Chương 5 : Nhập môn Assembly

Mục tiêu

- **Hiểu ngôn ngữ máy và ngôn ngữ Assembly.**
- **Trình hợp dịch Assembler.**
- **Lý do nghiên cứu Assembly.**
- **Hiểu các thành phần cơ bản của Assembly**
- **Nắm được cấu trúc của 1 CT Assembly.**
- **Biết viết 1 chương trình Assembly.**
- **Biết cách dịch, liên kết và thực thi 1 chương trình Assembly.**

Giới thiệu ngôn ngữ Assembly

- Giúp khám phá bí mật phần cứng cũng như phần mềm máy tính.
- Nắm được cách phần cứng MT làm việc với hệ điều hành và hiểu được bằng cách nào 1 trình ứng dụng giao tiếp với hệ điều hành.
- Một MT hay một họ MT sử dụng 1 tập lệnh mã máy riêng cũng như 1 ngôn ngữ Assembly riêng.

Assembler

- **Một chương trình viết bằng ngôn ngữ Assembly muốn MT thực hiện được ta phải chuyển thành ngôn ngữ máy.**

- Chương trình dùng để dịch 1 file viết bằng Assembly
➔ ngôn ngữ máy , gọi là Assembler.

Có 2 chương trình dịch:

MASM và TASM

Lý do nghiên cứu Assembly

- Đó là cách tốt nhất để học phần cứng MT và hệ điều hành.
- Vì các tiện ích của nó .
- Có thể nhúng các chương trình con viết bằng ASM vào trong cácchương trình viết bằng ngôn ngữ cấp cao .

Lệnh máy

- **Là 1 chuổi nhị phân có ý nghĩa đặc biệt – nó ra lệnh cho CPU thực hiện tác vụ.**
  - **Tác vụ đó có thể là :**
    di chuyển 1 số từ vị trí nhớ này sang vị trí nhớ khác.
    Cộng 2 số hay so sánh 2 số.

0 0 0 0 0 1 0 0 Add a number to the AL register

1 0 0 0 0 1 0 1 Add a number to a variable

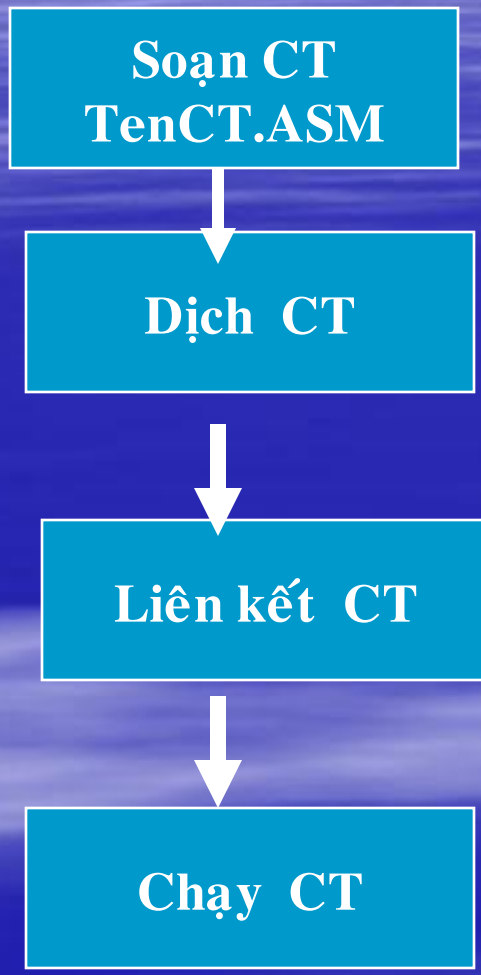
1 0 1 0 0 0 1 1 Move the AX reg to another reg

Lệnh máy (cont)

- Tập lệnh máy được định nghĩa trước, khi CPU được sản xuất và nó đặc trưng cho kiểu CPU .
  - Ex : B5 05 là 1 lệnh máy viết dạng số hex, dài 2 byte.
  - Byte đầu B5 gọi là Opcode
  - Byte sau 05 gọi là toán hạng Operand

Ý nghĩa của lệnh B5 05 : chép giá trị 5 vào reg AL

Cách viết 1 chương trình Assembly



Dùng 1 phần mềm soạn thảo VB bất kỳ để soan CT Assembly như : NotePad, NC, màn hình C, Pascal ...

CT có phần mở rộng là .ASM

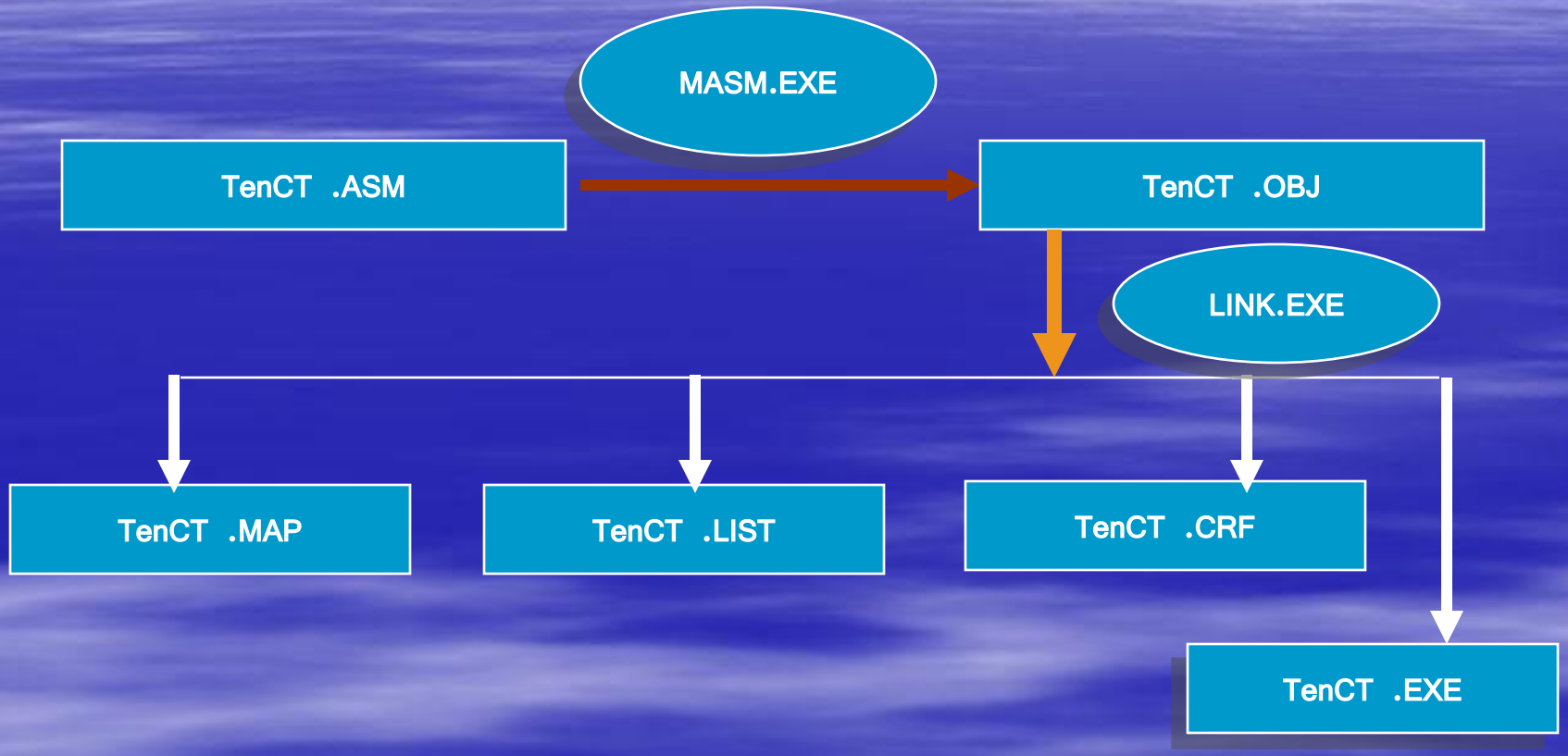
dùng MASM để dịch chương trình nguồn .ASM

➔ File Object.

dùng LINK để liên kết Object tạo tập tin thực hiện .EXE

Gỏ tên tập tin thực hiện .EXE từ dấu nhắc DOS để chạy

Dịch và nối kết chương trình



Một chương trình minh hoạ

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
MES DB “HELLO WORD”,’$’
.CODE
MAIN PROC
   MOV AX, @DATA
   MOV DS, AX
   MOV  DX,  OFFSET  MES
   MOV  AH,  9
   INT  21
   MOV  AH,4CH
   INT  21
MAIN  ENDP
END  MAIN
```

Các file được tạo

- Sau khi dịch thành công file nguồn.ASM, ta có các file :
- File listing : file VB , các dòng có đánh số thứ tự mã.
- File Cross reference
- File Map
- File Obj
- File EXE

File Listing

```
Microsoft (R) Macro Assembler Version 5.10                  10/11/4
                                                            Page     1-1


                        1                       DOSSEG
                        2                       .MODEL SMALL
                        3                       .STACK 100H
                        4                       .DATA
                        5 0000  48 45 4C 4C 4F 20       MES DB "HELLO WORD$"
                        6       57 4F 52 44 24
                        7                       .CODE
                        8 0000                      MAIN PROC
                        9 0000  B8 ---- R               MOV AX,@DATA
                       10 0003  8E D8                   MOV DS, AX
                       11 0005  B4 09                   MOV AH,9
                       12 0007  BA 0000 R                   MOV DX, OFFSET MES
                       13 000A  CD 21                   INT 21H
                       14 000C  B4 4C                   MOV AH,4CH
                       15 000E  CD 21                   INT 21H
                       16 0010                      MAIN ENDP
                       17                       END MAIN
♀◘Microsoft (R) Macro Assembler Version 5.10                10/11/4
```

Map File

- Start Stop Length Name Class
- 00000H 0001FH 00020H _TEXT CODE
- 00020H 0002AH 0000BH _DATA DATA
- 00030H 0012FH 00100H STACK STACK

- Origin Group
- 0002:0 DGROUP

- Program entry point at 0000:0010

Giải thích

- .model small : dùng kiểu cấu trúc <= 64 K bộ nhớ cho mã , 64K cho dữ liệu.
- .Stack 100h : dành 256 bytes cho stack của chương trình .
- .Data : đánh dấu phân đoạn dữ liệu ở đó các biến được lưu trữ.
- .Code : đánh dấu phân đoạn mã chứa các lệnh phải thi hành.
- Proc : khai báo đầu 1 thủ tục, trong Ex này ta chỉ có 1 thủ tục Main.

Giải thích (cont)

- **Chép địa chỉ đoạn dữ liệu vào thanh ghi AX.**
- **Sau đó chép vào thanh ghi DS**
- **Gọi hàm số 9 của Int 21h của Dos để xuất chuổi ký tự ra màn hình.**
- **Thoát khỏi CT .**
- **Main endp : đánh dấu kết thúc thủ tục**
- **End main : chấm dứt chương trình**

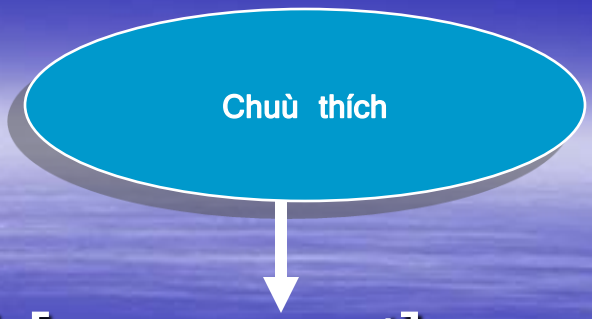
Cấu trúc của 1 CT ASM

```
DOSSEG
.MODEL  kiểu bộ nhớ
.STACK  kích thước
.DATA
        khai báo biến, hằng
.CODE
   MAIN PROC
        MOV  AX, @DATA
        MOV  DS,AX
             các lệnh của chương trình chính
       MOV AH,4CH  ; Thoát khỏi chương trình
       INT 21H
  MAIN ENDP
       các chương trình con khác nếu có
END MAIN
```

Các chế độ bộ nhớ

Kiểu	Mô tả
SMALL	Mã lệnh trong 1 đoạn.Dữ liệu trong 1 đoạn
MEDIUM	Mã lệnh nhiều hơn 1 đoạn.Dữ liệu trong 1 đoạn
COMPACT	Mã lệnh trong 1 đoạn. Dữ liệu nhiều hơn 1 đoạn
LARGE	Mã lệnh nhiều hơn 1 đoạn Dữ liệu nhiều hơn 1 đoạn,không có mảng nào > 64K
HUGE	Mã lệnh nhiều hơn 1 đoạn Dữ liệu nhiều hơn 1 đoạn, mảng có thể > 64K

Dạng lệnh



- [name] [operator] [operand] [comment]


```
Ex :   MOV  CX  ,  0
       LAP   :   MOV   CX,  4
       LIST   DB   1,2,3,4
```

Mỗi dòng chỉ chứa 1 lệnh và mỗi lệnh phải nằm trên 1 dòng

INT 21H

- Lệnh INT số hiệu ngắt được dùng để gọi chương trình ngắt của DOS và BIOS.

Ngắt 21h

Muốn sử dụng hàm nào của INT 21h ta đặt function_number vào thanh ghi AH, sau đó gọi INT 21h

Function_number	chức năng
1	nhập 1 ký tự từ bàn phím
2	Xuất 1 ký tự ra mà hình.
9	Xuất 1 chuổi ký tự ra màn hình

INT 21h (cont)

Hàm 1 : Nhập 1 ký tự

Input : AH =1
Output : AL = mã ASCCI của phím ấn
= 0 nếu 1 phím điều khiển được ấn

Hàm 2 : Hiển thị 1 ký tự ra màn hình

Input : AH =2
DL = Mã ASCII của ký tự hiển thị hay ký tự điều khiển

Thí dụ minh họa

```
DOSSEG
.MODEL SMALL
.STACK 100H
.CODE
   MAIN PROC
        MOV AH , 2
        MOV DL , '?'
        INT 21H
        MOV AH ,1
        INT 21H
        MOV BL,AL
```

```
    MOV   AH,2
    MOV DL, 0DH
    INT 21H
    MOV DL , 0AH
    INT 21H
    MOV DL , BL
    INT 21H
    MOV AX , 4C00H
    INT 21H
    MAIN ENDP
END MAIN
```

KẾT QUẢ

```
? N
N
```

Thí dụ minh họa các hàm của INT 21

- **In dấu ? ra màn hình :**

```
MOV AH, 2
MOV DL, '?'
INT 21H
```

- **Nhập 1 ký tự từ bàn phím :**

```
MOV AH, 1
INT 21H
```

Biến

- Cú pháp : [tên biến] DB | DW |.... [trị khởi tạo]
- Là một tên ký hiệu dành riêng cho 1 vị trí trong bộ nhớ nơi lưu trữ dữ liệu.
- Offset của biến là khoảng cách từ đầu phân đoạn đến biến đó.
- Ex : khai báo 1 danh sách aList ở địa chỉ 100 với nội dung sau :

```
.data
    aList  db “ABCD”
```

Biến (cont)

Lúc đó :

Offset	Biến
0000	A
0001	B
0002	C
0003	D

Khai báo biến

Từ gợi nhớ	Mô tả	Số byte	Thuộc tính
DB	Định nghĩa byte	1	Byte
DW	Từ	2	Word
DD	Từ kép	4	Doubleword
DQ	Từ tứ	8	Quardword
DT	10 bytes	10	tenbyte

Minh họa khai báo biến

KIỂU BYTE

- Char db ‘A’
- Num db 41h
- Mes db “Hello Word”,’$’
- Array_1 db 10, 32, 41h, 00100101b
- Array_2 db 2,3,4,6,9
- Myvar db ? ; biến không khởi tạo
- Btable db 1,2,3,4,5
  db 6,7,8,9,10

Minh họa khai báo biến

KIỂU WORD

```
DW   3   DUP  ( ? )
DW  1000h, 'AB', 1024
DW   ?
DW   5   DUP  ( 1000h )
DW   256*2
```

DẠNG LƯU TRỮ DỮ LIỆU KIỂU WORD :

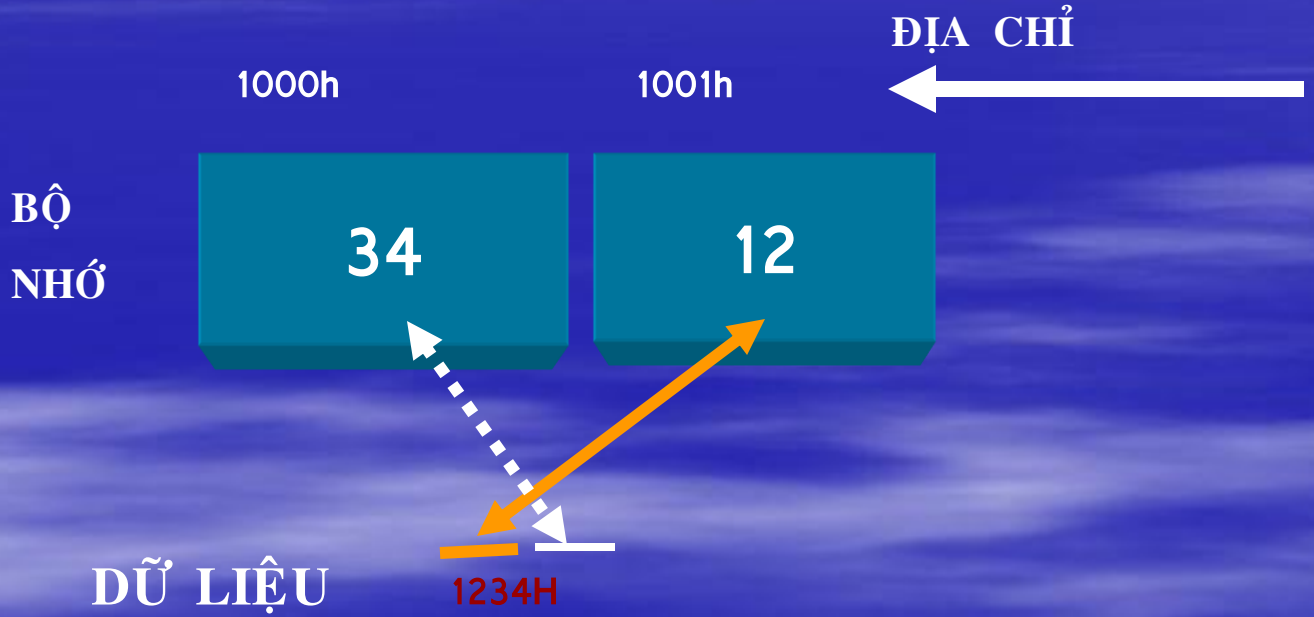
Trình hợp dịch đảo ngược các byte trong 1 giá trị kiểu WORD khi lưu trữ trong bộ nhớ :

Byte thấp lưu ở địa chỉ thấp Byte cao lưu ở địa chỉ cao

Minh họa khai báo biến

KIỂU WORD

Ex : 1234h được lưu trữ trong bộ nhớ như sau :



Toán tử DUP

- Lặp lại 1 hay nhiều giá trị khởi tạo.
- Ex :

```
Bmem   DB  50 Dup(?)
; khai báo vùng nhớ gồm 50 bytes.
  db  4 dup (“ABC”)
;12 bytes “ABCABCABCABC”
  db 4096 dup (0)
; Vùng đệm 4096 bytes tất cả bằng 0
```

Khởi tạo biến

- Lưu ý :

Khi khởi tạo trị là 1 số hex thì giá trị số luôn luôn bắt đầu bằng 1 ký số từ 0 đến 9. Nếu ký số bắt đầu là A.. F thì phải thêm số 0 ở đầu.

- Ex :

```
Db  A6H   ; sai
Db  0A6h  ; đúng
```

Toán tử DUP (cont)

```
Amtrix    dw    3 dup (4 dup (0) )
```

Taïo 1 ma traän 3x4

```
Atable    db    4 dup (3 dup (0), 2 dup ('X'))
```

Tạo 1 vùng nhớ chứa 000XX 000XX 000XX 000XX

Toán tử DUP

- **Chỉ xuất hiện sau 1 chỉ thị DB hay DW**
- **Với DUP ta có thể lặp lại 1 hay nhiều trị cho vùng nhớ.**
- **Rất có ích khi làm việc với mảng hay chuổi.**

Toán tử ?

- Muốn khai báo 1 biến hay 1 mảng mà không cần khởi tạo trị ta dùng toán tử ?

Ex : MEM8 DB ? ; khai báo 1 byte trống trong bộ nhớ

MEM16 DW ? ; khai báo 2 byte trống trong bộ nhớ

BMEM DB 50 DUP(?)

; khai báo 50 byte trống trong bộ nhớ

Chương trình dạng .COM

```
CODE SEGMENT
  ASSUME  CS:CODE , DS:CODE, SS:CODE
; toàn bộ chương trình chỉ nằm trong 1 segment
Org  100h ;; chỉ thị nạp thanh ghi lệnh IP=100h khi CT được nạp
Main proc
   mov ax,bx
 ……..
Main endp
Count db  10
……..
Code ends
End main
```

SUMMARY

- chương trình Assembly gồm nhiều dòng lệnh.
- Mỗi lệnh phải viết trên 1 dòng
- Lệnh có thể gồm [tên] [toán tử] [toán hạng]
- Các ký tự phải đặt trong dấu ‘ ‘ hay “ “
- DB dùng để định nghĩa biến kiểu BYTE
- DW dùng để định nghĩa biến kiểu WORD.
- Có 2 cách xuất nhập dữ liệu : liên lạc trực tiếp qua cổng hay dùng các phục vụ ngắt của DOS và BIOS.

Câu hỏi ôn tập

- Trong mã máy dưới đây được lấy từ tập tin liệt kê, hãy nêu ý nghĩa của R

5B 0021 R ADD BX, VAL1

- Nêu ý nghĩa của ký hiệu địa chỉ của biến dưới đây trong 1 tập tin liệt kê.

5B 0021 R ADD BX, VAL1

Câu hỏi ôn tập

- Chương trình sau có lỗi. Hãy tìm câu lệnh nào gây ra lỗi, giải thích và sửa lại cho đúng.

```
.MODEL SMALL
.STACK 100H
.DATA
  MOV AX, VALUE1
  MOV BX, VALUE2
  INC BX, 1
  INT 21H
  MOV 4C00H, AX
  MAIN ENDP
  VALUE1    0AH
  VALUE2    1000H
  END MAIN
```

Câu hỏi ôn tập

- Chương trình sau có lỗi. Hãy tìm câu lệnh nào gây ra lỗi, giải thích và sửa lại cho đúng.

```
.MODEL SMALL
.STACK 100H
.CODE
MAIN PROC
  MOV AX, @DATA
  MOV DS , AX
  MOV AX, VALUE1
   MOV AX, VALUE2
  MOV AX, 4C00H
INT 21H
MAIN ENDP
VALUE1 DB  0AH
VALUE2 DB 1000H
END MAIN
```

Bài tập lập trình

Bài 1 : Viết chương trình nhập 1 ký tự thường , in ra ký tự hoa tương ứng.

Bài 2 : Viết chương trình hoán vị 2 biến kiểu byte được gán sẵn trị.

Bài 3 : Viết chương trình tạo 1 array có các phần tử 31h,32h,33h,34h.

Nạp từng phần tử vào thanh ghi DL và xuất nó ra màn hình. Giải thích tại sao kết xuất trên màn hình là 1234.