



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
- ☐ Cách tạo và chạy chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
 - ❖ Cú pháp của chương trình hợp ngữ
 - ❖ Dữ liệu cho chương trình
 - ❖ Biến và hằng
 - ❖ Khung của một chương trình hợp ngữ
- ☐ Cách tạo và chạy một chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
 - ❖ **Cú pháp của chương trình hợp ngữ**
 - ❖ Dữ liệu cho chương trình
 - ❖ Biến và hằng
 - ❖ Khung của một chương trình hợp ngữ
- ☐ Cách tạo và chạy một chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Cú pháp của chương trình hợp ngữ

- **Tên Mã lệnh Các toán hạng ; chú giải**
- Chương trình dịch không phân biệt chữ hoa, chữ thường
- Trường tên:
 - ☐ chứa các nhãn, tên biến, tên thủ tục
 - ☐ độ dài: 1 đến 31 ký tự
 - ☐ tên không được có dấu cách, không bắt đầu bằng số
 - ☐ được dùng các ký tự đặc biệt: ? . @ _ \$ %
 - ☐ dấu . phải được đặt ở vị trí đầu tiên nếu sử dụng
 - ☐ Nhãn kết thúc bằng dấu :
 - ☐ Ví dụ:

TWO_WORD

?1

two-word

.@?

1word

Let's_go



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
 - ❖ Cú pháp của chương trình hợp ngữ
 - ❖ **Dữ liệu cho chương trình**
 - ❖ Biến và hằng
 - ❖ Khung của một chương trình hợp ngữ
- ☐ Cách tạo và chạy một chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Dữ liệu cho chương trình

- Dữ liệu:
 - ☐ các số hệ số 2: 0011B
 - ☐ hệ số 10: 1234
 - ☐ hệ số 16: 1EF1H, 0ABBAH
 - ☐ Ký tự, chuỗi ký tự: 'A', "abcd"



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
 - ❖ Cú pháp của chương trình hợp ngữ
 - ❖ Dữ liệu cho chương trình
 - ❖ **Biến và hằng**
 - ❖ Khung của một chương trình hợp ngữ
- ☐ Cách tạo và chạy một chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Biến và hằng

- **DB (Define Byte):** định nghĩa biến kiểu byte
- **DW (Define Word):** định nghĩa biến kiểu từ - 2 byte
- **DD (Define Double word):** định nghĩa biến kiểu từ kép - 4 byte

- **Biến byte:**

☐ **Tên** **DB** gia_trị_khởi đầu

☐ Ví dụ:

⇒ B1	DB	4
⇒ B1	DB	?
⇒ C1	DB	'\$'
⇒ C1	DB	34



Biến và hằng

• Biến từ:

☐ **Tên** **DW** **gia_trị_khởi đầu**

☐ Ví dụ:

⇒ W1 DW 4

⇒ W2 DW ?

• Biến mảng:

☐ M1 DB 4, 5, 6, 7, 8, 9

☐ M2 DB 100 DUP(0)

☐ M3 DB 100 DUP(?)

☐ M4 DB 4, 3, 2, 2 DUP (1, 2 DUP(5), 6)

☐ M4 DB 4, 3, 2, 1, 5, 5, 6, 1, 5, 5, 6

1300A		M1
13009		
13008	9	
13007	8	
13006	7	
13005	6	
13004	5	
13003	4	
13002		
13001		
13000		



Biến và hằng

• Biến mảng 2 chiều:

$$\begin{pmatrix} 1 & 6 & 3 \\ 4 & 2 & 5 \end{pmatrix}$$

□ M1 DB 1, 6, 3
DB 4, 2, 5

□ M2 DB 1, 4
DB 6, 2
DB 3, 5

1300A		M1
13009		
13008	5	
13007	2	
13006	4	
13005	3	
13004	6	
13003	1	
13002		
13001		
13000		



Biến và hằng

- **Biến kiểu xâu ký tự**

- ☐ STR1 DB 'string'
- ☐ STR2 DB 73h, 74h, 72h, 69h, 6Eh, 67h
- ☐ STR3 DB 73h, 74h, 'r', 'i', 6Eh, 67h

- **Hằng có tên**

- ☐ Có thể khai báo hằng ở trong chương trình
- ☐ Thường được khai báo ở đoạn dữ liệu
- ☐ Ví dụ:
 - ⇒ CR EQU 0Dh ; là carriage return
 - ⇒ LF EQU 0Ah ; LF là line feed
 - ⇒ CHAO EQU 'CR Hello'

 - ⇒ MSG DB CHAO, '\$'



Lập trình hợp ngữ với 8086

☐ Giới thiệu khung của chương trình hợp ngữ

- ❖ Cú pháp của chương trình hợp ngữ
- ❖ Dữ liệu cho chương trình
- ❖ Biến và hằng
- ❖ **Khung của một chương trình hợp ngữ**

☐ Cách tạo và chạy một chương trình hợp ngữ trên máy IBM PC

☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ

☐ Một số chương trình cụ thể



Khung của chương trình hợp ngữ

1.	.Model Small	khai báo kiểu kích thước bộ nhớ
2.	.Stack 100	khai báo đoạn ngăn xếp
3.	.Data	
4.	Tbao DB 'Chuoi da sap xep:', 10, 13	
5.	MGB DB 'a', 'Y', 'G', 'T', 'y', 'Z', 'U', 'B', 'D', 'E',	khai báo đoạn dữ liệu
6.	DB '\$'	
7.	.Code	khai báo đoạn mã lệnh
8.	MAIN Proc	bắt đầu chương trình chính
9.	MOV AX, @Data	;khởi đầu DS
10.	MOV DS, AX	
11.	MOV BX, 10	;BX: số phần tử của mảng
12.	LEA DX, MGB	;DX chỉ vào đầu mảng byte
13.	DEC BX	;số vòng so sánh phải làm
14.	LAP: MOV SI, DX	; SI chỉ vào đầu mảng
15.	MOV CX, BX	; CX số lần so của vòng so
16.	MOV DI, SI	;giá trị đầu là max
17.	MOV AL, [DI]	;AL chứa phần tử max
18.	TIMMAX:	
19.	INC SI	;chỉ vào phần tử bên cạnh
20.	CMP [SI], AL	; phần tử mới > max?
21.	JNG TIEP	;không, tìm max
22.	MOV DI, SI	; dùng, DI chỉ vào max
23.	MOV AL, [DI]	;AL chứa phần tử max
24.	TIEP: LOOP TIMMAX	;tìm max của một vòng so
25.	CALL DOICHO	;đổi cho max với số mới
26.	DEC BX	;số vòng so còn lại
27.	JNZ LAP	;làm tiếp vòng so mới
28.	MOV AH, 9	; hiển thị chuỗi đã sắp xếp
29.	LEA DX, Tbao	
30.	INT 21H	
31.	MOV AH, 4CH	;ve DOS
32.	INT 21H	
33.	MAIN Endp	kết thúc chương trình chính
34.	DOICHO Proc	bắt đầu chương trình con
35.	PUSH AX	
36.	MOV AL, [SI]	
37.	XCHG AL, [DI]	
38.	MOV [SI], AL	
39.	POP AX	
40.	RET	
41.	DOICHO Endp	kết thúc đoạn mã
42.	END MAIN	



Khung của chương trình hợp ngữ

- Khai báo quy mô sử dụng bộ nhớ
 - ❖ **.MODEL Kiểu kích thước bộ nhớ**
 - ❖ Ví dụ: **.Model Small**

Kiểu	Mô tả
Tiny (hẹp)	mã lệnh và dữ liệu gói gọn trong một đoạn
Small (nhỏ)	mã lệnh nằm trong 1 đoạn, dữ liệu 1 đoạn
Medium (trung bình)	mã lệnh nằm trong nhiều đoạn, dữ liệu 1 đoạn
Compact (gọn)	mã lệnh nằm trong 1 đoạn, dữ liệu trong nhiều đoạn
Large (lớn)	mã lệnh nằm trong nhiều đoạn, dữ liệu trong nhiều đoạn, không có mảng nào lớn hơn 64 K
Huge (đồ sộ)	mã lệnh nằm trong nhiều đoạn, dữ liệu trong nhiều đoạn, các mảng có thể lớn hơn 64 K



Khung của chương trình hợp ngữ

- Khai báo đoạn ngăn xếp
 - ❖ **.Stack** kích thước (bytes)
 - ❖ Ví dụ:
 - ❖ **.Stack 100** ; khai báo stack có kích thước 100 bytes
 - ❖ Giá trị ngầm định 1 KB
- ❖ Khai báo đoạn dữ liệu:
 - ❖ **.Data**
 - ❖ Khai báo các biến và hằng
- ❖ Khai báo đoạn mã
 - ❖ **.Code**



Khung của chương trình hợp ngữ

Khung một chương trình ra file .EXE

```
.Model    Small
.Stack    100
.Data
    ;các định nghĩa cho biến và hằng

.Code
MAIN      Proc
    ;khởi đầu cho DS
    MOV     AX, @data
    MOV     DS, AX
    ;các lệnh của chương trình

    ;trở về DOS dùng hàm 4CH của INT
    21H

    MOV     AH, 4CH
    INT     21H

MAIN      Endp
    ;các chương trình con nếu có

END MAIN
```

Khung một chương trình ra file .COM

```
.Model    Tiny
.Code
    ORG     100h
    START:  JMP     CONTINUE
    ;các định nghĩa cho biến và hằng

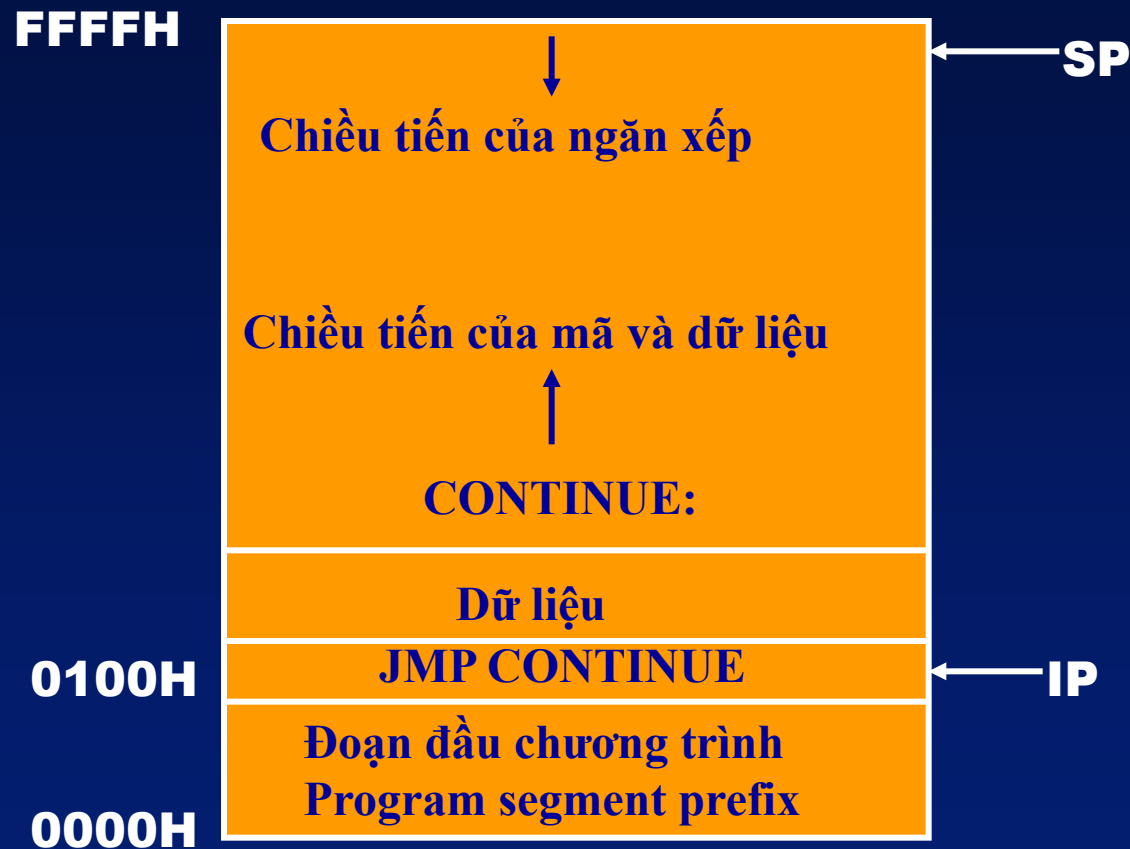
CONTINUE:
MAIN      Proc
    ;các lệnh của chương trình
    INT     20H    ;trở về DOS

MAIN      Endp
    ;các chương trình con nếu có

END START
```




Khung của chương trình hợp ngữ





Khung của chương trình hợp ngữ

- Chương trình Hello.EXE

```

.Model      Small
.Stack      100
.Data

        CRLF      DB      13,10,'$'
        MSG        DB      'Hello! $'

.Code
MAIN      Proc
;khởi đầu cho DS
MOV       AX, @data
MOV       DS, AX
;về đầu dòng mới dùng hàm 9 của INT 21H
MOV       AH,9
LEA       DX, CRLF
INT       21H
;Hiện thị lời chào dùng hàm 9 của INT 21H
MOV       AH,9
LEA       DX, MSG
INT       21H
;về đầu dòng mới dùng hàm 9 của INT 21H
MOV       AH,9
LEA       DX, CRLF
INT       21H
;trở về DOS dùng hàm 4CH của INT 21H
MOV       AH, 4CH
INT       21H

MAIN      Endp
END MAIN

```



Khung của chương trình hợp ngữ

- Chương trình Hello.COM

```
.Model      Tiny
.Code

          ORG      100H
START: JMP CONTINUE

          CRLF      DB      13,10,'$'
          MSG       DB      'Hello! $'

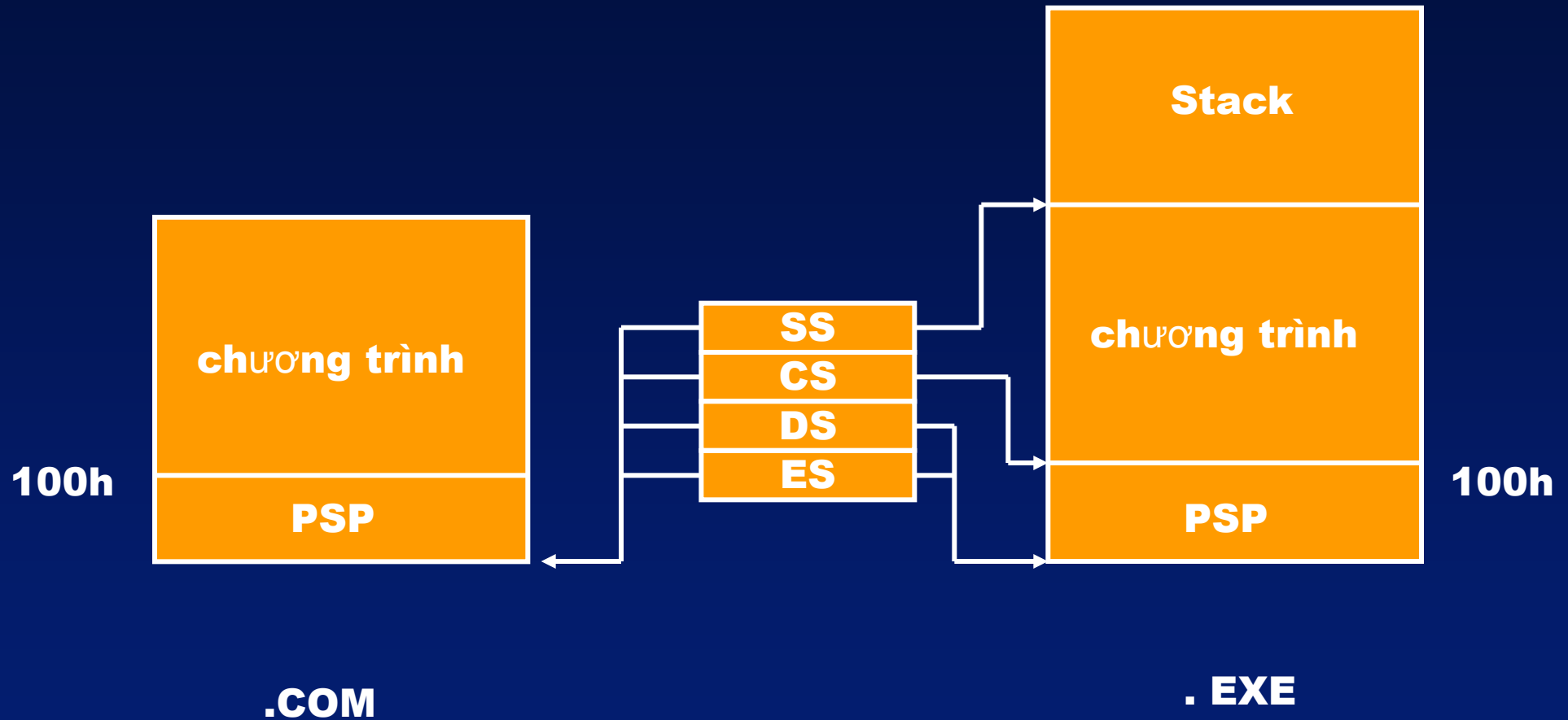
CONTINUE:

MAIN      Proc
          ;về đầu dòng mới dùng hàm 9 của INT 21H
          MOV       AH,9
          LEA       DX, CRLF
          INT       21H
          ;Hiển thị lời chào dùng hàm 9 của INT 21H
          MOV       AH,9
          LEA       DX, MSG
          INT       21H
          ;về đầu dòng mới dùng hàm 9 của INT 21H
          MOV       AH,9
          LEA       DX, CRLF
          INT       21H
          ;trở về DOS
          INT       20H

MAIN      Endp
          END START
```



Khung của chương trình hợp ngữ



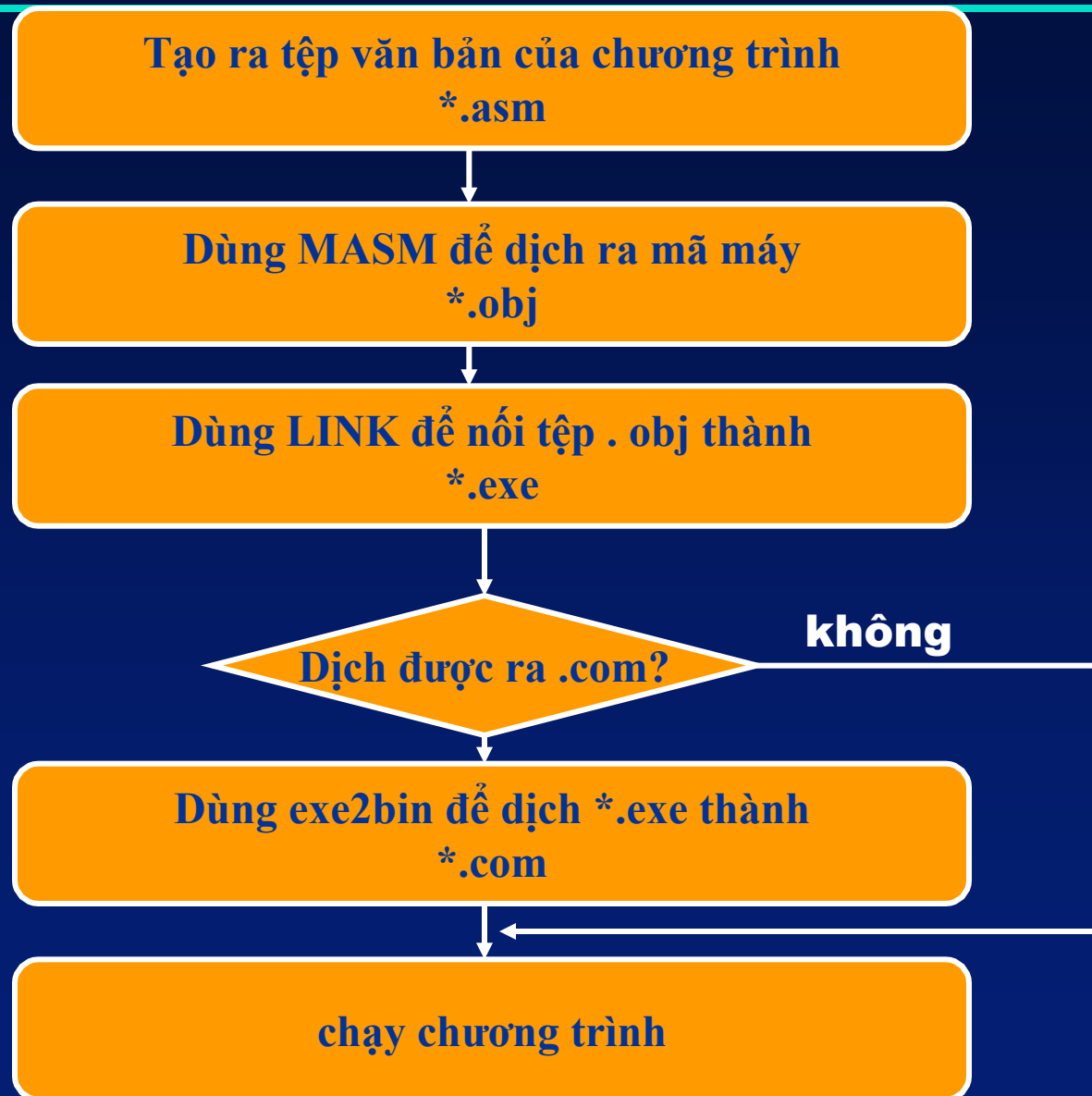


Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
- ☐ Cách tạo và chạy chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Cách tạo một chương trình hợp ngữ





Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
- ☐ Cách tạo và chạy chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
 - Cấu trúc lựa chọn
 - Cấu trúc lặp
- ☐ Một số chương trình cụ thể



Cấu trúc lựa chọn If-then

- If điều_kiện then công_việc
- Ví dụ: Gán cho BX giá trị tuyệt đối của AX

```
                                ; If AX<0
                                ; AX<0 ?
                                ; không, thoát ra
                                ; then
                                ; đúng, đảo dấu
End_if:  CMP    AX, 0
          JNL    End_if
          NEG    AX
          MOV    BX, AX        ;gán
```




Cấu trúc lựa chọn If-then-else

- If điều_kiện **then** công_việc1 **else** công_việc2
- Ví dụ: if AX<BX then CX=0 else CX=1

```
; if AX<BX
CMP      AX, BX      ; AX<BX ?
JL       Then_      ; đúng, CX=0
;else
MOV      CX, 1      ; sai, CX=1
JMP      End_if
Then_: MOV      CX, 0;
End_if:
```



Cấu trúc lựa chọn Case

- **Case Biểu thức**

Giá trị 1: công việc 1

Giá trị 2: công việc 2

...

Giá trị N: công việc N

END CASE

- Ví dụ:

Nếu $AX < 0$ thì $CX = -1$

Nếu $AX = 0$ thì $CX = 0$

Nếu $AX > 0$ thì $CX = 1$

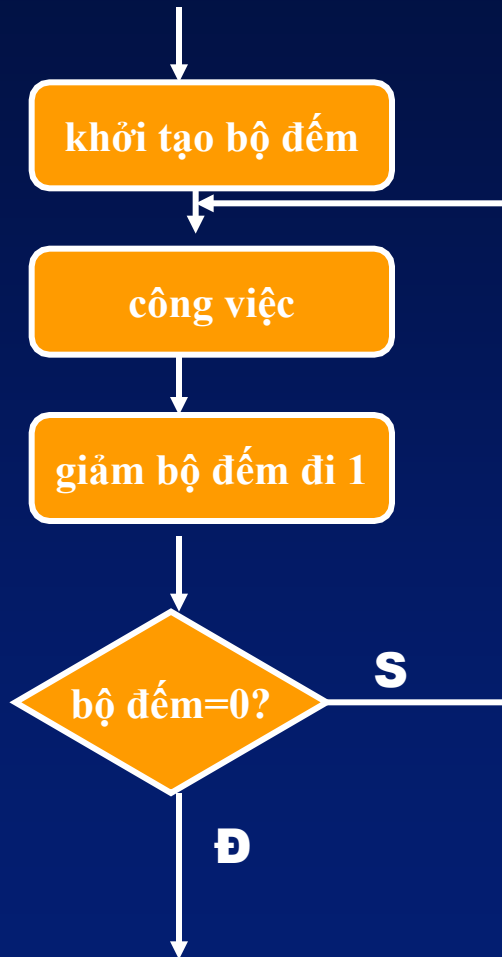
```
CMP      AX, 0 ;
JL       AM    ; AX<0
JE       Khong ; AX=0
JG       DUONG; AX>1
AM: MOV   CX, -1
        JMP   End_case
Khong: MOV CX, 0
        JMP   End_case

DUONG: MOV CX, 1
End_case:
```



Cấu trúc lặp For-Do

- **For** số lần lặp **Do** công việc



ví dụ: Hiển thị một dòng ký tự \$ trên màn hình

```
MOV CX, 80          ;số lần lặp
MOV AH, 2           ;hàm hiển thị
MOV DL, '$'         ;DL chứa ký tự cần hiển thị
HIEN: INT 21H        ; Hiển thị
      LOOP HIEN
;End_for
```



Cấu trúc lặp While-Do

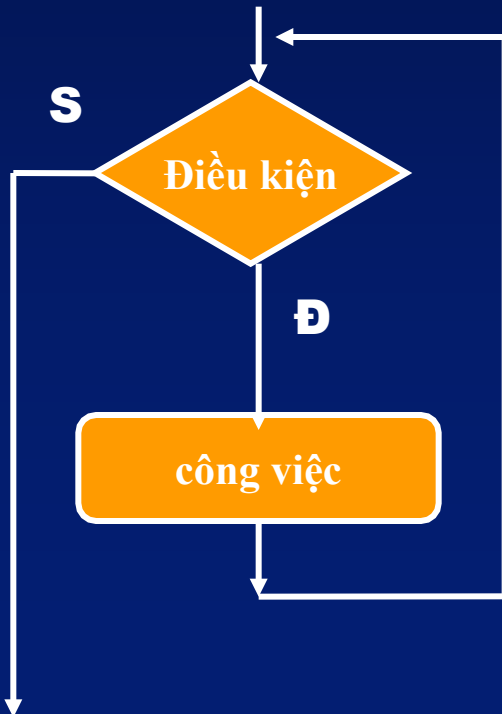
- While** điều kiện **Do** công việc

ví dụ:

Khởi tạo AX=0, BX=0

Trong khi AX<>10 thì

BX=BX+1 và AX=AX+2



```
XOR AX, AX      ;AX=0
```

```
XOR BX, BX      ;BX=0
```

TIEP:

```
CMP AX,10       ;so sánh AX với 10
```

```
JE End_while    ;kết thúc nếu AX=10
```

```
INC BX          ;BX=BX+1
```

```
ADD AX,2        ;AX=AX+2
```

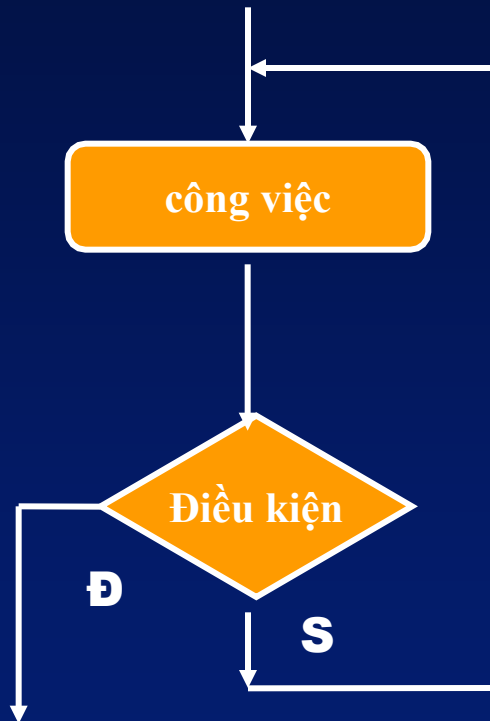
```
jmp tiep
```

End_while:



Cấu trúc lặp Repeat-until

- **Repeat** công việc **until** điều kiện



ví dụ: đọc từ bàn phím cho tới khi gặp ký tự CR thì thôi

```
MOV AH,1           ;hàm đọc ký tự từ bàn phím
TIEP:  INT 21H      ; đọc một ký tự vào AL
      CMP AL, 13    ; đọc CR?
      JNE TIEP      ; chưa, đọc tiếp
End_:
```



Lập trình hợp ngữ với 8086

- ☐ Giới thiệu khung của chương trình hợp ngữ
- ☐ Cách tạo và chạy chương trình hợp ngữ trên máy IBM PC
- ☐ Các cấu trúc lập trình cơ bản thực hiện bằng hợp ngữ
- ☐ Một số chương trình cụ thể



Xuất nhập dữ liệu

- 2 cách:
 - ❑ Dùng lệnh IN, OUT để trao đổi với các thiết bị ngoại vi
 - ⇒ phức tạp vì phải biết địa chỉ cổng ghép nối thiết bị
 - ⇒ Các hệ thống khác nhau có địa chỉ khác nhau
 - ❑ Dùng các chương trình con phục vụ ngắt của DOS và BIOS
 - ⇒ đơn giản, dễ sử dụng
 - ⇒ không phụ thuộc vào hệ thống
- Ngắt 21h của DOS:
 - ❑ Hàm 1: đọc 1 ký tự từ bàn phím
 - ⇒ Vào: AH=1
 - ⇒ Ra: AL=mã ASCII của ký tự, AL=0 khi ký tự là phím chức năng
 - ❑ Hàm 2: hiện 1 ký tự lên màn hình
 - ⇒ Vào: AH=2
 - DL=mã ASCII của ký tự cần hiển thị
 - ❑ Hàm 9: hiện chuỗi ký tự với \$ ở cuối lên màn hình
 - ⇒ Vào: AH=9
 - DX=địa chỉ lệch của chuỗi ký tự cần hiển thị
 - ❑ Hàm 4CH: kết thúc chương trình loại .exe
 - ⇒ Vào: AH=4CH



Một số chương trình cụ thể

- Ví dụ 1: Nhập 1 kí tự và hiển thị kí tự đó ra màn hình
- Ví dụ 2: Nhập 1 kí tự thường và in ra kí tự hoa
 - ❖ Ví dụ:
 - ⇒ Hay nhap vao mot chu cai thuong: a
 - ⇒ Kí tự hoa là: A
- Ví dụ 3: Nhập 1 chuỗi kí tự từ bàn phím và in chuỗi đó ra màn hình
- Ví dụ 4: Nhập 1 chuỗi kí tự từ bàn phím và xuất chuỗi ngược lại ra màn hình
- Ví dụ 5: Nhập 1 chuỗi kí tự từ bàn phím và in ra độ dài của chuỗi đó
- Ví dụ 6: Tính tổng 2 số kiểu byte
- Ví dụ 7: Tính hiệu 2 số kiểu byte



Một số chương trình cụ thể

- Ví dụ 8: Tính tổng 2 số kiểu word
- Ví dụ 9: Tính hiệu 2 số kiểu word
- Ví dụ 10: Nhập 1 số kiểu word và in ra mã nhị phân của số đó