**Lệnh Int**

**Cú pháp:**

**Int**          **<**n**>**

**Trong đó:**Trong đó <n> là số hiệu ngắt của một ngắt mềm cần gọi. Tức là, n có thể là một trong các giá trị từ 0 đến 255, đó chính là số hiệu ngắt của 256 ngắt mềm mà BIOS và MSDOS cung cấp.

**Tác dụng:**Lệnh Int (**Int**errupt) được sử dụng để gọi một ngắt mềm (của BIOS hoặc MSDOS) trong chương trình hợp ngữ. Khi một ngắt mềm được gọi thì hệ thống sẽ thực hiện chương trình con phục vụ ngắt tương ứng với nó.

**Ví dụ 1:**

                       Int           10h                 ; gọi ngắt 10h của BIOS

                       Int           20h                 ; gọi ngắt 20h của MSDOS

Một ngắt mềm, hay chính xác hơn là một chương trình con phục vụ ngắt, có thể thực hiện nhiều chức năng khác nhau, mỗi chức năng này được thể hiện thông qua một con số, được gọi là số hiệu hàm của ngắt. Do đó, trước khi gọi ngắt chương trình phải chỉ rõ gọi ngắt với hàm nào, bằng cách đặt số hiệu hàm cần gọi vào thanh ghi AH.

**Ví dụ 2:**

        Mov        Ah, 01                        ; gọi ngắt 21h với hàm 01, Hay nói ngược lại: gọi hàm

        Int           21h                 ; 01 của ngắt 21h

Trước khi gọi hàm/ngắt chương trình cần cung cấp đầy đủ dữ liệu vào (nếu có) cho nó, sau khi hàm/ngắt được thực hiện chương trình cần xác định rõ nó có trả về dữ liệu kết quả (dữ liệu ra) hay không, nếu có thì chứa ở đâu: thanh ghi hay ô nhớ, và có tác động đến các cờ hiệu hay không.

**Ví dụ 3:**

        Mov        Ah, 02                                    ; đặt số hiệu hàm cần gọi vào AH

        Mov        DH, 10                        ; cung cấp dữ liệu vào thứ nhất vào DH

        Mov        DL, 20                        ; cung cấp dữ liệu vào thứ hai vào DL

        Int           10h                             ; gọi ngắt 10h với hàm 02. Hàm/ngắt này không

                                                            ; trả về dữ liệu kết quả.

Dãy lệnh trên thực hiện việc gọi hàm 02 của ngắt 10h (ngắt của BIOS), nó thực hiện việc dịch chuyển con trỏ đến dòng 10 cột 20 của màn hình văn bản.

**Ví dụ 4:**

        Mov        Ah, 2Bh                      ; gọi ngắt 21h với

        Int           21h                             ; hàm 2Bh

Hàm này trả về ngày-tháng-năm hiện tại (theo đồng hồ hệ thống trên máy tính). Cụ thể: Thanh ghi CX (1980-2099) chứa năm hiện tại, thanh ghi DH (1-12) chứa tháng hiện tại, thanh ghi DL (1-31) chứa ngày hiện tại. Đồng thời AL cho biết ngày trong tuần (0 : chủ nhật, 6 : thứ 7).

**Một số hàm của ngắt 21h (MSDOS)**

*Ngắt 21h của MSDOS là ngắt thường được sử dụng nhất, nên ở đây chúng tôi chọn giới thiệu về  ngắt này, nhưng chỉ với các hàm vào/ra kí tự/xâu kí tự cơ bản. Chức năng của mỗi  ngắt, chức năng của các hàm trong một ngắt, các yêu cầu dữ liệu vào/ra của từng hàm trong mỗi ngắt,... dễ dàng tìm thấy trong các tài liệu về lập trình hệ thống.*

**Hàm 02 của ngắt 21h:**

**Tác dụng:**In một kí tự ra màn hình. Kí tự (hoặc mã ASCII của kí tự) cần in được đặt trước trong thanh ghi DL. Kí tự ở đây có thể là kí tự thường hoặc kí tự điều khiển.

**Sử dụng:**

**Vào:**Ah = 02

            Dl = <Kí tự cần in ra>

**Ra:**Không có

**Ví dụ 1:** Các lệnh sau đây in ra màn hình kí tự A:

            Mov          Ah, 02

            Mov          Dl, ‘A’                  ;có thể viết lệnh  **Mov   Dl, 41h**

            Int             21h                       ; 41h là mã ASCII của kí tự A

**Ví dụ 2:** Các lệnh sau đây in ra màn hình 10 kí tự, bắt đầu từ kí tự A:

            Mov          Cx, 10

            Mov          Ah, 02

            Mov          Dl, ‘A’

     Lap\_In:

            Int             21h

            INC           DL

            Loop         Lap\_In

**Ví dụ 3:** Các lệnh sau đây in xâu kí tự từ trong biến TieuDe ra màn hình. Giả sử rằng biến TieuDe đã được khai báo như sau:

            TieuDe     DB            ‘Khoa CNTT Truong DHKH Hue’

Các lệnh:

            Lea            DI, TieuDe

            Mov          Cx, 25

            Mov          Ah, 02

     Lap\_In:

            Mov          Dl, Byte PTR [DI]

            Int             21h

            INC           DI

            Loop         Lap\_In

            ;---------------------------

**Ví dụ 4:** Giả sử tại địa chỉ 0A00:0100 trong bộ nhớ có chứa một xâu kí tự ASCII, gồm 50 kí tự. Các lệnh sau đây sẽ in xâu kí tự nói trên ra màn hình.

            Mov          Ax, 0A00h

            Mov          ES, Ax

            Mov          DI, 0100h

            ;-------------------------

            Mov          Cx, 50

            Mov          Ah, 02

      Lap\_In:

            Mov          Dl, Byte PTR ES:[DI]

            Int             21h

            INC           DI

            Loop         Lap\_In

            ;--------------------------------------

**Hàm 09 của ngắt 21h:**

**Tác dụng:**In một dãy kí tự (một xâu kí tự) ra màn hình. Địa chỉ của xâu cần in này phải được chỉ bởi cặp thanh ghi DS:DX và xâu phải được kết thúc bởi dấu $.

**Sử dụng:**

**Vào:**Ah = 09

            DS:DX = <Segment:Offset của xâu cần in ra>

**Ra:**Không có

**Ví dụ 1:**Giả sử chương trình đã khai báo biến **TieuDe**. Viết lệnh in nội dung của biến **TieuDe** ra màn hình:

-     Trong trường hợp này biến TieuDe phải được khai báo trước như sau:

**TieuDe**     DB   ‘Truong DH Khoa hoc Hue$’

-     Và đoạn lệnh gồm các lệnh sau:

            Mov          Ah, 09

            Mov          Ax, Seg **TieuDe**

            Mov          DS, Ax

            Mov          Dx, Offset **TieuDe**; có thể dùng lệnh **Lea      TieuDe**

            Int             21h

Trong thự tế, với các chương trình hợp ngữ viết ở dạng đoạn giản đơn, thì không cần đưa địa chỉ Segment của biến cần in vào DS. Bởi vì:

-     Đối với các chương trình dạng COM:

-     Đối với các chương trình dạng EXE:

**Ví dụ 2:**Giả sử biến TieuDe đã được khai báo như sau:

**TieuDe**     DB            ‘Khoa CNTT Truong DHKH Hue$’

Các lệnh sau chỉ in các kí tự “Truong DHKH Hue” từ biến TieuDe ra màn hình:

            Mov          Ax, Segment **TieuDe**

            Mov          DS, Ax

            Mov          Dx, **TieuDe**

            Add           Dx, 11

            Mov          Ah, 09

            Int             21h

Các lệnh sau chỉ in các kí tự “Khoa CNTT” từ biến TieuDe ra màn hình:

            Mov          Ax, Segment **TieuDe**

            Mov          DS, Ax

            Mov          Dx, **TieuDe**

            Mov          DI, Dx

            Add           DI, 10

**Mov          Byte PTR DS:[DI], ‘$’**

            Mov          Ah, 09

            Int             21h

**Ví dụ 3:** Giả sử tại địa chỉ 0A00:0100 trong bộ nhớ có chứa một xâu kí tự ASCII, gồm 50 kí tự. Các lệnh sau đây sẽ in xâu kí tự nói trên ra màn hình.

            Mov          Ax, 0A00h

            Mov          DS, Ax

            Mov          Dx, 0100h

            ;-------------------------

            Mov          DI, Dx

            Mov          Cx, 50

            Add           DI, Cx

            Mov          Byte PTR DS:DX, ‘$’

            ;-------------------------------------

            Mov          Ah, 09

            Int             21h

            ;-------------------

**Hàm 01 của ngắt 21h:**

**Tác dụng:**Nhập một kí tự từ bàn phím vào lưu trong thanh ghi AL. Cụ thể là, AL sẽ chứa mã ASCII của kí tự ghi trên phím nhập.

**Sử dụng:**

**Vào:**Ah = 01

**Ra:**Al = 0: nếu phím nhập là một trong các phím chức năng

            Al = < mã ASCII của kí tự ghi trên phím nhập>

Cụ thể như sau: Khi chương trình gọi ngắt 21h với hàm 01 thì màn hình sẽ xuất hiện một con trỏ nhập, để người sử dụng nhập vào một kí tự từ bàn phím. Khi đó, nếu người sử dụng gõ một trong các phím chức năng thì AL nhận được giá trị 0, nếu người sử dụng gõ một phím kí tự nào đó thì AL nhận được mã ASCII của kí tự đó.

**Chú ý:**Hàm 08 của ngắt 21h có chức năng tương tự hàm 01 ngắt 21h nhưng kí tự trên phím gõ không xuất hiện trên màn hình, tất cả đều được xuất hiện bởi kí tự ‘\*’.

**Ví dụ 1:**

            Mov          Ah, 01                  ; với hai lệnh này màn hình sẽ xuất hiện con trỏ

            Int             21h                       ; nhập để người sử dụng nhập một kí tự vào AL

**Ví dụ 2:**Các lệnh sau đây cho phép nhập một xâu kí tự, với số lượng kí tự được ấn định trước, vào biến LuuXau đã khai báo trước trong chương trình

Giả sử biến LuuXau được khai báo như sau:

**LuuXau**         DB      30 Dup (‘ ‘)

Các lệnh:

            Mov          Ax, Seg **LuuXau**

            Mov          DS, Ax

            Lea            DI, **LuuXau**

            ;--------------------------------

            Mov          Cx, 30

            Mov          Ah, 01

     Nhap\_Xau:

            Int             21h

            Mov          Byte PTR DS:[DI], Al

            INC           DI

            Loop         Nhap\_Xau

            ;--------------------------------------

Trong trường hợp này chúng ta đã giả sử: Người sử dụng chỉ nhập các kí tự (gõ phím kí tự để nhập), không nhập các phím chức năng.

Trong thực tế, không bao giờ chúng ta sử dụng hàm 01 ngắt 21h để nhập xâu, vì nó tồn tại hai hạn chế: không thể kết thúc nhập bằng cách gõ phím Enter; số kí tự của xâu nhập phải được ấn định trước trong chương trình. Để khắc phục, MSDOS cung cấp hàm 0Ah của ngắt 21h để hỗ trợ nhập xâu kí tự.

**Hàm 0Ah của ngắt 21h:**

**Tác dụng:**Nhập một xâu kí tự vào một biến đệm cho trước trong chương trình, biến này phải được chỉ bởi cặp thanh ghi DS:DX. Và biến đệm phải có dạng như sau:

-         Byte 0: chứa số kí tự tối đa của xâu nhập vào

-         Byte 1: chứa một trị không (= 0)

-         Byte 2, byte 3, byte 4, ... chứa một trị rỗng (để trống), để chứa các kí tự sẽ được nhập vào sau này (khi chương trình thực hiện).

Để có được một biến như trên chương trình phải khai báo biến (tên biến là Xau\_Nhap) như sau:

**Xau\_Nhap     DB      256, 0, 256 Dup (‘ ‘)**

Như vậy Xau\_Nhap là một biến kiểu byte, gồm 258 byte. Byte đầu tiên (byte) chứa trị 256, byte 1 chứa trị 0, 256 byte tiếp theo chứa kí tự trống, được sử dụng để chứa các kí tự sẽ được nhập sau này. Xau\_Nhap chứa tối đa 256 kí tự.

Cũng có thể sử dụng hàm 0Ah/21h để nhập một xâu kí tự vào vùng nhớ có địa chỉ xác định trong bô nhớ.

**Sử dụng:**

**Vào:**  Ah = 0Ah

            DS:DX = <Địa chỉ Segment:Offset của xâu nhập>

**Ra:**DS:DX không thay đổi

Biến đệm bây giờ có dạng như sau:

-         Byte 0: không thay đổi

-         Byte 1: chứa tổng số kí tự đã được nhập vào

-         Byte 2, byte 3, byte 4, ... chứa các kí tự đã được nhập vào.

**Ví dụ 1:**Với khai báo biến đệm Xau\_Nhap như trên, nếu sau này chương trình nhập vào xâu: “Tin hoc” thì:

-         Byte 0: vẫn chứa số 256

-         Byte 1: chứa giá trị 7, đó chính là 7 kí tự trong xâu “Tin hoc”

-         Từ byte 2 đến 8 chứa lần lượt các kí tự trong xâu “Tin hoc.

**Ví dụ 2:**Giả sử chương trình đã khai báo xâu TieuDe như sau:

**TieuDe**           DB      100, 0, 100 Dup (‘ ‘)

            Các lệnh sau đây sử dụng hàm 0Ah/21h để nhập một xâu kí tự vào biến **TieuDe**:

            Mov          Ax, Seg **TieuDe**

            Mov          Ds, Ax

            Lea            Dx, **TieuDe**

            Mov          Ah, 0Ah

            Int             21h

Các lệnh sau đây lấy số kí tự thực sự nhập đưa vào lưu trữ trong thanh ghi Cx:

            Mov          Cx, 0

            Mov          Cl, TieuDe[1]

Các lệnh sau đây sử dụng hàm 02/21h để in xâu kí tự vừa nhập ra lại màn hình:

            Lea            DI, TieuDe

            Mov          Cx, 0

            Mov          Cl, TieuDe[1]

            Add           DI, 2

            Mov          Ah, 02

      Lap\_In:

            Mov          Dl, DS:[DI]

            Int             21h

            INC           DI

            Loop         Lap\_In

Các lệnh sau đây sử dụng hàm 09/21h để in xâu kí tự vừa nhập ra lại màn hình:

            Mov          Ax, Seg **TieuDe**

            Mov          Ds, Ax

            Lea            Dx, **TieuDe**

            Add           Dx, 2

            Mov          DI, Dx

            Mov          Cx, 0

            Mov          Cl, **TieuDe**[1]

            Add           DI, Cx

**Mov          Byte PTR DS:[DI], ‘$’**

            Mov          Ah, 09h

            Int             21h

**Ví dụ 3:** Chương trình dạng COM sau đây sử dụng hàm 0Ah ngắt 21h để nhập một xâu kí tự vào biến **Buff**. Sau đó sử dụng hàm 09h ngắt 21h để in xâu kí tự vừa nhập ra lại màn hình.

 Để đơn giản, chương trình khai báo biến **Buff** gồm toàn kí tự ‘$’, nhờ đó, khi dùng hàm 09h/21h để in ra chương trình không cần đưa thêm kí tự ‘$’ vào cuối xâu nhập, mà chỉ cần trỏ DS:DX về đầu vùng/xâu kí tự cần in ra.

**.Model**       small

**.Code**

**ORG**       100h

          Start:

                        JMP    Main

              TBN    DB      'Nhap vao mot xau ki tu: $'

              TBX    DB      0Ah,0Dh,'Xau vua nhap: $'

              Buff    DB      100,0,100 Dup ('$')

          Main          **Proc**

                    Mov        Ah,  09h

                    Lea          Dx, TBN

                    Int           21h

                    ;--------------------------

                    Mov        Ah, 0Ah

                    Lea          Dx, Buff

                    Int           21h

                    ;-------------------------

                    Mov        Ah,  09h

                    Lea          Dx, TBX

                    Int           21h

                    ;-------------------------

                    Mov        Ah,  09h

                    Lea          Dx, Buff

                    Add         Dx, 2

                    Int           21h

                    ;-------------------------

                    Int           20h

          Main          **Endp**

**End**         Start

**Một số ví dụ:**

**Ví dụ 1:**Giả sử hàm X của ngắt Y khi được gọi sẽ trả về trong cặp thanh ghi ES:SI địa chỉ của vùng nhớ chứa tên của nhà sản xuất (nhãn hiệu) của vi xử lý đang sử dụng trên máy tính hiện tại, tên này dài không quá 8 kí tự.

Các lệnh sau đây sẽ in tên nhà sản xuất nói trên ra màn hình:

                    Mov        Ah, X                          ; hàm cần gọi được đưa vào thanh ghi ah

                    Int           Y                                 ; gọi ngắt Y với hàm X

                    ;--------------------

                    Mov        Cx, 8                           ; tên dài không quá 8 kí tự

                    Mov        Ah, 02                                    ; dùng hàm 02/21h để in kí tự ra nàm hình

             LapIn:

                    Mov        Dl, Byte PTR ES:[SI]

                    Int           21h

                    INC         SI                                 ; đến kí tự tiếp theo

                    Loop       LapIn

                    ;------------------------------------

**Ví dụ 2:**Giả sử hàm X của ngắt Y khi được gọi với AL = 1 (được gọi là hàm con) sẽ trả về trong cặp thanh ghi ES:DI địa chỉ của vùng nhớ chứa *ngày-tháng-năm* sản xuất ROM-BIOS đang sử dụng trên máy tính hiện tại. Đồng thời hàm/ngắt này cũng cho biết số kí tự trong xâu *ngày-tháng-năm* trong thanh ghi BX.

Các lệnh sau đây sẽ in xâu *ngày-tháng-năm* nói trên ra màn hình:

                    Mov        Ah, X                          ; hàm cần gọi được đưa vào thanh ghi ah

                    Mov        Al, 1                            ; gọi hàm X với Al = 1

                    Int           Y                                 ; gọi ngắt Y với hàm X

                    ;--------------------

                    Mov        Cx, Bx                        ; đưa số kí tự của xâu *ngày-tháng-năm*vào Cx

                    Mov        Ah, 02                                    ; dùng hàm 02/21h để in kí tự ra nàm hình

             LapIn:

                    Mov        Dl, Byte PTR ES:[DI]

                    Int           21h

                    INC         DI                                ; đến kí tự tiếp theo

                    Loop       LapIn

                    ;-------------------------------------

**Ví dụ 3:** Hàm 39h của ngắt 21h được sử dụng để tạo thư mục con trên đĩa. Hàm này quy định: DS:DX chứa xâu tên thư mục cần tạo, bao gồm cả đường dẫntìm đến thư mục này, xâu này kết thúc bởi trị 0. Nếu việc tạo không thành công thì Cf = 1, khi đó thanh ghi Ax sẽ chứa mã lỗi.

Các lệnh sau đây sẽ tạo ra thư mục con BTCB trong thư mục ASSEM trên thư mục gốc ổ đĩa D.

Chương trình phải khai báo biến TenTM, chứa xâu tên thư mục cần tạo như sau:

            TenTM           DB      ‘D:\ASSEM\BTCB’,0

Các lệnh:

            Mov    Ax, Seg TenTM

            Mov    DS, Ax

            Mov    Dx, Offset TenTM

            Mov    Ah, 39h

            Int       21h

            ;-------------------------

            Jc        TB\_Loi                                   ; nếu CF = 1 thì việc tạo bị lỗi

                      <In ra thong bao hoan thanh>

            Jmp     KetThuc

       TB\_Loi:

                    <In ra thong bao khong tao duoc thu muc>

        KetThuc:

            ...

            ;----------------------------------------------------