

Python’a Merhaba ve Temel Kavramlar

Table of contents

Bölüm 1: Açık Kaynak Felsefesine Giriş	1
1.1. Giriş	1
1.2. Açık Kaynak ve Özgür Yazılım Nedir?	2
1.3. Açık Kaynağın Tarihçesi	2
1.4. Açık Kaynağın Avantajları ve Dezavantajları	3
1.5. Açık Kaynak Lisansları (Detaylı İnceleme)	4
1.6. Linux’un Doğuşu ve Linus Torvalds	5
1.7. Açık Kaynak Toplulukları ve İş Modelleri	6
1.8. Sonuç	7

Bölüm 1: Açık Kaynak Felsefesine Giriş

1.1. Giriş

Günümüzde bilgisayarlar, akıllı telefonlar, sunucular ve hatta otomobiller gibi hayatımızın her alanında kullandığımız cihazlar, yazılımlar sayesinde çalışır. Peki, bu yazılımlar nasıl geliştirilir? Kimler tarafından kontrol edilir? İşte bu noktada “açık kaynak” kavramı devreye giriyor.

Bu bölümde, açık kaynak ve özgür yazılım kavramlarını tanıyacak, tarihçesini öğrenecek, avantajlarını ve dezavantajlarını tartışacak ve açık kaynak topluluklarının nasıl çalıştığını keşfedeceğiz.

1.2. Açık Kaynak ve Özgür Yazılım Nedir?

Açık kaynak yazılım (Open Source Software - OSS), kaynak kodu herkes tarafından erişilebilir, incelenebilir, değiştirilebilir ve dağıtılabilir olan yazılımlardır. Bu, yazılımın “gizli” veya “kapalı” olmadığı, tam tersine şeffaf ve işbirliğine açık olduğu anlamına gelir.

Özgür yazılım (Free Software) ise, kullanıcılara dört temel özgürlük sunan yazılımlardır:

1. **Çalıştırma Özgürlüğü:** Yazılımı herhangi bir amaç için kullanma özgürlüğü.
2. **İnceleme Özgürlüğü:** Yazılımın kaynak kodunu inceleme ve nasıl çalıştığını anlama özgürlüğü.
3. **Değiştirme Özgürlüğü:** Yazılımı ihtiyaçlara göre değiştirme ve uyarlama özgürlüğü.
4. **Dağıtım Özgürlüğü:** Yazılımın kopyalarını (değiştirilmiş veya orijinal haliyle) başkalarıyla paylaşma özgürlüğü.

Önemli Not: “Özgür” kelimesi burada “ücretsiz” anlamına gelmez. Özgür yazılımlar genellikle ücretsiz olarak dağıtılsa da, ticari olarak da satılabilirler. Önemli olan, kullanıcılara yukarıdaki dört özgürlüğün sunulmasıdır.

Açık Kaynak ve Özgür Yazılım Arasındaki Fark:

Pratikte, açık kaynak ve özgür yazılımlar genellikle aynı yazılımlardır. Ancak aralarında felsefi bir fark vardır:

- **Özgür yazılım hareketi**, yazılımın özgür olmasının etik ve sosyal bir gereklilik olduğunu savunur. Kullanıcıların yazılımlar üzerinde kontrol sahibi olması gerektiğini vurgular.
- **Açık kaynak hareketi**, açık kaynak yazılım geliştirme modelinin daha pratik ve verimli olduğunu savunur. İşbirliği ve şeffaflığın daha iyi yazılımlar üretilmesini sağladığını vurgular.

1.3. Açık Kaynağın Tarihçesi

Açık kaynak felsefesi, bilgisayar biliminin ilk dönemlerine kadar uzanır. 1950’ler ve 1960’larda, üniversiteler ve araştırma laboratuvarlarındaki yazılımcılar, kodlarını birbirleriyle özgürce paylaşıyor ve birlikte geliştiriyorlardı.

Ancak 1970’lerin sonlarında ve 1980’lerin başlarında, yazılım şirketleri telif hakları ve lisanslar aracılığıyla yazılımları “kapalı” hale getirmeye başladı. Bu durum, yazılımın özgürce paylaşılmasını ve geliştirilmesini engelledi.

Richard Stallman ve GNU Projesi:

1983 yılında, Richard Stallman adlı bir bilgisayar bilimcisi, bu gidişata tepki olarak **GNU Projesi**’ni başlattı. GNU Projesi’nin amacı, tamamen özgür yazılımlardan oluşan bir işletim

sistemi yaratmaktı. Stallman, 1985 yılında **Özgür Yazılım Vakfı**'nı (Free Software Foundation - FSF) kurdu ve özgür yazılım hareketinin temellerini attı.

Linux'un Doğuşu:

1991 yılında, Linus Torvalds adlı Finlandiyalı bir üniversite öğrencisi, GNU Projesi'nin işletim sistemi çekirdeği (kernel) olan **Linux**'u geliştirmeye başladı. Linux, açık kaynak lisansı altında yayınlandı ve kısa sürede dünyanın dört bir yanındaki yazılımcıların ilgisini çekti. Linux, GNU Projesi'nin diğer bileşenleriyle birleştirilerek bugün kullandığımız GNU/Linux işletim sistemini oluşturdu.

Açık Kaynak Hareketi:

1990'ların sonlarında, "açık kaynak" terimi daha yaygın olarak kullanılmaya başlandı. Netscape'in tarayıcısı Mozilla'yı açık kaynak olarak yayınlaması, açık kaynak hareketinin önemli bir dönüm noktası oldu. Günümüzde, açık kaynak yazılımlar internetin altyapısından mobil cihazlara, süper bilgisayarlardan yapay zekâya kadar her alanda kullanılmaktadır.

1.4. Açık Kaynağın Avantajları ve Dezavantajları

Avantajları:

- **Şeffaflık:** Kaynak kodunun açık olması, yazılımın nasıl çalıştığının herkes tarafından incelenebilmesini sağlar. Bu, güvenlik açıklarının daha hızlı tespit edilmesine ve hataların daha kolay düzeltilmesine yardımcı olur.
- **Güvenlik:** Açık kaynak yazılımlar, daha fazla göz tarafından incelendiği için genellikle daha güvenlidir. Güvenlik açıkları daha hızlı tespit edilir ve kapatılır.
- **Esneklik:** Kullanıcılar, yazılımı ihtiyaçlarına göre değiştirebilir ve uyarlayabilirler. Bu, özellikle işletmeler ve kurumlar için büyük bir avantajdır.
- **Maliyet:** Birçok açık kaynak yazılım ücretsiz olarak kullanılabilir. Bu, özellikle bütçesi kısıtlı olan kullanıcılar ve kurumlar için önemlidir.
- **İşbirliği:** Açık kaynak yazılım geliştirme modeli, dünyanın dört bir yanındaki yazılımcıların işbirliği yapmasına olanak tanır. Bu, daha hızlı ve verimli bir geliştirme süreci sağlar.
- **Topluluk:** Açık kaynak projelerinin etrafında genellikle güçlü topluluklar oluşur. Bu topluluklar, kullanıcılara destek sağlar, bilgi paylaşımını teşvik eder ve yazılımın gelişimine katkıda bulunur.
- **Bağımsızlık:** Açık kaynak yazılımlar, kullanıcıları belirli bir şirkete veya satıcıya bağımlı olmaktan kurtarır.

Dezavantajları:

- **Destek:** Bazı açık kaynak yazılımlar için ticari destek bulmak zor olabilir. Ancak, birçok popüler açık kaynak proje için ücretli destek seçenekleri mevcuttur.

- **Kullanım Kolaylığı:** Bazı açık kaynak yazılımlar, özellikle yeni başlayanlar için kullanımı zor olabilir. Ancak, birçok açık kaynak proje, kullanıcı dostu arayüzler ve kapsamlı belgeler sunar.
- **Uyumluluk:** Bazı açık kaynak yazılımlar, belirli donanım veya yazılımlarla uyumlu olmayabilir. Ancak, açık kaynak toplulukları genellikle uyumluluk sorunlarını çözmek için hızlı bir şekilde çalışır.
- **Parçalanma:** Bazen açık kaynak kodun çatallanması (forklanması) ile çok sayıda farklı ve birbiri ile uyumsuz proje oluşabilir.

1.5. Açık Kaynak Lisansları (Detaylı İnceleme)

Açık kaynak yazılımlar, belirli lisanslar altında yayınlanır. Bu lisanslar, yazılımın nasıl kullanılabileceğini, değiştirilebileceğini ve dağıtılabilceğini belirler. Açık kaynak lisansları, genel olarak iki ana kategoriye ayrılır:

1. **Copyleft (Telif Feragatli) Lisanslar:** Bu lisanslar, yazılımın değiştirilmiş sürümlerinin de aynı lisans altında yayınlanmasını zorunlu kılar. Bu, yazılımın özgürlüğünün korunmasını ve “açık kalmasını” sağlar. En yaygın copyleft lisans, GNU General Public License (GPL)’dir.
2. **Permissive (İzin Verici) Lisanslar:** Bu lisanslar, yazılımın değiştirilmiş sürümlerinin farklı lisanslar altında yayınlanmasına izin verir. Bu, yazılımın daha geniş bir yelpazede kullanılmasına olanak tanır, ancak yazılımın “kapalı kaynak” hale gelme riskini de taşır. En yaygın permissive lisanslar MIT, Apache ve BSD lisanslarıdır.

Aşağıdaki tablo, en yaygın açık kaynak lisanslarını karşılaştırmaktadır:

Lisans Adı	Kategori	İzin Verilenler	Kısıtlamalar
GNU GPL (v2, v3)	Copyleft	Yazılımı kullanma, inceleme, değiştirme, dağıtma; ticari kullanım	Değiştirilmiş sürümler GPL altında yayınlanmalı; orijinal telif hakkı bildirimi ve lisans metni korunmalı.
GNU LGPL (v2.1, v3)	Copyleft	GPL’e benzer, ancak kütüphaneler (libraries) için daha esnek; LGPL lisanslı bir kütüphaneyi kullanan bir yazılımın tamamının LGPL altında yayınlanması zorunlu değildir.	Kütüphanede yapılan değişiklikler LGPL altında yayınlanmalı; orijinal telif hakkı bildirimi ve lisans metni korunmalı.
MIT License	Permissive	Yazılımı kullanma, inceleme, değiştirme, dağıtma; ticari kullanım; özel kullanım; alt lisanslama; patent hakkı verme (bazı sürümlerde)	Orijinal telif hakkı bildirimi ve lisans metni korunmalı.

Lisans Adı	Kategori	Verilenler	Kısıtlamalar
Apache License 2.0	Permissive	MIT'e benzer, ancak patent hakları konusunda daha açık ve kapsamlıdır; patent ihlallerine karşı koruma sağlar.	Orijinal telif hakkı bildirimi ve lisans metni korunmalı; değişiklikler belirtilmeli.
BSD License (2-clause, 3-clause)	Permissive	MIT'e çok benzer, ancak bazı sürümleri ek kısıtlamalar içerebilir (örneğin, yazılımı tanıtmak için orijinal geliştiricilerin adının kullanılmasına izin verilmemesi).	Orijinal telif hakkı bildirimi ve lisans metni korunmalı.
Mozilla Public License (MPL) 2.0	Zayıf	Diğer lisanslarla (GPL, LGPL, Apache vb.) Copy-uyumlu, hibrit bir lisanstır. Dosya bazında copyleft uygular, yani MPL lisanslı bir dosyada değişiklik yaparsanız, bu dosyayı MPL altında tutmalısınız.	Değişiklik yapılan dosyalar MPL altında kalmalı, orijinal telif hakkı bildirimi ve lisans metni korunmalı.

Lisans Seçimi:

Açık kaynak bir proje başlatırken doğru lisansı seçmek önemlidir. Lisans seçimi, projenin hedeflerine, hedef kitlesine ve geliştiricilerin tercihlerine bağlıdır.

- **Maksimum özgürlük ve açıklık isteyenler:** GPL gibi güçlü bir copyleft lisans tercih edebilir.
- **Daha geniş bir kitleye ulaşmak ve ticari kullanımı teşvik etmek isteyenler:** MIT veya Apache gibi izin verici bir lisans tercih edebilir.
- **Kütüphaneler için:** LGPL veya MPL gibi daha esnek copyleft lisanslar uygun olabilir.

1.6. Linux'un Doğuşu ve Linus Torvalds

1991 yılında, Helsinki Üniversitesi'nde bilgisayar bilimi öğrencisi olan **Linus Torvalds**, hobi olarak kendi işletim sistemi çekirdeğini (kernel) geliştirmeye başladı. O dönemde, ticari UNIX sistemleri oldukça pahalıydı ve GNU Projesi'nin özgür işletim sistemi çekirdeği (Hurd) henüz tamamlanmamıştı.

Torvalds, Minix adlı küçük bir UNIX benzeri işletim sisteminden ilham alarak, Intel 386 işlemciler için kendi çekirdeğini yazmaya başladı. Başlangıçta "Freax" adını vermeyi düşündüğü bu çekirdeğe, daha sonra arkadaşları tarafından **Linux** adı verildi.

Torvalds, Linux'un ilk sürümünü (0.01) 17 Eylül 1991'de internet üzerinden yayınladı ve kaynak kodunu **GNU GPL** lisansı altında paylaştı. Bu, Linux'un açık kaynak dünyasındaki yolculuğunun başlangıcı oldu.

Linux'un açık kaynak olarak yayınlanması, dünyanın dört bir yanındaki yazılımcıların ilgisini çekti. Kısa sürede, birçok gönüllü Linux'u geliştirmeye, hataları düzeltmeye, yeni özellikler eklemeye ve farklı donanım platformlarına uyarlamaya başladı.

Linux, GNU Projesi'nin diğer bileşenleriyle (derleyici, kabuk, kütüphaneler vb.) birleştirilerek tam teşekküllü bir işletim sistemi haline geldi. Bugün, **GNU/Linux** olarak bilinen bu işletim sistemi, sunuculardan süper bilgisayarlara, masaüstü bilgisayarlardan akıllı telefonlara kadar her alanda kullanılmaktadır.

Linus Torvalds ve Açık Kaynak Felsefesi:

Linus Torvalds, Linux'u geliştirmesindeki temel motivasyonunun "eğlence" olduğunu ve ticari bir amaç gütmeyeceğini her zaman vurgulamıştır. Linux'un başarısının ardındaki en önemli faktörlerden biri, Torvalds'ın açık kaynak felsefesine olan bağlılığı ve işbirlikçi geliştirme modelini benimsemesidir.

Torvalds, Linux'un geliştirme sürecini "kaotik" ama etkili bir şekilde yönetmiştir. "Release early, release often" (Erken yayımla, sık yayımla) prensibini benimseyerek, yeni sürümleri hızlı bir şekilde yayınlamış ve kullanıcılardan gelen geri bildirimleri dikkate almıştır.

Torvalds, Linux çekirdeğinin "patronu" olarak kabul edilse de, geliştirme sürecinde her zaman katılımcı ve demokratik bir yaklaşım sergilemiştir. Önemli kararlar genellikle toplulukla tartışılır ve fikir birliğine varılarak alınır.

1.7. Açık Kaynak Toplulukları ve İş Modelleri

Açık kaynak projeleri, genellikle gönüllü yazılımcılardan oluşan topluluklar tarafından geliştirilir. Bu topluluklar, forumlar, e-posta listeleri, sohbet odaları ve diğer çevrimiçi platformlar aracılığıyla iletişim kurar ve işbirliği yapar.

Açık kaynak yazılımlar, farklı iş modelleriyle gelir elde edebilir:

- **Destek ve Bakım:** Şirketler, açık kaynak yazılımlar için ücretli destek ve bakım hizmetleri sunabilir.
- **Özelleştirme ve Danışmanlık:** Şirketler, açık kaynak yazılımları müşterilerinin ihtiyaçlarına göre özelleştirebilir ve danışmanlık hizmetleri sunabilir.
- **Çift Lisanslama:** Yazılım, hem açık kaynak hem de ticari lisans altında yayınlanabilir. Bu, kullanıcılara yazılımı ücretsiz kullanma veya ek özellikler ve destek için ticari lisans satın alma seçeneği sunar.
- **Bağışlar:** Birçok açık kaynak proje, kullanıcıların ve kurumların bağışlarıyla desteklenir.
- **Bulut Hizmetleri:** Açık kaynak yazılım, bir bulut hizmeti olarak sunulur ve gelir elde edilebilir.

- **Eğitim ve Sertifikasyon:** Açık kaynak yazılım ile ilgili eğitimler verilebilir.

1.8. Sonuç

Açık kaynak, yazılım dünyasında devrim yaratan bir felsefedir. İşbirliği, şeffaflık ve özgürlük ilkeleri üzerine kurulu olan açık kaynak, daha iyi, daha güvenli ve daha erişilebilir yazılımların geliştirilmesini sağlamıştır. Linux'un başarısı, açık kaynak modelinin gücünün en somut örneklerinden biridir.

Bu bölümde öğrendiğimiz temel kavramlar, açık kaynak işletim sistemlerini ve Linux'u daha iyi anlamamıza yardımcı olacaktır. Açık kaynak lisanslarının çeşitliliği, geliştiricilere projelerinin ihtiyaçlarına en uygun olanı seçme esnekliği sunar. Açık kaynak toplulukları ise, bilgi paylaşımı, işbirliği ve yenilikçilik için önemli bir platform sağlar.

Tartışma Soruları:

1. Açık kaynak yazılımın, kapalı kaynak yazılıma göre ne gibi avantajları ve dezavantajları vardır? Kendi deneyimlerinizden örnekler veriniz.
2. Sizce açık kaynak felsefesi, yazılım dünyası dışındaki alanlarda (örneğin, eğitim, bilim, sanat) da uygulanabilir mi? Nasıl?
3. Hangi açık kaynak lisansının, hangi tür projeler için daha uygun olduğunu düşünüyorsunuz? Neden?
4. Açık kaynak topluluklarına katılmak, bir yazılımcıya ne gibi faydalar sağlayabilir?
5. Gelecekte açık kaynak yazılımların rolünün nasıl değişeceğini öngörüyorsunuz?

Ek Okuma Önerileri:

- “The Cathedral and the Bazaar” - Eric S. Raymond (Açık kaynak geliştirme modelini anlatan klasik bir eser)
- “Free as in Freedom” - Sam Williams (Richard Stallman'ın biyografisi)
- “Just for Fun” - Linus Torvalds ve David Diamond (Linus Torvalds'ın otobiyografisi)
- Açık Kaynak Girişimi (Open Source Initiative - OSI) web sitesi: <https://opensource.org/>
- Özgür Yazılım Vakfı (Free Software Foundation - FSF) web sitesi: <https://www.fsf.org/>

Proje Önerisi:

- İlginizi çeken bir açık kaynak projesi seçin (örneğin, bir işletim sistemi, bir metin düzenleyici, bir web tarayıcısı, bir oyun vb.).
- Projenin web sitesini, belgelerini ve kaynak kodunu inceleyin.
- Projenin tarihçesini, amacını, lisansını, topluluğunu ve iş modelini araştırın.

- Projenin nasıl çalıştığını anlamaya çalışın (mümkünse, projeyi kendi bilgisayarınızda derleyin ve çalıştırın).
- Projeye nasıl katkıda bulunabileceğinizi düşünün (örneğin, bir hata raporu gönderme, bir çeviri yapma, bir özellik önerme, kod yazma vb.).
- Araştırmanızı ve düşüncelerinizi özetleyen kısa bir rapor veya sunum hazırlayın.