CHƯƠNG TRÌNH GỠ RỐI DEBUG

Mục tiêu

- Dịch được 1 chương trình ngắn
- Xem các thanh ghi và cờ của CPU
- Xem sự thay đổi nội dung của các biến
- Dò tìm trị ở dạng nhị phân hoặc ASCII trong bộ nhớ
- **Hỗ trợ luyện tập viết chương trình bằng Assembly**

Dạng lệnh của Debug

- **<mã lệnh > <thông số>**

  Trong đó mã lệnh là 1 trong các chữ A,B,C,D,E, ... còn thông số thì thay đổi tùy theo lệnh.

Các thông số có thể là :

Địa chỉ : là 1 bộ địa chỉ đầy đủ segment : offset hay chỉ cần offset là đủ. Segment có thể dùng tên thanh ghi.

Ex : **F000:0100**

DS: 200

0AF5

Dạng **lệnh của Debug**

Tập tin : là 1 tham khảo tên tập tin đầy đủ, ít nhất phải có tên tập tin.

Danh sách :

Là 1 hay nhiều trị byte hoặc chuổi cách nhau bằng dấu phẩy.

Khoảng : là 1 tham khảo đến vùng bộ nhớ

Trị : là 1 số hệ 16 có tối đa có 4 chữ số

Tập lệnh của Debug

- A <Assemble> :
  cho phép viết từ bàn phím các lệnh mã máy dưới dạng gợi nhớ.
- A [<địa chỉ>]
  Ex : - A 100 dịch ở địa chỉ CS:100h
  - A dịch ở địa chỉ hiện tại
    (Debug lấy địa chỉ đoạn CS)
  - A DS:2000h
    dịch ở địa chỉ **DS:2000h**

Thí dụ minh họa lệnh A

- Phải nhập lệnh vào theo từng dòng một và kết thúc bằng Enter.
- Kết thúc nhập nhấn Enter ở dòng trống.
- **Ex : - A 100**

```
5514:0100      MOV  AH, 2
5514:0102      MOV  DL, 41
5514:0104        INT  21H
```

User gỏ vào

SEGMENT **OFFSET**

C (Compare)

- So sánh 2 vùng bộ nhớ và liệt kê các ô nhớ có nội dung khác nhau.

Cú pháp : C <khoảng> , < địa chỉ>

Ex : - C 100, 200, 3000 : 1000

So sánh ô nhớ DS:100h với ô nhớ 3000:1000h, ô nhớ DS:101h với ô nhớ 3000:1001h..... Cho đến ô nhớ DS :200h với ô nhớ 3000:1100h.

➔ So sánh 101 bytes

D (Dump)

- Hiện nội dung bộ nhớ theo dạng hệ 16 và ASCII.

Cách gọi : D <khoảng>

Ex : - D F000 : 0

- D ES : 100

- D 100

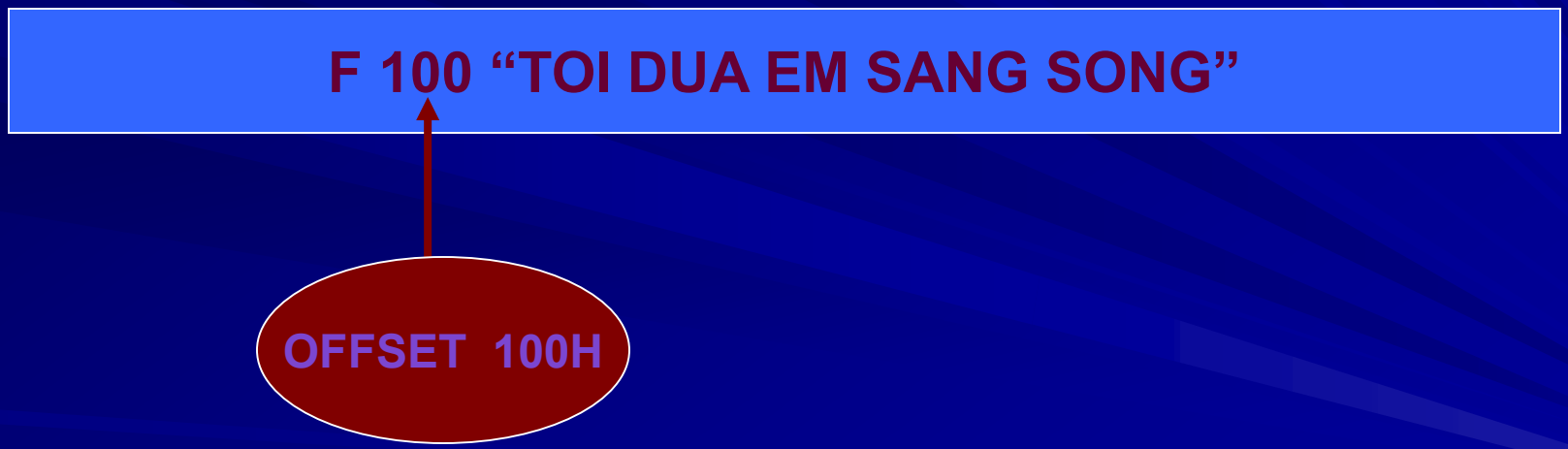
Lệnh F (**Fill**)

- Cú pháp : F <khoảng> <danh sách>
- Công dụng : lấp đầy trị vào vùng nhớ ngay tại địa chỉ mong muốn.

Trị nhập vào từng byte một theo hệ 16
Dấu trừ (-) dùng để lùi lại 1 địa chỉ.
SPACE BAR dùng để tới 1 địa chỉ.
ENTER để kết thúc.

Minh họa lệnh **F**

- Lắp đầy vùng nhớ tại địa chỉ offset 100h chuổi " Toi dua em sang song".



KẾT QUẢ

-F 100 "TOI DUA EM SANG SONG"
-D 100

```
0ADD:0100  54 4F 49 20 44 55 41 20-45 4D 20 53 41 4E 47 20   TOI DUA EM SANG
0ADD:0110  53 4F 4E 47 54 4F 49 20-44 55 41 20 45 4D 20 53   SONGTOI DUA EM S
0ADD:0120  41 4E 47 20 53 4F 4E 47-54 4F 49 20 44 55 41 20   ANG SONGTOI DUA
0ADD:0130  45 4D 20 53 41 4E 47 20-53 4F 4E 47 54 4F 49 20    EM SANG SONGTOI
0ADD:0140  44 55 41 20 45 4D 20 53-41 4E 47 20 53 4F 4E 47   DUA EM SANG SONG
0ADD:0150  54 4F 49 20 44 55 41 20-45 4D 20 53 41 4E 47 20   TOI DUA EM SANG
0ADD:0160  53 4F 4E 47 54 4F 49 20-44 55 41 20 45 4D 20 53   SONGTOI DUA EM S
0ADD:0170  41 4E 47 20 53 4F 4E 47-54 4F 49 20 44 55 41 20   ANG SONGTOI DUA
```

D (DUMP)

Mục đích : in nội dung bộ nhớ trong MT ra màn hình dưới dạng số hex.

Cú pháp : D [address]

D [range]

Ex : in nội dung vùng nhớ đã lắp đầy ở ví dụ trước ở địa chỉ 100h

Ex2 : xem nội dung vùng nhớ 16 bytes bắt đầu ở địa chỉ F000:100

```
- D F000:100  L10
```

Thí dụ minh họa lệnh **D**

- đánh vào lệnh D để xem nội dung vùng nhớ của 30h bytes bộ nhớ từ địa chỉ 0000:0040 đến 0000:006F

```
- D 0000:0040   006F
```

Địa chỉ bắt đầu

```
- D  0000:0040  L  30
```

Số bytes

E (ENTER)

- Dùng để đưa dữ liệu byte vào bộ nhớ ngay tại địa chỉ mong muốn.

Cách gọi :

- E <địa chỉ> <danh sách>

Trị nhập vào theo dạng số 16 từng byte một
Dấu - dùng để lùi lại 1 địa chỉ
Space Bar dùng để tới 1 địa chỉ
Enter dùng để kết thúc

Minh họa lệnh E

- Mục dích : thay đổi nội dung bộ nhớ.

Cú pháp : - E [address] [list]

Ex : thay đổi 6 bytes bắt đầu ở địa chỉ 100 thành "ABCDE"

- E 100 "ABCDE"

Debug lấy đoạn chỉ bởi DS
Nếu ta không qui định địa chỉ đoạn

Lệnh U (Unassemble)

- công dụng : in ra 32 bytes mã máy của chương trình trong bộ nhớ ra màn hình dưới lệnh gợi nhớ.
- cú pháp : U [address]

  U [range]

Ex : U 100 119

In ra màn hình các lệnh mã máy từ địa chỉ CS:100 đến CS:119

Lệnh R (Register)

- **Công dụng : xem và sửa nội dung thanh ghi.**
- **Cú pháp : - R enter (xem tất cả thanh ghi)**

xem thanh ghi AX : - R AX

xem thanh ghi cờ : R F

Ex : muốn bật thanh ghi cờ CF và ZF ta nhập CY và ZR.

Lệnh N (Name)

- Công dụng : tạo tập tin cần đọc hay ghi trước khi dùng lệnh L hay W.
- Cú pháp : - N <tên file> [thông số] L [địa chỉ]

Thí dụ minh họa lệnh N

Ex : tạo tập tin Love.txt .

- **Dùng lệnh R để xác định vùng địa chỉ dành cho User.**
- **Dùng lệnh để đưa câu thông báo " I love you more than I can say' ở địa chỉ 2000:100.**
- **Dùng lệnh D để kiểm tra vùng nhớ tại địa chỉ 2000:100.**
- **Dùng lệnh N để đặt tên tập tin trên đĩa.**
  - N Love.txt
- **Dùng lệnh R để định số byte cần thiết ghi lên đĩa trong 2 thanh ghi BX và CX. Cụ thể trong trường hợp này số byte cần ghi là 1Eh byte.**
  BX = 0000 CX = 1E
- **Dùng lệnh W 2000:100 để ghi dữ liệu đã nhập vào tập tin ở địa chỉ bộ nhớ 2000:100.**

- Thoát khỏi Debug và gọi lại tập tin theo cách sau :
  C :\> Debug Love.txt
  tìm xem Debug đã nạp tập tin Love.txt vào chỗ nào trong bộ nhớ.

Lệnh W (Write)

- Cú pháp : W [address]

Thường được sử dụng chung với lệnh N

Ex : tạo tập tin có tên Love.txt

Bước 1 : dùng lệnh E để đưa câu ‘I love you more than I can say” vào ô nhớ ở địa chỉ 100.

Bước 2 : dùng lệnh D để kiểm tra lại địa chỉ 100

Bước 3 : dùng lệnh N để đặt tên tập tin : - N Love.txt

Bước 4 : dùng lệnh R để định số byte cần ghi lên đĩa trong 2 thanh ghi BX và CX. (BX chứa 16 bit cao, CX chứa 16 bit thấp).

Ở đây số byte cần ghi là 1Eh.

Bước 5 : dùng lệnh W để ghi câu trên đã nhập vào vùng nhớ có địa chỉ bắt đầu là 100.

Lệnh T (Trace)và P

- **cú pháp : - T [= <địa chỉ>][số lần]**

Mục đích : dùng để chạy 1 hay nhiều lần các lệnh trong bộ nhớ

Ex : - T = 3000:1000

Ex : - T = 3000:1000 **<số lần>**

Lệnh L (Load)

- nạp tập tin hoặc nạp sector luận lý từ đĩa vào bộ nhớ.

Cú pháp : - L <địa chỉ> [<đĩa> <sector><số>]

Dạng 1 : nếu chỉ có địa chỉ dùng để nạp tập tin. Tên tập tin phải được gán trước bằng lệnh N.

Tập tin luôn luôn được gán ở địa chỉ offset 100h

Dạng 2 : nếu có đầy đủ các thông số , dùng để đọc sector luận lý trên đĩa vào bộ nhớ.

Đĩa : = 0 ổ đĩa A, =1 ổ đĩa B, =2 ổ đĩa C ....

Lệnh H (Hex Arithmethic)

- thực hiện phép cộng và trừ hệ 16

Cú pháp : - H <trị 1> <trị 2>

Kết quả : hiện ra tổng và hiệu của trị 1 và trị 2

Lệnh S (Search)

- Công dụng : tìm kiếm trị trong 1 vùng bộ nhớ.
- Cú pháp : - S <khoảng> <danh sách>
- Giải thích : tìm kiếm trị có hiện diện trong vùng bộ nhớ đã chỉ định hay không? Nếu có Debug hiện các địa chỉ đầu của những nơi có chứa danh sách.
  Ex : - S 100 L 1000 ‘DOS’
  18AF : 0154
  18AF : 0823

Ex2 : - S 2000 2200 13,15,8A, 8

Lệnh M (Move)

- Công dụng : chép nội dung vùng nhớ đến 1 địa chỉ khác.
- Cú pháp : - M <khoảng>
- Ex : - M 100 105 200

Chép 5 bytes từ DS:100 đến DS:200

Ex2 : - M CS:100 L 50 ES:300
Chép 50 bytes từ CS:100 đến ES:300

Lệnh I (Input)

- Công dụng : nhập 1 byte từ cổng xuất nhập và hiện ra màn hình.
- Cú pháp : - I <địa chỉ cổng> địa chỉ cổng là số hệ 16 tối đa 4 chữ số.
- Ex : - I 37E
  EC

Lệnh O (Output)

- Công dụng : xuất 1 byte ra cổng xuất nhập.
- Cú pháp :- O<địa chỉ cổng> <trị>
  địa chỉ cổng là số hệ 16 tối đa 4 chữ số.
- Ex : - O 378 5E

Summary

- Dùng lệnh D để xem nội dung vùng nhớ tại địa chỉ của ROM BIOS F000:0000.
- Tương tự xem nội dung vùng nhớ RAM màn hình ở địa chỉ B800:0000; bảng vector ngắt quãng 0000:0000
- Gỏ vào máy bằng lệnh A, đoạn chương trình sau ở địa chỉ 2000:0100

Summary

2000:0100 MOV AL,32
2000:0102 MOV AH, 4F
2000:0104 MOV CX, [200]
2000:0108 MOV WORD PTR [1800], 1
2000:010E MOV BYTE PTR [1800], 1
2000:0113

Xem lại đoạn chương trình vừa đánh trên bằng lệnh U. Chú ý quan sát phần mã máy. Tìm xem các toán hạng tức thời và các địa chỉ xuất hiện ở đâu trong phần mã máy của lệnh.

Phần mã máy của 2 câu lệnh cuối có gì khác nhaukhi dùng các toán tử WORD PTR và BYTE PTR.

Summary

- Dùng lệnh E nhập vào đoạn văn bản sau vào bộ nhớ tại địa chỉ DS:0100

8086/8088/80286 Assembly language.

Copyright 1988, 1886 by Brady Books, a division of Simon, Inc.

All right reserved, including the of reproduction in whole or in part, in any form.

(chú ý ký tự đầu dòng xuống dòng có mã ASCCI là 0D và 0A).