عنوان درس:

برنامه سازی سیستم

جلسه ۷: وقفه ها و برنامه های مقیم در حافظه

مدرس:

اعظم کبوتری کهنه شهری

وقفه

- مکانیسمی است که پردازنده را مجبور می کند تا اجرای برنامه جاری را موقتا متوقف کند و روال خاصی را اجرا کند. این روال خاص، روتین سرویس وقفه یا روال مدیریت وقفه نام دارد.
- انواع وقفه:
  - نرم افزاری: این وقفه ها توسط برنامه ها برای فراخوانی توابع DOS و BIOS تولید می شوند.
  - سخت افزاری: توسط اجزای سخت افزاری تولید می شوند و از طریق کنترلر وقفه به پردازنده منتقل می شوند. این وقفه ها بردونوعند:
    - قابل اغماض : وقفه هایی که قابل ناتوان سازی هستند تا سخت افزار نتواند به پردازنده وقفه بدهد. مانند وقفه های دستگاههای I/O. این وقفه ها را می توان با صفر کردن پرچم IF ناتوان کرد.
    - غیر قابل اغماض (NMI): وقفه هایی که حتما رسیدگی می شوند.

جدول بردار وقفه

- وقفه هایی که در اثر استفاده از دستور `int N` (N عددی بین صفر تا ۲۵۵) فعال می شوند هر یک دارای روتینی هستند که با فعال شدن وقفه این روتین اجرا می شود و سرویس مورد نظر را ارئه می دهد. به این روتین، روتین سرویس وقفه (ISR:Interrupt Service Routine) گفته می شود.
- جدول بردار وقفه محلی از حافظه است که برای نگهداری آدرس شروع روتین وقفه استفاده می شود.
- آدرس شروع هر روتین وقفه، در ۴ بایت نگهداری می شود:
  - دو بایت پایین برای آدرس شروع سگمنت روتین وقفه (CS)
  - دو بایت بالا برای آدرس افست روتین وقفه در داخل سگمنت (IP)

جدول بردار وقفه–ادامه

- برای جدول بردار وقفه حافظه ای به اندازه 256×4=1024=1KB فضا با شروع از ابتدای حافظه اصلی یعنی آدرس 0000:0000 اختصاص یافته است.

- شماره وقفه (N) به عنوان اندیسی برای مراجعه به جدول بردار وقفه استفاده می شود:
  - برای بدست آوردن آدرس افست روتین وقفه N به کدام آدرس در جدول بردار وقفه مراجعه می کنیم؟ N×4
  - برای بدست آوردن آدرس سگمنت روتین وقفه N به کدام آدرس در جدول بردار وقفه مراجعه می کنیم؟ N×4+2

نحوه اجرای وقفه ها

- زمانی که در برنامه ای یک سرویس وقفه فراخوانی می شود، چه مراحلی انجام می شود؟

۱. PUSH کردن محتوای فعلی ثبات ها در پشته:
   - قراردادن محتوای ثبات های CS:IP و ثبات پرچم در پشته
۲. مراجعه به جدول بردار وقفه و پیدا کردن آدرس روتین وقفه و قراردادن آن در CS:IP:
   - قراردادن آدرس سگمنت روتین وقفه در ثبات CS
   - قراردادن آدرس افست روتین وقفه در ثبات IP
۳. اجرای دستورات روتین وقفه
۴. POP کردن ثبات ها از پشته با رسیدن به دستور IRET (آخرین دستور روتین وقفه):
   - بازیابی مقادیر ثبات های CS:IP و ثبات پرچم از پشته و قراردادن آنها در ثبات ها
۵. ادامه برنامه

برنامه های مقیم در حافظه یا TSR

- TSR: **T**erminate and **S**tay **R**esident Programs

- برنامه ها روی دیسک قرار دارند و به هنگام اجرا توسط سیستم عامل از دیسک به حافظه بارگذاری می شوند، در پایان اجرا فضای حافظه تخصیص یافته به برنامه آزاد شده در اختیار برنامه دیگری قرار می گیرد.

- برنامه مقیم در حافظه برنامه ای است که بعد از اجرا، همچنان در حافظه باقی می ماند و به عبارت دیگر در حافظه مقیم می شود و تا وقتی کامپیوتر روشن است در حافظه قرار دارد:
  - مزیت: سرعت اجرای برنامه
  - عیب: اشغال حافظه

ساختن برنامه مقیم در حافظه

- روش کلی برای ساختن برنامه مقیم در حافظه **"بازنویسی وقفه"** است.

- بازنویسی وقفه بدین معنی است که:

  کاری که قرار است توسط برنامه مقیم در حافظه انجام شود، در داخل روالی نوشته شود سپس در جدول بردار وقفه، آدرس روتین یک وقفه، به آدرس روال نوشته شده تغییر یابد تا با فراخوانی آن وقفه، به جای روتین وقفه، روال کاربر اجرا شود.

  بنابراین بازنویسی وقفه، شامل جایگزین کردن آدرس روال کاربر با آدرس روتین وقفه است.

- معمولا وقفه هایی که استفاده می شوند:
  - `int 08h` (وقفه تایمر)
  - `int 09h` (وقفه صفحه کلید)

ساختار روال جایگزین روتین وقفه

ساختار روالی که قرار است جایگزین روتین وقفه شده و در حافظه مقیم شود به صورت زیر است:

```
newISR proc near
    push AX
    ;
    ; دستورات برنامه مقیم در حافظه
    ;
    pop AX
    jmp cs:oldvect

newISR endp
```

- تعریف روال جایگزین با نام دلخواه newISR
- ذخیره کردن مقدار ثبات AX در پشته تا از بین نرود.
- قرار دادن محتوای قبلی ثبات AX در آن
- مراجعه به مقدار قبلی روتین وقفه تا وقفه کار عادی خود را انجام دهد.
- oldvect متغیری است که آدرس روتین وقفه را در خود دارد.

بازنویسی وقفه

- بازنویسی وقفه در دو مرحله انجام می شود:
  - استخراج آدرس CS:IP روتین وقفه از جدول بردار وقفه و ذخیره کردن آن در یک متغیر
  - جایگزین کردن آدرس روتین وقفه با آدرس روال مقیم در حافظه در جدول بردار وقفه

استخراج آدرس روتین وقفه

این کار توسط سرویس 35h دستور int 21h به صورت زیر انجام می شود:

```
mov ah,35h

mov al,09h

int 21h
```

- قرار دادن 35h در ثبات AH
- قرار دادن شماره وقفه ای که می خواهیم آدرسش را استخراج کنیم در ثبات AL (08h یا 09h)
- اجرای دستور int 21h

- با اجرای دستورات فوق، مقدار CS:IP روتین وقفه 09h در ثبات های ES:BX قرار می گیرد.
- مقادیر استخراج شده در یک متغیر مثل oldvect ذخیره می شود.

جایگزین کردن آدرس روتین وقفه با آدرس روال

این کار توسط سرویس 25h دستور int 21h به صورت زیر انجام می شود:

```
mov ah,25h
mov al,09h
mov dx, offset newISR
int 21h
```

- قرار دادن 25h در ثبات AH
- قرار دادن شماره وقفه (08h یا 09h) در ثبات AL
- قراردادن افست آدرس شروع روال در ثبات DX
- اجرای دستور int 21h

- پس از اجرای دستورات فوق، آدرس شروع روال newISR به جای آدرس روتین وقفه 09h (در جدول بردار وقفه) قرار می گیرد.

مقیم کردن برنامه در حافظه

- در انتهای برنامه ای که شامل روال جایگزین است ، به جای برگشت به سیستم عامل که با دستورات زیر انجام می شود:

```
mov ax,4C00h
int 21h
```

۱. ابتدا اندازه روال جایگزین(برحسب پاراگراف) در ثبات dx قرار گیرد.
۲. سپس سرویس 31h دستور int 21h استفاده می شود:

```
mov ah,31h
int 21h
```

ساختار برنامه های مقیم در حافظه

- برنامه های مقیم در حافظه باید به صورت com نوشته شوند.
- برنامه از دو قسمت تشکیل می شود:
  - قسمت اول : تعریف روال جایگزین وقفه که در حافظه مقیم می شود.
  - قسمت دوم: این قسمت فقط یکبار در موقع مقیم کردن روال جایگزین اجرا می شود و شامل دستورات زیر است:
    - دستورات استخراج آدرس روتین وقفه
    - دستورات جایگزین کردن آدرس روتین وقفه با آدرس روال
    - مقیم کردن برنامه در حافظه

ساختار برنامه های مقیم در حافظه–ادامه

```
page 110,100
title 'TSRTemplate.asm'
;define code segment
codeSG segment 'code'
    assume ss:codeSG,ds:codeSG,cs:codeSG,es:codeSG
    org 100h
start: jmp load1
;-------------------------------------------------
oldvect dd ?        ;4Byte for save CS:IP of ISR
;define other data
;-------------------------------------------------
part1:
newISR proc near
    push ax
    ;
    ;
    ;
    pop ax
    jmp cs:oldvect
newISR endp
;-------------------------------------------------
```

```
part2:
load1 proc far
    ;1-Get vector of ISR and save on oldvect
    ;---------------------------------------
    mov ah,35h
    mov al,--    ;replace -- with INT number
    int 21h
    mov word ptr oldvect,bx
    mov word ptr oldvect+2,es
    ;---------------------------------------
    ;2-Set the offset of newISR in vector table
    ;---------------------------------------
    mov ah,25h
    mov al,--
    mov dx,offset newISR
    int 21h
    ;---------------------------------------
    ;3-make resident of newISR
    ;3-1-find size of newISR and save it in dx
    mov dx,(offset load1-offset codeSG)
    add dx,15
    mov cl,4
    shr dx,cl
    ;3-2-make newISR resident
    mov ah,31h
    int 21h
load1 endp
codeSG ends
end start
```

تمرین

- برنامه ای بنویسید که در حافظه مقیم شود و هر لحظه که کلیدهای Alt+F10 فشارداده شوند، کامپیوتر بوق بزند.
- برنامه ای بنویسید که در حافظه مقیم شود و هر لحظه که کلیدهای Alt+F10 فشارداده شوند، کامپیوتر برای مدتی بوق بزند.
- برنامه ای بنویسید که در حافظه مقیم شود و هر ۳۰ ثانیه کامپیوتر بوق بزند.

راهنمایی: برای برنامه هایی که باید با فشاردادن کلیدهایی اجرا شوند از وقفه 09h و برای برنامه هایی که باید در فواصل زمانی کاری انجام دهند از وقفه 08h استفاده کنید.