|  |
| --- |
| **Subject: MICROPROCESSOR LAB (MPL)**  **Name:Priyanka Salunke** |
| **Class: SE COMP A Roll No.: F19111151** |
| **Semester: Sem-II Year: 2020-21** |
| **Date of Performance: Date of Submission:** |
| **Examined:** |

**Assignment No-05**

**Title:-**Count no. of positive and negative numbers

**Assignment Name: -** Write an ALP to count no. of positive and negative numbers from the array.

**Objective-**

* To understand the assembly language program
* To understand 64 bit interrupt.

**Outcome-**

* Students will be able to write code for how to count positive and negative number from array
* Students will be able to understand different assembly language instruction.

### [Prerequisite](http://dictionary.reference.com/browse/prerequisite) -

System call of Unix for Assembly language Program.

**Hardware Requirement-**

Desktop PC

**Software Requirement-**

Ubuntu 14.04,

Assembler: NASM version 2.10.07

Linker: ld

**Introduction:-**

**Theory:**

**Algorithm:**

1. Start

2. Initialise section .data

3. Define variable for array,pcount,ncount

4. Count Positive and negative number using JS command.

5. Display counts

6. Terminate program using system call

6. Stop

**Conclusion:-** Hence we implemented an ALP to count positive and negative number from array and display count.

**Questions:-**

Q.1.Explain BT,JS,loop instruction with Example?

Q.2 Explain Paging in 80386?

Q.3 Draw control registers of 80386

**Program**

%macro print 2

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

section .data

m0 db "Counting +ve and -ve elements of an array.",10

l0 equ $-m0

m1 db "Positive nos. are : "

l1 equ $-m1

m2 db "Negative nos. are : "

l2 equ $-m2

array db -1h,2h,-3h,4h,-5h,-6h,-7h

pcount db 0

ncount db 0

newline db 0xa

section .bss

dispbuff resb 2

section .text

global \_start

\_start:

print m0,l0

mov rsi,array

mov rcx,07

again:

mov al, [rsi]

cmp al,00h

js next1

inc byte[pcount]

jmp pskip

next1: inc byte[ncount]

pskip: add rsi,1

loop again

print m1,l1

mov bl,[pcount]

call disp\_result

print newline,1

print m2,l2

mov bl,[ncount]

call disp\_result

print newline,1

mov rax,60 ;terminate program

xor rdi,rdi

Syscall

;procedure to convert hex number to its equivalent ASCII

disp\_result:

mov rdi,dispbuff

mov rcx,02

dispup1:

rol bl,4

mov dl,bl

and dl,0fh

add dl,30h

cmp dl,39h

jbe dispskip1

add dl,07h

dispskip1:

mov [rdi],dl

inc rdi

loop dispup1

print dispbuff,2

ret

**Output**

