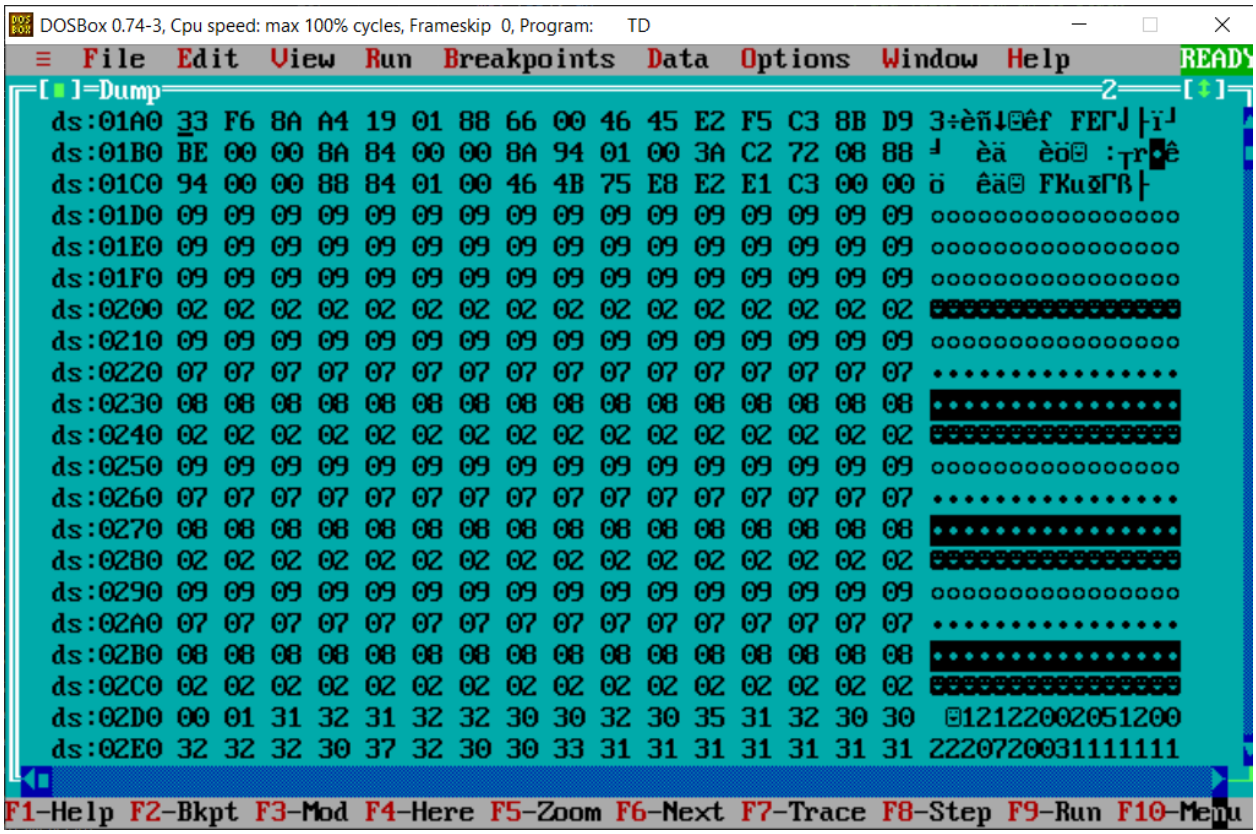
Скріншот дампу:

Вставка дат народжень:

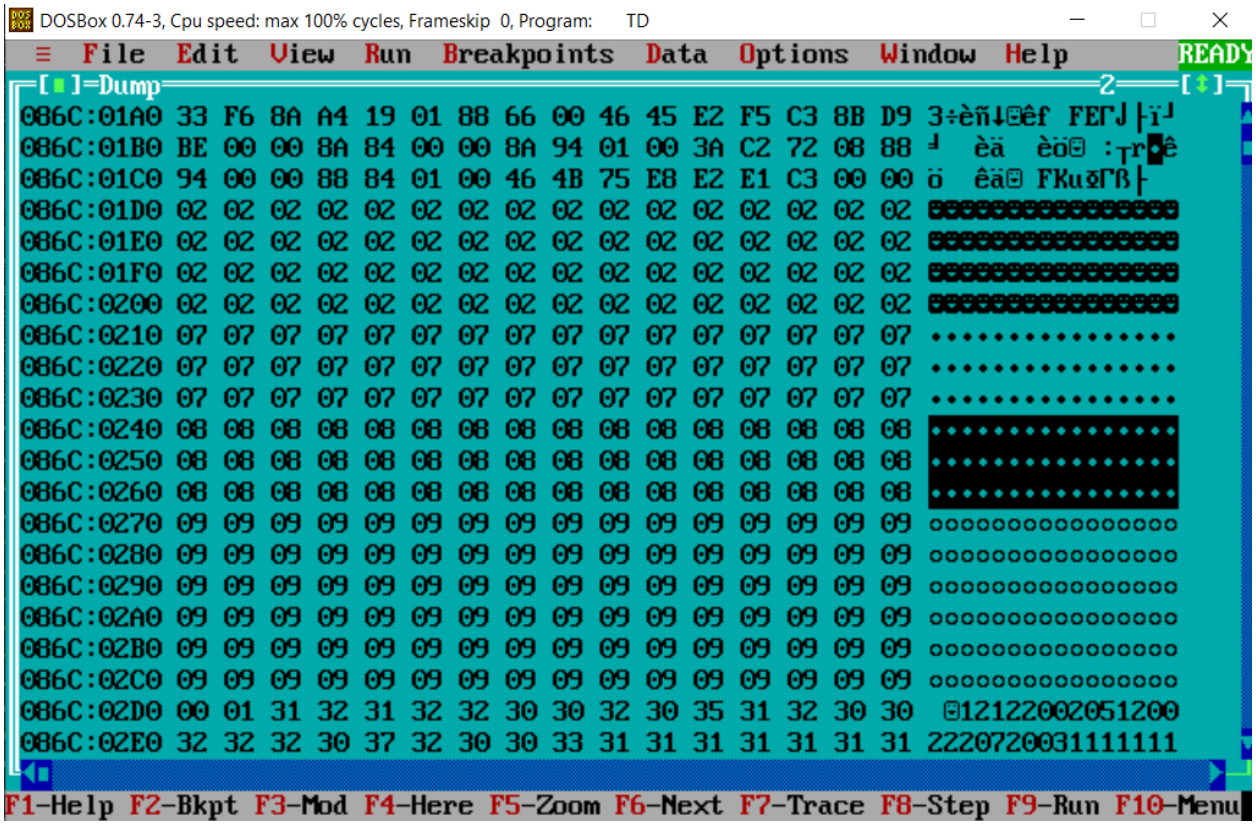


Сортування масиву за зростанням:

До


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
READY
[■]=Dump
ds:01A0 33 F6 8A A4 19 01 88 66 00 46 45 E2 F5 C3 8B D9
ds:01B0 BE 00 00 8A 84 00 00 8A 94 01 00 3A C2 72 08 88
ds:01C0 94 00 00 88 84 01 00 46 4B 75 E8 E2 E1 C3 00 00
ds:01D0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:01E0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:01F0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:0200 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
ds:0210 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:0220 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
ds:0230 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
ds:0240 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
ds:0250 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:0260 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
ds:0270 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
ds:0280 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
ds:0290 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
ds:02A0 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
ds:02B0 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
ds:02C0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
ds:02D0 00 01 31 32 31 32 32 30 30 32 30 35 31 32 30 30
ds:02E0 32 32 32 30 37 32 30 30 33 31 31 31 31 31 31 31
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Після


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
READY
[■]=Dump
086C:01A0 33 F6 8A A4 19 01 88 66 00 46 45 E2 F5 C3 8B D9
086C:01B0 BE 00 00 8A 84 00 00 8A 94 01 00 3A C2 72 08 88
086C:01C0 94 00 00 88 84 01 00 46 4B 75 E8 E2 E1 C3 00 00
086C:01D0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
086C:01E0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
086C:01F0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
086C:0200 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
086C:0210 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
086C:0220 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
086C:0230 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07
086C:0240 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
086C:0250 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
086C:0260 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08
086C:0270 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:0280 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:0290 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:02A0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:02B0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:02C0 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
086C:02D0 00 01 31 32 31 32 32 30 30 32 30 35 31 32 30 30
086C:02E0 32 32 32 30 37 32 30 30 33 31 31 31 31 31 31 31
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Результати дослідження: у ході виконання лабораторної роботи ми заповнили середину масиву 16x16 ділянкою з датами народження студентів нашої бригади відповідно до варіанту нашої групи.

Висновок: У ході лабораторної роботи попрацювали з різними видами адресації, створили двовимірний масив і розібралися як працює найшвидший спосіб роботи з даними: стек.