

Çekirdek Nedir? Ne yapar?

Erkan Esmel

Nisan, 2014

İçindekiler

1	Giriş	2
2	Çekirdek Tipleri	3
2.1	Mikrokernel	3
2.2	Monolitik Kernel	4
2.3	Hybrid Kernel	5

1 Giriş

Çekirdek, en alt katmanda çalışan ve donanımları kullanmamızı sağlayan uygulamalara arayüzlük yapan bir programdır diyebiliriz.

Bir çekirdek;

- Sistem kaynaklarının yönetilmesi
- Görev ve hafıza yönetimi
- Aygıt yönetimi
- Sistem çağrılarının yönetimi

gibi işlemleri yürütür.

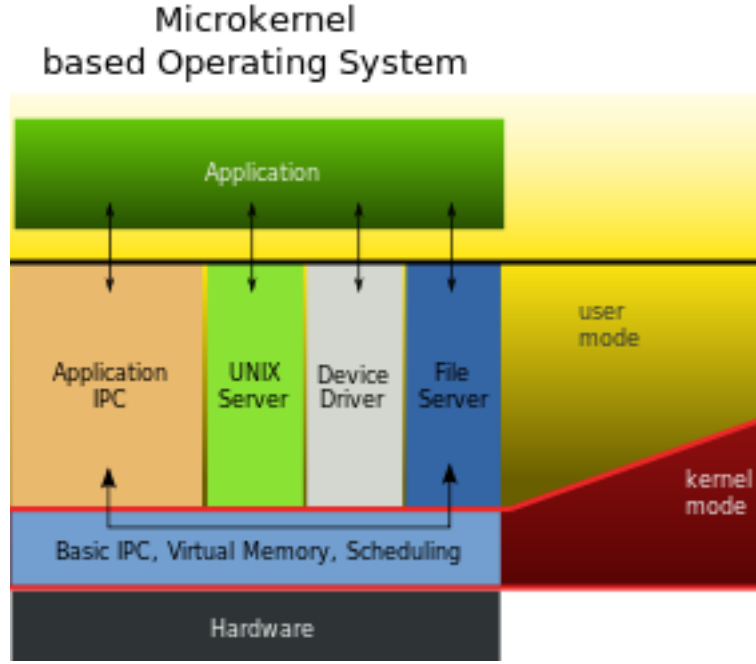
Eğer bir çekirdek yapısından bahsetmezsek, donanım kullanımından da bahsedemeyiz. Dolayısıyla aslında sistemi kullanmaktan bahsedemeyiz.

Linux çekirdeği en büyük açık kaynak projelerinden biridir. Belki de en büyüğüdür diyebiliriz. İçerisinde 13 milyondan fazla kod satırı barındırmaktadır.

2 Çekirdek Tipleri

Monolitik, **Mikrokernel** ve **Hybrid** olmak üzere 3 çekirdek yapısından bahsedebiliriz. Tabii ki duruma ve ihtiyaca göre çekirdek yapısı oluşturma durumu her zaman için geçerlidir. Biz, genel olarak sıklıkla tercih edilen yapı modellerine kısaca değineceğiz.

2.1 Mikrokernel



Şekil 1:

Mikrokernel, işlemci, geçici bellek gibi donanımları ve işlemler arası iletişimi -IPC- yönetir. Mikrokernel çekirdekte en alt seviyedeki fonksiyonlar, kullanıcı seviyesinde servisler ile sunulur. Mikrokernel modüler yapıdadır. Modüller arası trafik gereksinimi olduğundan diğer çekirdek tiplerine nispeten performans sorunları ortaya çıkabilir. İşlemler, kullanıcı seviyesinde sunulduğundan hatalar da burada gerçekleşir ve bütün sistemi etkilemez. Mikrokernel, kullanıcı modunda donanımı kullanır. Yüksek erişim gerektiren işlemlerde bulunmaz.

Artıları

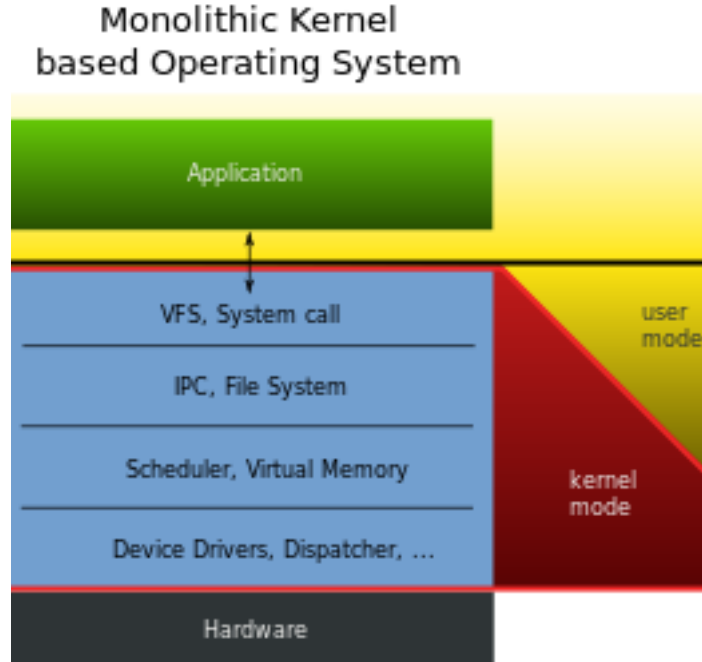
- Taşınabilirdir,
- Küçük alan kaplar.

Eksileri

- Donanım sürücülerinden soyutlanmıştır.
- Kullanıcı katmanında çalışır.

Süreçler kuyrukta beklemek zorundadır. Süreçlerden biri tamamlanmadan diğeri işleme alınmaz. Önem sırası veya duruma göre direkt işleme alma özelliği yoktur.

2.2 Monolitik Kernel



Şekil 2:

Monolitik Kernel, Mikrokernel gibi sadece işlemci, hafıza gibi bileşenleri yönetmez. Bunlarla beraber aygıt sürücülere, dosya sistemi yönetimi gibi konular da monolitik kernelin yönetimi kapsamındadır. Monolitik Kernel, tek bir dosyadan oluşur ve tüm gereksinimleri bu dosyadadır. Dolayısıyla modüler değildir. Bu da yapılacak işlemlerde süreç sırası dezavantajını ortadan kaldırır.

Monolitik Kernel'de Mikro Kernel'in aksine çok görevlilikten ve daha etkin donanım erişiminden bahsedebiliriz. Bir programdan bilgi almak veya başka bir işlemi gerçekleştirmek gerektiğinde süreç sırasında bekleme durumu yoktur. Bunun gibi durumlarda direkt gerçekleştirme veya erişim söz konusudur. Bu da performans konu olduğunda olumlu sonuç doğurur.

Monolitik Kernel'de birçok işlem yönetici modunda ve tek bir süreçmiş gibi çalıştığından bir hata meydana geldiğinde tüm sistem bundan etkilenir.

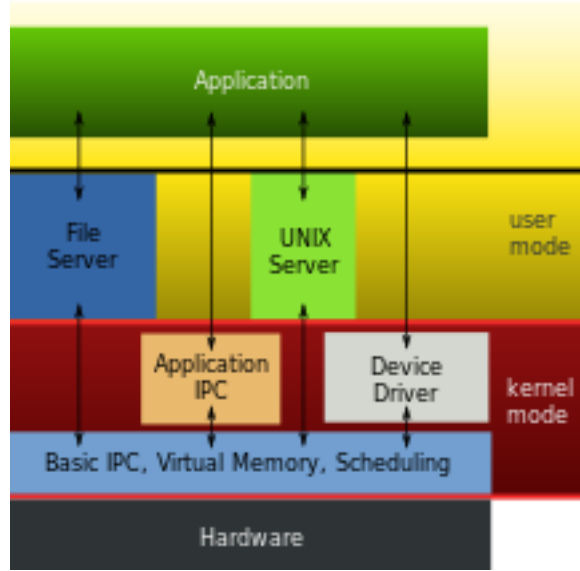
Artıları

- Programlar için donanıma direkt erişim vardır.
- Süreçlerin birbirleriyle olan iletişimi ve geçişi daha kolaydır.
- İşlemler daha hızlı gerçekleştirilir. Çünkü bir işlemin gerçekleştirilmesi için kuyrukta beklemesi gerekmez.

Eksileri

- Büyük alan gereksinimi vardır.
- Yüksek hafızaya ihtiyaç duyar.
- Herşey yönetici modunda çalışır.

"Hybrid kernel" based Operating System



Şekil 3:

2.3 Hybrid Kernel

Hybrid Kernel, diğer kernel tiplerindeki gibi tek bir modda çalışmaz. Yani duruma göre kullanıcı modu ve yönetici modu çalışma durumuna sahiptir. Genellikle donanım sürücülerini ve dosya sistemi gibi bileşenlerde kullanıcı modunda çalışır. IPC ve sunucu çağrılarını gibi bileşenlerde ise yönetici modunda çalışır.

Artıları

- Hem kullanıcı modunda hem de yönetici modunda çalışması seçilebilir.
- Monolitik Kernel'e göre daha az yer gereksinimi vardır.
- Diğer modellere göre her iki durumu da sağladığı için esnekler.

Eksileri

- Diğer iki kernelin de hatalarını barındırır.
- Aygıt sürücülerini kullanıcı modunda çalıştırır.

şimdi de konuya işletim sistemleri açısından ve Linux tarafından bakalım.

Linux monolitik çekirdek kullanır. Diğer bildiğimiz işletim sistemleri MacOS ve Windows ise hybrid çekirdek yapısında.

Linux'un monolitik çekirdek kullanması tartışmaya açık ve her zaman tartışılan bir konu olmuştur. Monolitik çekirdek, yapısı itibarıyla tek dosya üzerinde çalışır ve tüm gereksinimlerini bu dosya üzerinde bulundurur. Mesela bir aygıt sürücüsü, çekirdeğe bütünleşiktir. Bu sebeple Monolitik Çekirdek çok kararlı bir yapıya sahiptir.

Monolitik Çekirdek'te sistemimizde tanınmayan bir aygıt varsa o aygıtın sürücüsünü sisteme ekleme şansımız yoktur. Bu işlem için çekirdeği baştan derlememiz gerekir.

Örneğin Nvidia ve Ati sürücülerini çekirdeğe eklenmediği için sistem tarafından tanınmaz ve bunu düzeltebilmek için çekirdek katmanında işlemler yapmak, çekirdeği tekrar derlemek, inşa etmek gerekir. Bu işlemde çıkabilecek bir problemde istem kullanılmaz hale gelebilir.

Bu durum da son kullanıcı bazında işlem tipinin, sayısının durumuna göre bir dezavantaj sayılabilir. Fakat bu dezavantaj gibi görünen konulara rağmen bu mimari güncel işletim sistemleri arasında en çok platform desteği sağlayan mimaridir.

Bununla beraber PCI Hotplug, Cpu Hotplug gibi teknolojilerin kullanıldığı ilk mimaridir. Bu da mimarinin gelişme hızı ve duruma göre reaksiyon alma kabiliyetine önemli bir işarettir.

Hybrid Çekirdek yapısında ise yukarıda da söylediğimiz gibi diğer 2 çekirdeğin de özellikleri söz konusudur. Monolitik çekirdekten önemli ayrımı herhangi bir sürücü yüklemesinde çekirdek derlemeye gerek yoktur. Bu anlamda daha fazla esneklikten bahsedebiliriz fakat kararlılık, esnekliğe ters oranda ilerleyebilir ve tatışmaya açıktır.

Monolitik çekirdeğin sağladığı kararlılığın yanında esnekliğe açık olmadığını düşünenler Linux tarafında da Mikrokernel çalışması yapmaktalar. Fakat uzun süredir devam eden bir proje olmasna rağmen kararlı bir sürüme henüz kavuşmamıştır.

Merak ederseniz projenin sitesini inceleyebilirsiniz.\n<http://www.gnu.org/software/hurd/hurd.html>

Denemek isterseniz de yükleme şansınız var.\n<http://www.debian.org/ports/hurd/hurd-cd>

Belki ileriki sayılarımızda hurd konusuna daha detaylı değiniriz.

**Toparlayacak olursak **

Son kullanıcı ve sistem kullanımı tarafında çekirdek, yapı, mimari tartışmaları hep devam edecektir. Özellikle son kullanıcı bazında göreceli durumlar yeni tartışmalara ve dolayısıyla yeni fikirlere ve üretime sebep olacaktır.