

Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт
по лабораторной работе № 2
на тему:
“Условные переходы, циклы, сдвиги”
по дисциплине “Организация ЭВМ и Систем”
Вариант 4

Выполнил студент гр. 4306: Табаков А.В.
Принял: Манирагена Валенс

Цель

Ознакомиться с условными операторами, циклами, сдвигами на языке ассемблера intel 8086. Исследовать работу, удалённого участка программы в оперативной памяти, копированием участка кода в другое местоположение.

Текст программы

.Model tiny

.Data

```
greeting db "This programm count 1 bits in number from 0 to k bit", 0dh, 0ah, "$"
help db "Please, input natural numbers", 0dh, 0ah,
      db "Num is a number where we count 1 bits", 0dh, 0ah,
      db "K is a count of bits from 0", 0dh, 0ah, "$"
textNum db "Please input Num from 0 to 65535", 0dh, 0ah, "Num = $"
textK db "Please input K from 1 to 16", 0dh, 0ah, "K = $"
textRes db "1 bits in $"
equSym db " = $"
endl db 0ah, 0dh, "$"
pkey db "Press any key...$"
arr dw 4 dup(?)
```

```
buffer db 6      ;max num with 5 symbols
blength db ?
bconteg:      ;consistance of buf is over of prog
      hexstring equ bconteg
```

.Stack 0100h

.Code

start:

```
mov ax, @data
mov ds, ax
mov es, ax
```

call setDisp

```
lea dx, greeting ;greeting message
mov ah, 09h
int 21h
lea dx, help      ;help message
mov ah, 09h
int 21h
```

```
lea dx, textNum   ;Num=
mov ah, 09h
int 21h
```

```
call input        ;input Num
mov arr, ax
```

```

call endlp

lea dx, textK    ;K=
mov ah, 09h
int 21h

call input      ;input K
mov arr+2, ax
call endlp

; Main code
mov cx, arr+2
mov ax, arr
xor bx, bx
loopPoint:
test ax, 01h
jz endLoopPoint
inc bx
endLoopPoint:
shr ax, 1b
loop loopPoint
;*****
mov ax, bx
printing:
push ax
lea dx, textRes  ;1 bits in
mov ah, 09h
int 21h
mov ax, arr
call print
lea dx, equSym   ; =
mov ah, 09h
int 21h
pop ax
call print      ;output result
call endlp

call quit

proc setDisp
xor dx,dx      ;cursor's position
mov ah,02h     ;set at (0,0)
int 10h
mov bl,00001010b ;colors green on black
mov cx,25*80    ;count of simbols on display
mov ax,0920h    ;printing 25*80 spaces
int 10h
ret

```

endp

proc quit

lea dx, pkey

mov ah, 9h

int 21h ; output string at ds:dx

; wait for any key....

mov ah, 1h

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

endp

proc endlp ;press enter

push dx

push ax

lea dx, endl

mov ah, 09h

int 21h

pop ax

pop dx

ret

endp

proc input

lea dx,buffer ;buffer's address

mov ah,0ah ;write in buffer

int 21h

;from string to bin

xor di,di ;start of buffer

xor ax,ax ;clear ax

mov cl,blength

xor ch,ch

xor bx,bx

mov si,cx ;buffer's length

mov cl,10 ;multiplier

toHex:

mov bl,byte ptr bconteg[di]

sub bl,'0' ;num = num's code - 30h

jb badInp ;if symbol not a num

cmp bl,9 ;same

ja badInp ;try input again

mul cx ;multiply on 10

```

    add ax,bx      ;+new num to ax
    inc di         ;next symbol
    cmp di,si      ;if di<length + 1
    jb toHex
nM:
    jmp endInp

badInp:
    jmp start

endInp:
    ret
endp

proc print
    mov bx,0ah     ;divider
    xor cx,cx      ;clear count

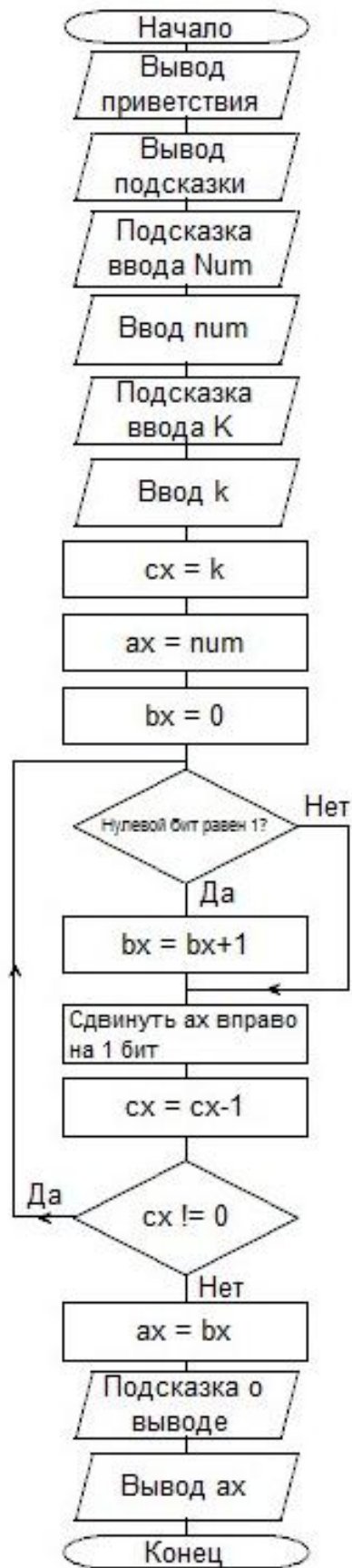
divloop:
    xor dx,dx      ;clear dx
    div bx         ;divide on 10
    add dx,'0'     ;make a symbol from num
    push dx        ;save dx
    inc cx
    test ax,ax     ;if ax!=0
    jnz divloop    ;continue to divide

restore:
    ;pop ax
    pop ax         ;read from stack
    mov dx, ax
    mov ah,2       ;print symbol from al
    int 21h        ;
    loop restore
    ret
endp

end start ; set entry point and stop the assembler.

```

Блок-схема алгоритма



Трассировка основного алгоритма программы

Адрес	Мнемокод	Двоичный код	Изменения регистров	Комментарий
003F	xor bx, bx	Байт 1: 00110011 001100 – операция XOR 1 – сначала приёмник потом источник 1 – слово (0 - 8 бит, 1 - 16 бит) Байт 2: 11011011 11 – указываем что работаем с регистрами 011 – код регистра приёмника (BX) 011 – код регистра источника (BX)	BX = 0	Флаги установлены: Нуля, Паритета.
0041	test ax, 1h	Байт 1: 10101001 – операция test ax Байт 2: число		Если нулевой бит установлен в 1, то устанавливается флаг нуля, иначе сбрасывается.
0044	jz 0047h	Байт 1: 01110100 – операция JZ Байт 2: 00000001 – смещение		Если установлен флаг нуля, то переходим
0046	inc bx	Байт 1: 01000011 01000 – операция INC 11 – код bx	BX=BX + 1	
0047	shr ax, 1h	Байт 1: 11010001 1101001 – циклически сдвинуть биты вправо один раз Байт 2: код регистра ax и дополнительные указания	AX=AX>>1	
0049	loop 0041h	Байт 1: 11100010 – операция loop Байт 2: Адрес	CX = CX - 1	

Вывод

Я ознакомился с условными операторами, циклами, сдвигами на языке ассемблера intel 8086. Исследовал работу, удалённого участка программы в оперативной памяти, копированием участка кода в другое местоположение.