

Sistem Programlama

Handel Kavramı:

Windows'da çalışırken pek çok durumda bir işe başlamadan önce işletim sisteminin o işe ilişkin bilgileri depolayacağı bir **alan** yaratması gerekir.

Alanın yaratıldığı bölge Windows'un kendi içerisindeki **heap** bölgesidir.

Handle yaratan CreateXXX isimli bir fonksiyon çağırıldığında verilen handle değerinin türü H ile başlayan türlerdendir. Örneğin:

HINSTANCE
HICON
HWND

Sistem Programlama

Handel Kavramı:

CreateXXX fonksiyonlarıyla tahsis edilen dinamik alanlara Windows sistem programlama terminolojisinde nesne(object) denir(Buradaki nesnenin nesne yönelimli programlamadaki nesneyle alakası yoktur). Windows'da sistem nesneleri 3 grup halinde incelenebilir:

Windows'da sistem nesneleri 3 grup halinde incelenebilir:

1. KERNEL
2. USER
3. GDI

Sistem Programlama

API(Application Programming Interface) Fonksiyonları:

Bir işletim sisteminin çalışırken kendisinin kullandığı, programcının da dışarıdan çağırabileceği fonksiyonları vardır. Bunlara sistem fonksiyonları denir. Sistem fonksiyonlarına unix'te System Call, Windows'da ise API denilmektedir. Sistem fonksiyonları işletim sistemiyle birlikte belleğe yüklenirler. API fonksiyonları bütün programlama dillerinde ve geliştirme ortamlarında işlemleri yapabilmek için derleyici sistemleri tarafından çağırılır. Örneğin Visual Basic'te programcı API kavramını bilmesede aslında derleyici sistemi bütün işlemlerini sistem fonksiyonlarını çağırarak gerçekleştirir. Windows API fonksiyonları dll dosyaları içerisinde ve sistemle beraber otomatik olarak yüklenirler. API fonksiyonları üç gruba ayrılır:

1. Kernel API'ler: Kernel32.dll içerisinde. Sistemin aşağı seviyeli çalışmalarıyla ilgili sistem fonksiyonları bulunur.
2. User API'ler: User32.dll içerisinde. Programcının doğrudan üzerinde çalıştığı konularla ilgili olan sistem fonksiyonlarıdır. Örneğin pencere işlemleri için gereken API'ler, kontrollere ilişkin API'ler vs.
3. Graphics API'ler: Gdi32.dll içerisinde. Bütün grafik çizimlerine ilişkin sistem fonksiyonlarını içerir.

Sistem Programlama

Kernel Nesneleri:

İşletim sisteminin içsel çalışmasına özgü olan nesnelere(handle alanlarına) kernel nesneleri denir. Örnek kernel nesneleri şunlardır:

Process(işlem)

Thread(alt işlem)

Mutex

Event

File

Semaphore

Sistem Programlama

Process(işlem):

Diskteki bir program çalışır duruma geldiğinde artık proses olarak isimlendirilir.

Sistem Programlama

Thread'ler:

Thread bir **alt işlemdir**. Thread bir process'in farklı bir programmış gibi çalışabilen parçalarına denir.

Process ana işlemdir ve bir process n adet thread'den oluşmaktadır.

Thread kullanımı Win32 sistemleriyle mümkün hale getirilmiştir.

Her process çalışmaya tek bir thread'le başlar. Buna ana thread denir.

Process'in çalışması ana thread'den başlar. Diğer thread'ler herhangi bir zamanda herhangi bir thread içerisinde CreateThread API fonksiyonuyla yaratılırlar.

Win32'nin zaman paylaşımli çalışma sistemi process temelinde değil thread temelinde yapılmaktadır.

Sistem Programlama

İşlevlerine Göre Thread'lerin Sınıflandırılması

Uyuyan Thread'ler(sleepers threads): Bu tür thread'ler bir olay oluşana kadar beklerler. Olay oluşunca bir faaliyet gösterirler. Sonra o olay oluşana kadar yeniden beklerler. Tabii bekleme işi thread bloke edilerek yapılmaktadır.

Tek İş Yapan Thread'ler(one shot threads): Bu thread'ler bir olay gerçekleşene kadar bekler. Olay gerçekleşince faaliyet gösterir ve thread'in çalışması biter.

Önceden Faaliyet Gösteren Thread'ler(anticipating threads): Burada ileride yapılacak bir işlem önceden yapılır. Eğer akış o işleme gerek kalmayacak bir biçimde geliyorsa işlem boşuna yapılmış olur. Eğer akış o işlemin farklı bir biçimde yapılmasını gerektirecek bir şekilde geliyorsa o işlem yeniden yapılır.

Beraber Faaliyet Gösteren Thread'ler: Burada spesifik bir iş vardır. CPU'dan daha fazla zaman alacak biçimde işlem birden fazla thread tarafından yapılır.

Bağımsız Faaliyet Gösteren Thread'ler: Bu thread'ler tamamen farklı amaçları gerçekleştirmek için yazılır. Genellikle bir senkronizasyon problemi oluşturmazlar. Tasarım kolaylığı sağlamak amacıyla kullanılırlar.

Sistem Programlama

Thread'ler ve Mesaj Sistemi:

Bir thread GUI uygulamalarından hiçbir pencere yaratmamışsa böyle thread'lere “Worker Thread” denir. Eğer thread en az bir pencere yaratmışsa bu tür thread'lere “User Interface Thread” denir. Win32'de her thread'in ayrı bir mesaj kuyruğu vardır. Yani bir thread içerisinde bir pencere yaratmışsak derhal WinMain'de yaptığımız gibi bir mesaj döngüsü oluşturmamız gerekir. Aslında GetMessage fonksiyonu her zaman o anda çalışmakta olan thread'in mesaj kuyruğundan mesajı alır.

Sistem Programlama

Event Senkronizasyon Nesnesi:

Bu senkronizasyon nesnesi belirli bir olay gerçekleşene kadar bir thread'i bloke durumda bekletmek için kullanılır. Örneğin klavyeden bir tuşa basılana kadar ya da bir push button'a click yapılana kadar bir thread bekletilebilir. Event senkronizasyon nesnesiyle işlem yapmak için kullanılan bir grup API fonksiyonu vardır. Event senkronizasyon nesnesi bir kernel nesnesidir. CreateEvent fonksiyonuyla yaratılır.

Sistem Programlama

Semaphore Senkronizasyon Nesnesi

Bu senkronizasyon nesnesi bir kaynağa istenilen sayıda kodun aynı anda erişebilmesini sağlamaktadır. Critical Section fonksiyonları paylaşılan kaynağa yalnızca bir kodun erişebilmesine olanak sağlar. Semaphore nesnesi CreateSemaphore API fonksiyonu ile yaratılır, CloseHandle API fonksiyonuyla yok edilir.

Sistem Programlama

Mutex Senkronizasyon Nesnesi

Process'lerin thread'leri arasındaki senkronizasyonda kullanılabilir. Mutex nesneleri thread temelinde senkronizasyon sağlar. Mutex nesneleri CreateMutex API fonksiyonu ile yaratılır. CloseHandle API fonksiyonu ile yok edilir.

Sistem Programlama

Dosya Sistemine İlişkin API Fonksiyonları

Win32'de dosyanın açılması, kapatılması, dosyadan okuma ve yazma yapılması bir grup API fonksiyonuyla gerçekleştirilir. Dosya da bir kernel nesnesidir. CreateFile API fonksiyonuyla yaratılır ve CloseHandle API fonksiyonuyla kapatılır. CreateFile API fonksiyonu yalnızca dosya yaratmakta değil, çeşitli donanım birimleriyle haberleşmek için mailslot, pipe gibi process'ler arası haberleşme mekanizmasının kurulması için de kullanılmaktadır. Ayrıca Win32'nin dosya API fonksiyonları her çeşit network bağlantısı için de içsel olarak işlem yapmaktadır. Örneğin network altında bir makinada çalışan bir process başka bir makinanın harddisk'inde dosya açabilir. Dosya fonksiyonları kendi içlerinde network haberleşme yöntemlerini kullanarak makinalar arası dosya işlemlerini gerçekleştirirler. Tabii network altında çalışan veri tabanı uygulamalarında dosya işlemlerinin de seri hale getirilmesi gerekir.

Sistem Programlama

Win32 Mesaj Sistemi:

Win32'de her thread'in ayrı bir mesaj kuyruğu vardır. Bir thread bir pencere yarattığında artık thread için bir mesaj kuyruğu yaratılır ve pencere ile ilgili işlemler mesaj biçiminde mesaj kuyruğuna yazılır. Bir thread'in yarattığı bütün pencerelerin mesajları aynı mesaj kuyruğuna yazılır. Örneğin bir thread on tane pencere yaratmış olsun. Bu pencerelerden herhangi birisine ilişkin mesaj thread'in mesaj kuyruğuna yazılacaktır. Yani bu durumda bir thread'de pencereler yaratıldıktan sonra mesajları işlemek için bir mesaj döngüsü oluşturmak gerekir.

Sistem Programlama

Process'ler Arası Haberleşme

Win32 sistemlerinde process'lerin bellek alanları izole edildiği için bir process'in diğer bir process'e bilgi göndermesi ancak işletim sisteminin sağladığı özel yöntemlerle yapılabilir. Tabii aynı process'in thread'leri arasındaki haberleşme global değişkenler yoluyla kolay bir biçimde yapılabilir. Process'ler arası haberleşme yöntemlerinin bir bölümü yalnızca bir makinanın process'leri arasındaki haberleşmede kullanılabilir. Diğer bir bölümü ise genel olarak network altında herhangi iki makinanın process'leri arasındaki haberleşmede kullanılır.

Sistem Programlama

Bellek Tabanlı Dosya Kullanımı:

Belek tabanlı dosya bir dosyanın RAM'e aktarılarak RAM üzerinde işlenmesine olanak sağlayan yeni bir tekniktir. Bir dosya bellek tabanlı olarak açılırsa dosyanın bellekteki başlangıcına ait doğrusal bir adres elde edilir. Sanki dosya bellekteymiş gibi dosya üzerinde işlem yapılabilir. Örneğin o bellek alanı güncellendiği zaman gerçekte dosya üzerinde işlem yapılmış olur. Bu teknik Windows programcılığında iyi bilinmemesi nedeniyle maalesef seyrek kullanılmaktadır.

Bellek tabanlı dosya kullanılmasının ilk adımı, dosyanın CreateFile API fonksiyonuyla normal olarak açılmasıdır. CreateFile ile dosya açıldıktan sonra dosyaya ilişkin bir handle elde edilir:

Sistem Programlama

Bellek tabanlı dosyalar üç durumda Win32'de kullanılmaktadır:

CreateProcess fonksiyonu exe ve dll dosyalarını belleğe yüklerken bellek tabanlı dosya kullanmaktadır.

Bellek tabanlı dosyalar ile klasik dosya işlemleri çok daha kolay bir biçimde ve kolay bir algı ile yürütülebilir.

Bellek tabanlı dosyalar process'ler arası haberleşmede de yoğun olarak kullanılmaktadır.

Sistem Programlama

Exe ve Dll Dosyalarının Bellek Tabanlı Dosya Kullanılarak Yüklenmesi

CreateProcess fonksiyonu çalıştırılacak olan exe dosyayı disk üzerinde arayarak onun için bellek tabanlı dosya oluşturur ve exe dosyayı default olarak 0x00400000(4Mb) adresinden itibaren belleğe yükler.

Default olarak dll dosyaları ise 0x10000000(256Mb) adresinden başlayarak yüklenmektedir.

Sonuç olarak exe ve dll dosyalarından swap işlemi page file üzerinden değil, doğrudan bu dosyalar üzerinden yürütülmektedir.

Bir exe dosya bölümlerden(section) oluşur. Bir bölüm koruma bilgileri aynı olan n ardışık sayfadan oluşur. Örneğin .text koda ilişkin bölüm, .data ilk değer verilmiş statik nesnelerin saklandığı bölüm, .bss ilk verilmemiş statik nesnelerin oluşturduğu bölüm, .const ise const nesnelerin tutulduğu bölümdür. Exe ve dll dosyaları yüklenirken aslında bütün dosya bellek tabanlı olarak açılmaz. Yalnızca data bölümleri bellek tabanlı açılır, .text bölümü VirtualAlloc tahsisatıyla page file üzerinde swap edilecek biçimde yüklenir.

Sistem Programlama

TCP/IP Network Haberleşmesi

Network haberleşmesinde internet ortamını da kapsayan ve en popüler olarak kullanılan yöntem TCP/IP yöntemidir. TCP/IP pek çok protokolün oluşturduğu bir IP yöntemidir.

Windows'da TCP/IP network haberleşmesi windows socket API fonksiyonlarıyla yapılır.aslında TCP/IP haberleşmesi için iki grup API fonksiyonu kullanılmaktadır.

Winsock API'leri
Wininet API'leri

Winsock API grubu aşağı seviyelidir ve bu gruptaki pek çok fonksiyon UNIX socket fonksiyonuyla uyumludur. Oysa wininet API grubu çok yüksek seviyeli işlemlere izin veren fonksiyonlara ilişikindir.