فصل نهم

وقفه ها (INTERRUPTS)

ریزپردازنده در یک کامپیوتر برنامه‌ها را بسادگی اجرا نمی‌کند. بلکه بعنوان تنظیم کننده سیستم درگیر چیزهائی میشود که اتفاق می‌افتند. بعنوان مثال وقتی که کلیدی روی صفحه کلید فشار می‌دهید ریزپردازنده بایستی دریابد که کدام کلید فشار داده شده و عمل مناسب آن کلید را انجام دهد. بعنوان مثال وقتی کلید Ctrl-Break را فشار می‌دهید عملی که بایستی انجام شود کاملاً متفاوت است با وقتی که کلید T را فشار می‌دهید. وقتی که data از disk به حافظه یا بالعکس منتقل می‌گردد این مسئولیت ریزپردازنده است که دستورالعملهای مناسب برای اینکار را اجرا نماید. همانطوریکه گفته شد ریزپردازنده در کلیه کارهای کامپیوتر نقشی ایفاء می‌نماید. حال سوالی که مطرح می‌شود اینست که ریزپردازنده چگونه با وسائل جانبی درگیر می‌گردد؟ حقیقت امر این است که ریزپردازنده‌ها و وسائل جانبی بطرق مختلف با هم ارتباط برقرار می‌کنند. کاری که انجام می‌شود بدین صورت است که ریزپردازنده شروع به اجرای برنامه می‌نماید و به اجرای برنامه ادامه می‌دهد تا زمانیکه یک وسیله جانبی مانند صفحه کلید، دیسک، یا مانیتور به ریزپردازنده اعلام نماید که به کمک ریزپردازنده نیاز دارد. البته وسائل جانبی در حقیقت با ریزپردازنده صحبت نمی‌کنند بلکه آنها سیگنال نیاز به کمک خود را از طریق وقفه یا Interrupt ارسال می‌نمایند.

نحوه کار وقفه ها

وقتی که یک وسائل جانبی اقدام به ارسال سیگنال وقفه می‌نماید شماره شناسائی خود را که type code نامیده می‌شود نیز ارسال می‌کند. هر وسیله جانبی از قبیل keyboard ، disk drive ، floppy drive و ... دارای type code مختلفی می‌باشد. در حقیقت 256 نوع مختلف type code وجود دارد. هر وقت درخواست کمکی از طرف یک وسیله جانبی شود (Interrupt) ریزپردازنده اگر کاری که در حال انجام آن می‌باشد بتواند موقتاً رها نماید، اینکار را انجام داده و به کمک وسیله جانبی می‌رود (البته با حفظ موقعیت فعلی). پس از تکمیل کار وسیله جانبی مجدداً کار قبلی خود را از سر می‌گیرد. در صورتیکه ریزپردازنده نتواند کار فعلی خود را رها نماید پس از تکمیل این کار به کمک وسیله جانبی می‌رود.

همانطور که متذکر شدیم 256 تا type code وجود دارد به شماره‌های 255 تا 0. ریزپردازنده از این type code استفاده نموده آدرسی را در ابتدای حافظه محاسبه نموده و از آدرس محاسبه شده آدرس دیگری را می‌خواند. این آدرس جدید Interrupt Vector نام دارد که در حقیقت آدرس برنامه‌ای است که کار آن وقفه را عهده‌دار می‌باشد. در مورد وسیله جانبی استاندارد، برنامه‌های سرویس دهنده وقفه‌ها (Interrupt servicing programs) در تراشه ROM ذخیره می‌گردد. در بسیاری از کامپیوترها این ROM به Basic Input/Output system یا BIOS معروفست.

حال که ریزپردازنده می‌داند کدام برنامه را بایستی اجرا کند این کار را انجام داده یعنی برنامه را اجرا نموده و پس از اتمام اجرای برنامه به کار قبلی خود برگشته و اجرای آن را دنبال می‌کند.

منابع وقفه ها

وقفه‌هائی که تاکنون بیان شده است وقفه‌های خارجی (External Interrupt) می‌باشد که توسط وسائل یا تجهیزات جانبی فعال می‌گردند. این نوع وقفه‌ها قسمتی از 256 نوع وقفه را پوشش می‌دهد. سایرین می‌توانند یکی از دو نوع وقفه ذیل باشند.

۱-برنامه‌ها همچنین می‌توانند با استفاده از دستورالعملهای خاص ایجاد وقفه در برنامه، وقفه‌ها را فعال نمایند.

۲-درموارد خاص ریزپردازنده حتی می‌تواند به خودش وقفه بدهد. بعنوان مثال وقتی که شما سعی در تقسیم بر صفر دارید.

وقفه های رزرو شده (Reserved Intruuput)

از 256 وقفه، 32 تای اول یعنی شماره 0 تا 31 بوسیله Intel رزرو گردیده است. وقفه‌های نوع 255 تا 32 برای موارد دیگر استفاده می‌گردند. نمونه‌ای از وقفه‌های رزرو شده در ذیل داده شده است:

Type 0,	Divide Error
Type 1,	Single – Step
Type 2,	Nonmaskable Interrupts
Type 3,	Breakpoint
Type 4,	Overflow
Type 5,	Bound Range Exceeded
Type 6,	Invalid Table Limit Too Small
Type 7,	Processor Extension Not Available
Type 8,	Interrupt Table Limit Too Small
Type 9,	Processor Extension Segment Overrun
Type 13,	Segment Overrun
Type 16	Processor Extension Error

وقفه های سیستم

در کامپیوترها، 1024 بایت اول حافظه یعنی محلهای حافظـه بـاآدرس 0 تـا 3FF تخصیص بـه جـدولی دارد کـه ایـن جـدول بنـام interrupt vector table معروفست. این جدولی است بـا آدرسـهای 32 بیتـی کـه بـه interrupt service routines در کامپیوتر اشاره می‌کنند. 256 وقفه مختلف به شماره‌های 0 تا 255 یـا FF تا 0 در مبنـای شـانزده وجـود دارد. ریزپردازنـده Intel ، 32 وقفـه اول یعنـی وقفه‌های شماره 1FH تا 0 را برای استفاده خودش در نظر می‌گیرد. 32 وقفه بعدی یعنی شماره‌های 3FH تا 20H برای استفاده سیستم عامل DOS در نظر گرفته شده است.

وقفـه از طریـق دسـتورالعمل‌هـای وقفـه در برنامـه یـا تجهیـزات خـارجی (external devices) در سیستم فعال می‌گردد.

وقتی که ریزپردازنده یک وقفه دریافت می‌نماید شماره وقفه را در 4 ضـرب نمـوده تـا آدرس interrupt vector در جـدول را بدسـت آورده سـپس محتـوی آدرس بدست آمده را در ثبات IP و ثبات CS قرار مـی‌دهـد و شـروع بـه اجـرای دستورالعملها در آن آدرس می‌نماید. بعنوان مثال اگر وقفه از نوع 4AH باشد.

$$4AH * 4 = 128H$$

محلی از حافظـه کـه آدرس آن 128H مـی‌باشـد شـامل آدرس Interrupt service routine وقفه نوع 4AH می‌باشد.

گرفتن اطلاعات از صفحه کلید

کلیدهای صفحه کلید به چهار دسته تقیسیم می شوند.

۱- کلیدهای حروف استاندارد که شامل حروف A تا Z رقم های 0 تا 9 و علامات مانند $ و% و# و....

۲- کلیدهای توابع شامل کیدهای F_1 تا F_{12} به همراه یکی از کلیدهای Shift, Ctrl, Alt و کلیدهای PgDn, PgUp

۳- کلیدهای عملیاتی صفحه کلید شامل ESC, Enter, Back space, Tab که کدهای اسکی و اسکن آنها به صورت هگزا به شرح زیر می باشند.

	AH کد اسکن	AL کد اسکی
Enter	1CH	0DH
ESC	01 H	1BH
Back Space	0EH	08 H
Tab	0FH	09H

۴- کلیدها کنترلی شامل کلیدهای Shift, Alt ,Ctrl که همراه با کلیدهای دیگر کار می کنند ولی خودشان کدهای اسکی ندارند.

برای عملیات صفحه کلید، از دستور INT16H و دستور INT21H و سرویس های مربوطه استفاده می نماییم.

الف - سرویس های دستور INT21 برای خواندن از صفحه کلید

- سرویس01H: خواندن یک حرف از صفحه کلید و نمایش آن برروی مانیتور

```
MOV AH,01
INT 21 H
```

- 06H : تشخیص فشار دادن کلید صفحه کلید
- **سرویس**07H: خواندن یک حرف از صفحه کلید بدون نمایش آن روی مانیتور

```
M0V AH,07
INT 21H
CMP AL,0DH
```

- **سرویس** 08H: خواندن یک حرف از صفحه کلید بدون نمایش آن برروی مانیتور (حساس به**ctrl Break**)
- **سرویس** 0AH: خواندن یک رشته اطلاعات از صفحه کلید، قرار دادن آن در محل در بـافر حافظـه و نشـان دادن آنها روی مانیتور

برای استفاده از این سرویس لازم است موارد زیر رعایت شود:

۱- با استفاده از شبه دستور LABEL در سگمنت داده،نامی به این محل حافظه،یا بافر حافظه داده میشود که در مثال زیر نام PARALST اختیار شده است.

۲- حداکثر تعداد حروف رشته ورودی باید در اولین بایت بعد از شبه دستور LABEL تعریف شود. که در مثال زیر حداکثر تعداد حروف رشته ورووردی ۲۰ بایت می باشد که به نام MAXLEN در سگمنت داده معرفی شده است.

۳- بعد از فشار دادن دکمه Enter(یعنی در پایان رشته حروف وارد شده از صفحه کلید) سیستم عامل تعداد واقعی حروف تایپ شده را در بایت دوم حافظه که در سگمنت داده تعریف شده قرار می دهد، که در مثال زیر یک بایت برای این محل به نام ACTLEN در نظر گرفته شده است.

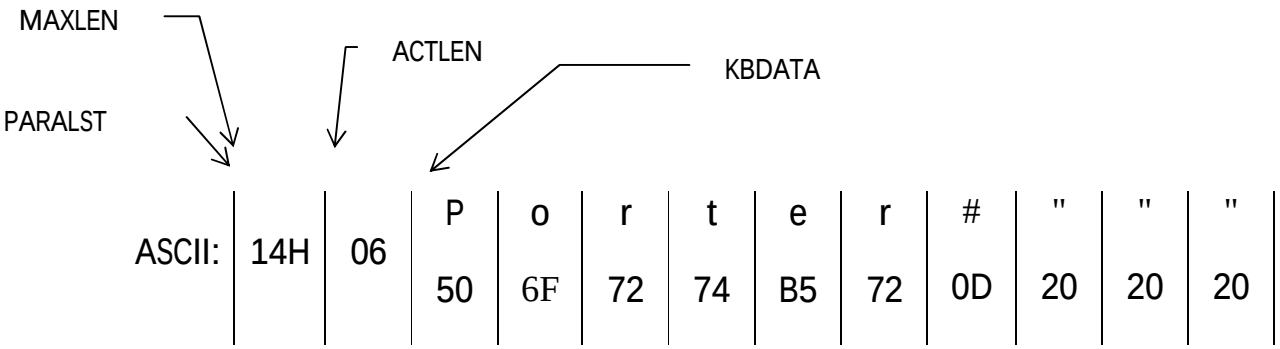
۴- سومین متغیری که در سگمنت داده بعد از شبه دستور LABEL باید تعریف شود، محلی است حروف تایپ شده صفحه کلید از چپ به راست وارد آن می شوند که در مثال زیر متغیر KBDATA با ۲۰ بایت (مقدار اولیه کاراکتر space ...) برای آن در نظر گرفته شده است.

```
PARALST LABEL BYTE;
MAXLEN DB 20;
ACTLEN DB?;
```

KBDATA DB 20 DVP ('') ;

قرار بدهیم و افست آدرس بافر OAH را برابر AHبرای گرفتن اطلاعات از صفحه کلید کافیست مقدار ثبات را اجرا نماییم INT21H بگذاریم و دستور DXحافظه در سگمنت داده را در ثبات

```
MOV AH, OAH
LEA DX,PARALST
INT 21 H
```



توجه: این سرویس کدهای کلیدهای کنترلی F_1 تا..... Home.....نمی پذیرد.

به یک پیغام می باشدN یا Yمثال تکه برنامه ای برای پاسخگویی

```
GET_KEY: MOV AH,01
         INT 21 H
         CMP AL,Y
         JE YES
         CMP  AL,N
         JE  NO
         JNE   GET_KEY
```

ب - سرویس های دستور INT 16 H (BIOS) برای خواندن از صفحه کلید

- **سرویس 00** : خواندن یک حرف از صفحه کلید بدون نمایش روی مانیتور

```
MOV  AH,0
INT  16H
CMP  AL,41H
JE serv
   ...

Serv:
```

- **سرویس 01H** : بررسی فشار دادن کلید صفحه کلید اگر کلیدی فشرده شده باشد ZF=0 در غیر اینصورت ZF=1
- **سرویس 02H دستور**: برای گزارش وضعیت بعضی کلیدهای صفحه کلید.
  - بیت ۷ ثبات AL اگر یک باشد یعنی کلید Insert فشار داده شد.
  - بیت ۶ ثبات AL اگر یک باشد یعنی کلیدcaps lock فشار داده شد.
  - بیت 5 ثبات AL اگر یک باشد یعنی کلید Nom Lock فشار داده شد.
  - بیت ۴ ثبات AL اگر یک باشد یعنی کلیدScroll Lock فشار داده شد.
  - بیت ۳ ثبات AL اگر یک باشد یعنی کلیدAlt فشار داده شد.
  - بیت ۲ ثبات AL اگر یک باشد یعنی کلیدCtrl فشار داده شد.
  - بیت ۱ ثبات AL اگر یک باشد یعنی کلیدLeft Shift فشار داده شد.
  - بیت ۰ ثبات AL اگر یک باشد یعنی کلیدRigh Shift فشار داده شد.

مثال:

```
BACK: MOV AH, 02H
      INT 16h
      TEST AL,00001000B
      JNZ OVER
      ...
```

OVER:

نمایش اطلاعات بر روی مانیتور در حالت متن

سرویسهای دستورات INT 10H (از BIOS) برای عملیات بر روی مانیتور

- سرویس 00H : تغییر حالت یا مد مانیتور
- سرویس 01H : تعیین اندازه مکان نما
- سرویس 02H : تغییر محل مکان نما
- سرویس 03H : بررسی موقعیت و اندازه مکان نما
- سرویس 05H : انتخاب صفحه فعال برای نمایش اطلاعات آن، روی مانیتور
- سرویس 06H : پاک کردن و چرخش اطلاعات مانیتور به بالا
- سرویس 07H : پاک کردن و چرخش اطلاعات مانیتور به پایین
- سرویس 08H : خواندن حرف و رنگ مربوط به آن در محل فعلی مکان نما
- سرویس 09H : نوشتن یک یا تعدادی حرف و رنگ آن در محل مکان نما، بدون تعییر محل مکان نما
- سرویس 0AH : نوشتن یک یا تعدادی حرف در محل مکان نما بدون تغییر رنگ و بدون تغییر محل مکان نما
- سرویس 0CH : روشن کردن یا نوشتن یک نقطه یا یک پیکسل مانیتور در حالت گرافیک
- سرویس 0DH : خواندن مشخصات یک نقطه یا یک پیکسل مانیتور در حالت گرافیک
- سرویس 0EH : نوشتن یک حرف روی مانیتور و تغییر محل خودکار مکان نما
- سرویس 0FH : تعیین حالت یا مد مانیتور

سرویسهای دستورات INT 21H (از DOS) برای عملیات بر روی مانیتور

- **سرویس 02H : نمایش یک حرف روی مانیتور و تغییر محل خودکار و مکان نما**
- **سرویس 09H : نمایش یک رشته اطلاعات یا یک پیغام روی مانیتور**

صفحه نمایش و حافظه مانیتور

- **مانیتور در حالت عادی دارای ۲۵ سطر و ۸۰ ستون می باشد که با ضرب این دو مقدار تعداد ۲۰۰۰ کاراکتر میتوان مشاهده نمود. برای نمایش هر کاراکتر دو بایت حافظه نیاز میباشد یک بایت برای کاراکتر و یک بایت برای صفت کاراکتر، بنابراین ۴۰۰۰ بایت حافظه برای مانیتور لازم می باشد.**

- **در حافظه کامپیوتر چهار قسمت به نام صفحه حافظه هر یک با ظرفیت 4 K بایت به نام:**
  - **صفحه صفر از آدرس B800:0000 یا آدرس فیزیکی B8000H.**
  - **صفحه یک از آدرس B900:0000 یا آدرس فیزیکی B9000H.**
  - **صفحه دو از آدرس BA00:0000 یا آدرس فیزیکی BA000H.**
  - **صفحه سه از آدرس BB00:0000 یا آدرس فیزیکی BB000H.**

نکته: در هر لحظه میتوان اطلاعات یکی از صفحات را روی مانیتور مشاهده نمود.

جدول مشخصات مکان نما روی مانیتور

محل مکان نما برای یک حرف در مانیتور	به صورت دهدهی		به فرم هگزا	
	سطر	ستون	سطر	ستون
گوشه سمت چپ بالا	00	00	00H	00H
گوشه سمت راست بالا	00	79	00H	4FH
وسط مانیتور	12	39,40	0CH	27H,28H
گوشه سمت چپ پایین	24	00	18H	00H
گوشه سمت راست پایین	24	79	18H	4FH

حالتهای مختلف مانیتور VGA در حالت متن کدهای تنظیم مطالب مانیتور

- **مانیتور می تواند در حالت ها یا مدهای ۰ و ۱ و ۲و ۳ و ۷ به صورت متن کار کند، که حداکثر تعداد حروف روی مانیتور، تعداد پیکسلهای هر حرف، دقت، و تعداد رنگ آنها در جدول زیر آمده است:**

در حالت متنVGAحالت های مختلف

حالت	تعداد پیکسل در هر حرف	تعداد حروف	دقت (برحسب پیکسل)	رنگ
00	9*16	40*25	360*400	تک رنگ
01	9*16	40*25	360*400	۱۶ رنگ
02	9*16	80*25	720*400	تک رنگ
03	9*16	80*25	720*400	۱۶ رنگ
07	9*16	80*25	720*400	تک رنگ

به عنوان مثال با دستورات زیر: شماره سرویس در AH قرار می گیرد. حالت ۳ برای استاندارد مانیتور 80X25 رنگی و دستور وقفه فعال می شود.

```
MOV AH,00
MOV AL,03
INT 10H
```

کدهای تنظیم مطالب مانیتور

- **به سر سطر برگرد** CR
- **یک خط فاصله** LF
- **به ستو بعد برو** TAB

دهدهی	اسکی (هگزا)	علامت
13	0D	CR
10	0A	LF
09	09	TAB

به عنوان مثال برای نوشتن پیغامی روی مانیتور، با یک خط فاصله، در سطر بعدی:

را به صورت زیر می نویسیم:LF جهت 0AH,CR برای 0DH**در سگمنت داده:** در ابتدای پیغام مذکور کد

```
PEYGHAM DB 0AH,0DH, 'TEHRAN IRAN', '$'
```

دستورات زیر را می نویسیم:INT 21H دستور 09H**در سگمنت کد:** با استفاده از سرویس

```
MOV AH,09H
LEA DX,PEYGHAM
INT 21H
```

رنگ حروف و زمینه مانیتور در حالت متن

		رنگ زمینه			رنگ حروف			
صفت	BL	R	G	B	I	R	G	B
شماره بیت	7	6	5	4	3	2	1	0

نشانگر رنگ های آبی، سبزو قرمز می باشند. همان طوری که ملاحظه می شود، بیت ۰ تا ۲ بایت صفت R,G,Bدر جدول مذکور حروف حروف حرف برای تعیین رنگ حرف، و بیت ۴ تا ۶ جهت تعیین رنگ زمینه، و بیت ۳ نیز برای شدت نور و بالاخره بیت ۷ برای حالت چشمک زدن حرف می باشد.

IRGB	رنگ	IRGB	رنگ
1000	خاکستری	0000	سیاه
1001	آبی کمرنگ	0001	آبی
1010	سبز روشن	0010	سبز
1011	آبی آسمانی کمرنگ	0011	آبی آسمانی
1100	قرمز روشن	0100	قرمز
1101	زرشگی روشن	0101	زرشگی
1110	زرد	0110	قهوه ای
1111	سفید براق	0111	سفید

مثال: ۱۲ علامت ستاره * قهوه ای رنگ که چشمک میزند در زمینه آبی

```
MOV AH,09H

MOV AL, '*'

MOV BH, 00H X ; صفحه صفر

MOV BL, 1EH    ; رنگ حرف قهوه اي چشمك زن بر آبي (00011110)

MOV CX,12 ; تعداد حروف تكرار

INT 10H
```

v سرویس 00 دستور INT 10

برای تغییر حالت مانیتور استفاده میشود.

```
MOV AH,OO

MOV AL,01

INT 10H
```

v سرویس 01 دستور INT 10

برای تعیین اندازه مکان نما. تا ۱۵ سطر میتوان آنرا بزرگ کرد.

شماره سرویس AH=01
خط شروع مستطیل =CH
خط پایان مستطیل =CL

```
MOV AH,O1
MOV CH,05
MOV CL,0AH
INT 10H
```

v سرویس 02 دستور INT 10

برای تغییر محل مکان نما

شماره سرویس AH=02
شماره سطر =DH
شماره ستون =DL
شماره صفحه فعال مانیتور =BH
فراخوانی سرویس INT 10 H

```
MOV AH,02H
MOV BH,00
MOV DH,10
MOV DL,8
INT 10H
```

v سرویس 03 دستور INT 10

برای بررسی موقعیت و اندازه مکان نما

```
MOV BH,00
MOV AH,03
INT 10H
```

v سرویس 05 دستور INT 10

برای انتخاب صفحه فعال برای نمایش اطلاعات آن روی مانیتور

```
MOV AH,05      فعال کردن سرویس
MOV AL,02      شماره صفحه ۲ فعال میشود
INT 10H
```

v سرویس 06 دستور INT 10

پاک کردن و چرخش اطلاعات مانیتور به بالا

AH = شماره سرویس

BH = رنگ زمینه و حروف

CL = شماره ستون گوشه سمت چپ بالای پنجره

CH = شماره سطر گوشه سمت چپ بالای پنجره

DL = شماره ستون گوشه سمت راست پایین پنجره

DH = شماره سطر گوشه سمت راست پایین پنجره

INT 10H

مثال : سابروتین به نام CLRSCR که صفحه نمایش را پاک می کند.

```
CLRSCR PROC NEAR
        MOV AX,O600
        MOV BH,61H
        MOV CX,0000
        MOV DX,184FH
        INT 10H
        RET
CLRSCR ENDP
```

v سرویس 07 دستور INT 10

پاک کردن و چرخش اطلاعات مانیتور به پایین:

دقیقاً مانند سرویس شماره 06H است ولی چرخش صفحه فعال از بالا به پایین است و خطوط پاک شده نیز از

بالا به پایین است.

مثال : پاک کردن پنجره ای از مانیتور، از گوشه بالا چپ به مختصات A(0,0) و به تعداد ۱۵ سطر B(4F,0F)

```
MOV AH,06H      ;  سرویس 06H دستور INT 10H
MOV AL,0F       ;  تعداد خطوط پنجره که پاک خواهد شد
MOV CX,0000     ;  CH,CL مختصات گوشه سمت چپ بالای پنجره در
MOV DH,0FH      ;  DH شماره سطر گوشه سمت راست پایین پنجره در
MOV DL,4FH      ;  DL شماره ستون گوشه سمت راست پایین پنجره در
MOV BH,70       ;  رنگ حروف سیاه، در زمینه سفید مانیتور
INT 10 H        ;  10 H اجرای دستور وقفه
```

v سرویس 08 دستور INT 10

خواندن حرف و رنگ مربوط به آن، در محل فعلی مکان نما

```
MOV BH,00
MOV AH,08H
INT 10H
```

کد اسکی حرف در ثبات AL و شماره رنگ در ثبات AH قرار میگیرند.

سرویس 09 دستور INT 10

نوشتن یک یا تعدادی حرف و رنگ آن در محل مکان نما، بدون تغییر محل مکان نما

AH= 09 **شماره سرویس**

AL= **کد اسکی حرف مورد نظر برای نمایش**

BL= **کد رنگ**

BH= **شماره صفحه فعال مانیتور**

CX= **تعداد دفعاتی که حرف میبایستی تکرار شود**

INT 10H

مثال : اگر بخواهیم حرف A در محل فعلی مکان نما، ۱۰ بار پشت سرهم نوشته شود، دستورات زیر را مینویسیم:

```
MOV AH, 09
MOV AL,'A'
MOV CX,10
MOV BH,00H
MOV BL,07H
INT 10H
```

v سرویس 0A دستور INT 10

نوشتن یک یا تعدادی حرف در محل مکان نما، بدون تغییر رنگ و محل مکان نما

شماره سرویس AH = 0A

کد اسکی حرف مورد نظر برای نمایش = AL

شماره صفحه فعال مانیتور = BH

تعداد دفعاتی که حرف میبایستی تکرار شود = CX

INT 10H

مثال : اگر بخواهیم تعداد ۶۰ عدد قلب در محل فعلی مکان نما، پشت سرهم نوشته شود، دستورات زیر را مینویسیم:

```
MOV AH, 0A
MOV AL,03H
MOV BH,00H
MOV BL,16H
MOV CX,60
INT 10H
```

v سرویس 0E دستور INT 10

نوشتن یک حرف در محل مکان نما، و تغییر محل خودکار مکان نما

شماره سرویس AH = 0EH

کد اسکی حرف مورد نظر برای نمایش = AL

کد رنگ مورد نظر برای نمایش = BL

شماره صفحه فعال مانیتور = BH

INT 10H

مثال : حرف A در صفحه نمایش دیده میشود

```
MOV AH, 0EH
MOV AL,41H
MOV BH,00H
```

```
MOV BL,07H
INT 10H
```

v سرویس 0F دستور INT 10

تعیین حالت یا مد مانیتور

```
MOV AH,0FH
INT 10H
```

تعداد حروف مانیتور در هر سطر در ثبات AH و شماره صفحه در ثبات BH قرار میگیرند.

: جهت نمایش حروف روی مانیتورINT 21H دستور 09Hمثال ۱: برنامه نمونه برای بکار بردن سرویس

```
        PAGE 110,100
TITLE   'INTE-DIS.ASM'  procedures & interrupts
;-----------------------------------------------
;           Defining Segment of Program
;           ---------------------------
CODESG  SEGMENT 'CODE'
        ASSUME
SS:CODESG,DS:CODESG,CS:CODESG
        ORG 100H
START:  JMP MAIN        ;1-Jump to instuctions
;-----------------------------------------------
CHAR    DB 00
;-----------------------------------------------
;           Main procedure
;           --------------
MAIN    PROC NEAR
        CALL CLRMON   ;2-Clear monitor
        CALL SETCUR     ;3-Set cursor
        CALL CHARA      ;4-Display characters
        MOV AX,4C00H    ;5-End of
        INT 21H         ;6-processing
MAIN    ENDP            ;7-End procedure
;           Clear monitor
;           -------------
CLRMON  PROC NEAR       ;8-Begin of procedure
        MOV AX,0700H    ;9-Scroll down completly
        MOV BH,07H      ;10-White on black
        MOV CX,0000     ;11-Upper left location
        MOV DX,184FH    ;12-Lower right location
        INT 10H         ;13-Call BIOS
        RET             ;14-Return to caller
CLRMON  ENDP            ;15-End of procedure
;                   Set cursor at 10,00
;                   -------------------
```

```
SETCUR PROC NEAR      ;16-Begin of procedure
       MOV AH,02H     ;17-Request set cursor
       MOV DH,10      ;18-Row 10
       MOV DL,00      ;19-Column 00
       INT 10H        ;20-Call BIOS
       RET            ;21-Return to caller
SETCUR ENDP           ;22-End of procedure
;              Display characters
;              ------------------
CHARA  PROC NEAR      ;23-Begin of procedure
       MOV CX,256     ;24-  256 iterations
       MOV AL,CHAR    ;25-  00 for ASCII
BACK1: MOV AH,0EH     ;26-Display ASCII
       INT 10H        ;27-characters
       INC AL         ;28-Next character
       LOOP BACK1     ;29-Loop nonzero
       RET            ;30-Return to caller
CHARA  ENDP           ;31-End of procedure
CODESG ENDS           ;32-End of segment
       END START      ;33-End of program
```

مثال ۲: برنامه ای بنویسید که :

الف: مانیتور را پاک کند

ب- پیغام SPEAKER MAKE SOUND. را روی مانیتور بنویسد

ج- با فشار دادن کلید D: :

- **حرف D را با استفاده از سرویس 02 دستور INT 21H روی مانیتور بنویسد**
- **با استفاده از سرویس 0E دستور INT 10H حرف D را روی مانیتور بنویسد**
- **با استفاده از سرویس 0E دستور INT 10H بوق بزند.**

```
       PAGE 100,110
TITLE 'KEY-DETE.ASM' showing different interrupt
     ----------------------------------------------;
          ;Define stack segment
-------------------          ;
STSEG  SEGMENT STACK 'STACK'
       DW 32 DUP(0)
STSEG  ENDS
;
DSEG   SEGMENT
```

```
MESSAGE DB 'SPEAKER MAKE SOUND'$','
DSEG    ENDS
;
CSEG    SEGMENT
START   PROC FAR
        ASSUME CS:CSEG, DS:DSEG, SS:STSEG
        MOV AX,DSEG   ; 1-Set address of
        MOV DS,AX     ; 2-data segment
;
          ;Interrupt for clear screen
-------------------------           ;
        MOV AH,6      ;3- Sercice number in AH
        MOV CX,0      ;4- Left corner
        MOV DX,18F4H  ;5- Right corner
        INT 10H       ;6- Active interrupt 10H
;
     ;Interrupt for making message on monitor
---------------------------------------     ;
        MOV AH,9H     ;7- Service number in AH
     LEA DX,MESSAGE   ;8-Address MESSAGE in DX
        INT 21H       ;9- Active Interrupt 21H
;
     ;Interrupt for detecting key of keyboard
---------------------------------------     ;
        MOV AH,7H     ;10- Service 7
        INT 21H       ;11- retun ASCII in AL
        CMP AL,'D'    ;12- Is pressed key D?
        JE SPEAK      ;13- If yes go to speak
        JMP QUIT1     ;14- othervise go QUIT1
;
          ;Display character
----------              ;
SPEAK:  MOV AH,02H    ;15- Service 02
        MOV DL,'D'    ;16- Show a charater on
        INT 21H       ;17- monitor
;
          ;Display Character
----------              ;
        MOV AH,0EH    ;18- Acitve service 0EH
        MOV AL,'D'    ;19- to show a character
        INT 10H       ;20- on monitor
;
          ;Make Beep
---------             ;
        MOV AH,0EH    ;21- Active service 0EH
        MOV AL,7H     ;22-Put ASCII of Beep
        INT 10H       ;23- which is 7 in DL
;
          ;Come back to operating system
----------------------------            ;
QUIT1:  MOV AX,4C00H  ;24-End of
        INT 21H       ; 25- processing
;
           ;End of progrm
-------------           ;
START   ENDP          ; End of main procedure
```

```
CSEG    ENDS          ; End of segment
        END START     ; End of program
```

مثال ۳: برنامه ای که تعداد کاراکترهای یک رشته را می شمارد:

```
; counts the number of characters of a zero terminated string.
name "counter"
org 100h
jmp start
str1 db 'abcdefg hijklmnop qrstvuwxyz', 0
start: lea bx, str1 ; load address of string.
    mov ax, 0 ; reset counter.
compare: cmp [bx], 0 ; is it end of string?
    je done  ; if zero, then it's the end.
    inc ax   ; count char.
    inc bx   ; next memory position in string.
    jmp compare
; print result in binary:
done:
mov bx, ax
mov cx, 8
print: mov ah, 2   ; print function.
    mov dl, '0'
    test bl, 10000000b  ; test first bit.
    jz zero
    mov dl, '1'
zero:  int 21h
    shl bl, 1
loop print
; print binary suffix:
mov dl, 'b'
int 21h

; wait for any key press....
mov ah, 0
int 16h
ret
```

مثال ۴: برنامه ای که تعداد فشار دادن کلیدهای صفحه کلید را می شمارد و در صورتیکه کلیدهای enter یا esc را بفشارید از برنامه خارج میشود.

```
;Count number of key presses. the result is in bx register.
;
;You must type into the emulator's screen,
;if it closes, press screen button to re-open it.
name "keycount"
org  100h
;print welcome message:
mov dx, offset msg
mov ah, 9
int 21h
xor bx, bx ; zero bx register    .
wait:  mov ah, 0   ; wait for any key....
       int 16h

       cmp al, 27  ; if key is 'esc' then exit.
       je stop

       mov ah, 0eh ; print it.
       int 10h

       inc bx ; increase bx on every key press.
       jmp wait


;print result message:
stop:  mov dx, offset msg2
       mov ah, 9
       int 21h

mov ax, bx
call print_ax

;wait for any key press:
mov ah, 0
int 16h

ret ; exit to operating system.

msg db "I'll count all your keypresses. press 'Esc' to stop...", 0Dh,0Ah"$" ,
msg2 db 0Dh,0Ah, "recorded keypresses$ :
```

منابع:

۱- کتاب زبان ماشین واسمبلی دکتر سید رضی

۲- کتاب زبان ماشین و برنامه سازی سیستم مهندس داریوش نیک مهر

۳- کتاب برنامه نویسی و زبان اسمبلی کامپیوترهای شخصی مهندس فرزانه کیمیایی