



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜ&ΜΥ
Εργαστήριο Μικροϋπολογιστών

4^η Εργαστηριακή Άσκηση
Ακ. έτος 2011-2012

Ομάδα C07:

Ελένη Ευαγγελάτου	A.M.: 03108050
Γρηγόρης Λύρας	A.M.: 03109687
Βασιλεία Φραγκιαδάκη	A.M.: 03108026

13 Ιανουαρίου 2012

Άσκηση (i)

Σε αυτή την άσκηση ζητείται αρχικά να φτιάξουμε δύο ρουτίνες, μία που να επιστρέφει τους ascii τιμές του δεκαεξαδικού αριθμού που υπάρχει στον al και μία που να επιστρέφει τον αριθμό στον al, όταν δέχεται ως είσοδο τις ascii τιμές στον ax (στον ah ο ascii των τεσσάρων MSB bits και στον al των τεσσάρων LSB bits). Γι' αυτό το σκοπό στην πρώτη περίπτωση απομονώνουμε ανά τετράδες τα ψηφία και ανάλογα με το αν ο αριθμός είναι μεγαλύτερος ή μικρότερος του 10 επιστρέφουμε την αντίστοιχη ascii τιμή. Στη δεύτερη ρουτίνα βρίσκουμε από την ascii τιμή τον αριθμό που αντιστοιχεί και αφού πολλαπλασιάσουμε τον MSB αριθμό επί 16, τον προσθέτουμε στον LSB αριθμό. Έπειτα φτιάξαμε μια βιβλιοθήκη -ένα αρχείο 'lib.inc'- που περιλαμβάνει όλες τις ρουτίνες που χρειαζόμαστε στο πρόγραμμα, βάζοντας τις κατάλληλες δηλώσεις των ρουτινών αλλά και τις αντίστοιχες εντολές στο κυρίως πρόγραμμα ώστε να συμπεριλάβουμε τη βιβλιοθήκη. Έπειτα συμπεριλάβαμε και το αρχείο 'macros.txt' όπου βάλαμε τις μακροεντολές που χρειαστήκαμε. Στο κυρίως πρόγραμμα διαβάζουμε τους δεκαεξαδικούς αριθμούς που δίνει ο χρήστης, μετρώντας κάθε φορά τα ψηφία ώστε να μπορούμε να "φτιάξουμε" τον αριθμό που αντιστοιχεί σ' αυτή την παράσταση, ακόμα διαβάζουμε την πράξη που θα εκτελέσουμε (πρόσθεση ή αφαίρεση) και σταματάμε με το <enter>. Στην πρόσθεση μπορεί να έχουμε υπερχείλιση, γεγονός που ελέγχουμε και στην αφαίρεση χρειάζεται να ελέγξουμε ποιο από τα δύο ορίσματα είναι μεγαλύτερο (σε unsigned παράσταση) ώστε σε κάθε περίπτωση να εκτυπώσουμε ή όχι το '-' και να αφαιρέσουμε από το μεγαλύτερο το μικρότερο αριθμό.

Κυρίως κώδικας:

```
1  DEFINE_ASCII MACRO
2      LOCAL GRAMMA,PSIFIO,SYNEXEIA,GRAMMA2,PSIFIO2,SYNEXEIA2
3      LOCAL SKIP_ASCII
4      JMP SKIP_ASCII
5
6      ASCII PROC NEAR
7      PUSH BX
8      MOV AH,AL          ; SAVE TON AL
9      AND AL,0FH         ; APOMONWSH TWN 4 LSB BITS
10     CMP AL,09H
11     JG GRAMMA          ; AN EINAI GRAMMA
12     JMP PSIFIO         ; AN EINAI ARITHMOS
13
14     GRAMMA:
15         ADD AL,37H      ; ASCII GIA KEFALAIA
16         JMP SYNEXEIA
17
18     PSIFIO:
19         ADD AL,30H
20
21     SYNEXEIA:
22         MOV BL,AL       ; SAVE TON AL
23         AND AH,0F0H     ; APOMONWSH TWN 4 MSB BITS
24         SHR AH,4        ; OLIS8HSH 4 SESEIS DEXIA GIA NA PARW TA 4 MSB BITS
25         CMP AH,09H
26         JG GRAMMA2
27         JMP PSIFIO2
28
29     GRAMMA2:
30         ADD AH,37H      ; AH ->ASCII 4 MSB BITS
31         JMP SYNEXEIA2
32
33     PSIFIO2:
34         ADD AH,30H
35
36     SYNEXEIA2:
37         MOV AL,BL       ; AL ->ASCII TWN 4 LSB BITS
38         POP BX
39     RET
40     ASCII ENDP
41
42     SKIP_ASCII:
43     DEFINE_ASCII ENDM
44
45     DEFINE_NUMBER MACRO
46         LOCAL CONT,CONT1
47         LOCAL SKIP_NUMBER
48         JMP SKIP_NUMBER
```

```

49
50     NUMBER PROC NEAR
51     PUSH BX
52     PUSH CX
53     PUSH DX
54     SUB AL,30H      ; AN O AL EINAI ASCII ARITHMOU
55     CMP AL,09H
56     JLE CONT
57     SUB AL,07H      ; ASCII KEFALAIΟΥ
58     CMP AL,09H
59     JLE CONT
60     SUB AL,20H      ; ASCII MIKROU GRAMMATOS
61
62     CONT:
63     SUB AH,30H      ; ANTISTOIXA GIA TON AH
64     CMP AH,09H
65     JLE CONT1
66     SUB AH,07H
67     CMP AH,09H
68     JLE CONT1
69     SUB AH,20H
70
71     CONT1:
72
73     SHL AH,4        ; POLLAPLASIASMOS TWN 4 MSB ME 16
74     ADD AL,AH       ; AL ->PROSSESI TWN DYO ARI8MWN
75     MOV AH,0
76     POP DX
77     POP CX
78     POP BX
79     RET
80     NUMBER ENDP
81
82     SKIP_NUMBER:
83     DEFINE_NUMBER ENDM
84
85     DEFINE_PRINT_HEX MACRO
86     ; TYPWNEI TON ANTISTOIXO HEX ARI8MO POU VRISKETAI STON AH
87     LOCAL ADDR,ADR2
88     LOCAL SKIP_PRINT_HEX
89     JMP SKIP_PRINT_HEX
90
91     PRINT_HEX PROC NEAR
92     PUSH BX
93     PUSH CX
94     PUSH DX
95     CMP AH,9        ; AN O ARI8MOS EINAI METAKSY O K 9 PROSSETW 30H
96     JG ADDR
97     ADD AH,30h
98     JMP ADR2
99
100    ADDR:
101    ADD AH, 37h      ; DIAFORETIKA PROSSETW 37H ('a' = 41H)
102    ADR2:
103    PRINT AH
104    POP DX
105    POP CX
106    POP BX
107    RET
108    PRINT_HEX ENDP
109
110    SKIP_PRINT_HEX:
111    DEFINE_PRINT_HEX ENDM
112
113    DEFINE_PRINT_DEC MACRO
114    LOCAL ADDR1,ADDR2
115    LOCAL SKIP_PRINT_DEC
116    JMP SKIP_PRINT_DEC
117
118    PRINT_DEC PROC NEAR
119    ; TYPWNEI TON ANTISTOIXO DEKADIKO POU VRISKETAI STON AX
120    PUSH DX
121    PUSH CX
122    PUSH BX
123    MOV CX,0

```

```

124
125 ADDR1:
126     MOV DX,0
127     MOV BX,10
128     DIV BX
129     PUSH DX
130     INC CX
131     CMP AX,0
132     JNZ ADDR1
133
134 ADDR2:
135     POP DX
136     PRINT_NUM DX
137     LOOP ADDR2
138     POP BX
139     POP CX
140     POP DX
141     RET
142 PRINT_DEC ENDP
143 SKIP_PRINT_DEC:
144 DEFINE_PRINT_DEC ENDM
145
146
147 DEFINE_READ_HEX MACRO
148 ; DIAVAZEI TOUS DYO ARISMOUS KAI THN PRAXH(+ H -)
149     LOCAL IGNORE,CONTIN,PROSTHE,CONT2,PSIFIO_1,PSIFIA_2,PSIFIA_3,PSIFIA_4,SYNEX,IGNORE1
150     LOCAL CONT3,CONT4,PSIFIO1_2,PSIFIA2_2,PSIFIA3_2,PSIFIA4_2,TELOS
151     LOCAL SKIP_READ_HEX
152     JMP SKIP_READ_HEX
153
154 READ_HEX PROC NEAR
155     MOV BX,0
156     MOV CX,0
157     MOV DX,0
158
159 IGNORE:
160     READ                ; DIAVAZEI MEXRI 4 HEX PSIFIA, STAMATAEI OTAN DIAVASEI + H -
161     CMP AL,2BH          ; ELEGXOS GIA +
162     JE PROSTHE
163     CMP AL,2DH          ; ELEGXOS GIA '-'
164     JE CONT2            ; 'q'/'Q' TERMATISMOS
165     CMP AL,'Q'
166     JE QUIT
167     CMP AL,'q'
168     JE QUIT
169     CMP BX,4            ; AGNOEI PERA TWN 4 PSIFIWN
170     JE IGNORE
171     CMP AL,30H
172     JL IGNORE          ; AGNOEI PERA TWN HEX PSIFIWN
173     CMP AL,39H
174     JLE CONTIN
175     CMP AL,'A'
176     JL IGNORE
177     CMP AL,'F'
178     JLE CONTIN
179     CMP AL,'a'
180     JL IGNORE
181     CMP AL,'f'
182     JLE CONTIN
183     JMP IGNORE
184
185 CONTIN:
186     MOV AH,0
187     PUSH AX
188     INC BL              ; METRHTHS PSIFIWN
189     JMP IGNORE
190
191 PROSTHE:
192     MOV PROSTH,1        ; SAN FLAG GIA PROS8ESH
193 CONT2:
194     CMP BX,00           ; FTIAXNEI TON ARI8MO ANALOGA ME TON ARI8MO TWN SPIFIWN
195     JZ QUIT
196     CMP BL,1
197     JE PSIFIO_1
198     CMP BL,2

```

```

199     JE PSIFIA_2
200     CMP BL,3
201     JE PSIFIA_3
202     CMP BL,4
203     JE PSIFIA_4
204
205     PSIFIO_1:
206     POP AX
207     MOV AH,30H      ; AH => '0'
208     CALL NUMBER
209     JMP SYNEX        ; 1 PSIFIO ->AX
210
211     PSIFIA_2:
212     POP AX
213     MOV CL,AL        ; ENDIAMESOS KATAXRHTHS GIA TO LSB BIT(OPWS DIVAZOYME AP' TH STOIVA)
214     POP AX           ; MSB BIT
215     MOV AH,AL
216     MOV AL,CL
217     CALL NUMBER
218     JMP SYNEX        ; 2 PSIFIA -> AX
219
220     PSIFIA_3:
221     POP AX           ; 30-20 (2 LSB BITS)
222     MOV BL,AL
223     POP AX
224     MOV AH,AL
225     MOV AL,BL
226     CALL NUMBER
227     MOV DX,AX
228
229     POP AX
230     MOV AH,30H      ; 10 PSIFIO
231     CALL NUMBER
232     MOV AH,AL        ; POL/ZOYME EPI 16~2= 2~8 TO MSB BIT <=> PHGAINETI STON MSB KATAXRHTH
233     MOV AL,DL
234     JMP SYNEX        ; 3 PSIFIA ->AX
235
236     PSIFIA_4:
237     POP AX
238     MOV CL,AL
239     POP AX
240     MOV AH,AL        ; 20 LSB BIT P ANASYROYME AP' TH STOIVA
241     MOV AL,CL
242     CALL NUMBER
243     MOV DX,AX        ; 40-30 PSIFIO STON DX
244
245     POP AX
246     MOV CL,AL
247     POP AX           ; 20-10 BIT
248     MOV AH,AL
249     MOV AL,CL
250     CALL NUMBER
251     MOV AH,AL
252     MOV AL,DL        ; SAN POL/SMOS EPI 256 TA 2 MSB PSIFIA
253                     ; 4 PSIFIA ->AX
254
255     SYNEX:           ; 20S ARI8MOS
256     PUSH AX          ; APO8IKEYSH 10U ARI8MOY STH STOIVA
257
258     MOV BX,0         ; MHDENISMOS METRHTH PSIFIWN
259     IGNORE1:
260     READ
261     CMP AL,0DH        ; ELEGXOS GIA <ENTER>
262     JE CONT4
263     CMP AL,'Q'
264     JE QUIT
265     CMP AL,'q'
266     JE QUIT
267     CMP BX,4          ; AGNOEI PERA TWN 4 PSIFIWN
268     JE IGNORE1        ; KAI TWN HEX PSIFIWN
269     CMP AL,30H
270     JL IGNORE1
271     CMP AL,39H
272     JLE CONT3
273     CMP AL,'A'

```

```

274     JL IGNORE1
275     CMP AL,'F'
276     JLE CONT3
277     CMP AL,'a'
278     JL IGNORE1
279     CMP AL,'f'
280     JLE CONT3
281     JMP IGNORE1
282
283     CONT3:
284     PUSH AX
285     MOV AH,0
286     INC BL
287     JMP IGNORE1
288
289     CONT4:
290     ; IDIA DIADIKASIA GIA NA "FTIAXOYME" TO 20 ARISMO
291     CMP BL,00H
292     JE QUIT
293     CMP BL,1
294     JE PSIFIO1_2
295     CMP BL,2
296     JE PSIFIA2_2
297     CMP BL,3
298     JE PSIFIA3_2
299     CMP BL,4
300     JGE PSIFIA4_2
301
302
303     PSIFIO1_2:
304     POP AX
305     MOV AH,30H
306     CALL NUMBER
307     JMP TELOS      ; 1 PSIFIO ->AX
308
309     PSIFIA2_2:
310     POP AX
311     MOV CL,AL      ; ENDIAMESOS KATAXRHTHS GIA TO LSB PSIFIO
312     POP AX
313     MOV AH,AL
314     MOV AL,CL
315     CALL NUMBER
316     JMP TELOS      ; 2 PSIFIA ->AX
317
318     PSIFIA3_2:
319     POP AX          ; 30-20
320     MOV BL,AL
321     POP AX
322     MOV AH,AL
323     MOV AL,BL
324     CALL NUMBER
325     MOV DX,AX
326
327     POP AX
328     MOV AH,30H      ; 10 PSIFIO
329     CALL NUMBER
330     MOV AH,AL
331     MOV AL,DL      ; POL/ZOYME EPI 256 TO 10 PSIFIO
332     JMP TELOS      ; 3 PSIFIA ->AX
333
334     PSIFIA4_2:
335     POP AX
336     MOV CL,AL
337     POP AX
338     MOV AH,AL
339     MOV AL,CL
340     CALL NUMBER
341     MOV DX,AX      ; 40-30 PSIFIO STON DX
342
343     POP AX
344     MOV CL,AL
345     POP AX
346     MOV AH,AL      ; 10-20
347     MOV AL,CL
348     CALL NUMBER

```

```

349     MOV AH,AL
350     MOV AL,DL           ; SAN POL/SMOS EPI 256 TA 2 MSB PSIFIA
351                           ; 4 PSIFIA ->AX
352
353     TELOS:
354         MOV DX,AX       ; 20S ARI8MOS ->DX
355         POP AX          ; 10S ARI8MOS ->AX
356     RET
357     READ_HEX ENDP
358     SKIP_READ_HEX:
359     DEFINE_READ_HEX ENDM

```

Κυρίως κώδικας εκτελέσιμου:

```

1     INCLUDE LIB.INC           ; PERILAMVANOYME TH VILVIO8HKH
2     INCLUDE MACROS.TXT       ; KAI TIS MAKROENTOLES
3
4     DATA SEGMENT
5         ; ADD YOUR DATA HERE!
6         PROSTH DB 0          ; FLAG GIA PROS8ESI // FLAG GIA PLHN STHN EKTYPWSH THS AFAIRESHS
7         MESSAGE DB 3DH,"$"
8         GRAMMI DB 0ah,0dh,"$"
9         PLHN DB "-$"
10    ENDS
11
12    STACK SEGMENT
13        DW 128 DUP(0)
14    ENDS
15
16    CODE SEGMENT
17    START:
18        ; SET SEGMENT REGISTERS:
19        MOV AX, DATA
20        MOV DS, AX
21        MOV ES, AX
22
23        MOV PROSTH,0          ; ARXIKOPOIHSH FLAG
24        CALL READ_HEX         ; 10S ->AX, 20S ->DX
25
26        PUSH AX
27        PUSH DX
28        PUSH BX
29        PRINT_STRING GRAMMI ; ALLAGI GRAMMHS KAI '='
30        MOV CL,3DH
31        PRINT CL
32        POP BX
33        POP DX
34        POP AX
35
36        CMP PROSTH,1          ; ELEGXOS FLAG GIA PRAXH
37        JE PROSTHESI
38        JMP AFARESH
39
40    PROSTHESI:
41        ADD AX,DX
42        JNC EKTYPWSH
43        PUSH AX
44        PUSH DX
45        PUSH BX
46        MOV CL,31H            ; YPERXEILISH STHN PROS8ESI -> 50 PSIFIO = 1
47        ;MOV PROSTH,02        ; FLAG GIA KRATOUMENO STHN PROS8ESH
48        PRINT CL
49        POP BX
50        POP DX
51        POP AX
52
53    EKTYPWSH:
54        ; EKTYPWSH APOTELESMATOS APO PROS8ESI H AFARESH
55        PUSH AX                ; APO8IKEUSH ARXIKOU ARI8MOU
56        CMP AH,0               ; MHN EKTYPWSEIS TA 2 PRWTA PSIFIA AN EINAI 0
57        JE EPOMENO
58        PUSH AX                ; ALLIWS KALESE THN ASCII KAI EKTYPWSE TA 2 PSIFIA
59        MOV AL,AH              ; ARI8MOS STON AL
60        CALL ASCII
61        PRINT AH
62        PRINT AL
63        POP AX

```

```

63  EPOMENO:
64      CALL ASCII          ; TO IDIO GIA TON ARI8MO STON AL
65      PRINT AH
66      PRINT AL
67
68      PUSH AX
69      PUSH DX
70      PUSH CX
71      MOV BL,3DH          ; '='
72      PRINT BL
73      POP CX
74      POP DX
75      POP AX
76
77      POP AX              ; EXAGOUME TON ARXIKO ARI8MO PROS EKTYPWSH
78
79      CMP PROSTH,03        ; ELEGXOS FLAG GIA EKTYPWSH PLHN
80      JNE ADDRESS
81      PUSH AX
82      PUSH DX
83      PRINT_STRING PLHN
84      POP DX
85      POP AX
86
87  ADDRESS:
88      CALL PRINT_DEC      ; EKTYPWSH ANTISTOIXOU DEKADIKOU ARI8MOU
89      PRINT_STRING GRAMMI ; ALLAGI GRAMMHS
90
91      JMP START
92
93  AFAIRESH:
94      CMP AX,DX           ; UNSIGNED SYGKRISH ARI8MWN
95      JB MIKROTEROS      ; AX < DX
96      SUB AX,DX          ; AX >= DX
97      JMP EKTYPWSH
98
99  MIKROTEROS:
100     MOV PROSTH,03        ; FLAG = 03 GIA TYPWSH PLHN
101     PUSH AX
102     PUSH DX
103     PRINT_STRING PLHN
104     POP DX
105     POP AX
106     SUB DX,AX
107     MOV AX,DX
108     JMP EKTYPWSH
109
110  QUIT:
111     MOV AX, 4C00H        ; EXIT TO OPERATING SYSTEM.
112     INT 21H
113
114  ENDS
115
116  DEFINE_READ_HEX
117  DEFINE_PRINT_HEX
118  DEFINE_PRINT_DEC
119  DEFINE_ASCII
120  DEFINE_NUMBER
121
122  END START

```

Τα macros που χρησιμοποιήσαμε:

```

1  READ MACRO
2      MOV AH,1
3      INT 21H
4  ENDM
5
6  PRINT_NUM MACRO CHAR
7      PUSH DX
8      PUSH AX
9      MOV DX,CHAR
10     ADD DX,30H
11     MOV AH,2
12     INT 21H
13     POP AX

```



```

14     POP DX
15
16 ENDM
17
18 PRINT MACRO CHAR
19     PUSH AX
20     PUSH DX
21     MOV DL,CHAR
22     MOV AH,02H
23     INT 21H
24     POP DX
25     POP AX
26 ENDM
27
28 PRINT_STRING MACRO STRING
29     PUSH AX
30     PUSH DX
31     MOV DX,OFFSET STRING ;ASSUME THAT STRING IS A VARIABLE OR CONSTANT, not AN ADDRESS
32     MOV AH,09H
33     INT 21H
34     POP DX
35     POP AX
36 Endm

```

Άσκηση (ii)

Κυρίως κώδικας:

```

1  INCLUDE MACROS.TXT
2
3  CODE SEGMENT
4      ASSUME CS:CODE,DS:CODE,ES:CODE
5      ORG 100H
6
7  MAIN PROC FAR
8      ECHO         DW 0
9      ECHO_LINE    DW 0
10     ECHO_COL     DW 0
11     MSG_LINE     DW 0
12     MSG_COL      DW 41
13     MESSAGE1 DB "FOR ECHO FUNCTION PRESS <Y> . OTHERWISE PRESS <N> ? $"
14     MESSAGE2 DB "PLEASE GIVE BAUD RATE. PRESS <1> FOR 300, <2> FOR 600, <3> FOR 1200, <4> FOR 2400, <5> FOR 4800, <6> FOR9600 ."
15     NEW_LINE DB 0AH, 0DH, '$'
16
17
18  START:
19      CLEAR                     ; CLEAR THE SCREEN
20      PRINT_STRING MESSAGE1
21      CALL READ_ECHO
22      PRINT_STRING NEW_LINE
23      PRINT_STRING MESSAGE2
24      CALL READ_BR ;BAUT RATE
25      CALL OPEN_RS232
26      CLEAR                     ; CLEAR THE SCREEN
27      CALL SCREEN_SPLIT         ; PRINT THE LINE IN THE MIDDLE
28      MOVE_THERE ECHO_LINE ECHO_COL ; MOVE TO ECHO LINE COLUMN
29
30  READ_RS232:
31      CALL RXCH_RS232           ; READ COM
32      CMP AL,0                  ; IF THERE IS NOTHING THERE
33      JE NOCHAR                 ; CHECK KEYBOARD
34      MOV BL,AL                 ; MOVE IT TO BL FOR PRINTING
35      ; === PRINT THE MESSAGE ===
36      CMP BL,0DH
37      JE INCMLINE
38      ; === IF GIVEN ENTER SHOULD INC LINE ===
39      MOVE_THERE MSG_LINE MSG_COL ; MOVE CURSOR TO APPROPRIATE LINE COL
40      PRINT_THERE BL             ; OK PRINT IT
41      MOV BP,OFFSET MSG_COL      ; AND INCREASE
42      MOV BL,DS:[BP]             ; THE COLUMN
43      INC BL                     ; IF IT REACHES 80
44      CMP BL,80                  ; IT SHOULD BE RESET TO 41
45      JNE NOINCMLINE
46  INCMLINE:
47      INCREASE_LINE MSG_LINE MSG_COL 41 ; INCREASE LINE AND SAVE NEW LINE/COLUMN

```

```

48     MOVE_THERE MSG_LINE MSG_COL          ; AND MOVE THE CURSOR THERE
49 ; === THIS SHOULD SCROLL UP WHEN NEEDED ===
50     MOV BP,OFFSET MSG_LINE
51     MOV BL,DS:[BP]
52     CMP BL,22
53 ; === COMPARE WITH 22 [LINES IN DOSBOX] ===
54     JNE NOCHAR
55     SUB BL,1                               ;
56     MOV DS:[BP],BL
57     SCROLL_UP 1 0 41 23 79
58     JMP NOCHAR                            ; IF YOU DON'T NEED TO SCROLL
59                                         ; THEN CHECK THE KEYBOARD
60 NOINCMLINE:
61     MOV DS:[BP],BL                       ; STORE THE COLUMN (NO LINE INCREASE NO SCROLL)
62
63
64 NOCHAR:
65     READNB
66 ; === NO CHAR THEN READ COM ===
67     JZ READ_RS232
68     MOV BL,AL
69 ; === PRINT ECHO ===
70     CMP BL,27
71 ; === COMPARE WITH ESCAPE ===
72     JE QUIT
73     CMP BL,8
74 ; === COMPARE WITH BACKSPACE ===
75     JE READ_RS232
76     CALL TXCH_RS232
77 ; === SHOULD I ECHO R SHOULD I NOT? ===
78     MOV BP,OFFSET ECHO
79     CMP DS:[BP],1
80     JNE READ_RS232
81     CMP BL,0DH
82     JE INCELINE
83 ; === IF GIVEN ENTER SHOULD INC LINE ===
84     MOVE_THERE ECHO_LINE ECHO_COL
85     PRINT_THERE BL
86     MOV BP,OFFSET ECHO_COL
87     MOV BL,DS:[BP]
88     INC BL
89     CMP BL,40
90     JNE NOINCELINE
91 INCELINE:
92     INCREASE_LINE ECHO_LINE ECHO_COL 0
93     MOVE_THERE ECHO_LINE ECHO_COL
94 ; === THIS SHOULD SCROLL UP WHEN NEEDED ===
95     MOV BP,OFFSET ECHO_LINE
96     MOV BL,DS:[BP]
97     CMP BL,22
98 ; === COMPARE WITH 22 [LINES IN DOSBOX] ===
99     JNE READ_RS232
100    SUB BL,1
101    MOV DS:[BP],BL
102    SCROLL_UP 1 0 0 23 39
103    JMP READ_RS232
104
105 NOINCELINE:
106    MOV DS:[BP],BL
107    JMP READ_RS232
108
109 QUIT:
110    CLEAR
111    MOV AL,0H
112    EXIT
113
114 MAIN ENDP
115
116
117 SCREEN_SPLIT PROC NEAR
118     MOV CX, 24 ; 24 FORES NA TREXEI TO LOOP
119
120 LOOP1:
121     MOVE_THERE_CX 40 ; PAEI STHN I GRAMMH KAI STHLH PANTA THN 40 KAI TUPWNEI''
122     PRINT_THERE 179

```

```

123     LOOP LOOP1
124     MOVE_THERE_CX 40 ; PAEI STHN I GRAMMH KAI STHLH PANTA THN 40 KAI TUPWNEI''
125     PRINT_THERE 179
126     RET
127
128     SCREEN_SPLIT ENDP
129
130     READ_ECHO PROC NEAR
131
132     ;PI8ANON NA XREIASTEI PUSH K POP TON AL
133     LOOP_E:
134         READ                ;DIABAZEI XARAKTHRA APO TO PLHKTROLOGIO
135         CMP AL, 'Y'        ;AN YES TOTE EXOUME ECHO
136         JE ECHO_ON
137         CMP AL, 'y'
138         JE ECHO_ON
139         CMP AL, 'N'
140         JE ECHO_OFF
141         CMP AL, 'n'
142         JE ECHO_OFF
143         CMP AL, 27
144         JE QUIT            ;AN PATH8EI ESCAPE PHGAINETI STO QUIT
145         JMP LOOP_E
146
147     ECHO_ON:
148         MOV BP,OFFSET ECHO
149         MOV DS:[BP],1
150         RET
151     ECHO_OFF:
152         RET
153
154     READ_ECHO ENDP
155
156     READ_BR PROC NEAR
157
158     ;PI8ANON NA XREIASTEI PUSH K POP TON AL
159     LOOP_BR:
160         READ                ;DIABAZEI XARAKTHRA APO TO PLHKTROLOGIO
161         CMP AL, '1'        ;AN YES TOTE EXOUME ECHO
162         JE BAUD_300
163         CMP AL, '2'
164         JE BAUD_600
165         CMP AL, '3'
166         JE BAUD_1200
167         CMP AL, '4'
168         JE BAUD_2400
169         CMP AL, '5'
170         JE BAUD_4800
171         CMP AL, '6'
172         JE BAUD_9600
173         CMP AL, 27
174         JE QUIT            ;AN PATH8EI ESCAPE PHGAINETI STO QUIT
175         JMP LOOP_BR
176
177     ;=====DIABASMA GIA TO BAUD RATE APO PLHKTROLOGIO=====
178
179     BAUD_300:
180         MOV AL,43H
181         RET
182     BAUD_600:
183         MOV AL,63H
184         RET
185     BAUD_1200:
186         MOV AL,83H
187     BAUD_2400:
188         MOV AL,0A3H
189         RET
190     BAUD_4800:
191         MOV AL,0C3H
192         RET
193     BAUD_9600:
194         MOV AL,0E3H
195         RET
196     READ_BR ENDP
197

```

```

198
199 ;==== INITIALIZE RS232 PORT ====
200 OPEN_RS232 PROC NEAR
201     JMP BEGIN
202     BAUD_RATE LABEL WORD
203         DW 1047
204         DW 768
205         DW 385
206         DW 192
207         DW 96
208         DW 48
209         DW 24
210         DW 12
211 BEGIN:
212     STI
213     MOV AH,AL
214     MOV DX,03FBH
215     MOV AL,80H
216     OUT DX,AL
217
218     MOV DL,AH
219     MOV CL,4
220     ROL DL,CL
221     AND DX,0EH
222     MOV DI,OFFSET BAUD_RATE
223     ADD DI,DX
224     MOV DX,03F9H
225     MOV AL,CS:[DI]+1
226     OUT DX,AL
227
228     MOV DX,03F8H
229     MOV AL,CS:[DI]
230     OUT DX,AL
231
232     MOV DX,03FBH
233     MOV AL,AH
234     AND AL,01FH
235     OUT DX,AL
236     MOV DX,03F9H
237     MOV AL,0H
238     OUT DX,AL
239     RET
240 OPEN_RS232 ENDP
241
242 ;==== READ 1 CHAR FROM RS232 PORT ====
243 RXCH_RS232 PROC NEAR
244     MOV DX,3FDH
245     IN AL,DX
246
247     TEST AL,1
248     JZ NOTHING
249     SUB DX,5
250     IN AL,DX
251     JMP EXIT2
252 NOTHING:
253     MOV AL,0
254 EXIT2:
255     RET
256
257 RXCH_RS232 ENDP
258
259 ;==== WRITE 1 CHAR TO RS232 PORT ====
260 TXCH_RS232 PROC NEAR
261     PUSH AX
262     MOV DX,03FDH
263
264 TXCH_RS232_2:
265     IN AL,DX
266     TEST AL,020H
267     JZ TXCH_RS232_2
268
269     SUB DX,5
270     POP AX
271     OUT DX,AL
272     RET

```

```

273 TXCH_RS232 ENDP
274
275 END MAIN

```

Τα macros που χρησιμοποιήσαμε:

```

1  ;This macro change registers AH,AL
2  READ MACRO
3      MOV AH,1
4      INT 21H
5  ENDM
6
7  ;Non blocking read
8  READNB MACRO
9      MOV AH,06H
10     MOV DL,OFFH
11     INT 21H
12 ENDM
13
14 MOVE_THERE_CX MACRO COL
15     PUSH AX
16     PUSH BX
17     PUSH DX
18     MOV AH,02H
19     MOV DH,CL
20     MOV DL,COL
21     MOV BH,0H
22     INT 10H
23     POP DX
24     POP BX
25     POP AX
26 ENDM
27
28 MOVE_THERE MACRO LINE COL
29     PUSH AX
30     PUSH BX
31     PUSH DX
32     MOV AH,02H
33     MOV BP,OFFSET LINE
34     MOV DH,DS:[BP]
35     MOV BP,OFFSET COL
36     MOV DL,DS:[BP]
37     MOV BH,0H
38     INT 10H
39     POP DX
40     POP BX
41     POP AX
42 ENDM
43
44
45 INCREASE_LINE MACRO LINE COLUMN MINC
46 ; === Zero Column ===
47     MOV BP,OFFSET COLUMN
48     MOV DS:[BP],MINC
49 ; === Increase Line ===
50     MOV BP,OFFSET LINE
51     INC DS:[BP]
52
53 ENDM
54
55 SCROLL_UP MACRO LINES UPLR UPLC LORR LORC
56     PUSH AX
57     PUSH CX
58     PUSH DX
59     MOV AL,LINES
60     MOV CH,UPLR
61     MOV CL,UPLC
62     MOV DH,LORR
63     MOV DL,LORC
64     MOV AH,06H
65     INT 10H
66     POP DX
67     POP CX
68     POP AX
69 ENDM
70

```

```

71  CLEAR MACRO
72      XOR CX,CX
73      MOV DH,24
74      MOV DL,80
75      MOV BH,7
76      MOV AX,700H
77      INT 10H
78  ENDM
79
80  ; This will use INT 10/02H
81  ; to move the cursor
82  ; to the appropriate position
83  PRINT_THERE MACRO CHAR
84      PUSH AX
85      MOV AL,CHAR
86      MOV AH,0EH
87      INT 10H
88      POP AX
89  ENDM
90
91  ;This macro uses registers AH,DX
92  PRINT_STRING MACRO STRING
93      MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
94      MOV AH,09H
95      INT 21H
96  ENDM
97
98  EXIT MACRO
99      MOV AH,4CH
100     INT 21H
101  ENDM

```