

Yazılım Patentleri Tehlikesi¹

Richard M. STALLMAN

1. Bölüm*

Özgür yazılım üzerine olan çalışmalarımı duymuşsunuzdur. Bu konuşma bunun hakkında değil. Bu konuşma, yasaların yanlış kullanımının, yazılım geliştirmeyi tehlikeli bir etkinlik haline getirmesi hakkında. Burada, patent yasalarının yazılım alanına uygulanmasının sonuçlarından bahsedeceğiz.

Söz konusu olan yazılımın patentlenmesi değil. Böyle bir tanım tamamen kötü ve yanıltıcı. Çünkü patentlenen tek tek programlar de-

ğil. Böyle olsaydı sorun değildi, çünkü temelde zararsız olacaktı. Fakat söz konusu olan fikirlerin patentlenmesi. Her patent bir fikrin patentlenmesi ile ilgilidir. Yazılım patentleri, yazılım üzerine fikirleri, diğer bir deyişle yazılım geliştirmede kullandığımız fikirleri kapsar. İşte bu, onları yazılım geliştirme için bir tehlike haline getirir.

İnsanların yanıltıcı bir terim olan "Fikri Mülkiyet Hakları" nı kullandığını duymuşsunuzdur. Bu terim önyargılıdır. Her ne ile ilgileniyorsanız, onun bir tür mülkiyet kavramı içinde ele alınabileceği varsayımından yola çıkar, oysa bu alternatiflerden yalnızca birisidir. "Fikri Mülkiyet" terimi, ilgili alanda peşinen bir hüküm vermektedir. Bu açık ve tarafsız bir düşünüş değildir.

Kullanılan terimde başka bir problem daha vardır. Olguları anlaşılabilir hale getirmektedir. "Fikri Mülkiyet" derleme bir terimdir: Telif hakları (copyright) ve patentler gibi farklı hukuksal alanlarda yer alan kavramları aynı yere yığmaktadır. Her ayrıntı farklıdır. Ticari markalar (Trademarks) gibi tamamen farklı ve az rastlanan daha birçok şeyi biraraya yığmaktadır. Hiçbirinin bir diğeriyle ortak bir yanı yoktur. Tarihsel olarak tamamen farklıdır; yasal tasarımları birbirinden bağımsızdır; hayatın başka alanlarını ve etkinliklerini kapsamaktadırlar. Kamusal politikalar açısından tamamen farklıdır; dolayısıyla

... kuşkusuz özgürce nefes
alabilirsin ama soluğunu
patent altına
aldık...



* Yazının 2. bölümü derginin 426. sayısında yayınlanacaktır.

1) 25 Mart 2002'de Stallman'ın Cambridge Üniversitesi'nde verdiği konuşma. Emre Uğur, İnan Kanbur ve İzlem Gözükeleş tarafından Türkçe'ye çevrilmiştir.

hepsini bir yere yığmaya çalışırsanız ulaşacağınız sonucun aptalca olacağı garantidir. "Fikri Mülkiyet" hakkında makul ve mantıklı bir kaniya ulaşmak tamamen olanaksızdır. Eğer açıkça anlamak istiyorsanız, hepsini bir yere yığmayın. Önce telif hakkını düşünün, sonra patentleri. Telif hakkı yasalarını öğrenin ve bundan ayrı olarak patent yasalarını düşünün.

Telif hakları ve patentler arasındaki temel farklara gelirsek:

- *Telif hakları, bir işin ifade-sindeki detaylarla ilgilidir; telif hakları fikirlerle ilgilenmez. Sadece patenter, fikirleri ve onların kullanımını kapsar.*
- *Telif hakları, otomatik olarak varolurlar. Patentler ise başvuruya göre patent ofislerince verilirler.*
- *Patentlerin maliyeti yüksektir. Hatta kimi zaman avukatlara verilen para, patentin uygulanmasından daha yüksek tutabilir. Patent ofisleri çoğu zaman üstünkörü bir inceleme yapsalar da bu incelemenin sonuçlanması yıllar alabilir.*
- *Telif hakları uzun ömürlüdür. Bazı durumlarda 150 yıla kadar çıkabilir. Patentler ise 20 yıllıktır, ömrümüzün yetebileceği uzunluktadır. Ancak yazılım gibi bir alanda yine de çok uzun bir zaman aralığıdır. 20 yıl önce PC'nin yeni birşey olduğu günleri düşünün. Yalnızca 1982'de varolan düşüncelerle yazılım geliştirmek zorunda kaldığınızı hayal edin.*
- *Telif hakları sadece kopyalamayı kapsar. Eğer "Rüzgar gibi geçti" ile kelimesi kelimesine aynı bir roman yazarsanız ve "Rüzgar gibi geçti" romanını görmediğinizi ispat ederseniz kendinizi mahkemede telif hakları ihlaline karşı savunabi-*

lirsiniz. Patent ise bir fikri tamamen kuşatan bir monopoldür. Fikrin sahibinin kendiniz olduğunu iddia etmeniz bile, eğer o fikir başkası tarafından patentlenmişse bunun bir anlamı yoktur.

Bu konuşma boyunca telif haklarını aklınızdan çıkaracağınızı ümit ediyorum. Çünkü bu konuşma tamamen patentler üzerine. Eğer yasal konuları tüm açıklığıyla anlamak istiyorsanız, telif haklarını ve patentleri birarada ele almamalısınız.

Suyu ve etanolü birbirinden ayırtedemediğinizde pratik kimyayı (ya da yemek pişirmeyi) ne kadar anlayabileceğinizi düşünün.

İnsanların patent sistemini tanımlamasına bakarsak genelde patent almayı umut eden birinin bakış açısıyla baktıklarını görüyoruz-patent almanın size şöyle bir faydası olacaktır, sokakta cebinizde bir patentle dolaşırken bunu cebinden çıkaracaksınız, sonra birine işaret edip sesleneceksiniz: "Bana para ver."

Bu önyargının bir nedeni var. Bunu söyleyen birçok insanın patent sisteminden bir çıkarı var ve sizden de bu sistemi sevmenizi istiyorlar. Bir başka nedeni daha var: Patent sistemi bir piyangoya benziyor, patentlerin sadece ufak bir bölümü onu elinde bulunduranlara bir fayda sağlıyor. Aslında bir keresinde The Economist dergisi patentleri zaman tüketen bir piyangoya benzetmişti. Piyango reklamlarını görmüşsünüzdür, sizi hep kazancı düşünmeye motive ederler. Kaybettiğinizi düşünmenizi istemezler, bu daha büyük bir olsalık olmasına rağmen. Patent reklamlarında da aynısı var. Sizin hep kazananların tarafında olduğunuzu düşünmenizi istiyorlar.



Bu önyargıyı dengelemek için, patent sistemini kurbanlarının bakış açısından tanımlayacağım. Bir diğer deyişle yazılım geliştirmek isteyen fakat her an patent sisteminin suçlamasıyla karşı karşıya kalabileceklerin bakış açısından.

Dolayısıyla aklınızda yazmak istediğiniz bir program olduğunda yapmanız gereken ilk şey nedir?

İlk başta patent sistemi ile uğraşmak zorundasınız. Yazacağınız programın hangi patentlerle ilişkili olduğunu bulacaksınız. Bu olanaksızdır.

Bunun nedeni, beklemekte olan bazı patent uygulamalarının gizli tutuluyor olabileceğidir. Belirli bir zaman sonra yayınlanırlar, 18 ay gibi. Fakat bu programı yazmak ve onu dağıtmak için oldukça uzun bir süre. Üstelik bir patentle karşılaşacağınızı ve bununla suçlanacağınızı bile bilmeden.

Ayrıca bu salt bir ekonomik sorun da değil. 1984'te compress adında bir veri sıkıştırma programı yazıldı. O sıralarda LZW algoritması üzerine bir patent yoktu. Fakat 1985'de ABD algoritmayı patentledi ve sonraki yıllarda programı dağıtanlar tehdit altında kaldılar.

Compress programının yazarının suçlanacağı aklından bile geçmezdi. Yaptığı yalnızca bir dergide gördüğü algoritmayı uygulamaktı, bir çok programcının her zaman yaptığı gibi. Dergide gördüğü fikirleri artık güvenli bir şekilde gerçekleştiremeyeceğinin farkında değildi.

Bu problemi unutalım. Kabul edilmiş patentler, patent ofisince basılır. Böylece bunların uzunca bir listesine ulaşabilir, ne söylediklerini inceleyebilirsiniz.

Elbette ki listenin tamamını okuyamazsınız, çünkü çok fazla. ABD'de yüzbinlerce yazılım patenti var. Bunların ne üzerine olduğunu takip etmek çok zor. İlgili patentleri bulabilmek için araştırmayı denemek zorundasınız.

Bazı insanlar bilgisayar çağında bunun kolay olacağını söyleyebilir. Anahtar kelimeleri aratabilirsiniz. Ancak bu sadece bir noktaya kadar olanaklı. İlgilendiğiniz alanla ilgili bazı patentleri bulabilirsiniz.

Örneğin, bir zamanlar hesap çizelgelerinde "doğal sıralamanın yeniden hesaplanmasına" dair bir patent vardı (Şimdi kalkmış olabilir). Bunun anlamı, basitçe, eğer bazı hücreleri diğerlerine bağımlı hale getirirseniz, ona bağlı herşeyi yeniden hesaplıyordu; örneğin bir yeniden hesaplamadan sonra herşey güncelleniyordu. İlk hesap çizelgeleri hesaplamaları yukarıdan aşağı yapıyordu, eğer bir hücreyi daha aşağıda bir hücreye bağımlı kılarırsanız ve böyle birkaç adımınız varsa, yeni değerlerin yukarıya tırmanması için çok sayıda yeniden hesaplama yapmanız gerekir (Yukarıdaki hücrelere bağımlı olduğunuz varsayılmakta).

O zamanlar birisi, neden her birşeyi, kendine bağlı olanlar yeniden hesaplandıktan sonra yeniden hesaplamıyorum, diye düşünmeye başladı. Bu algoritma, topolojik sıralama olarak adlandırılır. Buna yapılmış ilk referans bulduğum kadarıyla 1963'te. Patent, topolojik sıralamayı gerçekleştirmek için düzinelerce farklı yol belirtmiş.

Fakat bu patenti hesap çizelgesi diye aratsaydınız bir şeye ulaşamazdınız. Bunu "doğal sıralama" ya da "topolojik sıralama" şeklinde aramanız durumunda da bir şey elde edemeyecektiniz. (Patent) Bu

terimlerden hiçbirine sahip değildi. Aslında, "formüllerin hedeflenen koda yönelik derlenmesi" şeklinde tanımlanmış. İlk gördüğümde yanlış bir patentle ilgilendiğimi sanmıştım.

Şimdi varsayalım ki elinizde bir liste halinde patentler var ve ne yapamayacağınızı görmek istiyorsunuz. Bu patentler üzerine çalışmayı denediğinizde anlaşılmasının çok zor olduğunu göreceksiniz, anlaşılması oldukça zor hukuksal bir dille yazılmıştır. Patent ofislerinin söyledikleri, çoğu zaman asıl belirtmek istedikleri konular değildir.

Avustralya hükümeti 1980'lerde patent sistemi üzerine çalışıyordu. Uluslararası baskı dışında patent sistemine sahip olmak için herhangi bir gerekçe göremediler (kamu için bir faydası yoktu) ve uluslararası baskı olmasaydı bütünüyle ortadan kaldırılabilirdi. Vurguladıkları noktalardan biri mühendislerin patentleri okumakla uğraşmadıkları, çünkü patentlerin anlaşılmasının güç olduğu idi. Bir mühendisin sözlerini aktarmışlardı: "Patent kataloğunda yer alan kendi buluşlarımı anlayamıyorum".

Bu sadece teorik bir durum değildi. 1990 yılı civarında, Paul Heckel adında bir programcı, Hypercard'ın kendine ait birkaç patenti ihlal ettiğini iddia ederek Apple'ı suçladı. Hypercard'ı ilk gördüğünde kendi patentleri ile bir ilişki kuramamıştı. Benzer gözüküyorlardı. Fakat avukatı, kendi patentinin, Hypercard'ın kapak kısmıyla ilgili olduğunu söylediğinde, Apple'a saldırmaya karar verdi. Stanford'da bununla ilgili bir konuşma yaptığımda kendisi de izleyiciler arasındaydı. "Bunun doğru olmadığını sadece kendi haklarının boyutunun farkında ol-

madığını söyledi”. Ben de, “Evet, anlatmak istediğim tam da buydu” dedim.

Dolayısıyla patentlerin size neyi yasak ettiğini öğrenebilmeniz için, avukatlarla uzun uzun konuşmanız gerekiyor. Eninde sonunda diyecikleri aşağı yukarı şöyle olacak:

“Burda birşey yaparsanız kaybedersiniz; şurada birşey yaparsanız (Stallman eliyle havada genişçe bir alan çizmektedir), büyük ölçüde kaybedersiniz. Eğer mutlaka güvende olmak istiyorsanız, bu alandan uzak durun (daha da büyük bir alan çizer). Bu arada herhangi bir davada şansın da önemli payı olduğunu unutmayın.”

Şimdi öngörülebilir bir alanda iş (!) yürüteceksiniz, ne yapmanız gerekiyor? Deneyebileceğiniz 3 farklı yaklaşım var, herhangi biri farklı durumlarda bazen işinize yarayabilir. Bunlar:

1. *Patentten kaçınmak*
2. *Patentin lisansını edinmek*
3. *Patenti Mahkeme Aracılığıyla Geçersiz Kılmak*

İzin verirsiniz bunların hangi durumlarda işe yarayıp yaramadığını açıklayayım:

Patentlerden Kaçınmak

“Patentlerden kaçınmak”- patentler kapsamına giren fikirlerden kaçınma. Bu kolay ya da zor olabilir. Fikrin ne olduğuna bağlı.

Kimi zaman bir özellik patentlenmiştir. O zaman o özelliği kullanmayarak patentlerden kaçınabilirsiniz. Sorun, sözkonusu özelliğin sizin için ne kadar önemli olduğudur.

Bazı durumlarda onsuz yapabilirsiniz. Bir süre önce, XYWrite adlı

kelime işlemcisinin kullanıcıları, mesajları arasında bir eksiltim (downgrade) gördüler. Eksiltim, kısaltmaları önceden tanımlamanızı sağlayan bir özelliğin kaldırılması üzerineydi. Örneğin noktalama işaretleri ile ayrılmış bir kısaltma, hemen ardından açılımıyla yer değiştiriyordu. Böylece uzun kalıplara ait kısaltmaları önceden tanımlayabiliyordunuz. Programı geliştirenler, bunu bana yazdılar. Çünkü Emacs’ın da buna benzer bir özelliği vardı. Aslında, 1970’lerden beri vardı. Hayatımda en azından patentlenebilecek bir fikre sahip olduğumu göstermesi bakımından ilginçti. Patentlenebilir diyorum, çünkü çok sonra başka biri patentledi.

Bu durumda çözüm olarak üç farklı yaklaşımı da denediler. Öncelikle pek de anlaşmaya yanaşmayan patent sahibiyle bir anlaşma yolu arandı. Daha sonra patenti mahkemede geçersiz kılmanın yollarını düşündüler. En sonunda da o özelliği programlarından çıkarttılar.

Bu özellik olmadan da yapabilirsiniz. Eğer kelime işlem programı sadece bu özellikten yoksunsa, insanlar yine de programı kullanmaya devam edebilirler. Fakat bu özellikler birer birer gittiğinde, insanların elinde pek de iyi olduğunu düşünmedikleri bir program kalır ve daha sonra büyük bir ihtimalle de kullanmayı bırakırlar.

Bu, özel bir konuda, daha az kapsamı olan bir patentti. British Telekom’un, telefonla erişimle hiperlinkleri dolaşan patentine ne diyeceksiniz? Hiperlinkleri dolaşma, günümüzde bilgisayarlar için temel bir kullanımdır. Telefonla erişim de temeldir. Bunsuz nasıl yapacaksınız? Ayrıca sözkonusu olan tek bir patent de değil. Yanyana bulunan

iki özelliğin bir kombinasyonu. Bir odada kanepa ve televizyon patentlerine sahip olmak gibi birşey.

Bazen patentlenen düşünce oldukça geniş ve temeldir, tüm bir alanı kontrol altına alabilir. ABD’de patentlenen kamu anahtarlarının şifrelenmesi fikrini ele alalım. Patent 1997 yılında sonlandı. Bu vakte kadar, ABD’de kamu anahtarlarının şifrelenmesi engellendi. Bir sürü program, başlarken bitti; hiçbir zaman kullanılmadılar. Çünkü patent sahipleri, geliştiricileri tehdit etti. O zamanlar, PGP adında bir Özgür Yazılım ortaya çıktı. Anlaşılan, patent sahipleri bunu hedef almayı düşündüklerinde, çok tepki çekeceklerinin farkına vardılar. Bazı sınırlamalar getirdiler, ticari olmayan durumlarda kullanılmasına izin verdiler; bu da yazılımın çok tutulmasının önüne geçti. Böylece kamu anahtarlarının şifrelenmesinin kullanımını, on yıl ya da daha fazla bir süre büyük ölçüde engellediler. Bu patentten kaçış yoktu. Kamu anahtarlarının şifrelenmesi türünden birşey yapmanızın olanağı yoktu.

Bazen de özel bir algoritma patentlenir. Örneğin, HFT (Hızlı Fourier Transformasyon)’nin, optimize edilmiş hali patentlenmiştir. İki katı hızla çalışabilir. Normal HFT’yi kullanarak bu patentten kaçınabilirsiniz. Böylece programınızın o kısmı iki katı yavaş olur. Belki bu sizin için sorun değildir, belki de programınızın sadece ufak bir bölümüdür. İki katı yavaş olsa bile farketmeyeceksinizdir. Belki de bu durumda programınızı hiç çalışmayacaktır. Etkiler değişebilir.

Bazı durumlarda daha iyi bir algoritma bulabilirsiniz. Bu işinize yarar ya da yaramaz. Çünkü compress’i GNU projesinde kullanamayınca

farklı bir sıkıştırma algoritması aramıştık. Birisi elinde bir tane olduğunu söyledi; bir program yazmıştı ve katkıda bulunmak istiyordu. Sadece şans eseri, New York Times'in bir kopyasını gördüm; o zamanlar haftalık patent köşesi vardı (Times'e genelde birkaç ayda bir kereden fazla bakmam. Açtım baktım ve birinin yeni bir sıkıştırma algoritması için patent aldığından bahsediyordu. Bu patente daha dikkatli bakmam gerektiğini anladım, kopyasına baktım. Gördüm ki bizim kullanmak üzere olduğumuz programı kapsıyor. Böylece program doğmadan öldü.

Sonrasında patentlenmemiş farklı bir algoritma bulduk. Bu da şu anda fiili bir standart olan gzip oldu. Veri sıkıştırmada kullanılan bir algoritma olarak güzeldi. Veri sıkıştırmak isteyen herhangi biri

compress yerine gzip kullanabiliirdi.

Aynı LZW sıkıştırma algoritması grafik formatlarında da kullanılıyor, GIF gibi. Fakat orada sorun, sadece basitçe verinin sıkıştırılması değil, insanların bir yazılım aracılığı ile grafiği de görebilmesi gerektiğinden yeni bir algoritmaya geçmek o kadar kolay olmadı. 10 yıl bunun üstesinden gelemedik. Evet, insanlar, GIF formatını kullanmalarından dolayı mahkemelerle karşı karşıya kaldıklarında, farklı bir grafik formatı için gzip algoritmasını kullandılar. İnsanlara, GIF dosyalarını kullanmamalarını ve buna geçmelerini söylediğimizde, "Yapamayız, tarayıcılar henüz bunu desteklemiyor." dediler. Tarayıcı geliştiriciler de: "Bu konuda acelemiz yok. Hem kimse bu yeni formatı kullanmıyor" dedi.

Toplum, GIF formatının kullanımında büyük bir eylemsizlik içindeydi. İnsanları ikna edemedik. Topluluğun hala GIF formatını kullanıyor olması, onları tüm bu tehlikelere açık hale getiriyor.

Gerçekte durum biraz daha garip. LZW sıkıştırma algoritmasını kapsayan iki farklı patent var. Patent ofisi aynı şeyin iki patenti olduğunu da açıklayamıyor; durumu kontrol edemiyorlar. Bunun bir nedeni var. İki patentin gerçekte aynı şeyleri kapsadığını görmek zaman istiyor.

Eğer kimyasal süreçler üzerine patent olsaydı, durum çok daha basit olabilirdi. Hangi maddelerin kullanıldığını bilirdiniz, çıktılarını, fiziksel işlemleri bilirdiniz. Nasıl tarif edildikleri sorun değildi. Ne olduklarını ve benzerliklerini görebilirdiniz. Eğer birşey saf olarak



matematiksel bir haldeyse, onu tanımlamak için birçok yol vardır; çok farklı bir durumdur. Yüzeysel olarak benzer değildir. Onların aynı şeyden bahsettiğini görebilmek için iyice anlamak gerekir. Patent Ofisi'nin vakti yoktur. Birkaç yıl önce patent ofisi, patent başına ortalama 17 saat harcıyordu. Bu dikkatlice düşünmek için yeterli değildir, bu nedenden dolayı elbette benzer hatalar yapıyorlar. Size doğmadan ölen bir programdan bahsetmiştim. Bu programın algoritmalarının da ABD'ce vermiş iki farklı patenti var. Açıkça görüldüğü gibi bu sıradışı bir durum değil.

Patentlerden kaçınma kolay ya da zor olabilir. Kolay olabilir, ama programınız işlevsiz duruma gelebilir. Bu varolan koşullara göre değişir.

Bu da bahsetmek istediğim bir diğer noktadır. Bazen bir şirket ya da konsorsiyum bir formatı ya da protokolü fiili bir standart haline getirebilir. Eğer o format ya da protokol patentlenmişse bu sizin için bir felakettir. 2001'in eylülünde WWW konsorsiyumu patentlerce kapsanmış standartları kullanmaya kalktığında büyük bir gürültü koptu. Topluluk itiraz etti, bunun üzerine kurul tam tersini söylemeye başladı. Her patentin özgürce gerçekleştirilmesinin gerekliliğinde ve her standardın gerçekleştiriminin özgür olmasında ısrar etmeye başladılar. İlginç bir zaferdi. İlk kez standart koyucu bir kurul böyle bir karar veriyordu. Standart koyucu kurulların patentlerle sınırlanmış standartlar koymaları ve insanların özgürce gerçekleştirimini engellemeleri normaldir. Diğer standart kurullarına ulaşmaya ve

onları kurallarını değiştirmeye davet etmeye gereksinimiz var.

Patentin Lisansını Edinmek

Patentten kaçınmak yerine baş vurabileceğiniz ikinci yol patentin lisansını edinmektir. Bununla birlikte, patent sahibinin size lisans teklif etmek, vermek gibi bir yükümlülüğü yoktur. Bundan on yıl önce, *League for Programming Freedom** kendilerinden yardım talep eden bir mektupalmıştı. Mektubu gönderen kişi, kumarhaneler için oyun makineleri üreten bir ailedendi ve ta o zamanlarda, bu iş için bilgisayarları kullanıyorlardı. Bu kişi başka bir şirket tarafından tehdit ediliyordu. Bu şirket, ellerinde patentleri olduğunu ve başkalarının bu makineleri üretmeye izninin olmadığını söy-



* Yazılım patentlerine ve kullanıcı arayüzü telif haklarına karşı kurulmuş bir organizasyon. Bkz. <http://lpf.ai.mit.edu/> (çev.)

luyor; adamdan bu işi bir an önce bırakmasını istiyordu.

Bahsi geçen patenti sonradan inceledim. Patent, bir ağ üzerinde oyun oynamak için kurulmuş bir veya daha fazla sayıdaki bilgisayarı kapsıyordu. Herbir bilgisayar birden fazla oyunu destekliyordu ve belli bir zamanda birden fazla oyun oynamanıza olanak sağlıyordu.

Sizin de göreceğiniz gibi, patent ofisi, herhangi bir şeyin birden fazla yapılmasının gerçekten dahice bir şey olduğu fikrine sahip. Oysa ki, bilgisayar bilimlerinde bu olay, herhangi bir şeyin genelleştirilmesi için kullanılan en açık yoldur. Fakat onlar bu gerçeğin farkında değiller. Bir şeyi bir kere yaptıysanız eğer, aynı şeyi daha sonradan istediğiniz kadar tekrarlayabilirsiniz. Hatta, bunu her zaman kullanabileceğiniz bir çözüm yolu haline getirebilirsiniz. Patent ofisindekilere göre, bir şeyi birden fazla kere yapmış olmanızın anlamı, sizin bir şekilde dahi biri olduğunuz ve kimsenin bu konuda sizinle tartışamayacağıdır. Dahası, sizin, bu konuda başkalarına ne yapmaları gerektiğini söyleme hakkına sahip olduğunuzu düşünürler.

Her neyse. Adama patent verilmedi ve bu yüzden işi bırakmak zorunda kaldı. Mahkemeye gidecek durumu dahi yoktu. Ben olsam, ilgili patentin, herkesin aklına gelebilecek çok açık bir fikre dayalı olduğunu söyledim. Belki benimle aynı fikirde olan bir yargıca bile denk gelinebilirdi. Fakat bunu asla bilemeyeceğiz; çünkü o adamın mahkemeye gidecek gücü bile yoktu.

Fakat yine de, bir çok patent sahibi ellerinde bulundurdukları patentlerin lisanslarını sunmakta

bir sakınca görmezler; çoğu zaman hayli yüksek rakamlar talep etseler de. “doğal sıralamanın yeniden hesaplanması” patentini elinde bulunduran şirket, bu özelliği barındıran her hesap çizelgesi programının Birleşik Devletler’deki satışından elde edilecek toplam gelirin yüzde beşini lisans ücreti olarak istiyordu. Birileri bana bunun, mahkemeye çıkmaksızın ödenilecek en düşük lisans ücreti olduğunu söyledi -eğer gerçekten size dava açmalarına neden olursanız ve kazanırlarsa, muhtemelen daha fazlasını isterler.

Bu yüzde beşlik lisans ücretini ödeyebilecek gücünüz var diyelim; peki ya programınızı geliştirmek için 20 farklı patentin lisansına ihtiyacınız varsa?.. Kazandığınız bütün para bu lisanslara gidecektir. Peki ya 21 lisansa ihtiyaç duyarsanız? İşin içindeki bazı insanlar bana, bu tür iki veya üç lisansın herhangi bir işin gerçekleşmesini imkansız kılacağını söylemişti.

Lisans edinmenin iyi bir çözüm olduğu bir durumun varlığı da söz konusu: çokuluslu, dev bir anonim şirketseniz... Bu tür şirketler zaten bir çok patente sahiptir ve bu nedenle karşılıklı olarak birbirlerine patentlerini lisanslayabilirler. Bu şekilde, patent sisteminin sebep olduğu birçok zarardan uzak dururlar ve sadece iyi olan taraflarından yararlanırlar.

1990 yılının 5. sayıydı sanırım, IBM, *Think* dergisinde kendi patent portföyü ile ilgili bir makale yayımlamıştı. Makale, IBM’in Birleşik Devletler’de edindiği 9000 adet patentten iki tür kazanç sağladığını söylüyordu. (Eminim bu rakam günümüzde çok daha yüksektir.) Bunlardan ilki, patent ücretlerini toplamaktı, ikincisi ise, “diğerle-

rinin patentlerine ulaşmak”. Bu ikincisinin daha büyük düzeyde bir kazanç olduğunu söylüyorlardı. Yani, başkaları tarafından patentlenmiş fikirleri kullanmak için ücret ödemek zorunda olmamanın IBM’e sağladığı kazanç, şirketin elindeki patentleri lisanslamakla elde edebileceği doğrudan gelirin on katı değerindeydi.

Bu tam olarak ne anlama geliyor peki? IBM’in, “diğerlerinin patentlerine ulaşma” ile elde ettiği kazanç nedir? Aslında tüm kazanç, patent sisteminin başınıza açabileceği tüm sorunlardan uzak olma kazancıdır. Patent sistemi bir piyangoya benzer: patent, hiçbir işe yaramayabilir, patentin sahibi beklenmedik bir şekilde çok büyük kazançlar sağlayabilir, ya da sahibi dışındaki herkes için tam bir felaket halini alabilir. IBM’in bu kadar büyük bir şirket olması, tüm bu olasılıklardan ortalama bir düzeyde etkilenmelerini sağlamaktadır. Patent sisteminin ortalama kar ve zararını hesaplamak durumundalar. Aksi taktirde, patent sisteminin ortaya koyduğu sorunlar, bu sistemin faydalarından on kat daha fazla olurdu onlar için.

“Olurdu” diyorum, çünkü IBM, patentlerin karşılıklı lisanslanması yöntemi sayesinde, bu tür sorunlardan uzak kalmayı başarabilmektedir. Onlar için bu tür sorunlar sadece bir olasılık; gerçekte, asla başlarına böyle şeyler gelmez. Fakat yine de onlar bu sorunlardan uzak kalıyor olmanın getirdiği kazancı hesaplamaktan geri kalmazlar. Bu kazanç, ellerindeki patentlerden elde ettikleri gelirin neredeyse on katı değerinde.

Bu, “patentlerin karşılıklı olarak lisanslanması” olgusu, yaygın olarak bilinen bir miti de çürüt-

mektedir, “aç/yoksul deha”nın miti, patentlerin “küçük muictler”i “korudugunu” söyleyen mit. mit. (Bunlar tamamıyla propaganda sözcükleridir. Kesinlikle kullanmamalısınız.)

Hikaye aynen şöyledir: Farzedelim ki, belli bir “deha” diyebileceğimiz düzeyde bir tasarımcı var. Bu kişinin, yıllarını “bir tavan arasında yokluk içinde”, yepyeni ve gerçekten çok iyi bir şey tasarlayarak harcadığını ve şimdi bu yeni tasarımını imalata aktarmak istediğini varsayalım. Bu aşamada, büyük şirketlerin birden bire ortaya çıkıp bu insanla rekabete girişmeleri ve onu işinden etmeleri, sonuç olarak da, bu “deha”nın tekrar “yokluk” içine dönmesi utanç verici bir şey değil mi?

Bir şeye dikkatinizi çekmekte fayda görüyorum: yüksek teknoloji alanlarıyla ilgili insanlar, genellikle kendi başlarına çalışmazlar, kullandıkları fikirler de boşluktan doğmaz diğer insanların fikirlerine dayanırlar çoğunlukla. Ve bu insanlar, ihtiyaçları olması halinde, iş bulmakta hiç de zorluk çekmezler, özellikle bugünlerde. Bu nedenle, yokluk içinde tek başına çalışan şu “deha”nın hikayesi hiç de gerçekçi değil.

Ancak şöyle bir senaryo pekala mümkün olabilir; diyelim ki, adamın birinin çok parlak bir fikri var ve bu fikir, diğer yüz veya iki yüz fikirle birlikte, önemli bir ürünün temel taşlarını oluşturuyor ve büyük şirketler bu adamla rekabete girmek istiyorlar. Şimdi bu adamın, büyük şirketleri durdurmak için patent sistemini kullanmayı denemesi durumunda neler olacağına bir bakalım. Adam muhtemelen şöyle bir şeyler söyleyecektir: “Hayır IBM. Benimle rekabet edemezsin. Elimde patentim var.” IBM ise şöyle cevap verecektir: “Evet,

ürününe bir göz atıp görelim bakalım. Benim elimde şu patent var, sonra şu var; sonra şu, sonra şu, sonra şu ve sonra da şu. Ve senin ürününün bazı kısımları bu patentleri ihlal ediyor. Bu patentlere karşı mahkemede mücadele edebileceğini düşünüyorsan eğer, geri dönüp ürünün üzerinde biraz daha çalışarak, bir kaç patent ihlali daha bulacağıma şüphen olmasın. Peki neden patentlerimizi karşılıklı olarak lisanslamıyoruz.” Ve küçük dahinin cevabı şöyle olacaktır: “Pekala, tamam. Anlaştık. Öyle yapalım.” Böylelikle, bu adam işine geri dönebilir ve şu harika ürünü geliştirmeye devam edebilir. Ama artık IBM’in de buna yetkisi vardır. Gördüğünüz gibi, IBM bu adamın patentine “ulaştı” ve onunla rekabet etme hakkına kavuştu. Bunun anlamı, adamın patentinin onu hiçbir şekilde “koruyamadığı”dır. Gerçekte de patent sistemi böyle bir şey yapmaz zaten.


Dev şirketler çoğunlukla, bir şekilde, patent sisteminin zararlarından kaçınırlar; bu sistemin sadece faydasını görürler. Yazılım patentlerinden yana olmalarının ana nedeni de budur zaten; bu sistemden faydalanacak olanlar onlardır çünkü. Fakat küçük şirketler ya da tek başına çalışan insanlar bunu yapamazlar, patent sisteminden faydalanamazlar. Denerler, fakat başaramazlar. Asıl sorun, diğerleriyle karşılıklı olarak lisanslayacak kadar çok patentlerinin olmamasıdır.

Her patent belli bir yönü işaret etmektedir. Eğer küçük bir şirketin elinde, şu yönü, şu yönü ve şu yönü gösteren patentler varsa, (Stallman, başka bir tarafa işaret ederek) biri şu taraftan çıkacaktır ve onlara doğru bir patent doğrultarak şöyle diyecektir: “bana

paranızı verin!”. Bu durumda küçük şirket tamamen çaresizdir. Bu tür bir şeyi IBM yapabilir örneğin. Elleriindeki 9000 patent her yeri işaret etmektedir çünkü; ve nerede olursanız olun, büyük bir olasılıkla, orada size doğru işaret etmiş bir IBM patenti vardır. Yani, neredeyse her durumda IBM sizinle karşılıklı lisanslama yapabilir. Küçük şirketler bunu çok nadiren yapabilirler. Bu şirketler, savunma amaçlı olarak patentler edindiklerini söyleyecekler, fakat asla kendilerini savunabilmek için yeterli sayıda patent elde edemeyeceklerdir.

IBM’in bile birilerini kendisiyle karşılıklı lisanslama yapmaya ikna edemeyeceği durumlar vardır. Tek işi çeşitli patentler elde etmek ve bunlar aracılığıyla insanlardan para koparmak olan bir şirket karşılıklı lisanslama işine sıcak bakmayacaktır mesela. “Doğal sıralamanın yeniden hesaplanmasına” patentini elinde bulunduran şirket tam da bu tip bir kuruluştur. Yegane işleri, insanları mahkemeye vermekle tehdit edip paralarını almaktı. Üstelik paralarını aldıkları bu insanlar gerçek anlamda bir şeyler geliştiren kişilerdir.

Yasal işlemlere dair herhangi bir patent bulunmamaktadır. Avukatların, patent sistemiyle kendileri uğraşmak zorunda kalsalardı, ne tür sıkıntılar çekeceklerini tahmin edebildiklerini düşünüyorum.

Sonuç olarak, bu tür şirketlerin sizinle karşılıklı lisans yapmasını sağlamanın hiçbir yolu yoktur. Bu nedenle, bu şirketler etrafta dolaşmaya ve insanlardan para almaya devam ederler. Fakat öyle sanıyorum ki, IBM gibi şirketler bunu, iş yapıyor olmanın getirdiği bedelin bir parçası olarak görüyorlar ve işlerini bu şekilde de sürdürüyorlar. 

YAZILIM PATENTLERİ TEHLİKESİ¹

2. Bölüm*

Richard M. Stallman

Patenti Mahkeme Aracılığıyla Geçersiz Kılmak

Bir şeyin patentlenebilmesi için, sözümona o şeyin yeni ve faydalı olması, bunun yanında açık olmaması gerekiyor. (Yani Birleşik Devletler'de böyle söylüyorlar; sanırım diğer ülkelerde de bunlara benzer sözcükler kullanıyorlardır.) Tabii patent ofisindekiler bu “yeni” ve “açık olmayan” laflarını farklı şekilde yorumlarlar. Onlara göre “yeni” şu anlama gelmektedir: “dosyalarımız arasında bulunmayan”; “açık olmayan”ın anlamıysa şudur: “I.Q. seviyesi 50 civarlarında olan biri tarafından pek kolay anlaşılamayan”.

Birleşik Devletler'de verilmiş yazılım patentleri üzerine çalışmalar yapan biri -ya da en azından, bir zamanlar yapardı; hala bunları takip edebiliyor mu bilmiyorum- bu patentlerin %90 kadarının “Crystal City testi”ni geçememiş olması gerektiğini söylemişti. Yani, patent ofisindekiler, her gazete büfesinde bulunabilen şu bilgisayar dergilerinden herhangi birine şöyle bir gözatsalar, bu fikirlerin çoktan bilinen şeyler olduklarını kolaylıkla görebilirlerdi.

Patent ofisi o kadar aptalca işler yapar ki, bunların gerçekten aptalca olduğunu anlamanız için işi öğrenmek zorunda olmanıza bile gerek kalmazdı. Üstelik bu sadece yazılım patentleriyle sınırlı bir şey de değil. Bir keresinde, ünlü Harvard fare patentini görmüştüm. Bu patent Harvard'a, kansere neden olan bir gen kullanılarak, bir farenin genetik olarak incelenmesi üzerine vermişti. İncelemede kullanılan genin kansere yol açtığı zaten biliniyordu. Dahası, bu iş için seçilen farenin cinsi ve uygulanan teknikler de önceden biliniyordu. Ama aldıkları patent, kansere yol açan herhangi bir genin, herhangi bir yöntem kullanılarak, herhangi bir memelinin vücuduna yerleştirilmesini

*Bu yazının 1. bölümü Elektrik Mühendisliği Dergisi'nin 425. sayısında yayınlanmıştır.

1. 25 Mart 2002'de Stallman'ın Cambridge Üniversitesi'nde verdiği konuşma. Emre Uğur, İnan Kanbur ve İzlem Gözükeleş tarafından Türkçe'ye çevrilmiştir.

kapsiyordu. Bu durumun ne kadar gülünç olduğunu görebilmek için genetik mühendisi olmanıza gerek yok tabi. Bu “aşırı hak isteminin”-nin gayet normal bir şey olduğunu duymuştum. Hatta bazen Birleşik Devletler patent ofisinin, patent için başvurulara, taleplerini daha geniş tutmalarını tavsiye ettiği bile oluyormuş. Esasında, taleplerinizi, önceden yapılmış ve çok bilinen bir işle çakışmadığı sürece istediğiniz kadar genişletebilirsiniz. Akıl alanında ne kadar arazinin yanınıza kar kalabileceğini görebiliyor musunuz?

Programcılar bir çok yazılım patentine baktıklarında şöyle derler: “bu gülünç derecede açık bir şey!” Oysa patent bürokratları/kırtasiyecileri, programcıların ne düşündüklerini umursamamakla ne kadar haklı olduklarını kanıtlayacak her türlü bahaneye sahiptirler. Cevapları hazırdır: “Bütün bunları 10-20 yıl öncesinin şartlarına göre düşünmeniz gerekiyor.” Sonra sonra farkettiler ki, bir şeyi ölümüne konuşurlarsa, er geç kafanız karışabilir. Birçok parçaya bölünüp yeterince incelenirse, her şey karmaşık görünebilir. Bir süre sonra, ‘açık’ ve ‘karmaşık’ kavramları birbirine geçer; neyin açık, neyin karmaşık olduğuna nasıl karar verildiğini karıştırırsınız. Ve bu aşamada patent bürokratları size, patent sahiplerinin hepsinin ne kadar harika insanlar olduklarını anlatırlar; onların, yaptığımız işleri yönlendirme yetkilerini sorgulayamayız.

Yargıçların bu ‘açıklık-karmaşık-lık’ konusunda biraz daha ciddi olacakları muhtemeldir. Fakat buradaki sorun, mahkemeye başvurmanın maliyetinin milyon dolarları bulmasıdır.

Davalının Qualcomm olduğu bir patent davası biliyorum. Qualcomm kaybetmişti. Hatırladığım kadarıyla karar, 13 milyon Amerikan Doları ödenmesi olmuştu. Bu

paranın büyük bir kısmı iki tarafın avukatlarına gitmişti, davacıya ancak bir kaç milyon dolar kalmıştı.

Geniş bir açıdan bakacak olursak, bir patentin geçerliliği tarihi rastlantılara bağlıdır. Tam olarak neyin ne tarihte yayımlandığı, geçmişte yayımlanmış şeylerin ne kadarının bulunabildiği, ne kadarının kayıp olduğu, ve bunun gibi bir çok durum bir patentin geçerli olup olmadığı sorusunun yanıtının verilmesine yardımcı olur.

Doğrusu, *British Telecom*’un “telefonla erişimle hiperlinkleri takip etme” patentinin 1975 yılında uygulamaya konulmuş olması oldukça garip bir durum. Çünkü *Info* paketini ilk defa olarak 1974 yılında ben geliştirmiştim. *Info* paketi hiperlinkler boyunca hareket etmenizi sağlar; ve o günlerde insanlar sisteme ulaşmak için telefonları kullanıyorlardı. Görülüyor ki, bu patent alınmadan önce, ben kendim benzer bir iş yapmışım. Bu benim hayatım boyunca ürettiğim ikinci patentlenebilir fikir.

Bununla birlikte, bu durumu ispatlayacak herhangi bir kanıtım olduğunu sanmıyorum. Bunun yayımlamaya geçecek kadar ilginç bir şey olduğunu bile düşünmemiştim. Üstelik, hiperlinkleri takip etme fikrini, Englebart’ın editörünün demosundan almıştım. Yayımlanmaya değer bir fikri olan biri varsa, o da Englebart’tır. Yaptığım işe “zavallı adamın hypertexti” adını vermiştim; çünkü işi, TECO bağlamında yapmak durumunda kalmıştım. Yaptığım şey, Englebart’ın hypertexti kadar başarılı değildi, ama en azından doküman tarama işi için iyi bir araçtı; zaten işin bütün amacı da buydu. Telefonla erişim konusuna gelince: evet ortada gerçekten telefonla erişim hatları vardı ve sisteme bunlar sayesinde ulaşılıyordu. Ama bunun, işin diğer kısmıyla (zavallı adamın hypertexti) hiçbir

alakası yoktu. Sırf ortada telefonla erişim hatları var diye de makale yayımlamayacaktım herhalde.

Bu işi tam olarak hangi tarihte yaptığımı söyleyebilmem pek olanaklı değil sanırım. Herhangi bir şekilde, bu işin yayımlanmış olma ihtimali var mı peki? Aslında, ARPANET’e gelmeleri ve makinelerimize girmeleri için bir kaç kişiyi davet etmistik -böylelikle *Info*’yu kullanarak doküman taraması yapabilecekler ve olan biten hakkında fikir sahibi olabileceklerdi. Eğer bir şekilde bize sormuş olsalardı, telefonla erişimimiz olduğunu anlamış olacaktı. Gördüğünüz gibi, tarihsel rastlantılar bir işin önce kimin tarafından yapıldığını belirlemede oldukça etkili olabiliyor.

Hypertext konusunda Englebart tarafından yapılmış bir yayım şu an mevcuttur. Ve muhtemelen, davalar bu yayımı mahkemede kullanacaklardır. Bununla birlikte, bu yayımda, sisteme bağlı telefonla erişimli hatlardan bahsedildiğini sanmıyorum; o yüzden mahkemede yeterli olup olmayacağı kesin değil.

Patenti mahkeme aracılığıyla geçersiz kılma olasılığı bir seçenek. Fakat masrafının çok olmasından dolayı, çoğu zaman bu seçenek kullanılmamaktadır, elinizde patentin geçersiz olduğunu gösteren kesin kanıtlarınız olsa bile. Sonuç olarak, geçersiz bir patent, gerçekte asla varolmaması gereken bir patent (ki aslında patentlerin bir çoğu böyledir) tehlikeli bir silahtır. Geçersiz bir patentle üzerinize gelen birileri başınıza gerçekten büyük sorunlar açabilir. Benzer bir işin daha önce yapıldığını gösteren bir kanıtla bunları uzaklaştırmaya çalışabilirsiniz. Korkup kaçacakları garanti değildir ama. Şu şekilde de düşünebilirler: “Pekala, sadece blöf yapıyorsun, mahkemeye gidemeyeceğini biliyoruz; buna gücün yetmez. O yüzden, her şekilde, seni dava edeceğiz.”

Şimdiye kadar bahsettiğim yöntemlerin üçü de zaman zaman kullanabileceğiniz seçeneklerdir, fakat çoğu zaman bunları kullanamazsınız. O yüzden sürekli olarak patentlerle karşı karşıya kalırsınız. Ne zaman bu üç seçenekten birini kullanılabilir bulsanız, hemen arkasından başka bir patent çıkar karşınıza, onun arkasından da başka bir tane... İş, bir mayın tarlasından geçmeye benzemeye başlar. Attığınız her adım, aldığınız her dizayn kararı, bir patente denk gelmeyecektir muhtemelen, böylelikle bir kaç adım atabilirsiniz ve büyük bir ihtimalle bir patlama olmayacaktır. Fakat, program büyüdükçe, hiçbir patente denk gelmeden, mayın tarlasını boylu boyunca geçme ve istediğiniz programı geliştirme şansınız gitgide azalır.

İnsanlar bana diğer alanlarda da patent sisteminin olduğunu söylerler, ve yazılım alanının bu sistem dışında tutulması gerekliliğinin nedenini sorgularlardı. Buradaki tuhaf varsayıma dikkat edin lütfen: hepimizin patent sisteminin sıkıntısını çekmesi gerekiyormuş bir şekilde. Bu tam olarak, şöyle bir şey söylemeye benziyor: "Bazı insanlar kansere yakalanıyor. Neden sen bir istisna olmalıymışsın ki?" Gördüğünüz gibi, kansere yakalanmamış her insan iyi bir şey demektir.

Fakat, bunun ardında, daha az önyargılı bir soru var, iyi bir soru: Yazılım, diğer alanlardan farklı mı? Farklı alanlarda, farklı patent politikaları mı uygulanmalı? Eğer öyleyse, neden?

Cevaplamama izin veriniz: patentler farklı alanlarla farklı şekillerde ilgilidir, çünkü patentler farklı alanlardaki ürünlerle farklı şekilde ilgilidir.

İlaç sanayiini ele alalım örneğin: bir kimyasal formül için patent alabilirsiniz, ve patent sadece ve sadece bir ürünü kapsar. Yeni bir ilaç

varolan bir patentle çakışmaz. Bu yeni ürün için birilerine bir patent verilecekse eğer bir şekilde, bu kişi ürünü geliştiren kişi olacaktır.

Bu, patent sisteminin özündeki fikre uyan bir durumdur: yeni bir ürün geliştiriyorsanız şayet, patentini de alacaksınız demektir. Her ürün başına sadece bir patent düşer ve bu patent ürünün dayandığı fikri kapsar; ana fikir budur. Bazı alanlar için, bu durum, gerçeğe oldukça yakındır; fakat diğer alanlarda gerçeklikten çok uzaktır.

Yazılım sanayii ikinci duruma bir örnektir: bir program, bir çok patentle çakışır. Bunun temel nedeni, yazılım paketlerinin genellikle gerçekten çok geniş ve kapsamlı olmasıdır. Bir sürü farklı fikir birarada kullanılır. Eğer program yeniyse ve gerçekten bir yerlerden kopyalanmamışsa, büyük ihtimalle bilinen fikirlerin farklı bir birleşimi kullanılarak üretilmiştir. Ve tabii ki bu fikirler yeniden koda dökülmüştür, çünkü sadece bu fikirlerin isimlerini söylemek ve büyüğü bir şekilde çalışmalarını beklemek bir işe yaramayacaktır. Bütün fikirleri yeniden koda dökmeniz gerekir. Bütün fikirleri belli bir kombinasyona göre yeniden koda dökmeniz gerekir.

Sonuç olarak, yeni bir program yazarken bir çok fikirden yararlanırsınız. Bu fikirlerden herhangi biri başka bir şahıs tarafından patentlenmiş olabilir. Hatta bu kişi, kullandığınız fikirlerden birkaç tanesinin birden patentlerini elinde bulunduruyor da olabilir. Bir fikrin birden fazla, farklı ifade şekilleri olabilir ve bunların herbiri çeşitli insanlarca patentlenmiş olabilir. Gördüğünüz gibi, programınızda başkaları tarafından patentlenmiş binlerce nokta bulunması oldukça yüksek bir olasılık.

Bu durum gösteriyor ki, yazılım patentleri meselesi, yazılım geliştirme işinin önünü tıkama yönünde yol

almaktadır. Eğer gerçekten "ürün başına sadece bir patent" kuralı bu alanda da geçerli olsaydı, o zaman bu yazılım patentleri, yeni ürünlerin geliştirilmesi konusunda bu denli engel teşkil etmezdi. Çünkü geliştirdiğiniz yeni ürün, başkası tarafından önceden patentlenmiş olmazdı. Fakat birçok farklı fikrin birleşiminden oluşan yeni bir ürünün, bir süre sonra, bir kısmının veya tamamının, başkaları tarafından önceden patentlenmiş olduğu ortaya çıkacaktır.

Bu, neden yazılım patentlerinin yazılım süreçlerini engellemeye meyilli olduğunu açıklar. Eğer her ürüne bir patent ilkesi geçerli olsaydı, bu patentler ürünlerin gelişimini engellemeyecekti çünkü yeni bir ürün geliştirildiği zaman başka biri tarafından patentlenmemiş olacaktır. Fakat bir ürün değişik fikirleri içerdiği zaman, sizin yeni ürününüzün (bir kısmının ya da tümünün) başka birisi tarafından patentlenmiş olma ihtimali oldukça yüksektir.

Gerçekten bir patent sisteminin artan yaratıcılığın olduğu bir alana empoze etmenin o alandaki gelişimi yavaşlatacağını gösteren ekonomik araştırmalar bulunmaktadır. Yazılım patentlerinin olması gerektiğini savunanlar şunu söyler: "Tamam, problemler olabilir, fakat tüm bunlardan daha önemlisi, patentler yaratıcılığı ödüllendirir ve bu, neye neden olursa olsun çok önemlidir." Çok aptalca olduğu için tabii ki bunu yüksek sesle söylemiyorlar, fakat örtülü olarak sizin anlamanızı istedikleri şudur: Patent sistemi gelişimi ödüllendirdiği sürece, tüm bedellerin üzerindedir. Fakat, aslında gelişimi ödüllendirdiğine inanmak için bir neden yok. Şu anda elimizde patentlerin gelişimi nasıl geciktirebildiğini gösteren bir model var. Bu modelin uygulandığı durum, yani artımlı yaratıcılık durumu, yazılım alanını oldukça iyi tarif ediyor.

Neden yazılım, spektrumun bu kadar aşırı uçlarında yer alıyor? Çünkü yazılım yaparken, biz idealize edilmiş matematiksel nesnelere geliştiririz. Karmaşık ve büyük bir kale yapabilirsiniz, ve onu ince bir ipin üzerine koyabilirsiniz. Orada duracaktır çünkü fiziksel bir ağırlığı yoktur. Diğer alanlarda çalışma yapılırken, maddenin yada fiziksel nesnelere zorlukları ve inatçılıkları hesaba katılmak zorundadır. Madde yapması gerekeni yapar. Onu modellemeye çalışabilirsiniz, ama eğer gerçek davranış sizin modelinize uymuyorsa şansınıza küsün, çünkü asıl meydan okuma gerçekten çalışan fiziksel nesnelere yaratılmaktadır.

Bir while deyiminin içine if deyimini yazarken, if deyiminin belli bir sıklıkta salınım yapıp yapmayacağı ve while deyimine sürünerek onu parçalamayıp parçalamayacağı

üzerine endişelenmem. Aynen belli bir sıklıkta salınıp yaparken değişkenin değerini tümevarım yoluyla değiştirip değiştirmeyeceğin endişe etmemem gibi. If deyiminin ne kadar akım çekeceği, sonrasında while deyiminin içerisindeki ısıyı yok edip etmeyeceği ya da while deyiminde bir voltaj düşüşünün if deyiminin çalışmasına etkisi üzerine endişelenmem. Bu programı tuzlu su ortamında çalıştırırsam, tuzlu suyun if deyimi ile while deyimi arasında girip aşınmaya yol açıp açmayacağı üzerine de endişelenmem. [Bu olasılıklar sıralanırken izleyicilerden kahkahalar yükselir.]

Herhangi bir değişkeni ısınıp ısınmadığını düşünmeden yirmi kere kullanabiliriz. Değişkenin ne kadar kapasitansını olduğuyla yada değerine ulaşması için ne kadar zaman şarj edileceğiyle de ilgilenmeyiz.

Programı yazarken her kopyayı fiziksel olarak nasıl birleştirebileceğim yada while deyiminin içerisine bir if deyimi yazmak için nasıl erişebileceğim de endişelenmem gereken bir problem değildir. Oraya koyduğum if deyimi bozulursa ya da kırılırsa, onu oradan çıkarıp yerine yeni bir if deyimini nasıl yerleştirebileceğim de beni endişelendirmez. Yazılım yaparken beni kaygılandırmayacak çok fazla problem vardır ki, bu da temel olarak bir programın yazılımını fiziksel bir nesnenin tasarımından daha kolay yapar.

Bu garip görünebilir çünkü yazılım tasarılmanın ne kadar zor bir iş olduğundan, bunun ne kadar büyük bir problem olduğundan bahseden çok fazla insan görmüşsünüzdür. Aslında aynı sorundan bahsetmiyoruz. Ben fiziksel sistemlerle aynı karmaşıklığı ve aynı sayıda



(c) Fidget 2001

parçaya sahip yazılım sistemlerini karşılaştırıyorum. Bence yazılım sistemlerini tasarlamak fiziksel sistemleri tasarlamaktan çok daha kolaydır. Fakat bu farklı alanlarda çalışan insanlar aynı zeka seviyesinde olduklarına göre, daha kolay bir dalda çalışmak zorunda kalanlar ne olacak? Yetenekler sonuna kadar zorlanır. Aynı büyüklükteki iki sistemden biri basit, bu durumda ne yapacağız? Tabi ki basit bir sistemin büyüklüğünü on katına çıkaracağız, böylece zorlaşacaktır. Aslında şu anda yaptığımız da budur: fiziksel sistemlerden çok daha fazla parçaya sahip yazılım sistemleri üretmek.

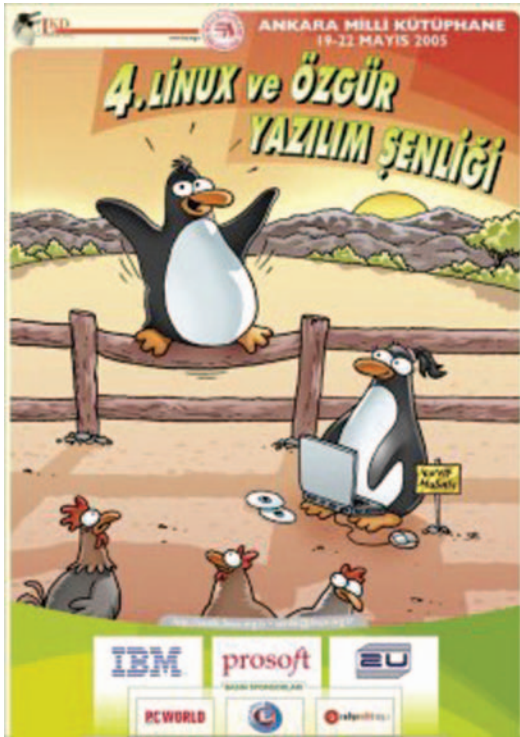
İçerisinde milyonlarca farklı parça barındıran fiziksel bir sistemin tasarımı çok büyük bir projedir. İçerisinde bir milyon parçası olan bir yazılım projesi belki de 300,000 satırdan oluşur ki az sayıda insan iki yılda bu projeyi tamamlayabilir. Aslında bu çok da büyük bir proje değildir. GNU Emacs'ın tasarımı birkaç milyon parçadan oluşuyor. Bu projede yaklaşık bir milyon satır

kod yazılmıştır. Bu proje çoğunlukla insanların boş zamanlarındaki üretimleriyle geliştirilmiş dolayısıyla herhangi bir parasal destek almamıştır.

Daha büyük başka bir tasarruf var. Fiziksel bir ürün tasarlanırken bir sonraki adım ürünün üretileceği fabrikanın tasarımını yapmaktır. Bu fabrikayı inşa etmek milyonlarca dolara mal olurken bir programın kopyalayabilmek için "copy" komutu yeterlidir. Aynı kopyalama komutu tüm programları kopyalamak için kullanılabilir. CD'ye mi kopyalamak istiyorsunuz? Sorun yok, asıl CD'den çok sayıda CD yapabilirsiniz. Tüm CD kopyalama işlemleri için aynı ekipman kullanılır. Ürüne özel fabrikalar yapmanıza gerek yoktur. Bu, işlerin inanılmaz basitleşmesine ve tasarım maliyetinin muazzam azalmasına denk gelir.

Yeni model bir araba üretmek için 50 milyon dolarlık bir fabrika harcaması gereken bir otomobil firması için, avukatlar kiralarak patent lisans görüşmeleri yapmak sorun değildir. Hatta açılan davalarla da başa çıkabilirler. Aynı karmaşıklıkta bir programın tasarımı 50 bin yada 100 bin dolara mal olacaktır. Karşılaştırma yapılacak olursa, patent sistemiyle böyle bir durumda karşılaşmak yıkıcı olacaktır. Bir arabanın mekanik tasarım karmaşıklığına denk bir program yazmak bir ay kadar bir zaman alacaktır. Herhangi bir bilgisayar sistemi gerektirmeyen bir araba kaç parçadan oluşabilir... Bu, iyi bir tane asarlamamanın kolay bir iş olduğundan söylemiyor²; sadece içinde çok da fazla parça içermediğini söylemek istiyoruz.

Bunu açıklamanın belki de en iyi yolu senfonilerle arasında benzerlik kurmaktır. Bir senfoni de uzundur, bir çok nota içerir ve müziğe dair pek çok fikir içerir. 1700'lerin Avrupa'sında hükümetlerin senfonik müzikteki ilerlemeleri ödüllendirmeye karar verdiklerini hayal edin. Kelimelere dökülebilen müzik ile ilgili tüm fikirleri patentleme hakkına sahip Avrupa Müzik Patent Ofisi adında bir kuruluş kurulduğundan.



1800'lere gelin ve Beethoven olduğunuzu düşünün. Bir senfoni yazmak istiyorsunuz. Herhangi bir patenti çiğnemeyen kendi senfonizi yazmak "iyi bir senfoni" yazmaktan zor olacaktır.

Bu durum hakkında yakınacak olursanız, patentleri ellerinde tutan insanlar size şöyle diyeceklerdir: "Ah Beethoven, şikayet ediyorsun çünkü hiç orijinal fikrin yok. Tek yapmak istediğin bizim buluşlarımızı yürütmek." Beethoven'in birçok yeni fikri vardı, fakat algılanabilir bir müzik üretmek için varolan fikirleri kullanmak zorunda kalmıştır. Bir müziği dinleyenlere sevdirebilmek için, onların dinledikleri şeyi müzik olarak algılaması gerekir. Hiç kimse müziği tamamen farklı bir şekilde yeniden icat edip, insanlara bunu sevdirecek kadar dahi değildir. Pierre Boulez, bunu yapmaya çalışacağını söylemiştir, peki şu anda Pierre Boulez'i kim dinliyor?

Kimse tüm bilgisayar bilimini, yepyeni ve bambaşka bir şekilde tekrar icat edecek kadar zeki değil. Eğer bunu yap-

2. Bir arabanın iletim (**transmission**) sisteminde yaklaşık olarak 300-400 birbirine benzemeyen parça bulunur ve iletim sistemi en arabaların en karmaşık bileşenlerinden biridir. Bir **transmission** tasarımı 6 aydan bir yıla kadar bir zaman diliminde tamamlanabilir, yapılması ve çalışır hale gelmesi ise büyük ihtimalle daha fazla zaman isteyecektir. Buna karşılık, 500-800 fonksiyonel parçası olan bir program 200-300 satır koddan oluşur ki bu da iyi bir programcı için yazım, test ve yanlış ayıklamalar dahil bir gün bir hafta arasında tamamlanabilir.

bilen biri çıksa bile, bu kullanıcılara o kadar değişik gelecek ki, kimse bunu kullanmak istemeyecektir. Bir kelime işleyicisini inceleyecek olursanız, içerisinde yüzlerce farklı özellik bulursunuz. Yeni yaratıcı bir kelime işleyicisi geliştirecek olursanız, bu içerisinde yeni fikirlerin olduğu anlamına gelir; fakat aynı zamanda yüzlerce eski fikri de barındırmak zorundadır. Eğer bunları kullanmak için izin verilmezse, yenilikçi bir kelime işleyicisi yaratamazsınız. Çünkü yazılım geliştirme işi o kadar büyüktür ki, sonuçta yeni fikirleri oluşturmak için yapay planlara gerek yoktur. Basitçe yazılım yapan insanlar vardır ve bunların kafalarında bazı yeni fikirler olacaktır. Eğer bir program yazmak istiyorsanız ve programın iyi olmasını istiyorsanız, bazı yeni fikirler bulursunuz ve bunları kullanmak için çalışırsınız.

Yazılım patentleri çıkmadan önce yazılımın içerisinde yer alan biri olarak biliyorum, daha önceleri, birçok yazılımcı, önemli olduğunu düşündükleri yeni fikirlerini, saygı görececeklerini düşündükleri için yayımlarlardı. Eğer fikirler çok ufak çaplı ise yada etkileyici değillerse yayımlamazlardı, çünkü bu ters sonuçlar doğurabilirdi. Şu anda patent sistemi fikirleri açıklamamayı özendirir hale gelmiştir. Doğru, kodlarını saklardı ama eski günlerde kimse fikirlerini saklamazdı. Sonuç olarak kod, yapılan işin tamamına denk gelir. Kodu saklardı ve fikirleri yayınlarlardı, böylece çalışanlar hem kendilerini iyi hisseder hem de tanınırlardı.

Yazılım patentleri çıktıktan sonra kodlarını saklamaya devam ettiler, ve fikirlerini patentlediler. Gerçekten de fikirleri açıklama herhangi bir yönden teşvik edilmiyor. Bugün, eskiden gizlenen şeyler

halen gizli tutuluyor, fakat eskiden yayımlanan ve bizim kullanabildiğimiz fikirler bugün patentleniyor ve 20 yıl boyunca ulaşılmazlıklarını koruyorlar.

Bir ülke bunu değiştirmek için ne yapabilir? Bu problemi çözmek için politikamızı ne şekilde değiştirmeliyiz?

Atağa geçebileceğimiz iki yer var. Birisi patentlerin dağıtıldığı yerler, yani patent ofisleri. Diğer yer ise patentlerin uygulama alanları, yapılacak şey ise patentlerin kapsamlarının sorgulanması.

Bir yol patentleri dağıtırken iyi kriterler koymak. Bu yöntem sadece, Avrupa'nın büyük kısmında olduğu gibi, daha önce yazılım patentleriyle ilgili bir kısıtlama olmayan ülkelerde uygulanabilir. Avrupa Paten Ofisi kuralları yazılım patentlememenin Avrupa için iyi bir çözüm olduğunu söylüyor, bizim yapabileceğimiz bunu pekiştirmeye çalışmak olabilir. Avrupa şu anda yazılım patentleri konusunda kendine bir yön çizmeye çalışıyor (Yönelimin daha geniş bir çerçevesi olduğunu ve bunların yazılım patentleri açısından önemli sonuçları olacağını düşünüyorum.). Yazılım fikirlerinin patentlenemeyeceğine karar vermek, Avrupa'nın büyük bir kısmını bu problemden kurtarır. Fakat ne yazık ki, Birleşik Krallık dahil bazı Avrupa Ülkeleri bu sorunu sadece kendi sorunları olarak görmektedir.

Bu yaklaşım Birleşik Devletler'de geçerli değildir. Çünkü Birleşik Devletler zaten çok fazla sayıda yazılım patenti elinde bulundurmaktadır ve patentlerle ilgili kriterlerdeki herhangi bir değişim halihazırda varolanları bir kenara itemez³. Bu nedenle Birleşik Devletler'de çözüme patentlerin uygulanabilirlik-

lerini ve kapsamalarını değiştirerek ulaşabiliriz. Söyle söylemeliyiz: Donanımsal olarak patent yasalarını çiğnemeyen bir genel amaçlı bilgisayarda çalışan saf bir yazılım gerçekleştirim'i patent kapsamına alınamaz ve dava konusu olamaz. Bu da başka bir çözüm.

İlk çözüm, yani geçerli olan patent tiplerini belirlemenin baskın olduğu çözüm, Avrupa için iyi bir çözüm olabilir.

Birleşik Devletler'de ilk yazılım patentleri verilmeye başlandığı zaman, politik bir tartışma olmadı. Gerçekte, kimse bunu farketmedi. Yazılım alanındaki insanların çoğunluğu farketmedi bile. 1981'de Anayasa Mahkemesi'nin "curing rubber" işleminin patentlenebileceğine dair bir kararı oldu. Kural şuna dayanıyordu: "curing rubber" işlemi sırasında kullanılan aygıtların bir bilgisayar ve program içermesi, bu sürecin patentlenemez olduğunu göstermez. Bir sonraki yıl, patentlerle ilgili tüm patent başvurularında argümanlar değişti: Eğer bir süreç bilgisayar ve program içeriyorsa, o patentlenebilir olmalıydı. Sonuç olarak bilgisayar ve program içeren herşey patentlenebiliyor. Bu Birleşik Devletler'de neden iş yöntem patentlerinin ortaya çıktığını açıklıyor: çünkü iş yöntemleri bilgisayar üzerinde yürüyor ve bu onları patentlenebilir kılıyor.

Bu kural böylece gerçekleşti, ve sanırım doğal sıralamanın yeniden hesaplanması patenti de bunlardan ilki oldu.

80'li yıllar boyunca bunların farkında değildik. İlk defa 90'ların başında Birleşik Devletler'deki programcılar yazılım patentleri tehlikesi ile karşı karşıya olduklarının farkına varmaya başladılar. Ben bu alanda tüm

3. "Yazılım patentleri"nden bahsederken gerçekten neyi kastediyorum? Birleşik Devletler Patent Ofisi patentlerde yazılım patentleri ve diğer patentler gibi bir ayırım yapmamaktadır. Böylece, gerçekten de bir patent, eğer yazılım alanında uygulanabilirse, size dava açılmasını sağlayabilir. O halde yazılım patentleri, yazılıma uygulanma potansiyeli olan ve potansiyel olarak bir program yazdığınızda size dava açılmasına yolaçan patentlerdir.

bunlar olmadan önceki çalışmaları da gördüm, bunlar olduktan sonrakileri de. 1990'dan sonra yazılım alanındaki ilerlemelerde herhangi bir hızlanmaya şahit olmadım.

Birleşik Devletler'de herhangi bir politik tartışma olmadı, fakat Avrupa'da büyük politik tartışmalar oldu. Birkaç yıl önce Avrupa Patent Ofisini kuran Munich Antlaşması üzerinde değişiklik yapmak istediler; antlaşma yazılımın patentlenemeyeceğine dair bir madde içerdiği için antlaşmayı yazılım patentlerini kabul edecek şekilde değiştirmek istediler. Özgür yazılım geliştiricilerin ve özgür yazılım kullanıcılarının başını çektiği topluluk durumun farkına vardı. Yazılım patentleri sadece bizi tehdit etmiyordu, tüm yazılım geliştiricilerini ve hatta kullanıcıları tehdit ediyordu.

Örnek olarak Paul Heckel'i verebiliriz; tehditleri Apple firmasını korkutmadı. Fakat Apple'ın müşterilerinin dava etmekle tehdit etmeye başladığında Apple durumu korkutucu buldu. Sonuç olarak müşterileri kazanacak olsa bile, tüm bu davalarla başa çıkamayacaklarının farkına vardı. Yazılım kullanıcıları da, geliştiricilere saldırmak için yada para sızdırmak için dava edilebilirdi. Tüm geliştiriciler ve kullanıcılar kolayca zarar görebilirdi.

Yasayı değiştirmek için yapılan baskıya karşı örgütlenmede başı çekenler Avrupa'daki özgür yazılım topluluğu oldu. Gerçekten de Avrupa ülkeleri patent düzenlemelerini ikinci kere reddetti. Daha sonra duruma Avrupa Topluluğu el koydu, topluluğun kuruluşları bu konuda ikiye bölündü. Bu kurumlardan biri, işi yazılımları ödüllendirmek olanı yazılım patentlerine karşı çıkıyordu; fakat bu konudayetkili değildiler. Yetkili Open Market Directorate (Açık Pazar Kurumu) idi ve kurumun

başkanı yazılımın patentlenmesini savunuyordu. Halkın fikrine aldırmadılar ve yazılım patentlerine izin veren bir talimat önerdiler.

Fransız hükümeti yazılımın patentlenmesine karşı olduğunu açıklamıştı bile. İnsanlar Avrupa'daki diğer hükümetlerinin yazılımın patentlenmesine karşı çıkması için çalışıyorlardı, özellikle bunu burada, İngiltere'de başarmak çok hayatiydi. Avrupa'daki mücadelenin liderlerinden biri olan Hartmut Pilch'e göre ana darbe Birleşik Krallık patent ofisinden geldi. Birleşik Krallık patent ofisi yazılımların patentlenmesinden yana bir eğilim sergilemişti. Yaptıkları bir kamuyu yoklamasında, insanların çoğu yazılım patentlerine karşı çıkmıştı. Buna rağmen insanların onlarla aynı fikirde göründüklerini söyleyen bir rapor hazırladılar. Özgür yazılım topluluğu sonuçların halka ve kendilerine gönderilmesini istedi. Böylece sonuçları yayınlamak zorunda kaldılar fakat genel olarak insanlar sonuçlara karşı çıktı. Rapora baktığınızda, bunun Birleşik Krallık patent ofisi tarafından yayımlandığı hemen anlaşılıyordu.

Çok fazla saptırılabilir bir şeye, "technical effect" (teknik etki) terimine başvurdular. Şöyle düşünmeliydik: Bir program ancak spesifik bir fiziksel harekete (physical acts) denk gelirse patentlenebilirdi. Eğer bu şekilde yorumlanırsa, problem çözülmüş oluyordu. Fakat problem şu ki bu terimi istediğiniz gibi saptırabilirsiniz. Bir program çalıştırarak elde ettiğiniz bir sonucu fiziksel bir sonuçmuş gibi açıklayabilirsiniz. Fiziksel bir sonuç diğerinden nasıl farklı olabilir ki? Tamam, o zaman bu sonuç bir hesaplamanın (computation) sonucu. Birleşik Krallık patent ofisinin önerdiği şey problemi tamamen çözüyormuş gibi görünse de, herşeyin patentlenebilmesinin yolunu açıyordu.

Aynı bakanlıktaki insanlar telif hakları konusuyla ilgilenmişlerdi, fakat bu konunun yazılım patentleriyle aynı insanların ilgilenmesi dışında ortak bir noktası yok. (Belki de "fikri mülkiyet" terimini kullanarak tüm bu konuları aynı yığında topladılar) Avrupa Topluluğu'nun yeni telif hakları yasası, Birleşik Devletler'deki Digital Millennium Copyright Act (DMCA) kanunu kadar korkunç olsa da, yasayı nasıl uygulayacağına karar verme konusunda ülkelere esneklik tanınmıştır. Birleşik Krallık direktifin uygulanması için en sert yollar öneriyor. Yasayı olabildiğince düzgün bir şekilde uygulayarak zararını en aza indirebilirsiniz. Birleşik Krallık bu yasanın kötü etkilerini mümkün olduğunca arttırmaya çalışıyor. Dizginlenmesi gereken belirli bir grup varmış gibi görünüyor - Department of Trade and Industry (Ticaret ve Sanayi Departmanı)-. Bunların yaptıkları işler gözlenmeli ve güçlenmeleri engellenmelidir.

Yazılım patentleri, geliştiricilerin ve tüm bilgisayar kullanıcılarının bir tür bürokrasi içinde ellerini bağlamaktadır. Bilgisayar kullanan iş çevreleri yazılımın patentlenmesinin başlarına nasıl bir iş açacağını farkedelerse, harekete geçip bunu durdurabilirler. İş çevreleri bürokrasi içerisinde sıkışıp kalmaktan hoşlanmazlar. Bazen tabi ki, bürokrasi önemli bir amaca hizmet edebilir. Birleşik Krallık hükümetinin belli bazı işleri bürokrasiyle kısıtlamasını isteyebiliriz, tıpkı hayvan taşımacılığında olduğu gibi⁴. Fakat yapay tekeller yaratmak dışında başka bir amaca hizmet etmiyorsa, yazılımların patentlenmesine karşı çıkmalıyız. Yöneticilerin yazılım patentlerinin nelere yol açabileceğinin farkında olmalarını sağlayıp, bu mücadelede onların desteğini almamız.

Savaş henüz bitmedi, hala kazanabiliriz.

4. Bulaşıcı hastalıkların yayılmasını engellemek için.