



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜ&ΜΥ
Εργαστήριο Μικροϋπολογιστών

3^η Εργαστηριακή Άσκηση
Ακ. έτος 2011-2012

Ομάδα C07:

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

26 Δεκεμβρίου 2011

Άσκηση (i)

Κυρίως κώδικας:

```
1  INCLUDE MACROS.TXT
2  ;=====DATA SEGMENT=====
3  DATA SEGMENT
4      MESSAGE1 DB "GIVE A 10-BINARY NUMBER: $" ;EDW PREPEI NA MPEI META TO INPUT DE KSERW AKRIBWS PWS
5      MESSAGE2 DB "DECIMAL: $" ; EDW PREPEI NA MPEI TO OUTPUT, OUTE K EDW KSERW PWS AKRIBWS :P
6      PKEY DB "PRESS ANY KEY TO START OR Q IN ORDER TO EXIT. $"
7      NEW_LINE DB 0AH, 0DH, '$' ;OI ASCII KWDIKOI GIA ALLAGH GRAMMHS
8  ENDS
9  ;=====STACK SEGMENT=====
10 STACK SEGMENT
11     DW 128 DUP(?)
12 ENDS
13
14 ;=====CODE SEGMENT=====
15 CODE SEGMENT
16     ASSUME CS:CODE, SS:STACK, DS:DATA, ES:DATA
17
18
19 MAIN PROC FAR
20
21 ; SET SEGMENT REGISTERS:
22     MOV AX, DATA
23     MOV DS, AX
24     MOV ES, AX
25
26 START:
27     PRINT_STRING MESSAGE1 ;PRINT_STRING APO THN MACRO
28     CALL BIN_KEYBOARD ;O XRHSTHS DINEI TON BINARY
29     PUSH DX
30     PRINT_STRING NEW_LINE ;ALLAGH GRAMMHS
31     PRINT_STRING MESSAGE2
32     POP DX
33     CALL DEC_CONVERSION ;TON METATREPW SE DEKADIKO
34     PRINT_STRING NEW_LINE ;ALLAGH GRAMMHS
35     PRINT_STRING PKEY ;ODHGIES PROS TON XRHSTH GIA TO TI NA PATHSEI
36     READ ;DIABAZEI AUTO POU EDWSE O XRHSTHS
37     CMP AL, 'Q' ;AN PATHSHKE TO Q
38     JE QUIT ; TELOS PROGRAMMATOS
39     CMP AL, 'q' ;AN PATHSHKE TO Q
40     JE QUIT ; TELOS PROGRAMMATOS
41     PRINT_STRING NEW_LINE ;ALLAGH GRAMMHS
42     JMP START
43 QUIT:
44     EXIT ;APO THN MACRO
45
46
47 MAIN ENDP
48 ;=====DIABASMA TOY BINARY APO TO KEAYBOARD=====
49 BIN_KEYBOARD PROC NEAR
50     ;PUSH DX ;EPHREAZETAI APO THN MACRO PRINT
51     MOV DX, 0
52     MOV CX, 10 ;O CX EINAI DEFAULT COUNTER GIA LOOPS. SELW 10 NOUMERA NA DIABASW
53
54 IGNORE:
55     READ ;DIABAZEI XARAKTHRA APO PLHKTROLOGIO XWRIS NA TO TUPWSEI
56     CMP AL, 'Q' ;BLEPW AN EINAI Q
57     JE QUIT ;AN EINAI TOTE KANOUME EXIT ; H' ALLIWS JE ADDR2
58     CMP AL, 'q' ;BLEPW AN EINAI Q
59     JE QUIT ;AN EINAI TOTE KANOUME EXIT ; H' ALLIWS JE ADDR2
60     SHL DX,1
61     CMP AL, '0' ;ALLIWS,BLEPW AN EINAI 0
62     JE ZERO ;JUMP IF LESS APO TO 0 (TOTE TO AGNOOUME)
63     CMP AL, '1' ;ALLIWS,BLEPW AN EINAI 1
64     JE ONE ;JUMP IF GREATER APO TO 1 (TOTE TO AGNOOUME)
65     JMP IGNORE
66 ONE:
67     INC DX ;GIA NA DIABASW 10 ARI8MOUS...
68 ZERO:
69     LOOP IGNORE
```

```

70  ADDR2:
71      ;POP DX
72      RET
73
74  BIN_KEYBOARD ENDP
75
76  ;=====METATROPH K PRINT SE DEKADIKO=====
77  DEC_CONVERSION PROC NEAR
78      ;STON DX EXW TON BINARY ARI8MO MOU
79      MOV AX,DX
80      MOV DX,0
81      ;BAZW TON DX (BINARY ARI8MO MOU STON AX GIATI O AX EINAI O DEFAULT DIAIRETEOS
82      MOV BX, 1000
83      DIV BX      ; DIAIRW ME 1000
84      PRINT_NUM AL ;TO AL EXEI TO PHLIKO DHLADH THN XILIADA
85      MOV AX,DX   ; DIAIRETHS 8A GINEI TO PROHGOUMENO UPOLOIPO
86      MOV DX,0
87      MOV BX, 100 ;DIAIRW ME 100
88      DIV BL
89      PRINT_NUM AL ;TO AL 8A EXEI TO PHLIKO POU 8A NAI OI EKATONTADES
90      MOV AL,AH   ; DIAIRETHS 8A GINEI TO PROHGOUMENO UPOLOIPO
91      MOV AH,0
92      MOV BX, 10  ; DIAIRW ME 10
93      DIV BL
94      PRINT_NUM AL ; TO PHLIKO EDW EXEI TIS DEKADES
95      PRINT_NUM AH ; TO UPOLOIPO EDW EXEI TIS MONADES\
96
97      RET
98
99  DEC_CONVERSION ENDP
100 ;=====
101 CODE ENDS
102
103 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  ;This macro changes registers AH,DL
8  PRINT MACRO CHAR
9      PUSH AX
10     PUSH DX
11     MOV DL,CHAR
12     MOV AH,02H
13     INT 21H
14     POP DX
15     POP AX
16 ENDM
17
18 ;This macro change registers AH,DX
19 PRINT_STRING MACRO STRING
20     PUSH AX
21     PUSH DX
22     MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
23     MOV AH,09H
24     INT 21H
25     POP DX
26     POP AX
27 ENDM
28
29 PRINT_NUM MACRO CHAR
30     PUSH DX
31     PUSH AX
32     MOV DL, CHAR
33     ADD DL, 30H
34     MOV AH, 2
35     INT 21H
36     POP AX
37     POP DX
38 ENDM
39

```

```

40 PAUSE MACRO
41     PUSH AX
42     PUSH DX
43     LEA DX,PKEY           ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
44     MOV AH,9
45     INT 21H               ;OUTPUT STRING AT DS:DX
46     MOV AH,8              ;WAIT FOR PRESSING OF A KEY
47     INT 21H               ;WITHOUT ECHO->8
48     PRINT OAH
49     PRINT ODH
50     POP DX
51     POP AX
52 ENDM
53
54 EXIT MACRO
55     MOV AH,4CH
56     INT 21H
57 ENDM

```

Άσκηση (ii)

Κυρίως κώδικας:

```

1  INCLUDE MACROS.TXT
2
3  DATA SEGMENT
4      ; ADD YOUR DATA HERE!
5      PKEY DB "INSERT 4 DECIMAL NUMS AND THEN <ENTER>...$"
6      MESSAGE1 DB "GIVE FOUR NUMBERS: $"
7      MESSAGE2 DB "HEX = $"
8      NEW_LINE DB OAH, ODH, '$' ;OI ASCII KWDIKOI GIA ALLAGH GRAMMHS
9  ENDS
10
11 STACK SEGMENT
12     DW 128 DUP(?)
13 ENDS
14
15 CODE SEGMENT
16
17 MAIN PROC FAR
18 ; SET SEGMENT REGISTERS:
19     MOV AX, DATA
20     MOV DS, AX
21     MOV ES, AX
22
23 START:
24
25     PRINT_STRING MESSAGE1
26     CALL DEC_KEYBOARD ; KATEUSEIAN ME TO READ TA BAZEIS STON BP A8ROIZONTAS
27     PRINT_STRING NEW_LINE
28 BCK:
29     READ
30     CMP AL,ODH ;koita gia enter
31     JE CNT
32     CMP AL,'Q'
33     JE QUIT
34     CMP AL,'q'
35     JE QUIT
36     JMP BCK
37
38 CNT:
39     PRINT_STRING MESSAGE2
40     ;TUPWSE TA HEX TOU 16BITOU BP
41     CALL DIGITS_TO_HEXS
42     PRINT_STRING NEW_LINE
43     JMP START
44
45
46 QUIT:
47     EXIT
48
49 MAIN ENDP
50
51
52 ;=====PROCEDURES=====

```

```

53 DEC_KEYBOARD PROC NEAR
54     MOV DX, 0
55     MOV CX, 4 ;GIATI 8A DEXTW 4 ARISMOUS
56
57     IGNORE:
58     READ
59     CMP AL,'Q'
60     JE QUIT
61     CMP AL,'q'
62     JE QUIT
63     CMP AL, '0'
64     JL IGNORE
65     CMP AL, '9'
66     JG IGNORE
67     SUB AL, 30H ;DEN EIMAI SIGOURH EDW
68     MOV BL,AL ;APO8UKEUSE TO TREXON STON BL
69     MOV BH,0
70     MOV AX,DX ; FORTWSE TO PROHGOU MENO A8ROISMA APO DX
71     MOV DX,10 ;BALE STO DL 10
72     MUL DX ;AX=AX*10
73     ADD AX,BX ;+BL
74     MOV DX,AX ;KAI BALTO PALI STO DX
75     LOOP IGNORE
76     MOV BP,DX
77     RET
78
79 DEC_KEYBOARD ENDP
80
81
82
83 ;=====MAKE 16 BITS TO HEX=====
84
85 DIGITS_TO_HEXS PROC NEAR
86     MOV BX, BP
87     MOV BL, BH ;APOMONWNW TA 4 MSB
88     SHR BL, 4 ;OLIS8HSE TA STIS 4 LEAST SIGNIF 8ESEIS
89     CALL PRINT_HEX
90     MOV BX, BP
91     MOV BL,BH
92     AND BL, 0FH
93     CALL PRINT_HEX
94     MOV BX, BP
95     AND BL, 0F0H
96     SHR BL, 4
97     CALL PRINT_HEX
98     MOV BX, BP
99     AND BL, 0FH
100    CALL PRINT_HEX
101
102
103    RET
104 DIGITS_TO_HEXS ENDP
105
106 PRINT_HEX PROC NEAR
107     CMP BL,9 ;AN 0 ARIS8MOS EINAI METAKSU 0 K 9 PROS8ETW 30H
108     JG ADDR1
109     ADD BL, 30H
110     JMP ADDR2
111
112 ADDR1:
113     ADD BL, 37H ;DIAFORETIKA PROS8ETW 37H ('A' = 41H)
114 ADDR2:
115     PRINT BL
116     RET
117
118 PRINT_HEX ENDP
119 ;=====END OF MAKE 16 BITS TO HEX=====
120
121
122 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  ;This macro changes registers AH,DL
8  PRINT MACRO CHAR
9      PUSH AX
10     PUSH DX
11     MOV DL,CHAR
12     MOV AH,02H
13     INT 21H
14     POP DX
15     POP AX
16 ENDM
17
18 ;This macro change registers AH,DX
19 PRINT_STRING MACRO STRING
20     PUSH AX
21     PUSH DX
22     MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
23     MOV AH,09H
24     INT 21H
25     POP DX
26     POP AX
27 ENDM
28
29 PRINT_NUM MACRO CHAR
30     MOV DL, CHAR
31     ADD DL, 30H
32     MOV AH, 2
33     INT 21H
34 ENDM
35
36 PAUSE MACRO
37     PUSH AX
38     PUSH DX
39     LEA DX,PKEY           ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
40     MOV AH,9
41     INT 21H              ;OUTPUT STRING AT DS:DX
42     MOV AH,8             ;WAIT FOR PRESSING OF A KEY
43     INT 21H              ;WITHOUT ECHO->8
44     PRINT 0AH
45     PRINT 0DH
46     POP DX
47     POP AX
48 ENDM
49
50 EXIT MACRO
51     MOV AH,4CH
52     INT 21H
53 ENDM
54
55 GETHON MACRO R
56     CALL GETHEX
57     MOV R,AX
58     CALL GETHEX
59     SHL R,4
60     OR R,AX
61     CALL GETHEX
62     SHL R,4
63     OR R,AX
64     CALL GETHEX
65     SHL R,4
66     OR R,AX
67 ENDM

```

Άσκηση (iii)

Κυρίως κώδικας:

```

1  INCLUDE MACROS.TXT
2
3  STACK_SEG SEGMENT STACK
4      DW 128 DUP(?)
5  ENDS
6

```

```

7
8 DATA_SEG SEGMENT
9     MSG DB "GIMME <=20 CHARS END PRESS RETURN '/' TO QUIT",0AH,0DH,"$"
10     SPACE DB " "
11     LINE DB 0AH,0DH,"$"
12     NUMS DB 20 DUP("$")
13     NCNT DW 0
14     LOWC DB 20 DUP("$")
15     LCNT DW 0
16     UPRC DB 20 DUP("$")
17     UCNT DW 0
18
19
20 ENDS
21
22 CODE_SEG SEGMENT
23     ASSUME CS:CODE_SEG,SS:STACK_SEG,DS:DATA_SEG,ES:DATA_SEG
24
25 MAIN PROC FAR
26     ;FOR SEGMENT REGISTERS
27     MOV AX,DATA_SEG
28     MOV DS,AX
29     MOV ES,AX
30
31 START:
32     PRINT_STRING MSG
33     MOV DX,0
34     MOV BX,0
35     CALL GET_INPUT
36
37 CNT:
38     PRINT_STRING NUMS
39     PRINT SPACE
40     PRINT_STRING LOWC
41     PRINT SPACE
42     PRINT_STRING UPRC
43     PRINT_STRING LINE
44     JMP START
45
46 EX:
47     EXIT
48 MAIN ENDP
49
50 GET_INPUT PROC NEAR
51     MOV DX,0
52     MOV CX,20
53 READL:
54     READ
55     CMP AL,0DH
56     JE CNT
57     CMP AL,'/'
58     JE EX
59     CMP AL,30H ;0
60     JL READL
61     CMP AL,40H ;9+1
62     JL NUMBERS
63     CMP AL,41H ;A
64     JL READL
65     CMP AL,5BH ;Z+1
66     JL ULETTER
67     CMP AL,61H ;a
68     JL READL
69     CMP AL,7BH ;z+1
70     JL LLETTER
71     JMP READL
72 NUMBERS:
73     MOV BX,OFFSET NUMS
74     ADD BX,NCNT
75     MOV [BX],AL
76     INC NCNT
77     LOOP READL
78     RET
79 LLETTER:
80     MOV BX,OFFSET LOWC
81     ADD BX,LCNT

```

```

82     MOV [BX],AL
83     INC LCNT
84     LOOP READL
85     RET
86 ULETTER:
87     MOV BX,OFFSET UPRC
88     ADD BX,UCNT
89     MOV [BX],AL
90     INC UCNT
91     LOOP READL
92     RET
93 GET_INPUT ENDP
94
95
96 CODE_SEG ENDS
97
98 END MAIN

```

Τα macros που χρησιμοποιήσαμε:

```

1  ;This macro change registers AH,AL
2  READ MACRO
3      MOV AH,8
4      INT 21H
5  ENDM
6
7  ;This macro changes registers AH,DL
8  PRINT MACRO CHAR
9      PUSH AX
10     PUSH DX
11     MOV DL,CHAR
12     MOV AH,02H
13     INT 21H
14     POP DX
15     POP AX
16 ENDM
17
18 ;This macro change registers AH,DX
19 PRINT_STRING MACRO STRING
20     PUSH AX
21     PUSH DX
22     MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
23     MOV AH,09H
24     INT 21H
25     POP DX
26     POP AX
27 ENDM
28
29 PRINT_NUM MACRO CHAR
30     MOV DL, CHAR
31     ADD DL, 30H
32     MOV AH, 2
33     INT 21H
34 ENDM
35
36 PAUSE MACRO
37     PUSH AX
38     PUSH DX
39     LEA DX,PKEY           ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
40     MOV AH,9
41     INT 21H              ;OUTPUT STRING AT DS:DX
42     MOV AH,8             ;WAIT FOR PRESSING OF A KEY
43     INT 21H              ;WITHOUT ECHO->8
44     PRINT 0AH
45     PRINT 0DH
46     POP DX
47     POP AX
48 ENDM
49
50 EXIT MACRO
51     MOV AH,4CH
52     INT 21H
53 ENDM

```


Άσκηση (iv)

Κυρίως κώδικας:

```
1  INCLUDE MACROS.TXT
2
3  STACK_SEG SEGMENT STACK
4      DW 128 DUP(?)
5  ENDS
6
7
8  DATA_SEG SEGMENT
9      FIRST DB "First number: $"
10     SECOND DB "Second number: $"
11     SPACE DB " "
12     LINE DB 0AH,0DH,"$"
13
14
15  ENDS
16
17  CODE_SEG SEGMENT
18      ASSUME CS:CODE_SEG,SS:STACK_SEG,DS:DATA_SEG,ES:DATA_SEG
19
20  MAIN PROC FAR
21      MOV AX,DATA_SEG
22      MOV DS,AX
23      MOV ES,AX
24      CALL GET_INPUT
25      MOV AX,BX
26      MOV DX,0
27      MUL SI
28      MOV BP,AX
29      PUSH BP
30      PUSH CX
31      MOV CX,DX
32      MOV AX,BX
33      MOV DX,0
34      MUL DI
35      ADD AX,CX
36      JNC NOTOVF1
37      INC DX
38  NOTOVF1:
39      POP CX
40      MOV BX,DX
41      PUSH BX
42      MOV BX,AX
43      MOV AX,CX
44      MOV DX,0
45      MUL SI
46      ADD AX,BX
47      JNC NOTOVF2
48      INC DX
49  NOTOVF2:
50      POP BX
51      MOV BP,AX    ;2ND DIGIT
52      PUSH BP
53      MOV AX,CX
54      MOV CX,DX    ; REALLY??
55      MOV DX,0
56      MUL DI
57      ADD AX,BX
58      JNC NOTOVF3
59      INC DX
60  NOTOVF3:
61      ADD AX,CX
62      JNC NOTOVF4
63      INC DX
64  NOTOVF4:
65      MOV BP,AX
66      PUSH BP
67      MOV BP,DX
68      ;now BP has the answer
69      CALL DIGITS_TO_HEX
70      POP BP
71      CALL DIGITS_TO_HEX
72      POP BP
```

```

73     CALL DIGITS_TO_HEX5
74     POP BP
75     CALL DIGITS_TO_HEX5
76
77
78
79     EXIT
80 MAIN ENDP
81
82 GET_INPUT PROC NEAR
83     PRINT_STRING FIRST
84     GETHON CX
85     GETHON BX
86     PRINT_STRING LINE
87     PRINT_STRING SECOND
88     GETHON DI
89     GETHON SI
90     PRINT_STRING LINE
91     RET
92 GET_INPUT ENDP
93
94
95 GETHEX PROC NEAR
96 R:  READ
97     MOV AH,0
98     CMP AL,30H ;0
99     JL R
100    CMP AL,40H ;9+1
101    JL NUM
102    CMP AL,41H ;A
103    JL R
104    CMP AL,47H ;F+1
105    JL CAPS
106    CMP AL,61H ;a
107    JL R
108    CMP AL,67H ;f+1
109    JL SMALL
110    JMP R
111 NUM:
112     SUB AL,30H
113     RET
114 CAPS:
115     SUB AL,37H
116     RET
117 SMALL:
118     SUB AL,57H
119     RET
120
121 GETHEX ENDP
122
123
124 ;=====MAKE 16 BITS TO HEX=====
125
126 DIGITS_TO_HEX5 PROC NEAR
127     MOV BX, BP
128     MOV BL, BH ;APOMONWNN TA 4 MSB
129     SHR BL, 4 ;OLIS8HSE TA STIS 4 LEAST SIGNIF 8ESEIS
130     CALL PRINT_HEX
131     MOV BX, BP
132     MOV BL,BH
133     AND BL, 0FH
134     CALL PRINT_HEX
135     MOV BX, BP
136     AND BL, 0FOH
137     SHR BL, 4
138     CALL PRINT_HEX
139     MOV BX, BP
140     AND BL, 0FH
141     CALL PRINT_HEX
142
143
144     RET
145 DIGITS_TO_HEX5 ENDP
146
147 PRINT_HEX PROC NEAR

```

```

148     CMP BL,9 ;AN O ARI8MOS EINAI METAKSU O K 9 PROS8ETW 30H
149     JG ADDR1
150     ADD BL, 30H
151     JMP ADDR2
152
153 ADDR1:
154     ADD BL, 37H ;DIAFORETIKA PROS8ETW 37H ('A' = 41H)
155 ADDR2:
156     PRINT BL
157     RET
158
159 PRINT_HEX ENDP
160 ;=====END OF MAKE 16 BITS TO HEX=====
161
162
163
164
165
166
167
168 CODE_SEG ENDS
169
170 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  ;This macro changes registers AH,DL
8  PRINT MACRO CHAR
9      PUSH AX
10     PUSH DX
11     MOV DL,CHAR
12     MOV AH,02H
13     INT 21H
14     POP DX
15     POP AX
16 ENDM
17
18 ;This macro change registers AH,DX
19 PRINT_STRING MACRO STRING
20     PUSH AX
21     PUSH DX
22     MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
23     MOV AH,09H
24     INT 21H
25     POP DX
26     POP AX
27 ENDM
28
29 PRINT_NUM MACRO CHAR
30     MOV DL, CHAR
31     ADD DL, 30H
32     MOV AH, 2
33     INT 21H
34 ENDM
35
36 PAUSE MACRO
37     PUSH AX
38     PUSH DX
39     LEA DX,PKEY           ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
40     MOV AH,9
41     INT 21H               ;OUTPUT STRING AT DS:DX
42     MOV AH,8              ;WAIT FOR PRESSING OF A KEY
43     INT 21H               ;WITHOUT ECHO->8
44     PRINT OAH
45     PRINT ODH
46     POP DX
47     POP AX
48 ENDM
49
50 EXIT MACRO

```

```
51         MOV AH,4CH
52         INT 21H
53     ENDM
54
55     GETHON MACRO R
56         CALL GETHEX
57         MOV R,AX
58         CALL GETHEX
59         SHL R,4
60         OR R,AX
61         CALL GETHEX
62         SHL R,4
63         OR R,AX
64         CALL GETHEX
65         SHL R,4
66         OR R,AX
67     ENDM
```