Chapter 3

Basic I/O in NASM

In this chapter, we will learn how to obtain user input and how to return the output to the output device. The input from the standard input device (Keyboard) and Output to the standard output device (monitor) in a NASM Program is implemented using the Operating System's read and write system call. Interrupt no: 80h is given to the software generated interrupt in Linux Systems. Applications implement the System Calls using this interrupt. When an application triggers int 80h, then OS will understand that it is a request for a system call and it will refer the general purpose registers to find out and execute the exact Interrupt Service Routine (ie. System Call here).The standard convention to use the software 80h interrupt is, we will put the system call no: in eax register and other parameters needed to implement the system calls in the other general purpose registers. Then we will trigger the 80h interrupt using the instruction 'INT 80h'. Then OS will implement the system call.

1. Exit System Call
    - This system call is used to exit from the program
    - System call number for exit is 1, so it is copied to eax reg.
    - Output of a program if the exit is successful is 0 and it is being passed as a parameter for exit( ) system call. We need to copy 0 to ebx reg.
    - Then we will trigger INT 80h

```
mov eax, 1        ; System Call Number
mov ebx, 0        ; Parameter
int 80h           ; Triggering OS Interrupt
```

2. Read System Call

   - Using this we could read only string/character
   - System Call Number for Read is 3. It is copied to eax.
   - The standard Input device(keyboard) is having the reference number 0 and it must be copied to ebx reg.
   - We need to copy the pointer in memory, to which we need to store the input string to ecx reg.
   - We need to copy the number of characters in the string to edx reg.
   - Then we will trigger INT 80h.
   - We will get the string to the location which we copied to ecx reg.

```
mov eax, 3                ; Sys_call number for read
mov ebx, 0                ; Source Keyboard
mov ecx, var               ; Pointer to memory location
mov edx, dword[size]      ; Size of the string
int 80h                   ; Triggering OS Interrupt
```

   - This method is also used for reading integers and it is a bit tricky. If we need to read a single digit, we will read it as a single character and then subtract 30h from it(ASCII of 0 = 30h). Then we will get the actual value of that number in that variable.

   (a) Reading a single digit number

```
mov eax, 3
mov ebx, 0
mov ecx, digit1
mov edx, 1
int 80h
sub byte[digit1], 30h   ;Now we have the actual number in [digit1]
```

   (b) Reading a two digit number

```
;Reading first digit
mov eax, 3
mov ebx, 0
mov ecx, digit1
mov edx, 1
int 80h

;Reading second digit
mov eax, 3
mov ebx, 0
mov ecx, digit2
mov edx, 2              ;Here we put 2 because we need to read and
int 80h                  omit enter key press as well

sub byte[digit1], 30h
sub byte[digit2], 30h

;Getting the number from ASCII
; num = (10* digit1) + digit2

mov al, byte[digit1]     ; Copying first digit to al
mov bl, 10
mul bl                   ; Multiplying al with 10
movzx bx, byte[digit2]   ; Copying digit2 to bx
add ax, bx

mov byte[num], al        ; We are sure that no less than 256, so we can
omit higher 8 bits of the result.
```

3. Write System Call
    - This system call can be used to print the output to the monitor
    - Using this we could write only string / character
    - System Call Number for Write is 4. It is copied to eax.
    - The standard Output device(Monitor) is having the reference number 1 and it must be copied to ebx reg.
    - We need to copy the pointer in memory, where the output sting resides

to ecx reg.

- We need to copy the number of characters in the string to edx reg.
- Then we will trigger INT 80h.

  See the below given example which is equivalent to the C statement printf()

  Eg:

```
mov eax, 4          ;Sys_call number
mov ebx, 1          ;Standard Output device
mov ecx, msg1       ;Pointer to output string
mov edx, size1      ;Number of characters
int 80h             ;Triggering interrupt.
```

- This method is even used to output numbers. If we have a number we will break that into digits. Then we keep each of that digit in a variable of size 1 byte. Then we add 30h (ASCII of 0) to each, doing so we will get the ASCII of character to be print.