ASSEMBLY - CONSTANTS

There are several directives provided by NASM that define constants. We have already used the EQU directive in previous chapters. We will particularly discuss three directives −

- EQU
- %assign
- %define

The EQU Directive

The **EQU** directive is used for defining constants. The syntax of the EQU directive is as follows −

```
CONSTANT_NAME EQU expression
```

For example,

```
TOTAL_STUDENTS equ 50
```

You can then use this constant value in your code, like −

```
mov   ecx,   TOTAL_STUDENTS
cmp   eax,   TOTAL_STUDENTS
```

The operand of an EQU statement can be an expression −

```
LENGTH equ 20
WIDTH  equ 10
AREA   equ length * width
```

Above code segment would define AREA as 200.

Example

The following example illustrates the use of the EQU directive −

```
SYS_EXIT  equ 1
SYS_WRITE equ 4
STDIN     equ 0
STDOUT    equ 1
section   .text
   global _start     ;must be declared for using gcc

_start:              ;tell linker entry point
   mov eax, SYS_WRITE
   mov ebx, STDOUT
   mov ecx, msg1
   mov edx, len1
   int 0x80

   mov eax, SYS_WRITE
   mov ebx, STDOUT
   mov ecx, msg2
   mov edx, len2
   int 0x80

   mov eax, SYS_WRITE
   mov ebx, STDOUT
   mov ecx, msg3
   mov edx, len3
```

```
    int 0x80

    mov eax,SYS_EXIT     ;system call number (sys_exit)
    int 0x80             ;call kernel

section   .data
msg1 db 'Hello, programmers!',0xA,0xD
len1 equ $ - msg1

msg2 db 'Welcome to the world of,', 0xA,0xD
len2 equ $ - msg2

msg3 db 'Linux assembly programming! '
len3 equ $- msg3
```

When the above code is compiled and executed, it produces the following result −

```
Hello, programmers!
Welcome to the world of,
Linux assembly programming!
```

The %assign Directive

The **%assign** directive can be used to define numeric constants like the EQU directive. This directive allows redefinition. For example, you may define the constant TOTAL as −

```
%assign TOTAL 10
```

Later in the code, you can redefine it as −

```
%assign  TOTAL  20
```

This directive is case-sensitive.

The %define Directive

The **%define** directive allows defining both numeric and string constants. This directive is similar to the #define in C. For example, you may define the constant PTR as −

```
%define PTR [EBP+4]
```

The above code replaces *PTR* by [EBP+4].

This directive also allows redefinition and it is case-sensitive.