UNIT-II
8086 Assembler Directives

Assembler:

Is a program that accepts an assembly language program as input and converts it into an object module and prepares for loading the program into memory for execution.

Loader (linker) further converts the object module prepared by the assembler into executable form, by linking it with other object modules and library modules.

The final executable map of the assembly language program is prepared by the loader at the time of loading into the primary memory for actual execution.

Procedure for assembling a program

Assembling a program proceeds statement by statement sequentially.

The first phase of assembling is to analyze the program to be converted.

It also analyses the segments used by the program types and labels and their memory requirements.

The second phase looks for the addresses and data assigned to the labels.It also finds out codes of the instructions from the instruction machine, code database and the program data.

It processes the pseudo operands and directives.

It is the task of the assembler designer to select the suitable strings for using them as directives,pseudo operands or reserved words and decides syntax.

Directives:

Also called as pseudo operations that control the assembly process.

They indicate how an operand or section of a program to be processed by the assembler.

They generate and store information in the memory.

Assembler Memory models:

Each model defines the way that a program is stored in the memory system.

Tiny: data fits into one segment written in .COM format

Small: has two segments data and memory.

There are several other models too.

Directive for string data in a memory segment:

DB (DEFINE BYTE)

The DB directive is used to declare a byte type variable, or a set aside one or more storage locations of type byte in memory.

PRICES DB 49H, 98H, 29H;Declare array of 3 bytes named PRICES and initialize them with specified values.

NAMES DB “SAKSHI”; Declare array of 6 bytes and initialize with ASCII codes for the letters in SAKSHI.

Segment:

The SEGMENT directive is used to indicate the start of a logical segment. Preceding the SEGMENT directive is the name you want to give the segment.

Example:

```
Name SEGMENT
Variable_name DB …….
Variable_name DW …….
Name ENDS
Data SEGMENT
Data1 DB …….
Data2 DW …….
Data ENDS
Code SEGMENT
START: MOV AX,BX
```

ENDS (END SEGMENT)

This directive is used with the name of a segment to indicate the end of that logical segment.

CODE SEGMENT; Start of logical segment containing code instruction statements

CODE ENDS; End of segment named CODE

END (END PROCEDURE)

The END directive is put after the last statement of a program to tell the assembler that this is the end of the program module

ALIGN

Memory array is stored in word boundaries.

Example: ALIGN 2 means storing from an even address

ASSUME

ASSUME tells the assembler what names have been chosen for Code, Data Extra andStack segments.

Informs the assembler that the register CS is to be initialized with theaddress allotted by the loader to the label CODE and DS is similarly initialized with theaddress of label DATA.

Example

ASSUME CS: Name of code segment

ASSUME DS: Name of the data segment

ASSUME CS: Code1, DS: Data1

THANK YOU