# 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:

```
INCLUDE MACROS.TXT
;============DATA SEGMENT===============
DATA SEGMENT
    MESSAGE1 DB "GIVE A 10-BINARY NUMBER: $" ;EDW PREPEI NA MPEI META TO INPUT DE KSERW AKRIBWS PWS
    MESSAGE2 DB "DECIMAL: $"  ; EDW PREPEI NA MPEI TO OUTPUT, OUTE K EDW KSERW PWS AKRIBWS :P
    PKEY DB "PRESS ANY KEY TO START OR Q IN ORDER TO EXIT. $"
    NEW_LINE DB 0AH, 0DH, '$'    ;OI ASCII KWDIKOI GIA ALLAGH GRAMMHS
ENDS
;============STACK SEGMENT==============
STACK SEGMENT
    DW 128 DUP(?)
ENDS

;============CODE SEGMENT===============
CODE SEGMENT
    ASSUME CS:CODE, SS:STACK, DS:DATA, ES:DATA


MAIN PROC FAR

; SET SEGMENT REGISTERS:
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX

START:
    PRINT_STRING MESSAGE1       ;PRINT_STRING APO THN MACRO
    CALL BIN_KEYBOARD           ;O XRHSTHS DINEI TON BINARY
    PUSH DX
    PRINT_STRING NEW_LINE       ;ALLAGH GRAMMHS
    PRINT_STRING MESSAGE2
    POP DX
    CALL DEC_CONVERTION         ;TON METATREPW SE DEKADIKO
    PRINT_STRING NEW_LINE       ;ALLAGH GRAMMHS
    PRINT_STRING PKEY           ;ODHGIES PROS TON XRHSTH GIA TO TI NA PATHSEI
    READ                        ;DIABAZEI AUTO POU EDWSE O XRHSTHS
    CMP AL, 'Q'                 ;AN PATH8HKE TO Q
    JE QUIT                     ; TELOS PROGRAMMATOS
    CMP AL, 'q'                 ;AN PATH8HKE TO Q
    JE QUIT                     ; TELOS PROGRAMMATOS
    PRINT_STRING NEW_LINE       ;ALLAGH GRAMMHS
    JMP START
QUIT:
    EXIT                        ;APO THN MACRO


MAIN ENDP
;========DIABASMA TOY BINARY APO TO KEAYBOARD============
BIN_KEYBOARD  PROC NEAR
    ;PUSH  DX ;EPHREAZETAI APO THN MACRO PRINT
    MOV DX, 0
    MOV CX, 10   ;O CX EINAI DEFAULT COUNTER GIA LOOPS. 8ELW 10 NOUMERA NA DIABASW

IGNORE:
    READ         ;DIABAZEI XARAKTHRA APO PLHKTROLOGIO XWRIS NA TO TUPWSEI
    CMP AL, 'Q'  ;BLEPW AN EINAI Q
    JE QUIT      ;AN EINAI TOTE KANOUME EXIT   ; H' ALLIWS JE ADDR2
    CMP AL, 'q'  ;BLEPW AN EINAI Q
    JE QUIT      ;AN EINAI TOTE KANOUME EXIT   ; H' ALLIWS JE ADDR2
    SHL DX,1
    CMP AL, '0'  ;ALLIWS,BLEPW AN EINAI 0
    JE ZERO    ;JUMP IF LESS APO TO 0 (TOTE TO AGNOOUME)
    CMP AL, '1'  ;ALLIWS,BLEPW AN EINAI 1
    JE ONE    ;JUMP IF GREATER APO TO 1 (TOTE TO AGNOOUME)
    JMP IGNORE
ONE:
    INC DX   ;GIA NA DIABASW 10 ARI8MOUS...
ZERO:
    LOOP IGNORE
```

```
ADDR2:
    ;POP DX
    RET

BIN_KEYBOARD ENDP

;=========METATROPH K PRINT SE DEKADIKO==================
DEC_CONVERTION PROC NEAR
    ;STON DX EXW TON BINARY ARI8MO MOU
    MOV AX,DX
    MOV DX,0
    ;BAZW TON DX (BINARY ARI8MO MOU STON AX GIATI O AX EINAI O DEFAULT DIAIRETEOS
    MOV BX, 1000
    DIV BX       ; DIAIRW ME 1000
    PRINT_NUM AL ;TO AL EXEI TO PHLIKO DHLADH THN XILIADA
    MOV AX,DX    ; DIAIRETHS 8A GINEI TO PROHGOUMENO UPOLOIPO
    MOV DX,0
    MOV BX, 100  ;DIAIRW ME 100
    DIV BL
    PRINT_NUM AL ;TO AL 8A EXEI TO PHLIKO POU 8A NAI OI EKATONTADES
    MOV AL,AH    ; DIAIRETHS 8A GINEI TO PROHGOUMENO UPOLOIPO
    MOV AH,0
    MOV BX, 10   ; DIAIRW ME 10
    DIV BL
    PRINT_NUM AL ; TO PHLIKO EDW EXEI TIS DEKADES
    PRINT_NUM AH ; TO UPOLOIPO EDW EXEI TIS MONADES\

    RET

DEC_CONVERTION ENDP
;=========================================================
CODE ENDS

END MAIN
```

Τα macros που χρησιμοποιήσαμε:

```
;This macro change registers AH,AL
READ MACRO
    MOV AH,1
    INT 21H
ENDM

;This macro changes registers AH,DL
PRINT MACRO CHAR
        PUSH AX
        PUSH DX
        MOV DL,CHAR
        MOV AH,02H
        INT 21H
        POP DX
        POP AX
ENDM

;This macro change registers AH,DX
PRINT_STRING MACRO STRING
        PUSH AX
        PUSH DX
        MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
        MOV AH,09H
        INT 21H
        POP DX
        POP AX
ENDM

PRINT_NUM MACRO CHAR
    PUSH DX
    PUSH AX
        MOV DL, CHAR
        ADD DL, 30H
        MOV AH, 2
        INT 21H
    POP AX
    POP DX
ENDM

```

```
PAUSE MACRO
    PUSH AX
    PUSH DX
    LEA DX,PKEY         ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
    MOV AH,9
    INT 21H             ;OUTPUT STRING AT DS:DX
    MOV AH,8            ;WAIT FOR PRESSING OF A KEY
    INT 21H             ;WITHOUT ECHO->8
    PRINT 0AH
    PRINT 0DH
    POP DX
    POP AX
ENDM

EXIT MACRO
        MOV AH,4CH
        INT 21H
ENDM
```

## Άσκηση (ii)

Κυρίως κώδικας:

```
INCLUDE MACROS.TXT

DATA SEGMENT
    ; ADD YOUR DATA HERE!
    PKEY DB "INSERT 4 DECIMAL NUMS AND THEN <ENTER>...$"
    MESSAGE1 DB "GIVE FOUR NUMBERS: $"
    MESSAGE2 DB "HEX = $"
    NEW_LINE DB 0AH, 0DH, '$' ;OI ASCII KWDIKOI GIA ALLAGH GRAMMHS
ENDS

STACK SEGMENT
    DW   128  DUP(?)
ENDS

CODE SEGMENT

MAIN PROC FAR
; SET SEGMENT REGISTERS:
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX

START:

    PRINT_STRING MESSAGE1
    CALL DEC_KEYBOARD   ; KATEU8EIAN ME TO READ TA BAZEIS STON BP A8ROIZONTAS
    PRINT_STRING NEW_LINE
BCK:
    READ
    CMP AL,0DH    ;koita gia enter
    JE CNT
    CMP AL,'Q'
    JE QUIT
    CMP AL,'q'
    JE QUIT
    JMP BCK

CNT:
    PRINT_STRING MESSAGE2
    ;TUPWSE TA HEX TOU 16BITOU BP
    CALL DIGITS_TO_HEXS
    PRINT_STRING NEW_LINE
    JMP START


QUIT:
    EXIT

MAIN ENDP


;==========PROCEDURES=============
```

```
DEC_KEYBOARD PROC NEAR
    MOV DX, 0
    MOV CX, 4 ;GIATI 8A DEXTW 4 ARI8MOUS

IGNORE:
    READ
    CMP AL,'Q'
    JE QUIT
    CMP AL,'q'
    JE QUIT
    CMP AL, '0'
    JL IGNORE
    CMP AL, '9'
    JG IGNORE
    SUB AL, 30H ;DEN EIMAI SIGOURH EDW
    MOV BL,AL   ;APO8UKEUSE TO TREXON STON BL
    MOV BH,0
    MOV AX,DX   ; FORTWSE TO PROHGOUMENO A8ROISMA APO DX
    MOV DX,10   ;BALE STO DL 10
    MUL DX      ;AX=AX*10
    ADD AX,BX   ;+BL
    MOV DX,AX   ;KAI BALTO PALI STO DX
    LOOP IGNORE
    MOV BP,DX
    RET

DEC_KEYBOARD ENDP



;======MAKE 16 BITS TO HEX=======

DIGITS_TO_HEXS PROC NEAR
     MOV BX, BP
     MOV BL, BH ;APOMONWNW TA 4 MSB
     SHR BL, 4  ;OLIS8HSE TA STIS 4 LEAST SIGNIF 8ESEIS
     CALL PRINT_HEX
     MOV BX, BP
     MOV BL,BH
     AND BL, 0FH
     CALL PRINT_HEX
     MOV BX, BP
     AND BL, 0F0H
     SHR BL, 4
     CALL PRINT_HEX
     MOV BX, BP
     AND BL, 0FH
     CALL PRINT_HEX


     RET
DIGITS_TO_HEXS ENDP

PRINT_HEX PROC NEAR
    CMP BL,9 ;AN O ARI8MOS EINAI METAKSU 0 K 9 PROS8ETW 30H
    JG ADDR1
    ADD BL, 30H
    JMP ADDR2

ADDR1:
    ADD BL, 37H ;DIAFORETIKA PROS8ETW 37H ('A' = 41H)
ADDR2:
    PRINT BL
    RET

PRINT_HEX ENDP
;======END OF MAKE 16 BITS TO HEX========


END MAIN
```

Τα macros που χρησιμοποιήσαμε:

```
;This macro change registers AH,AL
READ MACRO
    MOV AH,1
```

```
    INT 21H
ENDM

;This macro changes registers AH,DL
PRINT MACRO CHAR
        PUSH AX
        PUSH DX
        MOV DL,CHAR
        MOV AH,02H
        INT 21H
        POP DX
        POP AX
ENDM

;This macro change registers AH,DX
PRINT_STRING MACRO STRING
        PUSH AX
        PUSH DX
        MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
        MOV AH,09H
        INT 21H
        POP DX
        POP AX
ENDM

PRINT_NUM MACRO CHAR
        MOV DL, CHAR
        ADD DL, 30H
        MOV AH, 2
        INT 21H
ENDM

PAUSE MACRO
    PUSH AX
    PUSH DX
    LEA DX,PKEY        ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
    MOV AH,9
    INT 21H            ;OUTPUT STRING AT DS:DX
    MOV AH,8           ;WAIT FOR PRESSING OF A KEY
    INT 21H            ;WITHOUT ECHO->8
    PRINT 0AH
    PRINT 0DH
    POP DX
    POP AX
ENDM

EXIT MACRO
        MOV AH,4CH
        INT 21H
ENDM

GETHON MACRO R
    CALL GETHEX
    MOV R,AX
    CALL GETHEX
    SHL R,4
    OR R,AX
    CALL GETHEX
    SHL R,4
    OR R,AX
    CALL GETHEX
    SHL R,4
    OR R,AX
ENDM
```

## Άσκηση (iii)

Κυρίως κώδικας:

```
INCLUDE MACROS.TXT

STACK_SEG SEGMENT STACK
    DW 128 DUP(?)
ENDS

```

```
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  CNT:
37      PRINT_STRING NUMS
38      PRINT SPACE
39      PRINT_STRING LOWC
40      PRINT SPACE
41      PRINT_STRING UPRC
42      PRINT_STRING LINE
43      JMP START
44
45  EX:
46      EXIT
47  MAIN ENDP
48
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
```

```
    MOV [BX] ,AL
    INC LCNT
    LOOP READL
    RET
ULETTER:
    MOV BX,OFFSET UPRC
    ADD BX,UCNT
    MOV [BX] ,AL
    INC UCNT
    LOOP READL
    RET
GET_INPUT ENDP


CODE_SEG ENDS

END MAIN
```

Τα macros που χρησιμοποιήσαμε:

```
;This macro change registers AH,AL
READ MACRO
    MOV AH,8
    INT 21H
ENDM

;This macro changes registers AH,DL
PRINT MACRO CHAR
        PUSH AX
        PUSH DX
        MOV DL,CHAR
        MOV AH,02H
        INT 21H
        POP DX
        POP AX
ENDM

;This macro change registers AH,DX
PRINT_STRING MACRO STRING
        PUSH AX
        PUSH DX
        MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
        MOV AH,09H
        INT 21H
        POP DX
        POP AX
ENDM

PRINT_NUM MACRO CHAR
        MOV DL, CHAR
        ADD DL, 30H
        MOV AH, 2
        INT 21H
ENDM

PAUSE MACRO
    PUSH AX
    PUSH DX
    LEA DX,PKEY        ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
    MOV AH,9
    INT 21H            ;OUTPUT STRING AT DS:DX
    MOV AH,8           ;WAIT FOR PRESSING OF A KEY
    INT 21H            ;WITHOUT ECHO->8
    PRINT 0AH
    PRINT 0DH
    POP DX
    POP AX
ENDM

EXIT MACRO
        MOV AH,4CH
        INT 21H
ENDM
```

## Άσκηση (iv)

Κυρίως κώδικας:

```
INCLUDE MACROS.TXT

STACK_SEG SEGMENT STACK
    DW 128 DUP(?)
ENDS


DATA_SEG SEGMENT
    FIRST DB "First number: $"
    SECOND DB "Second number: $"
    SPACE DB " "
    LINE DB 0AH,0DH,"$"


ENDS

CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG,SS:STACK_SEG,DS:DATA_SEG,ES:DATA_SEG

MAIN PROC FAR
    MOV AX,DATA_SEG
    MOV DS,AX
    MOV ES,AX
    CALL GET_INPUT
    MOV AX,BX
    MOV DX,0
    MUL SI
    MOV BP,AX
    PUSH BP
    PUSH CX
    MOV CX,DX
    MOV AX,BX
    MOV DX,0
    MUL DI
    ADD AX,CX
    JNC NOTOVF1
    INC DX
NOTOVF1:
    POP CX
    MOV BX,DX
    PUSH BX
    MOV BX,AX
    MOV AX,CX
    MOV DX,0
    MUL SI
    ADD AX,BX
    JNC NOTOVF2
    INC DX
NOTOVF2:
    POP BX
    MOV BP,AX    ;2ND DIGIT
    PUSH BP
    MOV AX,CX
    MOV CX,DX  ; REALLY??
    MOV DX,0
    MUL DI
    ADD AX,BX
    JNC NOTOVF3
    INC DX
NOTOVF3:
    ADD AX,CX
    JNC NOTOVF4
    INC DX
NOTOVF4:
    MOV BP,AX
    PUSH BP
    MOV BP,DX
    ;now BP has the answer
    CALL DIGITS_TO_HEXS
    POP BP
    CALL DIGITS_TO_HEXS
    POP BP
```

```
    CALL DIGITS_TO_HEXS
    POP BP
    CALL DIGITS_TO_HEXS



    EXIT
MAIN ENDP

GET_INPUT PROC NEAR
    PRINT_STRING FIRST
    GETHON CX
    GETHON BX
    PRINT_STRING LINE
    PRINT_STRING SECOND
    GETHON DI
    GETHON SI
    PRINT_STRING LINE
    RET
GET_INPUT ENDP


GETHEX PROC NEAR
R:  READ
    MOV AH,0
    CMP AL,30H ;0
    JL R
    CMP AL,40H ;9+1
    JL NUM
    CMP AL,41H ;A
    JL R
    CMP AL,47H ;F+1
    JL CAPS
    CMP AL,61H ;a
    JL  R
    CMP AL,67H ;f+1
    JL  SMALL
    JMP R
NUM:
    SUB AL,30H
    RET
CAPS:
    SUB AL,37H
    RET
SMALL:
    SUB AL,57H
    RET

GETHEX ENDP


;======MAKE 16 BITS TO HEX=======

DIGITS_TO_HEXS PROC NEAR
     MOV BX, BP
     MOV BL, BH ;APOMONWNW TA 4 MSB
     SHR BL, 4  ;OLIS8HSE TA STIS 4 LEAST SIGNIF 8ESEIS
     CALL PRINT_HEX
     MOV BX, BP
     MOV BL,BH
     AND BL, 0FH
     CALL PRINT_HEX
     MOV BX, BP
     AND BL, 0F0H
     SHR BL, 4
     CALL PRINT_HEX
     MOV BX, BP
     AND BL, 0FH
     CALL PRINT_HEX


     RET
DIGITS_TO_HEXS ENDP

PRINT_HEX PROC NEAR
```

```
    CMP BL,9 ;AN O ARI8MOS EINAI METAKSU 0 K 9 PROS8ETW 30H
    JG ADDR1
    ADD BL, 30H
    JMP ADDR2

ADDR1:
    ADD BL, 37H ;DIAFORETIKA PROS8ETW 37H ('A' = 41H)
ADDR2:
    PRINT BL
    RET

PRINT_HEX ENDP
;======END OF MAKE 16 BITS TO HEX========




CODE_SEG ENDS

END MAIN
```

Τα macros που χρησιμοποιήσαμε:

```
;This macro change registers AH,AL
READ MACRO
    MOV AH,1
    INT 21H
ENDM

;This macro changes registers AH,DL
PRINT MACRO CHAR
        PUSH AX
        PUSH DX
        MOV DL,CHAR
        MOV AH,02H
        INT 21H
        POP DX
        POP AX
ENDM

;This macro change registers AH,DX
PRINT_STRING MACRO STRING
        PUSH AX
        PUSH DX
        MOV DX,OFFSET STRING ;Assume that string is a variable or constant, NOT an address
        MOV AH,09H
        INT 21H
        POP DX
        POP AX
ENDM

PRINT_NUM MACRO CHAR
        MOV DL, CHAR
        ADD DL, 30H
        MOV AH, 2
        INT 21H
ENDM

PAUSE MACRO
    PUSH AX
    PUSH DX
    LEA DX,PKEY        ;<=>MOV DX, OFFSET PKEY;GIVES THE OFFSET OF PKEY TO DX
    MOV AH,9
    INT 21H            ;OUTPUT STRING AT DS:DX
    MOV AH,8           ;WAIT FOR PRESSING OF A KEY
    INT 21H            ;WITHOUT ECHO->8
    PRINT 0AH
    PRINT 0DH
    POP DX
    POP AX
ENDM

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
```