

# Συστήματα Μικρουπολογιστών

5η Ομάδα Ασκήσεων

Μετζάκης Ιωάννης

A.M.: 03116202

ΣΗΜΜΥ 8°

```
001 print macro char
002 push dx
003 push ax
004
             mov dl, char
mov ah, 2
int 21h
006
008
             pop ax
pop dx
010
011 endm
012
013 new_line macro
014 print Oah
015 print Odh
     endm
017
018 data segment
019 ends
020
021
      stack segment
022
      ends
023
024 code segment
025 start:
             mov ax, data ; segment registers
mov ds, ax
mov es, ax
026
027
028
029
             call hex_key
                                       ;diavazoume to 10 pshfio;an einai to 'Q', stamata
030
             cmp al, 'Q
je stop
mov bl, al
031
032
033
034
             call hex_key cmp al, 'Q'
035
                                       ;diavazoume to 20 pshfio
036
037
              cmp al.
                                        ;omoia me prin
              je stop
038
             print ""
print ""
print ""
print "0"
print "0"
                                       ;twra pia o bl exei to 1o;kai o al to 2o
039
040
041 \\ 042
              mov dl,bl ;printaroume to 1o(bl) call print_digit
044
045
046
              mov cl.4 shl bl.cl
047
                                       ;afou valoume ta 4LSB sta 4MSB
049
050
             051
052
053
054
             print "h"
print " "
print " "
print " "
055
057
             call print_dec ;kaloume tis antistoixes routines print "d" ;gia metatroph se dekadikh, oktadikh print "" ;kai diadikh morfh print "" print " " call print oct
058
059
060
061
062
             call print_oct
print "o"
print " "
print " "
print " "
Ø63
064
065
066
067
             call print_bin print bin
068
070
071
072
              new_line
              jmp start
```

```
076
077
078
      print_digit proc near

cmp dl, 9 ; o dl exei thn timh poy tha printaroume
jg addr1
add dl, 30h ; an einai pshfio(0..9) tote prosthetoum
jmp addr2
addr1:
080
081
                                      ;an einai pshfio(0..9) tote prosthetoume 30h
082
      add dl, 37h
addr2:
print dl
083
                                      ;an einai gramma(A..F) tote prosthetoume 37h
084
085
087
088
      print_digit endp
990 PRINT_DEC proc near

991 push dx

992 push cx

993 push ax
093
094
095
096
                mov ah, 00h
mov al, bl
mov cl, 100 ; kratame tis 100des
div cl ; diairesh me 100
mov dl, al
call print_digit ;printaroyme tis 100des
098
099
100
101
102
                mov cl, 10
mov al, ah
mov ah, 0
                                            ; kratame tis 10des
103
104
105
                div cl
mov dl, al
call print_digit ;printaroyme tis 10des
106
107
109
                mov dl, ah
call print_digit ;printaroyme tis monades
110
111
112
113
114
115
                 pop ax
                pop cx
pop dx
ret
117
118 PRINT_DEC endp
PRINT_OCT proc near
push dx
push cx
                mov cl, 6
mov dl, bl ;
sar dl, cl
and dl, 03h
call print_digit
124
125
                                          ; print 1o pshfio
126
127
128
129
130
131
132
133
134
                mov cl, 3
mov dl, bl
sar dl, cl
and dl, 07h
call print_digit ;print 2o pshfio
135
136
137
                mov dl, bl
and dl, 07h
call print_digit ;print 3o pshfio
138
139
140
141 pop cx
142 pop dx
143 ret
144 PRINT_OCT endp
146
147
148
      PRINT_BIN proc near
              push bx
push cx
149
150
              mov cx, 8
                                       ;8 fores tha kanei rol
```

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

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



```
901 print macro char

902 mov dl, char

903 mov ah, 2

904 int 21h
7 plithos arithmwn sth mnhmh
1008 N EQU 255
010 data segment
011 TABLE dw N dup(?)
012 ends
011
012
013
014
      stack segment
015 ends
016
017 code
018
      code segment
019
020
021
022
     start:
             mov ax, data
mov ds, ax
mov es, ax
023
024
025
026
027
             ; fortwsh mnhmhs me sinexomenous arithmous
927 ; fortwsh mnhmhs me s
928 mov cl, N
929 cld
930 mov di, OFFSET TABLE
931 mov al, 254
932 write_again:
934 dec al
935 cmp al, 254
936 jnz write_again
937
                                                    ; df = 0
                                                  ; fortwsh epomenou dedomenou
037
038
; df = 0
                                                    ; arxikopoihsh kataxwrhtwn athroismatos
                                                     ; fortwsh se kataxwrhth
051 mooove:
052
053
054
             mov ax, bx
mov bx, N+1
div bx
                                                    ; metakinhsh diairetaiou
                                                    ; diaireths
; ektelesh diaireshs
055
056
057
058
             mov dl, ah
call print_hex_full
mov dl, al
call print_hex_full
060
061
062
063
; fortwsh se kataxwrhth
071 cmp al, dl
072 ja local_max_calc
073 mov dl, al
074 local_max_calc:
075 cmp dl, bl
```

```
076
              jb next
077 mo
078 next:
079 cr
              mov dl, al
     cmp cl,00h
jz mooove_2
loop load_again_2
mooove_2:
080
081
082
083
Й84
                ; pairnei ena 2adiko pshfio kai to tipwnei 16dika
085
              print ""
print "m"
print ":"
086
087
088
              push dx
mov dl, dh
call print_hex_full
pop dx
call print_hex_full
089
090
091
092
093
094
              print " "
print "M"
print ":"
095
096
097
              mov dl, bh
call print_hex_full
mov dl, bl
call print_hex_full
098
                                                          ; ektipwsh 8MSB
099
100
                                                          ; ektipwsh 8LSB
101
102
103
              mov ax, 4c00h; exit int 21h
104
105
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 d1, 37h
117 addr2:
118
             print dl
              ret
120 print_hex endp
121
122
123
124
125 print_hex_full proc near
126 push dx
127 push ax
128 push bx
               push dx
push ax
push bx
push dx
129
130
                                         ; save to apotelesma
130
131
132
133
134
135
136
                sar dx. 1
sar dx. 1
sar dx. 1
                sar dx. 1 and dl. Ofh
                call print_hex
137
                pop dx
and dl, Ofh
call print_hex
138
139
140
                pop bx
pop ax
141
142
                pop dx
ret
143
144
145 print_hex_full
                                      endp
147 end start
```

```
001 new_line macro
002 print 0ah
ииз:
              print Odh
004 endm
005
996 print_str macro string
mov dx, offset string
mov ah, 9
int 21h
010 endm
011 endm
011 print macro char
013 mov dl, char
014 mov ah, 2
         mov dl, char
mov ah, 2
int 21h
015 i
016 endm
017
018
data segment
X_print
Y_print
ADD_print
SUB_print
                                  db "X = ", ' $ '
db "Y = ", ' $ '
db "X + Y = ", ' $ '
db "X - Y = ", ' $ '
025
026 stack segment
027 dw 128
| 128 | dw | 128 | dup(0)
029
030 code segment
031 start:
032 mov
             mov ax, data
mov ds, ax
mov es, ax
M33
034
035
036 main proc far
037
038
               mov cl, 00h;
039 loop:
               call read_num
mov ah, 00h
push ax
inc cl ; cl++
cmp cl, 04h ;otan ginei 4, exw 4 egkira pathmena
jz print_results
jmp loop
040
041
042
043
044
Й45
046
047
049
050 print_results:
               results:
new_line
print_str Y_print
pop bx
pop cx
mov dl, cl
call print_bin
mov dl, bl
call print_bin
051
052
053
054
055
056
057
058
                                              ;EDW EXW STON BL TO LOW TOU Y ;KAI STON CL TO HIGH TOU Y
059
                 mov al, cl
sal al,1
sal al,1
061
062
063
                 sal al.1
sal al.1
add al. bl
064
065
066
067
                                              ; a1 = Y
068
069
070
071
072
073
074
                 push ax
print " "
                 pop ax
                 pop bx
pop cx
push ax
075
```

```
076
077
078
079
                print_str_X_print
               mov dl, cl
call print_bin
mov dl, bl
call print_bin
080
081
082
                                          ;EDW EXW STON BL TO LOW TOU X ;KAI STON CL TO HIGH TOU X
083
084
085
                mov al, cl
sal al,1
086
087
088
                sal al.1
                sal al.1
sal al.1
089
090
091
                add al. bl
                                          ; a1 = X
092
093
094
                push ax
new_line
095
096
097
                print_str ADD_print
               pop ax
mov bx,ax
pop ax
xchg ax,bx
098
099
100
101
102
103
               mov ah, 00h
mov bh, 00h
push ax
add ax, bx
mov dx,ax
104
105
106
107
108
                                        ; d1 = a1 + b1
109
               push bx
call print_decimal
print " " _____
110
111
112
113
                print_str SUB_print
114
                pop bx
pop ax
115
116
117
118
119 cmp a

120 JB ari

121 mov d

122 sub d

123 jmp pr

124

125 arnhtikos:

126 mov d

127 sub d

128 push d

129 print
119
                cmp al.bl
                JB arnhtikos
                mov dl.al
sub dl.bl
                jmp print_abs
               mov dl.bl
sub dl.al
                push dx
print "-"
pop dx
131 print_abs:
              call print_decimal
new_line
133
134
135 jmp start
136
137 main endp
138
139
140 read_num proc near
141
              ; Reads key asci code in al mov ah, 01h int 21h
142
143
144
145
              cmp al, 30h
jl read_num
                                     ;elegxos an o arithmos einai <0..9>
146
147
148
              cmp al, 39h
jle ok_dec
149
150
```

```
cmp al, 39h
jle ok_dec
151
152
153
154
            cmp al, 41h
jl read_num
                                 ;elegxos an to gramma einai (A..F)
155
156
157
             cmp al, 46h
158
159
             jg read_num
            sub al, 37h
jmp end_num
160
                                 ;afairw 37h an gramma
161
162
163
164 ok_dec:
165
            sub a1, 30h
                                ;afairw 30h an pshfio
166
167
     end_num:
168
169
            ret
      read_num endp
170
171
172
173
174
175
     print_asci proc near
            mov ax, 4c00h int 21h
175
176
ends
177
178
print_hex proc near
179
cmp dl, 9
jg addr1
jg addr1
181 add dl, 30
182 jmp addr2
183 addr1:
                         30h
184 add dl, 37h
185 addr2:
186
187
            print dl
188 print_hex endp
189
190 print_bin proc near
191
192
193
192

193

194

195

b_addr1

195

b_addr1:
196
197
198
            add d1, 37h
      b_addr2:
            print dl
199 ret
200 print_bin endp
201
202
203 print_dec proc near
204 add dl, 30h
205 print dl
206
207
208
     print_dec endp
209
210 print_decimal proc near
211 push ax
              push ax push bx
212
213
214
              push dx
215
216
              mov al.00H
mov ah.00H
217
218
              jmp ekato
219
220 ekato_plus:
221 inc ah
222 sub dx,0064H
222
224 ekato:
```

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

```
001 new_line macro
002 print Oah
003 print Odh
004 endm turn
          print_str macro string
mov dx, offset string
mov ah, 9
int 21h
   006
007
012 print macro char mov dl. char mov ah. 2 int 21h endm
   019 data segment
020 input db 21 dup(?)
021 ends
   N22
   023 stack segment
024
025 ends
   026
027 code segment
028 start:
                  mov ax, data
mov ds, ax
mov es, ax
   R29
   031
032
   033
034
                   mov di, offset input
cld ;df = 0
mov cx, 16 ;Metrhths
mov bl,00h
push bx
   035
   036
   038
   040 read_loop:
   041
                   pop bx
push bx
call read_key
   042
043
                                                      ;diavazw
                    pop bx
cmp al. 80h
   044
045
                                                      ;plhktro termatismou ENTER
                   je stop
stosh
cmp al.39h
                                                      ; opou exw orisei na vazei to 80h
; save sth mnhmh
  cmp al,391
jg print0
050 print0:
052 push bx
053 print al
                                                      ;an einai <39h exoume arithmo -> bl++
                                                      ;printaroume oti diavasoume
                    loop read_loop
   056
057
058
                                                      ;edw exoume apothikevmena oti diavasame,
;kai ston bl, to plhthos twn arithmwn
  957 process: ;edw
new_line ;kai
cld
060 mov si, offset input
mov cx, 16
962 pop bx
mov dl,bl
964 push dx
push bx
100p_1:
10dsb ;diavazoum
668 call hig to small :e
          process:
                   lodsb ;diavazoume apo mnhmh
call big_to_small ;edw kanoume ta kefalaia peza
cmp al, 39h
jle print_digits ;kai an einai con
stosb ;sth make?
   069
070
071
072
                                                  ts ;kai an einai grammata, ta ksanaapothikevoume
;sth mnhmh giati prwta printaroume ta pshfia
                    loop loop_1
   073
074 print_digits:
075 print al
```

```
076
077
078
079
                   pop bx
dec bl
                   dec bl ;gia kathe arithmo pou printaroume, meiwnoume ton bl push bx ;pou exei to plhthos twn arithmwn, opote otan bl=0 cmp bl,00h ;teleiwsame me tous arithmous kai printaroume "-'
                   je add_pavla
loop loop_1
 083 add_pavla:
084 print "-"
 084
085
                  pop bx
pop dx
push dx
cmp cl.0
 086
087
        cmp cl.0
je therest
dec cx
therest:
                                                  ;edw vlepoume posa grammata exoun meinei pou den
;exoume epeksergastei akoma (an dld o teleftaios
;arithmos htan o 14os xarakthras, shmainei oti
;iparxoun 2 akoma grammata pou prepei na kanoume
art ;pezous char, kai na prosthesoyme sth stiva)
                rest.
cmp cl,0 ;ipa
je letters_start
lodsb
call big_to_small
stosb
loop therest
 092
093
094
095
096
097
098
099
100
101
        letters_start:
                  pop dx
mov bx.dx
mov dx.16
sub dx.bx
 102
 104
                  mov cx.dx
                                                ;edw exoume ston cx pia, to plhthos twn grammatwn
        print_letters:
cmp cl,00h
je restart
lodsb
                                               ;kai kanw thn antistoixh diadikasia gia
;grammata exw
 108
109
;an den einai (A..Z) tote den kanw tipota
                                                ;alliws prosthetw 32h, gia na exw (a..z)
 131 big_to_small endp
132
133 read_key proc near
         read_key proc near
ignore:
mov ah, 8
int 21h
 134
 136
137
138
139
                  cmp al, Odh
jne contin
mov al,80h
jmp exit
                                                  ;elegxos an paththei to ENTER (vazw al=80h);gia na mhn mperdeftw me to 'D' sth sinexeia
 140
142
143 contin:
144 cmp
145 jl i
146
147 cmp
148 jg i
149
150 jmp
 142
                 cmp al, 30h
jl ignore
                                                 ;oi klasikoi elegxoi
                 cmp al, 39h
jg its_letter
                  jmp exit
 151
152
153
154
        its_letter:
    cmp al, 'A'
    jl ignore
    cmp al, 'Z'
    jg ignore
exit:
 156
157
158
ret
159 read_key endp
 161
162 end start
```

Και ένα στιγμιότυπο από το output της οθόνης για διαδοχικές εισοδους *A8X9S1FETD73A8KL* (παράδειγμα εκφώνησης) και *5GIU9O04WN7FG3D6*:

