Successive addition.

section .data

msg db 10,"Enter First Number:",10

msg\_len equ $-msg

msg1 db 10,"Your Second Number:",10

msg1\_len equ $-msg1

msg2 db 10,"Addition is:",10

msg2\_len equ $-msg2

menu db 10,"1.successive addition:"

db 10,"2.Add shift method:"

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

n1 resb 2

n2 resb 2

ansh resb 2

ansl resb 2

buf resb 16

ans resd 1

char\_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 add

jmp menu1

c2: cmp al,'2'

jne c3

call Ad\_sh

jmp menu1

c3: cmp al,'3'

mov rax,60

mov rdx,00

syscall

Ad\_sh:

mov dword[ans],0000

print msg,msg\_len

call accept

mov [n1],bx

print msg1,msg1\_len

call accept

mov [n2],bx

xor rax,rax

xor rbx,rbx

mov ax,[n1]

mov bx,[n2]

mov cx,16

mov ebp,0

back4:

shl ebp,1

shl ax,1

jnc next4

add ebp,ebx

next4:

dec cx

jnz back4

display4:

mov [ans],ebp

print msg2,msg2\_len

mov eax,[ans]

call display32

ret

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

add:

mov word[ansl],00

mov word[ansh],00

print msg,msg\_len

call accept

mov [n1],bx

print msg1,msg1\_len

call accept

mov [n2],bx

mov cx,[n1]

mov ax,[n2]

cmp cx,0

je next

back:

add [ansl],ax

jnc next1

inc word[ansh]

next1:

dec cx

jnz back

next:

print msg2,msg2\_len

mov ax,[ansh]

call display

mov ax,[ansl]

call display

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,4

ret

display32:

mov rbx,16

mov rcx,8

mov rsi,char\_ans+7

cnt5:

mov rdx,0

div rbx

cmp dl,09h

jbe add301

add dl,07h

add301:

add dl,30h

mov [rsi],dl

dec rsi

dec rcx

jnz cnt5

print char\_ans,8

ret