**Conversion techniques.**

section .data

msg db 10,"Enter Number:",10

msg\_len equ $-msg

msg1 db 10,"Your Number:",10

msg1\_len equ $-msg

menu db 10,"1.Hexadecimal to BCD conversion:"

db 10,"2.BCD to Hexadecimal conversion:"

db 10,"3.Exit"

db 10,"Enter your choice(1-3):"

menu\_len equ $-menu

%macro print 2

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

%macro read 2

mov rax,0

mov rdi,0

mov rsi,%1

mov rdx,%2

syscall

%endmacro

section .bss

char\_ans resb 16

buf resb 8

ans resb 16

section .text

global \_start

\_start:

menu1: print menu,menu\_len

read buf,2

mov al,[buf]

c1: cmp al,'1'

jne c2

call hextobcd

jmp menu1

c2: cmp al,'2'

jne c3

call bcdhex

jmp menu1

c3: cmp al,'3'

mov rax,60

mov rdx,00

syscall

accept:

read buf,5

mov rsi,buf

mov rcx,4

mov bx,0

next\_byte:

shl bx,4

mov al,[rsi]

cmp al,'0'

cmp al,'9'

jbe sub30

cmp al,'A'

cmp al,'F'

jbe sub37

cmp al,'a'

cmp al,'f'

jbe sub57

sub57: *sub al,20h*

*sub37: sub al,07h*

*sub30: sub al,30h*

add bx,ax

inc rsi

dec rcx

jnz next\_byte

ret

display:

mov rbx,16

mov rcx,4

mov rsi,char\_ans+3

cnt:

mov rdx,0

div rbx

cmp dl,09h

jbe add30

add dl,07h

add30:

add dl,30h

mov [rsi],dl

dec rsi

dec rcx

jnz cnt

print char\_ans,5

ret

bcdhex:

print msg,msg\_len

read buf,5

mov rsi,buf

xor ax,ax

mov rbp,4

mov ebx,10

next:

xor cx,cx

mul bx

mov cl,[rsi]

sub cl,30h

add ax,cx

inc rsi

dec rbp

jnz next

mov [ans],ax

print msg1,msg1\_len

mov ax,[ans]

call display

ret

hextobcd:

print msg,msg\_len

call accept

mov ax,bx

mov bx,10

xor bp,bp

back:

xor dx,dx

div bx

push dx

inc bp

cmp ax,0

jne back

print msg1,msg1\_len

back1:

pop dx

add dl,30h

mov [char\_ans],dl

print char\_ans,1

dec bp

jnz back1

ret