



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 ІА-32(X86) У REAL ADDRESS MODE”*

Виконали
студенти групи ІТ-01:

Перевірів:

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

Дата здачі: 22.04.2021

Захищено з
балом: _____

Бердник Ю.М.

Посилання на github репозиторії учасників:

https://github.com/Criticalic/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.

```
1  TITLE LP_5
2
3  ;-----
4  ; Комп'ютерна архітектура
5  ; ВУЗ:      НТУУ "КПІ"
6  ; Факультет:  ФІОТ
7  ; Курс:      1
8  ; Група:      IT-01
9  ;-----
10
11  IDEAL
12  MODEL small
13  STACK 2048
14  ;-----
15
16  DATASEG
17  matrixArray db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
18               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
19               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
20               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
21               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
22               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
23               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
24               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
25               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
26               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
27               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
28               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
29               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
30               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
31               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
32               db 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
33
34  LEN DW 256
35
36  birthdate1 db "12122002"
37  birthdate2 db "0512002"
38  birthdate3 db "22072003"
39  birthdate0 db "11111111"
40  ;-----
```

```

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 ;-----Add birthdate to stack-----
140 ; Input: cx - birthday date,
141 ;         bp - offset
142 ;-----
143     PROC set_bdates3
144         xor si, si                ; set si to 0
145         birthdate3_label:
146             mov ah, [birthdate3+si] ; put value of birthdate3 to ah with offset si
147             mov [bp], ah           ; add value from ah to stack
148             inc si                 ; si++
149             inc bp                 ; bp++
150             loop birthdate3_label
151
152         ret
153     ENDP
154
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:

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[]=Dump 2 []

086C:03F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
086C:0400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
086C:0410	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
086C:0420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
086C:0430	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
086C:0440	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0450	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0460	09 09 31 31 31 31 31 31 31 31 09 09 09 09 09 09	oo11111111oooooooo
086C:0470	09 09 31 31 31 31 31 31 31 31 09 09 09 09 09 09	oo11111111oooooooo
086C:0480	09 09 31 32 31 32 32 30 30 32 09 09 09 09 09 09	oo12122002oooooooo
086C:0490	09 09 30 35 31 32 30 30 32 32 09 09 09 09 09 09	oo05120022oooooooo
086C:04A0	09 09 32 32 30 37 32 30 30 33 09 09 09 09 09 09	oo22072003oooooooo
086C:04B0	09 09 31 31 31 31 31 31 31 31 09 09 09 09 09 09	oo11111111oooooooo
086C:04C0	09 09 31 31 31 31 31 31 31 31 09 09 09 09 09 09	oo11111111oooooooo
086C:04D0	09 09 31 31 31 31 31 31 31 31 09 09 09 09 09 09	oo11111111oooooooo
086C:04E0	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:04F0	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0500	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0510	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0520	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo
086C:0530	09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09	oooooooooooooooooooo

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 16×16 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.