



Συστήματα Μικροπολογιστών

5^η Ομάδα Ασκήσεων

Μετζάκης Ιωάννης

A.M.: 03116202

ΣΗΜΜΥ 8^ο

1η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας σε **assembly** με τα απαραίτητα σχόλια:

```
001 print macro char
002     push dx
003     push ax
004
005     mov dl, char
006     mov ah, 2
007     int 21h
008
009     pop ax
010     pop dx
011 endm
012
013 new_line macro
014     print 0ah
015     print 0dh
016 endm
017
018 data segment
019 ends
020
021 stack segment
022 ends
023
024 code segment
025 start:
026     mov ax, data ; segment registers
027     mov ds, ax
028     mov es, ax
029
030     call hex_key ;diavazoume to 1o pshfio
031     cmp al, 'Q' ;an einai to 'Q', stamata
032     je stop
033     mov bl, al
034
035     call hex_key ;diavazoume to 2o pshfio
036     cmp al, 'Q' ;omoia me prin
037     je stop
038
039     print " " ;twra pia o bl exei to 1o
040     print "=" ;kai o al to 2o
041     print " "
042     print "0"
043     print "0"
044     mov dl,bl ;printaroume to 1o(bl)
045     call print_digit
046
047     mov cl,4
048     shl bl,cl ;afou valoume ta 4LSB sta 4MSB
049
050     add bl, al ;prosthetoyme ton al sto bl
051     mov dl, al ;printaroume to 2o(al)
052     call print_digit
053
054     print "h"
055     print " "
056     print "="
057     print " "
058     call print_dec ;kaloume tis antistoixes routines
059     print "d" ;gia metatroph se dekadikh, oktadikh
060     print " " ;kai diadikh morfhi
061     print "="
062     print " "
063     call print_oct
064     print "o"
065     print " "
066     print "="
067     print " "
068     call print_bin
069     print "b"
070     new_line
071
072     jmp start
073
074
075
```

```

076
077 print_digit proc near
078     cmp dl, 9      ;o dl exei thn timh poy tha printaroume
079     jg addr1
080     add dl, 30h    ;an einai pshfio(0..9) tote prosthetoume 30h
081     jmp addr2
082 addr1:
083     add dl, 37h    ;an einai gramma(A..F) tote prosthetoume 37h
084 addr2:
085     print dl
086     ret
087 print_digit endp
088
089
090 PRINT_DEC proc near
091     push dx
092     push cx
093     push ax
094
095
096     mov ah, 00h
097     mov al, bl
098     mov cl, 100    ; kratame tis 100des
099     div cl         ; diairesh me 100
100     mov dl, al
101     call print_digit ;printaroyme tis 100des
102
103     mov cl, 10    ; kratame tis 10des
104     mov al, ah
105     mov ah, 0
106     div cl
107     mov dl, al
108     call print_digit ;printaroyme tis 10des
109
110     mov dl, ah
111     call print_digit ;printaroyme tis monades
112
113     pop ax
114     pop cx
115     pop dx
116     ret
117
118 PRINT_DEC endp
119
120 PRINT_OCT proc near
121     push dx
122     push cx
123
124     mov cl, 6
125     mov dl, bl    ; print 1o pshfio
126     sar dl, cl
127     and dl, 03h
128     call print_digit
129
130
131     mov cl, 3
132     mov dl, bl
133     sar dl, cl
134     and dl, 07h
135     call print_digit ;print 2o pshfio
136
137     mov dl, bl
138     and dl, 07h
139     call print_digit ;print 3o pshfio
140
141     pop cx
142     pop dx
143     ret
144 PRINT_OCT endp
145
146 PRINT_BIN proc near
147     push bx
148     push cx
149
150     mov cx, 8      ;8 fores tha kanei rol

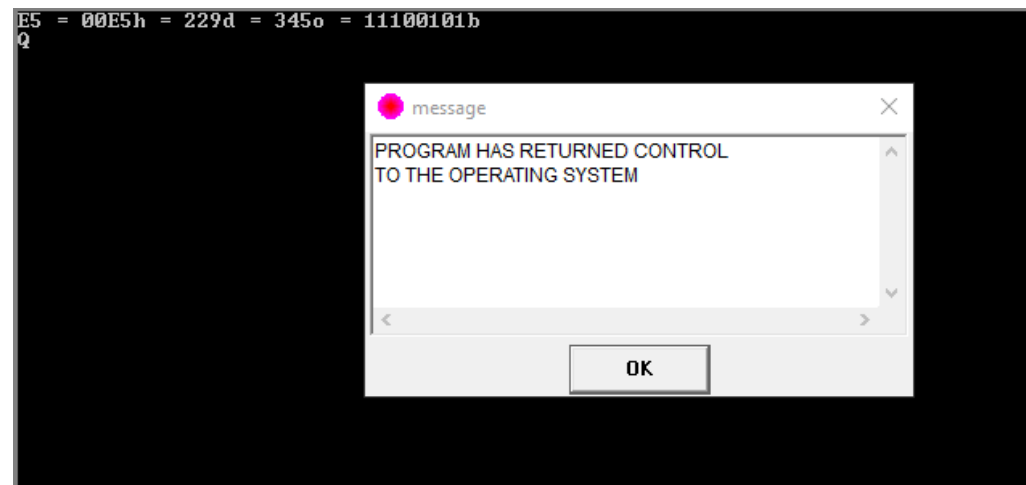
```

```

151 again:
152     rol bl, 1
153     jc print1
154     mov dl, 00h      ; print 0 an meta to rol
155     call print_digit ;den iparxei kratoumeno
156     LOOP again
157     pop cx
158     pop bx
159     ret
160 print1:
161     mov dl, 01h
162     call print_digit ;alliws print 1
163     LOOP again
164     pop cx
165     pop bx
166     ret
167 PRINT_BIN endp
168
169 hex_key proc near
170 not_good:
171     mov ah, 1
172     int 21h
173
174     cmp al, 'Q'      ;elegxos gia termatismo
175     je exit
176
177     cmp al, 30h
178     jl not_good
179
180     cmp al, 39h      ;paei sthn letter an einai
181     jg letter        ;megalitero apo 9 gia 2o elegxo
182
183     sub al, 30h      ;alliws einai arithmos -> -30h
184     jmp exit
185
186 letter:
187     cmp al, 'A'      ;an profanws <A h >F de mou kanei
188     jl not_good
189     cmp al, 'F'
190     jg not_good
191
192     sub al, 37h      ;einai gramma, ara afairw 37h
193 exit:
194     ret
195 hex_key endp
196
197
198 stop:
199     mov ax, 4c00h
200     int 21h
201 ends
202 end start

```

Παρακάτω βρίσκεται ένα στιγμιότυπο από το output της οθόνης για είσοδο E5, και μετέπειτα εισαγωγή του χαρακτήρα 'Q' για τερματισμό:



2η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας σε **assembly** με τα απαραίτητα σχόλια:

```
0001 print macro char
0002     mov dl, char
0003     mov ah, 2
0004     int 21h
0005 endm
0006
0007 ; plithos arithmwn sth mnhmh
0008 N EQU 255
0009
0010 data segment
0011     TABLE dw N dup(?)
0012 ends
0013
0014 stack segment
0015 ends
0016
0017 code segment
0018
0019
0020
0021 start:
0022     mov ax, data
0023     mov ds, ax
0024     mov es, ax
0025
0026
0027     ; fortwsh mnhmhs me sinexomenous arithmous
0028     mov cl, N
0029     cld                                ; df = 0
0030     mov di, OFFSET TABLE
0031     mov al, 254
0032 write_again:
0033     stosb
0034     dec al                                ; fortwsh epomenou dedomenou
0035     cmp al, 254
0036     jnz write_again
0037
0038
0039     mov cl, N
0040     cld                                ; df = 0
0041     mov di, OFFSET TABLE
0042     mov dx, 0                            ; arxikopoihsh kataxwrthwn athroismatos
0043     mov bx, 0
0044     mov ah, 0
0045 load_again:
0046     lodsb                                ; fortwsh se kataxwrthh
0047     add bx, ax
0048     cmp cl, 00h
0049     jz moove
0050     loop load_again
0051 moove:
0052
0053     mov ax, bx                            ; metakinshsh diairetaiou
0054     mov bx, N+1                            ; diaireths
0055     div bx                                ; ektelesh diaireshs
0056
0057     mov dl, ah
0058     call print_hex_full
0059     mov dl, al
0060     call print_hex_full
0061
0062
0063     mov cl, N
0064     cld                                ; df = 0
0065     mov di, OFFSET TABLE
0066     mov dl, 255                            ; dx = topiko min
0067     mov bl, 0                                ; bx = topiko max
0068 load_again_2:
0069     lodsb                                ; fortwsh se kataxwrthh
0070 local_min_calc:
0071     cmp al, dl
0072     ja local_max_calc
0073     mov dl, al
0074 local_max_calc:
0075     cmp dl, bl
```

```

076         jb next
077         mov dl, al
078 next:
079         cmp cl, 00h
080         jz moove_2
081         loop load_again_2
082 moove_2:
083
084         ; pairnei ena 2adiko pshfio kai to tipwnei 16dika
085
086         print " "
087         print "m"
088         print ":"
089         push dx
090         mov dl, dh
091         call print_hex_full
092         pop dx
093         call print_hex_full
094
095         print " "
096         print "M"
097         print ":"
098         mov dl, bh                ; ektipwsh 8MSB
099         call print_hex_full
100         mov dl, bl                ; ektipwsh 8LSB
101         call print_hex_full
102
103
104         mov ax, 4c00h ; exit
105         int 21h
106
107
108 ends
109
110         print_hex proc near
111             cmp dl, 9
112             jg addr1
113             add dl, 30h
114             jmp addr2
115 addr1:
116             add dl, 37h
117 addr2:
118             print dl
119             ret
120         print_hex endp
121
122
123
124
125         print_hex_full proc near
126             push dx
127             push ax
128             push bx
129             push dx                ; save to apotelesma
130
131             sar dx, 1
132             sar dx, 1
133             sar dx, 1
134             sar dx, 1
135             and dl, 0fh
136             call print_hex
137
138             pop dx
139             and dl, 0fh
140             call print_hex
141             pop bx
142             pop ax
143             pop dx
144             ret
145         print_hex_full endp
146
147 end start

```

3η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας σε **assembly** με τα απαραίτητα σχόλια:

```
001 new_line macro
002     print 0ah
003     print 0dh
004 endm
005
006 print_str macro string
007     mov dx, offset string
008     mov ah, 9
009     int 21h
010 endm
011
012 print macro char
013     mov dl, char
014     mov ah, 2
015     int 21h
016 endm
017
018
019 data segment
020     X_print db "X=", '$'
021     Y_print db "Y=", '$'
022     ADD_print db "X+Y=", '$'
023     SUB_print db "X-Y=", '$'
024 ends
025
026 stack segment
027     dw 128 dup(0)
028 ends
029
030 code segment
031 start:
032     mov ax, data
033     mov ds, ax
034     mov es, ax
035
036 main proc far
037
038     mov cl, 00h;
039 loop:
040     call read_num
041     mov ah, 00h
042     push ax
043     inc cl      ; cl++
044     cmp cl, 04h ;otan ginei 4, exw 4 egkira pathmena
045     jz print_results
046     jmp loop
047
048
049
050 print_results:
051     new_line
052     print_str Y_print
053     pop bx
054     pop cx
055     mov dl, cl
056     call print_bin
057     mov dl, bl
058     call print_bin
059     ;EDW EXW STON BL TO LOW TOU Y
060     ;KAI STON CL TO HIGH TOU Y
061
062     mov al, cl
063     sal al, 1
064     sal al, 1
065     sal al, 1
066     sal al, 1
067     add al, bl ; al = Y
068
069     push ax
070     print " "
071     pop ax
072
073     pop bx
074     pop cx
075     push ax
```

```

076
077
078     print_str X_print
079     mov dl, cl
080     call print_bin
081     mov dl, bl
082     call print_bin
083                                     ;EDX EXW STON BL TO LOW TOW X
084                                     ;KAI STON CL TO HIGH TOW X
085
086     mov al, cl
087     sal al, 1
088     sal al, 1
089     sal al, 1
090     sal al, 1
091     add al, bl           ; al = X
092
093
094
095     push ax
096     new_line
097     print_str ADD_print
098     pop ax              ;X
099     mov bx, ax
100     pop ax
101     xchg ax, bx
102
103
104     mov ah, 00h
105     mov bh, 00h
106     push ax
107     add ax, bx
108     mov dx, ax          ; dl = al + bl
109
110     push bx
111     call print_decimal
112     print " "
113     print_str SUB_print
114
115     pop bx
116     pop ax
117
118
119     cmp al, bl
120     JB arnhtikos
121     mov dl, al
122     sub dl, bl
123     jmp print_abs
124
125 arnhtikos:
126     mov dl, bl
127     sub dl, al
128     push dx
129     print "-"
130     pop dx
131 print_abs:
132     call print_decimal
133     new_line
134
135     jmp start
136
137 main endp
138
139
140 read_num proc near
141
142     ; Reads key ascii code in al
143     mov ah, 01h
144     int 21h
145
146     cmp al, 30h         ;elegxos an o arithmos einai <0..9>
147     jl read_num
148
149     cmp al, 39h
150     jle ok_dec

```



```

151     cmp al, 39h
152     jle ok_dec
153
154     cmp al, 41h    ;elegxos an to gramma einai <A..F>
155     jl read_num
156
157     cmp al, 46h
158     jg read_num
159
160     sub al, 37h    ;afairw 37h an gramma
161     jmp end_num
162
163
164 ok_dec:
165     sub al, 30h    ;afairw 30h an pshfio
166
167 end_num:
168     ret
169 read_num endp
170
171
172 print_ascii proc near
173
174     mov ax, 4c00h
175     int 21h
176 ends
177
178 print_hex proc near
179     cmp dl, 9
180     jg addr1
181     add dl, 30h
182     jmp addr2
183 addr1:
184     add dl, 37h
185 addr2:
186     print dl
187     ret
188 print_hex endp
189
190 print_bin proc near
191     cmp dl, 9
192     jg b_addr1
193     add dl, 30h
194     jmp b_addr2
195 b_addr1:
196     add dl, 37h
197 b_addr2:
198     print dl
199     ret
200 print_bin endp
201
202
203 print_dec proc near
204     add dl, 30h
205     print dl
206     ret
207 print_dec endp
208
209
210 print_decimal proc near
211     push ax
212     push bx
213
214     push dx
215
216     mov al, 00H
217     mov ah, 00H
218     jmp ekato
219
220 ekato_plus:
221     inc ah
222     sub dx, 0064H
223
224 ekato:

```

```

225         jmp dekades
226
227 dekades_plus:
228     inc al
229     sub dx,000AH
230
231 dekades:
232     cmp dx,000AH
233     JGE dekades_plus
234     push ax
235     push dx
236
237     mov dl,ah
238     cmp dl,00h
239     jz pass0
240     call print_dec
241
242     pop dx                ;edw bainoume an einai >=100
243     pop ax                ;ara theloume na tipwsoume tis
244     push dx               ;dekades akoma kai 0 na einai
245     mov dl,al
246     call print_dec
247     jmp pass1
248
249 pass0:
250     pop dx                ;edw bainoume an <100, ara elegxoume
251     pop ax                ;an einai kai <10 wste na tipwsoume
252     push dx               ;mono tis monades
253     mov dl,al
254     cmp dl,00h
255     jz pass1
256     call print_dec
257 pass1:
258     pop dx
259     and dl, 0fh
260     call print_dec
261     pop dx
262     pop bx
263     pop ax
264     ret
265
266 print_decimal    endp
267
268 end start

```

Παρακάτω βρίσκεται ένα στιγμιότυπο από το output της οθόνης για διαδοχικές εισόδους 2F3A (παράδειγμα εκφώνησης), FF05 και R0FTe1J1 (για να φανεί ο έλεγχος των έγκυρων HEX ψηφίων, δηλαδή των 0F11):

```

2F3A
Y=3A X=2F
X+Y=105 X-Y=-11
FF05
Y=05 X=FF
X+Y=260 X-Y=250
R0FTe1J1
Y=11 X=0F
X+Y=32 X-Y=-2

```

4η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας σε **assembly** με τα απαραίτητα σχόλια:

```
001 new_line macro
002     print 0ah
003     print 0dh
004 endm     turn
005
006 print_str macro string
007     mov dx, offset string
008     mov ah, 9
009     int 21h
010 endm
011
012 print macro char
013     mov dl, char
014     mov ah, 2
015     int 21h
016 endm
017
018
019 data segment
020     input db 21 dup(?)
021 ends
022
023 stack segment
024
025 ends
026
027 code segment
028 start:
029     mov ax, data
030     mov ds, ax
031     mov es, ax
032
033
034     mov di, offset input
035     cld                     ;df = 0
036     mov cx, 16             ;Metrhths
037     mov bl, 00h
038     push bx
039
040 read_loop:
041     pop bx
042     push bx
043     call read_key          ;diavazw
044     pop bx
045     cmp al, 80h            ;plhktro termatismou ENTER
046     je stop               ;opou exw orisei na vazei to 80h
047     stosb                 ;save sth mnhmh
048     cmp al, 39h
049     jg print0             ;an einai <39h exoume arithmo -> bl++
050     inc bl
051 print0:
052     push bx
053     print al              ;printaroume oti diavasoume
054
055     loop read_loop
056
057 process:                  ;edw exoume apothikevmena oti diavasame,
058     new_line              ;kai ston bl, to plthos tw n arithmw n
059     cld
060     mov si, offset input
061     mov cx, 16
062     pop bx
063     mov dl, bl
064     push dx
065     push bx
066 loop_1:
067     lodsb                 ;diavazoume apo mnhmh
068     call big_to_small    ;edw kanoume ta kefalaia peza
069     cmp al, 39h
070     jle print_digits     ;kai an einai grammata, ta ksanaapothikevoume
071     stosb                 ;sth mnhmh giati prwta printaroume ta pshfia
072     loop loop_1
073
074 print_digits:
075     print al
```

```

076     pop bx
077     dec bl      ;gia kathe arithmo pou printaroume, meiwoune ton bl
078     push bx     ;pou exei to plthos twn arithmw, opote otan bl=0
079     cmp bl,00h  ;teleiwsame me tous arithmous kai printaroume "-"
080     je add_pavla
081     loop loop_1
082
083 add_pavla:
084     print "-"
085     pop bx
086     pop dx
087     push dx
088     cmp cl,0
089     je therest   ;edw vlepoume posa grammata exoun meinei pou den
090     dec cx       ;exoume epeksergastei akoma (an dld o teleftaios
091 therest:        ;arithmos htan o 14os xarakthras, shmainei oti
092     cmp cl,0     ;iparxoun 2 akoma grammata pou prepei na kanoume
093     je letters_start ;pezous char, kai na prosthesoyme sth stiva)
094     lodsb
095     call big_to_small
096     stosb
097     loop therest
098
099 letters_start:
100     pop dx
101     mov bx,dx
102     mov dx,16
103     sub dx,bx
104     mov cx,dx    ;edw exoume ston cx pia, to plthos twn grammatwn
105
106 print_letters:
107     cmp cl,00h   ;kai kanw thn antistoixh diadikasia gia
108     je restart   ;grammata exw
109     lodsb
110     print al
111
112     loop print_letters
113 restart:
114     new_line
115     jmp start
116
117 stop:
118     mov ax, 4c00h
119     int 21h
120 ends
121
122
123 big_to_small proc near
124     cmp al, 'A'
125     jl cap_end    ;an den einai (A..Z) tote den kanw tipota
126     cmp al, 'Z'
127     jg cap_end
128     add al, 32    ;alliws prosthetw 32h, gia na exw (a..z)
129 cap_end:
130     ret
131 big_to_small endp
132
133 read_key proc near
134 ignore:
135     mov ah, 8
136     int 21h
137
138     cmp al, 0dh   ;elegxos an paththei to ENTER (vazw al=0dh)
139     jne contin    ;gia na mhn mperdefw me to 'D' sth sinexeia
140     mov al,80h
141     jmp exit
142
143 contin:
144     cmp al, 30h   ;oi klasikoi elegxoi
145     jl ignore
146
147     cmp al, 39h
148     jg its_letter
149
150     jmp exit
151
152 its_letter:
153     cmp al, 'A'
154     jl ignore
155     cmp al, 'Z'
156     jg ignore
157 exit:
158     ret
159 read_key endp
160
161
162 end start

```

Και ένα στιγμιότυπο από το output της οθόνης για διαδοχικές εισόδους A8X9S1FETD73A8KL (παράδειγμα εκφώνησης) και 5GIU9004WN7FG3D6:

```

A8X9S1FETD73A8KL
891738-axsfetdakl
5GIU9004WN7FG3D6
5904736-giuownfgd

```