

## Linux işletim sistemi nasıl çalışıyor?

Kullanıcıların çoğu **Windows** işletim sistemi kullanıyor. Daha az bir kesim **Mac OS** ve daha da az bir kesim **Linux** kullanıyor.

Linux işletim sisteminin çalışma prensipleri diğerlerinden oldukça farklı. Kullanımı biraz daha zorlaştırmasına karşın çeşitli avantajları söz konusu. Biz bu makalemizde hangisi en iyi sorusuna hiç girmiyor, sizleri Linux işletim sisteminin altında yatan **detaylar** ile tanıştırıyoruz.

Açık kaynak kodlu projeler pek çok farklı geliştirici tarafından oluşturuluyor ve sürdürülüyor. Linux çeşidi de, uygulama ve uygulamaların kullandığı yazılım çeşidi de oldukça kalabalık. Ancak neyin, nereden geldiğini ve nasıl çalıştığını biraz dikkatle anlamak mümkün. **Bir kez giriş yaptıktan sonra da oldukça ilginç ve çekici olduğunu farkedebilirsiniz...**

## Kurulum zorluğunun esas sebebi...

Linux işletim sisteminin temelinde kullanıcı izinleri ve yetkileri yatıyor. Bu, işletim sistemini kullanırken sık sık **şifre** girmeyi gerektirse de sonuçta güvenlik açısından büyük bir artı sağlıyor.

Uygulama kurarken Windows gibi **.exe** dosyası indirip, çift tıklayıp ardından hızlıca kurulumu tamamlamak mümkün değil. Buna yaklaşan kurulumlar olsa da bunlar Linux **mantığına** ters düşüyor.

Kullanıcıların **/root** dizinini görememesi ve değişiklik yapamaması da güvenlik amaçlı. Ya **Administrator** hesabından girmek ya da günlük kullanımda şifre girmek gerekiyor.

Linux dosya sisteminde 9 özellik var. Bir kullanıcı, bir grup ya da herkese, **okuma, yazma ya da çalıştırma izni** şeklinde seçenekler ile kim o dosyaya, hangi amaçla erişebilir belirlemek mümkün. Bu ayarları koruyarak yeni bir sürüme aktarabilmek, işleri çok kolaylaştırıyor.

## Masaüstü ortamları

Linux işletim sistemindeki **masaüstü** aslında bir uygulamadır ve kullanıcı ile uygulamalar arasındaki etkileşimi sağlar. Yani masaüstü işletim sisteminin kendisi değil, kullanıcı erişimi için bir penceredir.

**API** denilen Application Programming Interface'ler kullanılarak, programlara farklı yetenekler kazandırılabilir. Bu işlevler örneğin masaüstünün pencere çizim yeteneklerini bir programa kazandırmakta kullanılır.

Aynı API'lerin kullanılması işletim sistemini daha temiz, düzenli ve verimli tutar. Linux'ta kullanılan **Gnome** ve **KDE** masaüstü uygulamaları da bu API'lerin standartlaşmasını sağlar.

Elbette uygulamalar tek bir API kullanmakla yetinmez. Çok çeşitlidir ve programcılar için **Clutter** ve **Cairo** gibi araç kitleleri API kullanımını kolaylaştırır. Clutter hem **Ubuntu**, hem **Moblin**'de kullanılır çünkü düşük donanım gücüne sahip cihazları yormadan donanım hızlandırmalı masaüstü grafikleri sunabilir. Bu tam bir karşılaştırma amacı taşımayan, basit bir örnek.

## KDE ve Gnome masaüstleri

Uygulamaların haberleşmesi de işletim sisteminin çalışmasında kritik bir parçadır. Taktığınız **donanımlar** algılanır, otomatik sonuçlar üretilir. Örneğin kablosuz ağı açarsınız ve otomatik olarak bağlanır, uygulamalar güncelleme yapmaya başlar.

Gnome masaüstünde **GTK** uygulamaları, KDE uygulamaları **KDE** masaüstünde daha iyi performans sunar. Ancak her ikisi de uygulamalar arasında haberleşmek için aynı D-BUS sistemini kullanır.

En büyük fark masaüstünün ve pencerelerin görünümü ve çalışma biçimidir. KDE pencere yöneticisi **KWin** pencereleri sekmeli çerçevelere yerleştirebilir ve gölgeler, yansımalar gibi efektler sunabilir. Gnome'da standart olan **Compiz** ise daha gösterişlidir. Farklar ve detaylar ayrı ve uzunca bir başka incelemenin konusu olduğu için, Linux'un diğer önemli özellikleriyle makalemize devam ediyoruz...

## Linux'un Windows'tan en büyük farkı

Linux'un hem en büyük başarılarından, hem de en büyük sorunlarından birisi, internette indirilen bir **uygulamanın** kolayca kurulamamasıdır.

İndirilen uygulamalar, **Windows'ta** bütün gerekli parçaları tek pakette barındırır ve buradan çalıştırır. Windows içerisinden kullanılacak olan **API'ler** vb için ise dosya bağılıkları listesi statik olarak **setup.msi** içinde bulundurulur. Linux'ta ise bu parçalar işletim sisteminin kütüphanelerinde durur ve ortak olarak kullanılır.

Dağıtımların bağlı olduğu farklı dosyaları buldurmak ve bütün bu bağılıkların gereklerini yerine getirmek göz korkutucu olabilir. Ancak bu sayede gerekli parçalar güncellendiğinde, tek tek bütün uygulamaların **güncellenmesi** gerekmez. Bağlı bütün uygulamalar, kapatılan bir güvenlik açığından korunmuş olur. Bu, kolaylık pahasına sağlanan çok büyük bir avantajdır.

## X ve PAM

Linux'ta kullanılan **X** teknolojisi, ekrana pencereleri çizen, klavye ve fareyi kontrol eden teknoloji ve 20 seneye yakın süre dayandı.

X altında **client** ve **server** yani istemci ve sunucu tabirleri kullanılır. Bu mantık, işletim sisteminin Unix zamanlarından kalmadır ve genellikle ana sisteme bağlı klavye-fare ikilisi ve ekranlar döneminden kalmadır.

X sisteme bağlı **istemcilerin** arasında veri alışverişini sağlar, ana sistem hesaplar ve istemciler ekrana veriyi getirir ve etkileşimi sağlar. Ancak bu tanım Linux'ta farklıdır, sunucu ekran ve klavyeye sahip olan parçadır ve istemci de **işlemci** gücüne sahip parçadır. Bu normal istemci-sunucu mantığının tam tersi şekilde adlandırılmıştır. X çapraz bir şekilde adlandırır çünkü X kullanıcıya **kaynakları** sunar ve uygulamalar istemciler olarak kaynakları kullanır.

Sunucu ve istemci aynı bilgisayardadır ve bu mantık köklere kadar geçerlidir. **PAM** isimli (Pluggable Authentication Modules) sistem ile giriş yapılır ve PAM da güvenliği sağlar.

**Modüler** güvenlik özellikleri ile konfigürasyon dosyalarının ve verilerin ele geçmesini engeller.

### **Komut satırının avantajları**

**Grafik** arayüzler için içine girmeden önce bilgisayar işlemleri yazılı komutlar ile başlatılıyordu.

Şimdi ikonlara ve kısayollara tıklayarak bu komutları kolayca çalıştırabiliyoruz. Ancak **arayüzün kısayollar** ve komutlarla dolmaması için bazı komutlar halen komut satırı üzerinden yazılarak kullanılıyor.

Linux kullanmak biraz da **komut** satırı kullanmayı gerektiriyor. Grafik arayüzün yapamadığı derecede derin kontrol sağlamak, komut satırı ustalığı ile mümkün.

İster komut satırı tabanlı, ister grafik bir arayüz kullanın, şimdi Linux temellerine biraz daha inmenin zamanı. Bilgisayarı açar açmaz başlayan **sysinit**, kullanacağınız uygulamaları sırası ile başlatır. Bunlar yüklendikten sonra kullanılabilirler ki, arayüz de bunlardan birisidir.

### **Linux Kernel**

**Linux kernel**'i işletim sisteminin çekirdeğini oluşturur. Kernel bütün sistem kaynaklarının kontrolünden sorumludur. Dosya sistemi, uygulamalar, sürücüler, ağ bağlantısı, kullanıcı alanları, hafıza ve dahası kernel tarafından yönetilir.

Kullanıcı için en önemlilerinden birisi **sürücülerdir**. Çoğu Linux sürümünde sürücüler kernel'in bir parçası olarak verilmez, modüller olarak bulunur. Modüllerde eksik varsa da kullanıcı kendi modüllerini oluşturabilir ya da paket yöneticisi içerisinde kuararak ihtiyaçlarını karşılayabilir.

Sistem yöneticisi olarak **lsmod** komutunu kullanarak kernel modüllerini görmek ve dosya bağılıklarını anlamak mümkündür. Kernel'e özel olan **modüller**, kernel güncellemesinde sorun çıkartarak ekran kartı sürücüsünün hata vermesine yol açabilir. Bu gibi durumlarda sorun modülün farklı sürümünü kurarak çözülür.

`/lib/modules` dizininde bulunan bu modülleri ve dahasını kurcalayıp keşfedebilirsiniz. Daha fazlası ise `/lib/modules/kernel_name/kernel/drivers` dizini altında yer alır. Hangi modülü kuracağınız konusunda genellikle Linux bilgi kaynaklarına ve araştırma yapmaya ihtiyacınız vardır.

Bir süre uğraşırsınız ama neyi nasıl yapacağınızı anladığınızda işiniz kolaylaşır. Linux dünyasına hoşgeldiniz...