MaAL

Lecture 6
20-Sep
Theory

Named Constants

→ To make assembly language code easier to understand, symbolic name is used for a constant quality.

→ EQU (equates) pseudo-op is used to assign a name to a constant.
The syntax is.
Name EQU Constant.
For example. LF EQU 0Ah

→ Assigns the name LF to 0Ah, the ASCII code of line feed character.

→ Name LF may now be used in place of 0Ah anywhere in the program

→ The assembler translates the given instruction into same machine instruction.
MOV, DL, 0Ah & MOV.

Memory Models:

Directives: Memory allocate

→ The size of code and data program can have is determined by specifying a memory model using .Model directive.

.Model memory_model

→ .Model directive should come before any segment definition

Small code in one segment, data in one

Medium code in more than one segment, data in one segment

Compact code in one segment, data in more than one segment

Large code in more than one segment, data in more than one segments, no array larger that 64k byte

Huge code in more than one segment, data in more than one segments, arrays may be larger than 64k bytes.

Data and Stack segment:

→ Data segment contains all the variable and constant definitions.

→ Since no memory allocation is involved, constant can be declare in the code.

→ To declare a data segment, we use the directive .DATA, followed by variable and constant declarations.

→ Purpose of the stack segment declarations is to set aside a block of memory (the

→ stack area) to store the stack. Stack area should be big enough to contain the stack at its maximum size. The declaration syntax is.

.STACK SIZE

→ where size is an optional number that specifies the stack area size in bytes.

→ If size is omitted, 1KB is set aside for the stack area.

Code Segment:

→ Code segment contains a program instructions. The declaration syntax is.

```
.CODE name
```

→ where name is an optional name of segment.

→ Inside a code segment, instructions are organized as procedures. The simplest procedure definition is

```
name PROC
; body of the procedure
name ENDP
```

The last line in the program should be the END directive, followed by name of the main procedure.

Input and Output Instructions:

→ There are two categories of I/O service routines:

- Basic input/output system (BIOS) routines
- DOS routines.

→ To invoke a DOS or BIOS routine, the INT (interrupt) instruction is used; it has the format

INT interrupt number

→ where interrupt number is a number that specifies a routine

→ INT 21h may be used to invoke a large number of DOS functions.

→ A particular function is requested by placing a function number in the AH register and invoking INT 21h.