

Özgür Yazılım, Özgür Türkiye

Yazan:

Oktay Altunergil

<oktay (at) ozguryazilim.org>

<http://www.altunergil.com>

Eylül 2003

Özet

Özgür yazılım kavramına ve bize neler getirebileceğine kısa bir göz atmaya çalışacağız.

Geçmiş

1.0

Eylül 2003

OA

İlk sürüm.

Yasal Uyarı

Bu belgenin, *Özgür Yazılım, Özgür Türkiye*, **telif hakkı © 2003 Oktay Altunergil**'e aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan **GNU Genel Kamu Lisansı**^(B2)'nin 2. ya da daha sonrakı sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticarî isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

Bilgisayarın Bileşenleri

Özgür yazılımın ne olduğunu anlayabilmek için öncelikle bir bilgisayarın yapı taşlarının neler olduğunu tekrarlamakta fayda var. Gündelik hayatımızda giderek daha da kalıcı bir yer edinen bilgisayarlar donanım ve yazılım olmak üzere iki ana gruptaki ürünlerin birlikte çalışmaları sonucu kendilerine verilen görevi yerine getirirler.

Donanım, bilgisayarımızı bir araya getiren fiziki (elle tutulur) parçalara verilen genel addır. Örnek olarak bilgisayarın ana işlemcisi, hafızası, sabit diski, klavyesi ve monitörü donanım ürünleridir. Ana donanım bileşenlerinden bazılarının eksik olması durumunda bilgisayarımız işlevselliğini yitirecektir. Örneğin ana işlemcisi olmayan bir bilgisayarın çalışması beklenemez. Öte yandan ikincil dereceden önemli bazı donanım bileşenlerinin yokluğunda bilgisayarımız daha az becerili de olsa çalışmasına devam edecektir. Örneğin, ses kartı olmayan bir bilgisayar, ses üretme yeteneğinden yoksun olsa da diğer amaçlar için hala kullanılabilir.

Bilgisayarlarımız ana kartları, işlemcileri ve son model ekran kartları ile bir teknoloji harikası olabilirler. Fakat bu bileşenlerin işlev kazanmasını sağlayacak diğer önemli bileşen yazılım olmadan bilgisayarımız 2 ile 2'yi çarpıp 4 sonucuna ulaşmaktan bile acizdir.

Yazılım, insanların anlayacağı dilden olan problemleri, istekleri ve işleri bilgisayarların anlayacağı dile çeviren araçtır. Verimli bir bilgisayar oturumu için, kullandığınız yazılımın ihtiyaçlarınıza uygun olması en az kullandığınız donanım kadar önemlidir. Hatta, bilgisayar teknolojisinin ileri bir seviyeye ulaştığı günümüzde, iyi donanım bulmak nispeten kolay iken, iyi yazılıma ulaşmada büyük sorunlarla karşılaşılabilir.

Yazılımı da kendi içinde çeşitli sınıflara ayırmak mümkündür. Günümüz bilgisayarları asgari olarak "işletim sistemi" denen ve çeşitli donanım bileşenleri ile iletişim ve işbirliğini sağlayan ana yazılıma gereksinim duymaktadırlar. Temelde işletim sistemi yazılımının görevleri oldukça kısıtlı olduğu halde genelde bütün işletim sistemi üreticileri bu işletim sistemini kullanıcıların günlük hayatını kolaylaştıracak çeşitli yazılım ürünleri ile pekiştirirler. Hoşça vakit geçirmek için oynadığımız bilgisayar oyunları, arkadaşlarımızla iletişim için kullandığımız anında ileti sistemleri, web sitelerini gezmek için kullandığımız tarayıcılar, bilgisayarda müzik dinlemek için kullandığımız programlar da işletim sistemi dışında olan yazılımlara örnektir. (Gerçekte yazılımın sınıflandırması biraz daha karmaşık ama bizim konumuz için bu kadar bir ayırım yeterli.)

Yazılım Ciddi ve Zor Bir İştir

Yazılım üretmek belli bir bilgi birikimi ve bu alanda yeterli beceri ve yetenek gerektiren bir iştir. Program yazarken yazılımcı akla gelebilecek bütün olasılıkları hesaba katıp, bu olasılıkların tamamını uygun bir şekilde ele almalı ve programın karşılaşılabilecek her koşulda gerektiği gibi tepki vermesini sağlamalıdır. Bilgisayar genel kanının aksine zeka sahibi bir makine olmadığından bizim karşı karşıya kaldığımızda akıl ve tecrübelerimizden bir çıkarıma giderek doğru yolu bulacağımız durumlarda dahi kör bir şekilde kendisine yazılımcı tarafından iletilen adımları takip edecektir. Örneğin, beklenmedik bir şekilde, yazılımcının öngörmediği halde gerçekleşebilecek bir durum örneği olarak, bilgisayarınız kendisini üzerinde çalıştığınız bütün dokümanları silmeye yönelik bir adım ile karşı karşıya bulunduğunda, zeka sahibi varlıkların tersine, bu adımı seve seve takip edecektir.

Makul bütün olasılıkları öngörmeye ilaveten, yazılımcının ayrıca bilgisayarın gerçekleştirmesi gereken her işlemi tek tek ve açık olarak sıralaması gerekmektedir. Bilgisayar ne bir adımın eksik olduğunu ne de takip ettiğinde istenmeyen sonuçlar doğuracak bir adımla karşı karşıya olduğunu farketme yeteneğine sahiptir. Bu yönden bakıldığında bilgisayar yazılımcının emirlerine itaat eden, olağan üstü hızı ve sıkılmadan ve yorulmadan çalışma azmine rağmen kendi kendine düşünme yeteneği olmayan bir makineden ibarettir. Bilgisayarın bütün bu özelliklerinden dolayı, yazılım üretmek – özellikle iyi yazılım üretmek – neredeyse kendince bir sanat sayılabilecek, **zor** bir iştir.

Özgürlük Nedir?

Bilgisayarımızın bileşenleri, yazılımın bilgisayar kullanımındaki yeri ve kaliteli yazılım üretimindeki zorluklar konusuna kısaca değindik. Fakat esas olarak Özgür Yazılım'ın ne olduğunu anlamak için yola çıkmıştık. Yazılımı anladığımıza göre, geriye 'özgürlük' kavramı kalıyor. Nedir bu özgürlük? Yazılım gibi bir konuyla ne ilgisi olabilir?

Türk Dil Kurumu sözlüğü'ne göre:

özgürlük

1. Herhangi bir kısıtlamaya, zorlamaya bağlı olmaksızın düşünme veya davranma, herhangi bir şarta bağlı olmama durumu.
2. Her türlü dış etkidен bağımsız olarak insanın kendi iradesine, kendi düşüncesine dayanarak karar vermesi durumu, **hürriyet**.

olarak tanımlanmıştır.

Özgür Yazılım Derneği'ne göre bu özgürlük tanımı yazılım kavramına aşağıdaki şekilde uyarlanabilir:

Özgür yazılımın temelinde kullanıcının bir yazılımı çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme özgürlükleri yatar. Daha kesin ve açık bir ifadeyle, kullanıcılara şu haklar tanınmıştır:

- Her türlü amaç için programı çalıştırma özgürlüğü.
- Programın nasıl çalıştığını inceleme ve kendi gereksinimleri doğrultusunda değiştirme özgürlüğü. Program kaynak koduna erişim bunun için bir ön şarttır.
- Yeniden dağıtma ve toplumla paylaşma özgürlüğü.
- Programı geliştirme ve gelişmiş haliyle topluma dağıtma özgürlüğü.

Böylece yazılım bütün toplum yararına geliştirilmiş olur. Program kaynak koduna erişim bunun için de bir önşarttır.

Nedir bu Kaynak Kodu denen şey?

Yukarıda yer verdiğimiz Özgür Yazılım Derneğine göre özgür yazılımın gereklerine baktığımızda derneğin tekrar tekrar **kaynak kodu** denen bir şeyden bahsettiğini ve ısrarla bu kaynak kodunu talep ettiğini görüyoruz. Peki nedir bu kaynak kodu? Özgürlükten bahsederken bu kadar ısrarla üzerinde durulmasının sebebi ne olabilir? Acaba Özgür Yazılım derneği laf arasında bir punduna getirerek normalde vermeyeceğimiz bir şeyi bizden hile ile almaya mı çalışıyor?

Yazılım üretmenin zor olduğundan bahsederken yazılımcının bizim anlayacağımız işleri, amaçları ve çözülmesi gereken sorunları bilgisayarın anlayacağı bir şekilde sıralaması gerektiğinden bahsetmiştik. Bizim kelimeleri bir araya getirerek konuşmamız ve bu şekilde – çoğu zaman – anlaşabilmemize benzer olarak bilgisayarlar da kendi dillerini konuşurlar. Bu dil bizim dilimize göre oldukça kısıtlıdır ve yalnızca iki kelime, iki harf veya iki seçenek olarak basite indirgeyebileceğimiz 0 (sıfır) ve 1 (bir) değerlerinden oluşur. Bu iki yapı taşı ile donanım seviyesinde, açık veya kapalı, var veya yok, doğru ya da yanlış kavramları kullanılarak bilgisayara ne yapması gerektiği söylenmelidir. Bilgisayar cevap verirken yine aynı dili kullanacaktır. Bilişim tarihinin ilk günlerinde yazılım üretme işini sadece bu dili gerçek anlamda bilen, başka bir deyişle bilgisayarca konuşabilen insanlar gerçekleştiriyorlardı. Yazılımcı 0 ve 1’ce konuşarak bilgisayardan ne istediğini adım adım anlatıyor, bilgisayarın cevabını ise yine aynı şekilde 0 ve 1’ce olarak alıp sonuca ulaşıyordu. Bu 0 ve 1’ce zamanında yazılım üretmek **gerçekten zordu**. Bilişim alanında gelişmeler oldukça yine bu bilgisayarca konuşabilen yazılımcılar tarafından tercüme aracı denebilecek yazılımlar üretilmeye başladı. Derleyici (compiler) tabir ettiğimiz bu yazılımların görevi yarı insanca yarı bilgisayarca olarak tanımlayabileceğimiz yeni bir dilde yazılan adımları, bilgisayarın anlayacağı 0 ve 1’ceye çevirmektir. Bilgisayar kendi bünyesinde ve bileşenleri arasında hala bilgisayarca konuşmaya devam etmekte fakat bu tercüman yardımı ile en azından yeni nesil yazılımcıların ne dediğini anlayabilmektedir.

İşte bu insanların da anlayabileceği tercüme öncesi metne, kaynak kodu diyoruz. Kaynak kodunun güzelliği, gerekli eğitimi almış insanların bu metni okuyup anlayabilmesi gerekirse kendi metnini üretilip bilgisayara derdini anlatabilmesidir. Dileyen yazılımcı başka birinin yazdığı kaynak kodunu alıp ona ufak tefek değişiklikler yaparak bilgisayara derdini uzun uzun yeniden anlatmak zorunda kalmadan kendi işini de halledebilir.

Program çalıştığına göre kaynak kodu bende zaten var. Değil mi?

Bilgisayarım gayet olağan bir şekilde çalıştığına ve bilgisayar da programlarımın dilini anladığına göre bende zaten bu kaynak kodu denen metin hali hazırda yok mu? Özgür Yazılım savunucuları neden bahsediyorlar? Daha ne istiyorlar?

Buraya kadar okuyup bu sonuca ulaşmanız gayet doğal çünkü yukarıdaki tercüme işleminde gerçekleşen bir adımı atladık. Gerçekte yeni nesil yazılımcı bilgisayarca’ya tercüme edilmek üzere özel bir dilde yazdığı metni kullanıma hazır hale getirmek için bir kaç belli işlemi gerçekleştirmeli. Birinci adım, metnin dil kurallarına uygun olduğundan emin olmak. Metin dilin kurallarına uyuyorsa, tercüme edilmeye hazır demektir. Tercüme işlemini derleyici ile yapıyoruz ve bilgisayarca’ya tercüme edilmiş bir ürüne ulaşıyoruz. Bu aşamada elimizde başta kendimiz yazdığımız kaynak kodu ve tamamen bilgisayarca olan son ürün var. Son ürünün ne demek

istediğini bilgisayarımız çok iyi anlıyor ve ondan istediğimiz her işlemi güzelce gerçekleştiriyor. Bu aşamada bilgisayarca olan ürünü bir arkadaşımıza versek elinde kaynak kodu olmadığı halde bunu kendi bilgisayarında çalıştırdığında bilgisayar bize itaat ettiği gibi ona da itaat edecek ve bizim için gerçekleştirdiği işlemleri onun için de gerçekleştirecektir. Gördüğünüz gibi bir programın çalışması için kaynak kodunun da hazır bulunması gerekmiyor. Tercüme edilmiş metni aynı iş için yeniden kullanabilmek çok güzel bir özellik, fakat malesef özgürlük sözkonusu olduğunda yeterli değil.

Program zaten çalışıyor. Kaynak kodunu ne yapayım ben?

Arkadaşınızdan size böyle bir yazılım ürünü geldiğini düşünün. Elinizde kaynak kodu yok. Programı kullanıyorsunuz ve aynen arkadaşınızın size anlattığı gibi çalışıyor. Fakat program ekranında arkadaşınızın size bahsetmediği bir tuş var. Sizce bu tuş ne yapıyor olabilir? Programı açıp okumaya teşebbüs etseniz, daha önce bahsettiğimiz 0 ve 1 'lerle karşılaşacaksınız ve ne demek istediğini anlayamayacaksınız. Tuşa basarak deneyebilirsiniz, fakat ya bu tuş istemediğiniz bir işlem gerçekleştirirse? Tuş örneğinden daha kötüsü, programda varlığını görmediğiniz fakat yine de orada olan bir özellik veya bir hata olmadığını nereden biliyorsunuz? Arkadaşınıza güvenebilirsiniz diyelim, fakat yüzlerce dolar vererek aldığınız programınızın üreticisine ne kadar güvenebilirsiniz? Daha programı kullanmaya başlarken lisans anlaşmasında her türlü sorumluluğun size ait olduğunu ve üreticinin hiç bir şekilde sorumlu tutulamayacağını söyleyen onlar değil miydi? Oysa elinizde kaynak kodu olsaydı, o kodu kendiniz okuyabilir veya arkadaşınıza okutup neler olup bittiğini öğrenebilirdiniz. Belki dünyanın en kolay işi olmazdı, ama en azından mümkün olurdu.

Kaynak Kodu ve açıklık prensibi

Demek ki sadece kaynak kodunda olan, bu koddan derlenen programda olmayan ilk özellik, kaynak kodunun insanlar tarafından anlaşılabilir bir dilde yazılmış olması. Bu özellik kaynak kodundan üretilen programa olağan üstü bir açıklık kazandırıyor. Kaynak kodu elinizde olduğu sürece program sizden gizli bir işlem veya üreticinin herhangi bir kötü niyeti olmadığı halde yine de programda mevcut olan istenmeyen bir adım gerçekleştirilmeden önce kaynak koduna bakıp önlem alabiliyorsunuz. Beklenmeyen bir sonuçla karşılaştığınızda kaynak kodunu okuyarak sebebini araştırabiliyorsunuz. Kısaca yazılım amacını, kullandığı yöntemleri, belli şartlarda izleyeceği yolu, beklenmedik bir olayda hareket edeceği şekli ve zayıf ve sorun yaratabilecek yerlerini sizden gizleyemiyor, sizi gözü kapalı bir şekilde, ne olacağından emin olmadan programı kullanmaya zorlamıyor.

Yukarıda gözler önüne sermeye çalıştığımız açıklık ilkesi, bir çok yönden özgürlük ilkesi ile örtüşmektedir. Seçme özgürlüğünün gerek şartlarından belki en önemlisi mevcut seçeneklerin açık olarak tetkik edilebilmesi ve seçenekler hakkındaki önemli bütün bilgilere seçim aşamasında ulaşılabilmesidir.

Yazılımı Değiştirebilme Özgürlüğü

Açıklık kavramı tek başına vazgeçilmemesi gereken bir niteliktir. Kullandığı yazılımın kaynak koduna erişimi olan kullanıcı veya yazılımcı, yazılımı kullandıkça, yazılımda eksikler, gelişmeye açık noktalar olduğunu farketmeye başlayacaktır. Benzer bir şekilde, kullanıcı yazılımı gündelik olarak kullandıkça, yazılımın özgün halini yazıp derleyen üreticinin öngörmediği ihtiyaçlarını da bu yazılım ile kolayca giderebileceğini farkedecektir. Bu ihtiyaçları yazılıma ilave etmek her zaman üreticinin menfaatleri bakımından tercih edilen bir durum olmayabilir. Üreticinin istenen konuda yeterli tecrübesi olmayabilir. Üretici başka bir konu üzerinden bütün kaynaklarını kullanıyor ve sizin istediğiniz konuya ayıracak zamanı olmayabilir. Üreticinin bu ek özellik için talep ettiği ücreti karşılayamayabilirsiniz. En önemlisi üretici bir kere yazıp milyonlarca kere çoğaltıp dağıttığı bu yazılım üzerinde sizin istediğiniz değişikliği yapmaya tenezzül etmeyebilir. En nihayetinde, sizin ihtiyacınız diğer şahıslar için size olduğu kadar önemli olmayacaktır.

Böyle bir durumda elinizde kaynak kodu yoksa, hayalinizin gerçekleşme ihtimali yok denecek kadar azdır. Mevcut teknoloji ile kaynak kodu olmayan son kullanıcıya yönelik bir program üzerinde değişiklikler yapmak veya programa yeni özellikler eklemek mümkün değildir!

Eğer şanslı iseniz ve elinizde yazılımın kaynak kodu varsa, olaylar çok farklı ve sizin menfaatinize gelişebilecektir. Derleyicilerin anladığı yazılım dillerini biliyorsanız, kaynak kodunu okuyup, ihtiyacınız olan değişikliği yapıp yeni kaynak kodunu tekrar derleyerek işinizi görebilirsiniz. Kendiniz yazılımcı olmayabilirsiniz. Yazılım işinin ciddi ve zor bir iş olduğunu başta söylemiştik. Şimdi de kesinlikle herkesin yazılımı kendi ihtiyaçlarına göre kendisinin değiştirebileceğini imâ etmiyoruz. Eğer kendiniz yazılımı değiştirecek bilgi ve beceriye sahip değilseniz sizin seçeceğiniz kişi veya kuruluşlara gerekli değişiklikleri ücretini ödeyerek yaptırabilirsiniz. Sonuçta esas olan değişikliği kimin yaptığı değildir. Konu sizin kullandığınız bir programa herhangi bir değişikliği gerçekten yapıp yapmayacağınız da değildir. Esas olan gerekli olabilme ihtimaline karşı bu yazılımı değiştirme özgürlüğünün mevcut olmasıdır. Zaten özgürlükleri özgürlük yapan özelliklerden biri de uygulama zorunluluğu olmayıp, uygulamak isteyenlere de kısıtlama getirmemesi değil midir ?

Benim işime yaradı, alın siz de kullanın.

Dünyadaki hiç bir bilgisayar kullanıcısı, bilgisayarını bir diğeriyle tamamen aynı amaçlarla ve tamamen aynı şekilde kullanmaz. Bu kişilerin bilgisayar kullanış amaçları da aynı değildir. Bu kadar farklılık gösteren bir kullanıcı kitlesine rağmen yine de bir çok noktada ihtiyaçlar ve amaçlarda bir ortaklık mevcuttur. Sizin kaynak kodunu değiştirerek eklediğiniz özellik belki başkalarının da işini kolaylaştırabilir. Bunu anlamanın en güzel yolu, değiştirilmiş programı Internet ortamına koyup, ihtiyacı olanların erişebilmesini sağlamaktır. Bu değişikliği başkalarıyla paylaşmanın size bir zararı olamayacağına göre, neden yapmıyorsunuz? Yazılımı halka açtığınız zaman ihtiyacı olan insanlar bu yazılımı alıp kullanmaya başlayacaklardır. Bir çok kişi için yazılımı kullanan kişilerden alınan bir teşekkür kişinin harcadığı bütün çabaların karşılığı olmaya yeterlidir. Yazılımınızı kullanan insanların bir başka yapacağı da kendileri programlama yapamadıkları için size yeni fikirler sunmak olacaktır. Hatta yazılımı kullanan bir kısım insanlar ona kendi değişikliklerini ekleyerek halka açma yoluna gideceklerdir. Bu şekilde önüne geçilmez bir gelişme ve daha iyiye gitme süreci yazılımın hayatı boyunca ve o yazılıma ilgi olduğu sürece devam edecektir.

Malesef insanlığın bir bütün olarak ilerlemesini garanti eden bu kaynak kodunda değişiklik yapıp dağıtma işlemi gerçek hayatta bu kadar kolay gerçekleştirilememektedir. Öncelikle günümüzde geniş çapta kullanılan yazılımların ezici bir çoğunluğu kaynak kodu olmaksızın dağıtılmaktadır. Bu yazılımlar ücretli veya ücretsiz dağıtılabılır. Ödenen ücrete bağlı olmaksızın kaynak kodu olmadan dağıtılan yazılımın üreticinin kendisinin yapmayı tercih edebileceği geliştirmeler hariç olmak üzere geliştirilmesi mümkün değildir. Neredeyse bütün bir yazılım sektörü ücretli veya ücretsiz dağıtılan yazılımların kaynak kodunun gizli tutulup sadece derlenmiş ürünün dağıtma üzerine kurulmuştur.

Kaynak kodunun program ile birlikte dağıtılması her ne kadar sadece derlenmiş ürünün dağıtılmasından nispeten iyi bir durum ise de, tek başına beklentileri karşılamaya yeterli bir yöntem değildir. Günümüzde yazılım konusunu düzenleyen kanunlar kaynak kodu elinizde olsa bile ona değişiklik yapmanızı, değiştirilmiş veya özgün hali ile kaynak kodunu ve derlenmiş ürünü dağıtmanızı yasa dışı sayabilecek bir yapıdadır. Bu gibi durumlarda kaynak kodunun elinizde olup olmaması ulaşılabilecek sonucu değiştirmeyecektir.

Sonuç

Özgür yazılım kavramına ve bize neler getirebileceğine kısa bir göz attık. Bütün toplumun yararına olduğu açık olan bu akımın neden hala bütün dünyadaki yazılım konusundaki tek düşünce tarzı olmadığını merak ediyorsunuz olabilir. Malesef gerçek hayatta bizim değindiğimizden çok daha fazla değişkeni göz önüne almak gerekiyor. İlerleyen günlerde, yeni yazılarla sizlere özgür yazılım konusunda bilgi vermeye ve bu değişkenleri incelemeye devam edeceğiz.

Notlar

- a) Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- b) Konsol görüntüsünü temsil eden sarı zeminli alanlarda metin genişliğine sığmayan satırların sığmayan kısmı `↵` karakteri kullanılarak bir alt satıra indirilmiştir. Sarı zeminli alanlarda `↵` karakteri ile başlayan satırlar bir önceki satırın devamı olarak ele alınmalıdır.

(B2) [../howto/gpl.pdf](http://howto/gpl.pdf)

Bu dosya (ozguryazilim.pdf), belgenin XML biçiminin \TeX Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

18 Şubat 2007