

Subject: Microprocessor Lab

Assignment No. 05

Roll No: **21118**

Batch: E-1

Problem Statement:

Write X86/64 ALP to count number of positive and negative numbers from the array.

Hardware of PC:

- Manufacturer and model: Acer Swift-3
- Processor: Intel core i5 – 8265U @1.60 GHz
- Memory: 8GB of DDR4 RAM and 512GB of ROM
- System Type: 64-bit OS, x-64 based PC

Software Used:

- Operating system: Windows 10 and Ubuntu 20.04LTS. Using WSL 2 (Windows Subsystem for Linux) and Installed WSL plugin in VS Code.
- Text editor: VS Code (open source edition, version: 1.26.2)
- Assembler: NASM (version: 2.14.02)

Theory:

Instructions:

- **CMP:** The CMP instruction compares two operands. It is generally used in conditional execution. This instruction basically subtracts one operand from the other for comparing whether the operands are equal or not. It does not disturb the destination or source operands. It is used along with the conditional jump instruction for decision making.
 - Syntax: CMP destination, source
 - Example: CMP CX, 00 ; Compare the CX value with zero
- **JS:** It is a conditional jump instruction. It operates when sign flag is set. The sign flag (SF) is set when the result of an arithmetic or logical operation generates a negative result.

Procedure:

- Procedures in assembly are equivalent to functions in c++.
- Syntax:
proc_name:

- ```

 // procedure body
 // ..
ret

```
- Procedure can be called from another procedure by  
CALL *proc\_name*
  - The called procedure returns the control to the calling procedure by using the RET instruction.

#### Algorithm:

- *Counting Positive and Negative Numbers:*
  1. Start.
  2. Declare an array of 5 numbers.
  3. Initialize pos\_counter=0, neg\_counter=0, index\_reg=array address, counter=5
  4. Read the number from index\_reg into a register.
  5. Compare register with 00H and check sign bit
  6. If sign bit==1 then increment neg\_counter=neg\_counter+1 else
  7. increment pos\_counter=pos\_counter+1
  8. Increment index\_reg= index\_reg+1
  9. Decrement counter=counter-1
  10. If counter!=0 then goto step number 4 else continue
  11. Print message "Positive numbers are:" and print pos\_counter.
  12. Print message "Negative numbers are:" and print neg\_counter.
  13. Exit.

#### Program:

; Shubham (Roll No: 21118)  
; Subject MPL

```

%macro rwm 3
 mov rax, %1
 mov rdi, 01
 mov rsi, %2
 mov rdx, %3
 syscall
%endmacro

```

```

section .data
 arr dq -1111111111111111h, -1111111100000000h, 7999999999999999h,
0AAAAAAAAAAAAAAAAAAAh, 555555555555555h
 n equ 5

```

```
pmsg db "The Count of Positive No: "
plen equ $-pmsg
```

```
nmsg db "The Count of Negative No: ",
nlen equ $-nmsg
```

```
nwln db 0xA
```

```
section .bss
pcnt resq 1
ncnt resq 1
ascii_num resb 16
```

```
section .text
global _start
```

```
_start:
 mov rsi,arr
 mov rdi,n
 mov qword[pcnt], 0
 mov qword[ncnt], 0
```

```
pos_neg_cnt:
 mov rax,[rsi]
 cmp rax,0000000000000000h
 js neg_inc
```

```
pos_inc:
 inc qword[pcnt]
 jmp next
```

```
neg_inc:
 inc qword[ncnt]
```

```
next:
 add rsi,8
 dec rdi
 jnz pos_neg_cnt
```

```
 ; printing positive nums
 rwm 1,pmsg,plen
 mov rax,[pcnt]
 call conv_and_display ; hex to ascii conversion
 rwm 1,nwln,1
```

```
 ; printing neg nums
 rwm 1,nmsg,nlen
 mov rax,[ncnt]
```

```

 call conv_and_display ; hex to ascii conversion
 rwm 1,nwln,1

 ; exit syscall

 mov rax,60
 mov rbx,0
 syscall

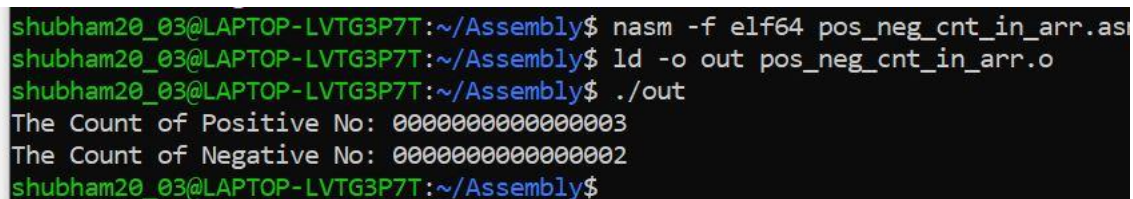
;conv_and_disp procedure -> convert hex to ascii
conv_and_display:
 mov rsi, ascii_num+15
 mov rcx,16

again:
 mov rdx,0
 mov rbx,16h ; 16 in hex == 10 in decimal
 ;(quotient and rem will be stored in rax and rdx resp)
 div rbx ; on divide rem will be last digit
 cmp dl,09h
 jbe add30
 add dl,07h
add30:
 add dl,30h
 mov [rsi],dl
 dec rsi
 dec rcx
 jnz again

 rwm 1,ascii_num,16
ret

```

### Output:



```

shubham20_03@LAPTOP-LVTG3P7T:~/Assembly$ nasm -f elf64 pos_neg_cnt_in_arr.asm
shubham20_03@LAPTOP-LVTG3P7T:~/Assembly$ ld -o out pos_neg_cnt_in_arr.o
shubham20_03@LAPTOP-LVTG3P7T:~/Assembly$./out
The Count of Positive No: 0000000000000003
The Count of Negative No: 0000000000000002
shubham20_03@LAPTOP-LVTG3P7T:~/Assembly$

```

### Conclusion:

In this assignment I learned arithmetic operations on 64-bit numbers by using switch case, macro and procedure in assembly language and written the assembly program for the same.

