

Noosferi İskana Açmak

Yazan:
Eric Steven Raymond

Thyrsus Enterprises^(B1)
<esr (at) thyrsus.com>

Çeviren:
Deniz Akkuş

2002

Özet

Hackerların gerçek hayatta davranışlarıyla, açık kaynak kod lisansları tarafından tanımlanan 'resmi' ideoloji arasında çelişkiler gözlemledikten sonra açık kaynak kodlu yazılımların sahiplik ve kontrolünü belirleyen gerçek gelenekleri inceliyorum. Bu geleneklerin Locke'nin toprak iskanı teorisiyle benzeşen bir mülkiyet hakları teorisine dayandıklarını gösteriyorum. Bu çıkarımı, hacker kültürünü, üyelerin zaman, enerji ve yaratıcılıklarını vererek prestij elde ettiği bir "hediye kültürü"ne benzeten analize bağlıyorum. Son olarak bu analizin sorun çözümü açısından ne manaya geldiğini inceliyor ve bazı çıkarımlarda bulunuyorum.

Konu Başlıkları

1. Çeviriler	3
2. Bir Başlangıç Çelişkisi	3
3. Hacker İdeolojisinin Çeşitleri	3
4. Çoğulcu Teori, Püriten Uygulama	5
5. Mülkiyet ve Açık Kaynak Kod	6
6. Locke ve Toprak Mülkiyeti	7
7. Bir Hediye Kültürü Olarak Hacker Ortamı	9
8. Hackerlığın Zevki	10
9. Şöhretin Değişik Yüzleri	11
10. Mülkiyet Hakları ve Şöhret Teşvikleri	11
11. Ego Problemi	12
12. Alçakgönüllülüğün Değeri	13
13. Şöhret Oyunu Modelinin Global Sonuçları	14
14. Ne Kadar İyi Bir Hediye?	15
15. Noosferin Özellikleri ve Mekan Sahiplenmesi	17
16. Çatışma Sebepleri	18
17. Proje Yapıları ve Mülkiyet	19
18. Çatışma ve Çatışma Çözümlemesi	20
19. Kültüre Giriş Mekanizmaları ve Akademi ile Bağlar	20
20. Hediye'nin Takastan Üstün Gelmesi	22
21. Sonuç: Gelenekten Yasaya	23
22. Araştırma Soruları	23
23. Notlar	24
24. Teşekkür	26
A. Kaynakça	27

Sürüm Bilgileri

Sürüm 3.0 Türkçe Çevirisi

Geçmiş

3.0.tr Türkçe çevirisi.	2002-07-05	dak
3.0 Orjinali.	2000-08-24	esr

Telif Hakkı © 2002 Türkçe Çeviri: Deniz Akkuş <deniz (at) arayan.com>

Telif Hakkı © 2000 Eric S. Raymond <esr (at) thyrsus.com>

Yasal Uyarı

Bu dökümanı kopyalamak, dağıtmak ve değiştirmek izni, Open Publication License, 2.0 sürümü altında verilmiştir.

1. Çeviriler

- Orjinal metin^(B2) Docbook biçemi^(B3) Yazar: Eric S. Raymond <esr (at) thyrus.com>
- Türkçe (bu metin)^(B4) Docbook biçemi^(B5) Çeviren: Deniz Akkuş <deniz (at) arayan.com>
- Çince^(B6)
- Danimarkaca^(B7)
- Fransızca^(B8)
- Almanca^(B9)
- İtalyanca^(B10)
- Japonca^(B11)
- İspanyolca^(B12)

2. Bir Başlangıç Çelişkisi

İnternet açık kaynak dünyasının üretken ve meşgul hayat tarzını bir müddet izleyen herkes açık kaynak kod hackerlarının inandıklarını söyledikleri şeyler ve gerçekte olan uygulamalar arasında ilginç bir çelişki gözlemleyecektir. Açık kaynak kod kültürünün resmi ideolojisi ve gerçekte yaşananlar farklıdır.

Kültürler uyum gösterebilen makinalardır. Açık kaynak kod kültürü tanımlanabilecek bir güdü ve baskı kümesine verilen tepkiden kaynaklanmaktadır. Kültürün çevresine uyumu hem bilinç üstü bir ideoloji hem de zımnen kabul edilen, bilinç altı bilgi olarak kendini göstermektedir. Ve sık rastlandığı şekilde, bilinçaltı adaptasyonları, kısmen de olsa bilinç üstü ideolojisi ile çelişkilidir.

Bu makalede çelişkilerin köklerini inceleyecek ve bunları oluşturan güdü ve baskıları ele alacağım. Hacker kültürü ve gelenekleri konusunda bazı ilginç çıkarımlarda bulunacağım. Sonunda ise, kültürün zımni bilgilerinin daha verimli kullanılabilmesi için yollar önereceğim.

3. Hacker İdeolojisinin Çeşitleri

İnternet açık kaynak kod kültürünün ideolojisi (hackerların inandıklarını söyledikleri şeyler) kendi başına oldukça derin bir mevzuudur. Bütün üyeler açık kaynak kodun (yani serbestçe dağıtılabilen ve değişen ihtiyaçlara cevap vermesi için rahatça değiştirilebilen yazılımlar) iyi bir şey olduğu ve önemli kolektif bir efora layık olduğunda hem fikirdirler. Bu kabul, efektif olarak kültüre üyeliği tanımlar. Fakat bireylerin ve çeşitli alt kültürlerin bu kabule varmakta kullandıkları nedenler oldukça farklıdır.

Görüşlerde etkin bir parametre fanatizm derecesidir; açık kaynak kod geliştirmenin sonuca varan uygun bir yoldan ibaret mi (iyi araçlar, eğlenceli oyuncaklar ve oynaması zevkli bir oyun) veya kendi başına bir hedef mi görüldüğü önemlidir.

Çok inançlı bir kişi “Serbest yazılım hayatımın amacıdır! Kullanışlı, güzel yazılımlar ve bilgi kaynakları oluşturmak ve bunları dağıtmak için varım.” diyebilir. Orta derecede inanç taşıyan birisi ise “Açık kaynak kod iyi bir şey ve gerçekleşmesi için zamanımı vermeye hazırım.” diyebilir. Düşük inançlı birisi ise “Açık kaynak kodun faydalı olduğu yerler var, bununla oynuyorum ve bunu oluşturan kişilere saygı duyuyorum.” diyebilir.

Diğer bir parametre, ticari yazılımlara ve/veya ticari yazılım pazarını elinde tuttuğu düşünülen şirketlere duyulan düşmanlık derecesidir.

Ticaret karşıtı birisi “Ticari yazılım hırsızlık ve istifçiliktir. Bu kötülüğü sona erdirmek için serbest yazılım geliştiriyorum.” diyebilir. Orta derecede ticaret karşıtı birisi ise “Ticari yazılım genelde kabul edilebilir çünkü programcıların ücretlerinin ödenmesi gerekli fakat kötü ürünler çıkaran ve güçlerini kötü yönde kullanan şirketler kötüdür.” diyecektir. Ticaret karşıtı olmayan birisi ise “Ticari yazılım kabul edilebilir, ben yalnızca daha fazla sevdiğim için açık kaynak kodlu yazılım kullanıyorum/yazıyorum.” diyecektir. (bu makalenin yazılışından beri endüstrinin açık kaynak kod tarafının gelişmesi ile bu günlerde “Kaynak kodu bana verildiği veya istediğim şeyi yaptığı müddetçe ticari yazılım kabul edilebilir” söylemi de duyulabilir.)

Bu kategorilerin kombinasyonlarından oluşan altı ana eğilimin hepsi de açık kaynak kod kültüründe mevcuttur. Bunlar arasındaki farkları belirtmek, her eğilim değişik bir gündemi ve değişik adaptasyon ve uzlaşma davranışlarını getirdiği için önemlidir.

Tarihsel olarak hacker kültürünün en görünen ve en iyi organize kesimi hem çok inançlı, hem de çok ticaret karşıtı olmuştur. Richard M. Stallman (RMS) tarafından kurulan Free Software Foundation, 1980’lerden beri, Internet açık kaynak dünyası için temel teşkil eden ve görülebilir gelecekte de temel olmaya devam edecek olan Emacs, GCC gibi araçlar da dahil olmak üzere, büyük miktarlarda açık kaynak kod gelişimine destek vermiştir.

Uzun yıllar boyunca açık kaynak kodlu hacker faaliyetlerinin tek ve en önemli odağı, halen kültür için büyük önem taşıyan çok sayıda aracı geliştiren FSF idi. FSF uzun zaman boyunca, hacker kültürünün dışından görülebilecek tek kurumsal kimlik sahibi açık kaynak kod destekçisi oldu. ‘Serbest yazılım’ kavramını tanımlayan ve ona bilerek çatışmacı bir ağırlık veren (tıpkı şimdi daha yeni ‘[açık kaynak kodlu](#)^(B13)’ teriminin bilinçli olarak uzlaşmacı seçilmiş olması gibi) de onlar oldu.

Dolayısıyla hacker kültürünün hem içinden, hem dışından izlenimi, kültürün FSF’in inançlı ve ticaret karşıtı zannedilen amaçlarıyla özdeşleştiği yolunda idi. RMS’in kendisi ticaret karşıtı olmadığını ifade etmektedir, fakat öne sürdüğü program, kendisinin en kuvvetli taraftarları da dahil olmak üzere pek çok kişi tarafından bu şekilde algılanmaktadır. FSF’in “Yazılım İstifçiliğine Son” çağırısı, bir hacker ideolojisine en yakın olgu, RMS’in kendisi de hacker kültürünün liderliğine en yakın kişi oldu.

FSF’in hazırladığı lisans olan “Genel Kamu Lisansı” (GPL) FSF’in tutumuna ışık tutar. Bu lisans açık kaynak kod dünyasında yaygın olarak kullanılmaktadır. North Carolina’daki [Metalab](#)^(B14) (daha önce Sunsitede) Linux dünyasının en popüler ve geniş yazılım arşividir. Temmuz 1997’de, Sunsitede herhangi bir lisans ibaresi taşıyan yazılımların yaklaşık yarısı GPL kullanıyordu.

Fakat FSF hiç bir zaman tek alternatif değildi. Hacker kültürünün içerisinde hep devam eden daha sessiz, daha az çatışmacı ve daha pazara dost bir grup mevcuttu. Pragmatik düşünen hackerlar bir ideolojiye değil, FSF’den önceki erken açık kaynak kod eforlarının üzerine kurulan bir mühendislik geleneğine bağlı idiler. Bu gelenekler Unix teknik kültürü ve ticarileşme öncesi Internet teknik kültürlerini de kapsıyordu.

Tipik pragmatik düşünüş tarzı ancak orta derece ticaret karşıtıdır ve ticari dünya ile başlıca problemi “istifçilik” değil, ticari dünyanın Unix, açık standartlar ve açık kaynak kodlu yazılım kullanmayı garip bir şekilde reddedişidir. Eğer pragmatik hacker bir şeyden nefret ediyorsa, bu, “istifçiler” değil, yazılım dünyasının o zamanki Kral’ıdır, ilk önce IBM, şimdi ise Microsoft.

Pragmatik hackerlar için GPL kendisi için önemli değil, bir araç olarak önemlidir. Esas değeri, “istifçiliğe” karşı bir silah değil, yazılım paylaşımı ve [pazar türü](#)^(B15) yazılım geliştirme topluluklarının büyümesini sağlamasıdır. Pragmatik hacker ticarileşmeyi sevmemekten ziyade güzel araçlar ve oyuncakları kullanabilmeyi sever ve ideolojik sorun yaşamaksızın ticari yazılım kullanabilir. Bir yandan da, açık kaynak kod tecrübesi, kendisine çok az kapalı yazılımın karşılayabileceği teknik kalite standartlarını öğretmiştir.

Uzun yıllarca hackerlığın içindeki pragmatik bakış açısı kendisini özelde GPL, genelde FSF’in gündemini tamamen desteklemeyi inatla reddederek gösterdi. 1980’ler ve 1990’ların ilk dönemleri boyunca bu tutum Berkeley Unix taraftarları, BSD lisansının kullanıcıları ve BSD kaynak temelinden açık kaynak kodlu Unix’ler yazma eforları ile bağdaştırıldı. Fakat bu eforlar yeterli büyüklükte pazar toplulukları oluşturamadılar, ciddi bölünmeler yaşadılar ve tesirli olamadılar.

1993–1994’lerin Linux patlamasına kadar pragmatiklik gerçek bir kuvvet toplayamadı. Linus Torvalds hiç bir zaman RMS’e özellikle karşı çıkmadıysa da, ticari Linux endüstrisinin gelişimine sıcak bakarak, yüksek kaliteli ticari araçların bazı işler için kullanımını kamu önünde onaylayarak ve kültürün içindeki daha pürist ve fanatik kesimleri yumuşak bir üslupla uyarak örnek oldu.

Linux’un hızlı gelişiminin bir yan etkisi de, öncelikli gönül bağları Linux’a olan çok sayıda yeni hackerın topluluğa katılması idi. Bunlar için FSF’in planları yalnızca tarihsel bir ilgi konusu idi. Her ne kadar Linux hackerları sistemi “the choice of a GNU generation” (yeni/GNU neslinin tercihi — Pepsi reklamına nazire) olarak nitelendirse de, çoğu Stallman yerine Torvalds’ı örnek alıyordu.

Giderek kendilerini azınlıkta bulanlar ticaret karşıtı püristlerdi. Durumun ne kadar değiştiği ise, 1998 Şubatı’nda Netscape’in Navigator 5.0’ı kaynak kodlarıyla dağıtacağı haberine kadar belli olmayacaktı. Bu haber ticari dünyada ‘serbest yazılım’a karşı daha büyük bir ilgi oluşturdu. Hacker kültürüne bu fırsatı değerlendirme ve ürününü ‘serbest yazılım’ yerine ‘açık kaynak kodlu’ olarak yeniden etiketlendirme çağrısı, bu çağrıyı yapan herkesi şaşırtacak derecede büyük bir kabul görecekti.

Paralel bir gelişmede, kültürün pragmatik tarafı 1990’ların ortalarına doğru çok merkezli hale geliyordu. Unix/Internet kökünden başka karizmatik liderler ve kendi bilincine sahip yarı bağımsız topluluklar oluşmaya başladı. Linux’dan sonra bunların en önemlisi, Larry Wall’ın yönetimindeki Perl kültürü idi. Daha küçük, fakat yine de önemli topluluklar arasında John Osterhour’un Tcl’i ve Guido van Rossum’un Python dili etrafında oluşan gelenekler sayılabilir. Bütün bu toplulukların üçü de GPL haricinde kendi lisanslarını oluşturarak kendi ideolojik bağımsızlıklarını dile getirdiler.

4. Çoğulcu Teori, Püriten Uygulama

Bütün bu değişiklikler boyunca ‘serbest yazılım’ veya ‘açık kaynak kod’ un ne olduğuna dair genel bir konsensus devam etti. Bu ortak teorinin en açık ifadesi çeşitli açık kaynak kodlu lisansların birbirleri ile ortak bölümlerinde gözlemlenebilir.

1997 yılında bu ortak öğeler Debian Serbest Yazılım Kuralları halinde toplandı ve daha sonra [Açık Kaynak Kod Tanımı – AKKT^{\(B16\)}](#) haline geldi. AKKT altında tanımlanan kurallar gereğince, açık kaynak kodlu bir lisans, açık kaynak kodlu yazılımların herhangi bir üçüncü şahıs tarafından kayıtsız ve şartsız olarak değiştirilebilme (ve değiştirilmiş halini yeniden dağıtma) özgürlüğünü içermelidir.

Dolayısıyla AKKT’nın (ve GPL, BSD, Perl Artistic License gibi AKKT uyumlu lisansların) öngördüğü teori, herkesin herşey üzerinde istediği gibi çalışabilmesidir. Herhangi bir açık kaynak kodlu ürünü (örneğin Free Software Foundation’un gcc C derleyicisini), yarım düzine farklı hacker’ın alıp, değişik yönlerde geliştirmesini ve hepsinin de *asıl* ürününün bu olduğunu iddia etmesini engelleyen hiç bir şey yoktur.

Pratikte ise bu tarz ‘çatallama’ neredeyse hiç olmamaktadır. Önemli projelerde ayrışmalar seyrek ve daima hem yeniden adlandırılarak, hem de kamusal ortamda büyük miktarda mazeret beyanı ile olmuştur. GNU Emacs/XEmacs ayrışmasında, gcc/egcs ayrışmasında veya BSD’nin kimi uç gruplarının değişik çatallanmalarında, ayrılan grubun oldukça kuvvetli bir toplum geleneğine karşı hareket ettiklerini hissettikleri açıktır. [\[SP\]](#) (sayfa: 24).

Aslında, (ve herkes–herşeyi–yapabilir teorisiyle çelişki içerisinde) açık kaynak kod kültüründe zengin fakat genelde açıkça dile getirilmeyen mülkiyet gelenekleri mevcuttur. Bu gelenekler yazılımları kimin değiştirebileceği, hangi durumlarda değiştirmenin hoş karşılandığı ve (özellikle) topluma değiştirilmiş sürümü kimin dağıtma hakkına sahip olduğunu düzenler.

Bir kültürün tabuları, normlarını açıkça ortaya koyar. Dolayısıyla burada bazı önemli tabuları ortaya koymamız ileride kolaylık sağlayacaktır.

- Projelerin çatallanmasına karşı kuvvetli bir sosyal baskı vardır. Gerçekten çok gerekmedikçe yapılmadığı gibi kamu önünde gerekçelerin ortaya konması ve projenin yeniden adlandırılması beklenmektedir.

- Moderatörlerin izni ve rızası olmaksızın bir projeye yapılan değişiklikleri dağıtmak hoş karşılanmamaktadır. Buna tek istisna, esasen önemli kabul edilmeyen, yeni bir platforma taşınırken gereken ufak tefek değişikliklerdir.
- Proje tarihçesi, katkılar bölümü veya güncelleyiciler listesinden birisinin adını silmek, ilgili kişinin açık şekilde ifade edilmiş rızası alınmaksızın, kesinlikle *yapılmaz*.

Bu makalenin geri kalan kısmında bu mülkiyet geleneklerini ve tabuları daha detaylı inceleyeceğiz. Bu unsurların nasıl işlediğini ele almanın yanı sıra, arkaplandaki sosyal dinamikler ve açık kaynak kod topluluğunun motivasyonları konularında ne gösterdiklerini inceleyeceğiz.

5. Mülkiyet ve Açık Kaynak Kod

Malın sonsuz defa yeniden kopyalanabildiği, kolayca değiştirilebildiği ve etrafındaki kültürün zorlayıcı güç ilişkileri veya önemli zor bulunurluk ekonomisi olmadığı zaman ‘mülkiyet’ ne demektir?

Aslında, açık kaynak kod kültüründe bu sorunun cevabı basittir. Bir yazılım projesinin sahip(ler)i, toplum önünde genel kabul görür şekilde o projenin *değiştirilmiş sürümlerini yeniden dağıtma* hakkına sahip olanlardır.

(Bu bölümde ‘mülkiyet’ten bahsederken, sanki her proje tek bir kişiye aitmiş gibi tekil şahıs kullanacağım. Bazı projelerin gruplara ait olduğu varsayılmalıdır. Böyle grupların iç dinamiklerini ayrıca ele alacağız.)

Standart açık kaynak kod lisanslarına göre evrim oyununda herkes eşittir. Pratikte ise, evrim geçiren yazılıma, genel kabul gören güncelleyiciler tarafından ‘resmen’ kabul edilen ve eklenen ‘yama’lar (patch) ile üçüncü şahıslar tarafından ortaya konan ‘korsan’ yamalar arasında fark vardır. Korsan yamalara sık rastlanmaz ve güvenilmez. [\[RP\]](#) (sayfa: 25).

Genel yeniden dağıtımın temel bir mesele olduğu kolayca görülebilir. Gelenek, şahsi kullanım için kişilerin yazılımları gerektiği şekilde yamamalarını teşvik etmektedir. Gelenek, değişikliklerini kapalı bir kullanıcı veya geliştirici kitlesi içerisinde dağıtan kişilere aldırış etmemektedir. Mülkiyet meselesi ancak değişiklikler, orijinali ile rekabet etmek için, açık kaynak kod toplumunun geneline gönderildiği zaman bir problem haline gelmektedir.

Genelde, bir açık kaynak kod projesinin mülkiyetini elde etmek için üç yol vardır. Birinci ve en bariz olanı, projeyi kurmaktır. Bir projenin başlangıcından beri halen aktif olan tek bir güncelleyicisi olduğu zaman, gelenek o projenin kime ait olduğu hususunda bir *soru* dahi kabul etmemektedir.

İkinci yol, projenin mülkiyetinin bir önceki sahibi tarafından devredilmesidir. (bu bazen ‘bayrağı devretmek’ olarak isimlendirilir) Geliştirme veya güncelleme işlerine ayıracak zamanları olmadığı durumlarda veya ilgilerini kaybettikleri zaman, proje sahiplerinin yetkin birisine projelerini devretme yükümlülükleri olduğu, toplulukta genel kabul görmüş bir fikirdir.

Önemli projelerde böyle kontrol devirlerinin oldukça şatafatlı şekilde anons edilmesi dikkate şayandır. Açık kaynak kod topluluğunun mevcut sahibin devir tercihine karıştığı görülmemişse de, uygulama, genelin kabul edebileceği bir tercih yapılması doğrultusundadır.

Küçük projelerde, proje dağıtımında yer alan değişiklik tarihçesinde mülkiyet değişikliğinin not edilmesi yeterlidir. Burada açık varsayım, mülkiyet değişikliğinin bir önceki sahibin rızası dahilinde yapılmaması halinde, makul bir zaman içerisinde kamuya açıklamada bulunarak, toplum desteği ile birlikte projenin kontrolünün geri alınabileceğidir.

Bir projenin mülkiyetini almanın üçüncü yolu, projede yapılması gerekenler olduğu ve sahibinin kayıp olduğu veya ilgilenmediğini ortaya koymaktır. Eğer bunu yaparsanız, projenin sahibini bulma gayretini göstermek size aittir. Eğer bu gayret başarısız olursa, konu ile alakalı bir ortamda (örneğin uygulama ile ilgili tartışmaların yapıldığı bir Usenet haber grubunda) projenin sahipsiz olduğu ve sizin sorumluluğunu yüklenmeyi düşündüğünüzü belirtirsiniz.

Gelenekler bu noktada kendinizi yeni sahip ilan ettiğinize dair bir bildiri yayınlamadan önce bir müddet beklenmesini emretmektedir. Bu süre zarfında eğer birisi proje üzerinde çalıştıklarını bildirirse, onların hakkı sizinkinden üstündür. Niyetlerinizi birden fazla defa tekrarlamak adap gösterir, takdir edilir. Niyetlerinizi birden fazla ortamda belirtmeniz (ilgili haber grupları, posta listeleri) daha da fazla takdir edilir; cevap beklerken sabır göstermeniz de ha keza. Kısacası, bir önceki sahibi veya başka hak sahiplerini bulmakta gösterdiğiniz efor ne kadar görünür ise, cevap alınamaması halinde hak iddia etmeniz o kadar kabul edilir.

Eğer bu süreci projenin kullanıcı kitlesinin gözü önünde yaşadığınız iseniz ve itirazlarla karşılaşmadıysanız, o zaman yetim kalmış projenin sahipliğini üstlenebilir ve tarihçesine bununla ilgili bir ibare düşebilirsiniz. Bu yöntem, yine de, bayrağı devralmaktan daha az emin bir yoldur ve kullanıcı kitlesinin gözlemleyebileceği şekilde projeye hatırı sayılır katkılarda bulunmadığınız müddetçe gerçek sahip sayılmayabilirsiniz.

Bu gelenekleri yirmi yıllık bir süre zarfında, bilfiil uygulamada, FSF öncesi dönemlerden beri gözlemledim. Bir kaç çok ilginç unsuruna dikkat çekmek istiyorum. Bir tanesi, hacker'ların bu gelenekleri tam olarak ne yaptıklarının farkında olmadan uygulamalarıdır. Hatta, yukarıdaki izah, bu usullerin ilk defa bilinçli olarak kağıda aktarılmasıdır.

Bir başka dikkat çekici husus, bilinç altı olmasına karşın, bu geleneklerin takdire şayan bir şekilde eksiksiz uygulanmasıdır. Zaman içerisinde izlediğim yüzlerce açık kaynak kod projesi içerisinde bu geleneklerin çiğnenmesi vakalarını bir elin parmaklarında sayabilirim.

Üçüncü bir özellik de bu geleneklerin zaman içerisinde değişmelerine rağmen, tutarlı bir yönde gelimeleridir. Gidilen yön, kamuya daha fazla hesap vermek, daha fazla bilgi vermek, katkıların ve değişiklik tarihçelerinin korunmasına daha fazla özen göstermek olmuştur.

Bu özellikler, söz konusu alışkanlıkların tesadüfi olmadığı, açık kaynak kod kültürünün işleyişinde temelden etkili olan bir oluşturuç özellik, gündem veya kalıbın sonucu olduğu çıkarımına yol açmaktadır.

Bir okuyucu, Internet hacker kültürünü, korsan kültürüyle (korsan bülten tahtaları ve oyun kırma etrafına toplanan "warez d00dz"larla) karşılaştırmamanın bu iki farklı kültürün ayrı oluşturuç özelliklerini ortaya koyduğuna işaret etti. Makalede d00dz'lara geri dönerek bu karşılaştırmayı yapacağız.

6. Locke ve Toprak Mülkiyeti

Bu oluşturuç özelliği anlamak için bu geleneklerin, hacker'ların genelde ilgi sahalarının çok dışında, tarihte benzeştiği bazı şeylerden bahsetmek faydalı olabilir. Hukuk tarihi ve siyaset felsefi öğrencilerinin bileceği gibi, bu geleneklerin işaret ettiği mülkiyet teorisi, birebir Anglo–Amerikan 'common–law'daki toprak iskanı ile aynıdır! (Çevirmen not: Bizim hukuk sistemimiz, kıta Avrupa'sının Jüstinyen kanunlarından başlayan hukuk geleneğini takip eder. Bu hukuk sisteminde, yazılı kanun esastır. Farklı bir hukuk geleneğinden gelen Anglo–Amerikan sistemi ise, yazılı kanun haricinde "örf/töre" kabul edebileceğimiz, toplumun genel geçer kabullerini de kanunmuşcasına kabul eder. Bu, kaynağı yazılı kanun değil, Anglo–Sakson örf ve töresi olan hukuk sistemine 'common–law' denir.)

Bu teori, toprağın mülkiyetinin üç farklı yoldan elde edilebileceğini kabul eder.

Serhatta, daha önce hiç sahibi olmamış olan toprağın bulunduğu noktada, isteyen, toprağı *iskan* ederek sahiplenebilir. Bu iskan için, kendi emeğini toprakla karıştırıp toprağın etrafını çitle çevirmek ve mülkün üzerinde sahiplik iddiasında bulunmak gereklidir.

Önceden iskan edilmiş bölgelerde, mülkiyet, *tapunun* el değiştirmesi ile yapılır — yani tapu bir önceki sahibinden yeni sahibine verilmelidir. Bu teoride, tapunun sahiplik zinciri önemlidir. Mülkiyet iddiasının en kuvvetli delili, toprağın ilk iskan edildiği zamana kadar geri giden bir dizi tapu ve el değiştirme kayıtlarıdır.

Son olarak, 'common–law', toprak mülkiyetinin kaybedilebileceği veya terkedilebileceğini kabul eder (örneğin sahibin arkasında mirasçı bırakmadan ölmesi veya mülkiyeti ilk iskan tarihine kadar geri götüren kayıtların kaybı halinde). Bu şekilde sahipsiz kalmış bir parça toprak, herhangi birisi tarafından tekrar iskan edilebilir — bu

iskanın kabulü için, iddia sahibinin toprağı işgal etmesi, üzerinde işleme/iyileştirme yapması ve tıpkı ilk iskanda olduğu gibi hak iddia etmesi gerekir.

Tıpkı hacker gelenekleri gibi, bu teori, merkezi otoritenin olmadığı veya zayıf olduğu uzun bir dönem içerisinde organik olarak gelişmiştir ve kaynağı bin yıl önceki Germen ve İskandinav kabile geleneklerine dayanır. Modern dönemin başlarında İngiliz siyaset filozofu John Locke tarafından sistematize ve rasyonalize edildiği için, bazen Locke'nin mülkiyet teorisi olarak adlandırılır.

Aynı mantığı taşıyan teoriler, az bulunan malın dağıtımını düzenleyebilecek kuvvette bir merkezi otoritenin olmadığı ve malın yüksek ekonomik veya yaşamsal değeri olduğu her noktada ortaya çıkmıştır. Bazen romantik bir görüşle mülkiyet kavramlarının olmadığına inanılan avcı-toplayıcı kültürlerde dahi benzer teorilere rastlanır. Örneğin Kalahari Çölü'nde yaşayan !Kung San kabilesinde avlanma bölgelerinin mülkiyeti yoktur. Fakat pınarların ve içme suyu bulunan gölcüklerin mülkiyeti, Locke teorisine benzer bir usulde *vardır*.

!Kung San örneği, Locke mülkiyet geleneklerinin, ancak kaynaktan beklenen getirinin, kaynağı savunma giderlerinden daha büyük olduğu zaman ortaya çıktığını göstermesi açısından önemlidir. Avlanma bölgeleri mülk değildir, çünkü avlanmanın getirisi değişkendir ve (değerli olmasına rağmen) günlük yaşamı devam ettirmek için gerekli değildir. Sulama kaynakları ise günlük yaşam için elzemdir ve savunulabilecek kadar küçüktür.

Bu makalenin başlığında yer alan "noosfer", düşüncelerin, fikirlerin oluşturduğu bölge, mekan anlamına gelmektedir. [N] (sayfa: 24). Hacker mülkiyet gelenekleri, bütün yazılımların oluşturduğu uzayda, yani noosfer'in bir alt kümesinde, Locke mülkiyet teorisinin geçerli olduğunu göstermektedir. Yeni bir açık kaynak kod projesini hayata geçiren herkes, noosfer'in yeni bir bölgesini mesken tutmakta, iskana açmaktadır.

Faré Rideau <fare@tunes.org> haklı olarak hacker'ların tam olarak soyut fikir dünyasında çalışmadıklarını söylemektedir. Rideau, hacker'ların sahip oldukları şeyin *programlama projeleri* olduğunu belirtmekte ve bunların, somut emeğin (geliştirme, hizmet vs.) odak noktaları olduğunu ve şöhret, güvenilirlik gibi kavramların bu projelerle alakalı olduğunu söylemektedir. Dolayısıyla, hacker projelerinin oluşturduğu uzay, noosfer *değil*, onun bir izdüşümü olan, noosferi araştıran program projeleridir. (Astofizikçilerden özür dileyerek, bu izdüşüm uzayına ergosfer, yani 'emeğin küresi' demenin etimolojik olarak doğru olacağını düşünüyorum.)

Pratikte ise, noosfer ve ergosfer arasındaki fark bu makalenin kapsamında önemli değildir. Faré'nin tanımladığı soyutlukta bir 'noosfer'in anlamlı bir şekilde var olup olmadığı tartışılabilir; bu kadar soyut hale gelmiş bir kavramı kabul etmek için neredeyse Eflatun ekolünden bir filozof olmak gerekmektedir. Ve noosfer ve ergosfer arasındaki fark, *pratikte* yalnızca fikirlerin (noosfer'in öğeleri) sahiplenilemeyeceği, yalnızca fikirlerin somuta dönüştürülmüş hali olan projelerin sahiplenilebileceğini göstermek için önemlidir. Bu soru, fikri mülkiyet teorilerindeki kavramlara girmekte ve bu makalenin kapsamı dışına çıkmaktadır. (bakınız [DF] (sayfa: 24))

Kavram kargaşasını önlemek için noosfer veya ergosfer'in bazen (çoğu hacker'ın bu terimden hoşlanmamasına rağmen) 'siber uzay' olarak adlandırılan, elektronik medyada bulunan sanal yerlerden farklı olduğunu söylemekte yarar vardır. Siber uzayda mülkiyet, gerçek hayata daha yakın olan ve bizim bahsettiklerimizden tamamen farklı olan kurallara dahilinde işlemektedir — esas olarak, medyanın ve siber uzayın bir parçası olan makinaların sahibi olan kişi, siber uzayın o parçasının sahibi olmaktadır.

Locke mantığı, gelenekleri değerlendirirken, açık kaynak kod hacker'larının geleneklere uyma sebebinin eforlarından elde etme beklentisi içinde oldukları bir getiriyi savunmak olduğuna işaret etmektedir. Bu getiri, iskan projeleri için gereken efordan, mülkiyet zincirini belgeleyen sürüm tarihçeleri tutma maliyetinden ve sahipsiz bir projeyi devralmak için kamuya açık bildirimde bulunmak ve beklemenin zaman maliyetinden daha önemli olmalıdır.

Dahası, açık kaynağın 'getirisi', yalnızca yazılımı kullanmaktan daha ileride, projenin çatallanması halinde azalacak bir şey olmalıdır. Eğer tek problem kullanım olsa idi, çatallamaya karşı bir tabu bulunmaz ve açık kaynak kod mülkiyeti, toprak mülkiyeti ile benzeşmezdi. Hatta, açık kaynak kod lisansları, yalnızca kullanımın tek getiri olduğu ve çatallamanın kabul edilebilir olduğu bir dünyaya işaret etmektedirler.

Bazı getiri türlerini hemen eleyebiliriz. Bir ağ bağlantısı üzerinden doğru dürüst baskı kurulamadığından dolayı, güç elde etme beklentisi olamaz. Benzer şekilde, açık kaynak kod kültürünün para benzeri bir kavramı veya içsel bir az bulunan mal ekonomisi yoktur, dolayısıyla hacker'lar materyal zenginliğe paralel bir şeyin (örneğin az bulunan bir şeyin biriktirilmesi) peşinde olamazlar.

Açık kaynak kod faaliyetlerinin insanları zenginleştirmesi için bir yol mevcuttur — bu yol, arkada yatan motivasyona da ışık tutabilir. Zaman zaman, hacker kültüründe elde edilen şöhret, gerçek dünyaya ekonomik açıdan önemli şekillerde yansiyabilir. Bu şöhret, bazen daha iyi bir iş teklifi, bir danışmanlık kontratı veya bir kitap yazma teklifi şeklinde gerçek dünyaya yansiyabilir.

Bu yan etki, pek çok hacker için son derece az rastlanır ve marjinaldir. Bu kadar az rastlanan bir etkinin tek motivasyon olması, hacker'ların sürekli olarak yaptıkları şeyi para için değil, sevgi veya idealizm uğruna yaptıkları iddialarını reddetsek bile inandırıcılıktan uzaktır.

Fakat, bu ekonomik yan etkilerin pazarlık ve anlaşma süreci, incelemeye değerdir. Aşağıda açık kaynak kod kültürünün *kendi içinde* şöhret dinamiklerini anlamanın açıklayıcı güce sahip olduğunu göreceğiz.

7. Bir Hediye Kültürü Olarak Hacker Ortamı

Açık kaynak kod kültüründe şöhretin yerini anlamak için, tarihten antropoloji ve iktisada geçiş yapmak ve *takas kültürleri* ile *hediye kültürleri* arasındaki farkı incelemek zorundayız.

Sosyal statü için rekabet içinde olmak, doğal bir insan güdüsüdür. Tarımın icadından önceki tarihin %90'ında, atalarımız küçük, göçebe, avcı kabileler halinde yaşadılar. Yüksek statüye sahip bireyler (koalisyonlar oluşturmakta ve başkalarını kendileri ile ittifaka ikna etmekte en başarılı olanlar), en sağlıklı eşlere ve en iyi yiyeceğe erişebiliyorlardı. Bu statü elde etme güdüsü, yaşam için gerekli şeylerin azlık derecesine bağlı olarak kendisini başka şekillerde gösterebiliyordu.

İnsanların organizasyon şekillerinin pek çoğu, çeşitli azlıklara adaptasyon süreçlerinden kaynaklanmaktadır. Her değişik organizasyon, sosyal statü elde etmek için değişik yollar içerir.

En basit organizasyon, *emir hiyerarşisidir*. Emir hiyerarşilerinde, az bulunan malın dağıtımı bir merkezden yapılır ve güç kullanarak itaat sağlanır. Emir hiyerarşileri, büyük ölçeklere çıkmakta sorun yaşar [Mal], büyüdükçe verimsizleşir ve zorbalashırlar. Bu sebepten dolayı, geniş aile boyundan daha büyük emir hiyerarşilerinin neredeyse tümü, daha büyük bir ekonominin parazitleri olarak bulunur. Emir hiyerarşilerinde sosyal statü ana olarak zorlama gücüne sahiplikle eşittir.

Toplumumuz ağırlıklı olarak bir *takas ekonomisidir*. Bu, emir hiyerarşilerine kıyasla büyük ölçeklere rahatlıkla çıkabilen, malların az bulunurluğuna karşı ileri düzeyde bir adaptasyondur. Az bulunan mallar ticaret ve gönüllü ittifaklar yolu ile merkezden bağımsız bir şekilde paylaştırılır (bu sistemde rekabet güdüsünün yarattığı başlıca etki, daha ittifaka uygun hareketler oluşturmaktır). Bir takas ekonomisinde sosyal statü, ağırlıklı olarak malların (bunların somut olması gerekli değil) kontrolüne sahip olmak ve bunları kullanmak veya ticaretini yapmak serbestisine sahip olmakla belirlenir.

İnsanların çoğunluğu kafalarında bu iki farklı sistemin soyut modellerini taşır ve birbirleri ile nasıl etkileşim içerisinde bulunabileceklerini bilirler. Devlet, askeriye ve (örneğin) organize suç, adına 'serbest pazar' dediğimiz daha geniş takas ekonomisi içerisinde yaşayabilen emir hiyerarşileridirler. Çok daha az bilinen ve genelde yalnız antropologlar tarafından tanınan bir üçüncü sistem vardır ve bu sistem diğerlerinden radikal derecede farklıdır: *hediye kültürü*.

Hediye kültürleri, mal azlığına değil, çokluğuna bir adaptasyon sürecidir. Yaşam için gerekli malların azlığı ile ilgili bir problem yaşamayan toplumlarda ortaya çıkarlar. Hediye kültürlerini, ılıman iklim ve bol gıdalı bölgelerde yaşan yerlilerde görebiliriz. Bizim toplumumuzun kimi katmanlarında da hediye kültürleri yer alır, özellikle eğlence dünyası ve çok zenginler arasında.

Bolluk, emir hiyerarşilerinin devamını güçleştirir ve takas ilişkilerini anlamsız bir oyun haline getirir. Hediye kültüründe, sosyal statü, neyi kontrol ettiğinizle değil, *neyi dağıttığınızla* belirlenir.

Ve ancak bu şekilde Kwakiutl kabile şefinin "potlach" şölenini, multi-milyonerlerin kapsamlı ve genelde kamusal ortamda yayınlanan hayır işlerini ve hacker'ın yüksek kaliteli açık kaynak kod üretmek için harcadığı uzun saatleri açıklayabilirsiniz.

Bu şekilde incelendiği zaman, açık kaynak kod hackerlarının toplumunun bir hediye kültürü olduğu son derece açıktır. Bu toplumda 'yaşam için gerekli olan' malların — yani disk alanı, bant genişliği, işlem gücü — ciddi bir azlığı yoktur. Yazılımlar serbestçe paylaşılabilir. Bu bolluk, rekabetçi başarı kriteri için yalnızca eşitleri arasındaki şöhretin önemli olduğu bir kültür doğurmaktadır.

Bu gözlem, hacker kültürünün gözlenen tüm özelliklerini açıklamak için tam olarak yeterli değildir. Cracker'lar (şifre kırıcılar) ve warez d00dz'lar (korsan yazılım dağıtıcıları), hackerlar ile aynı elektronik ortamda yaşayan hediye kültürlerine sahiptirler fakat davranışları tamamen farklıdır. Şifre kırıcılar ve korsan yazılım dağıtıcılarında grup mentalitesi çok daha kuvvetli ve hackerlardan daha dışlayıcıdır. Bu gruplar sırları paylaşmak yerine gizli tutmaktan hoşlanırlar; korsan gruplar içerisinde yazılım kırmaya yarayan kaynak kodsuz uygulamaları dağıtmak, bunu nasıl başardıklarını açıklayan bilgi bulmaktan çok daha kolaydır. (Bu davranışların içeriden gelen bir açıklaması için: [\[LW\]](#) (sayfa: 25)).

Bu, eğer yeterince açık değilse, bir hediye kültürünün birden fazla şekilde biçimlenebileceğini göstermektedir. Tarih ve paylaşılan değerler önemlidir. Hacker kültürünü *Hackerlığın Kısa Tarihçesi*[\[HH\]](#) makalesinde özetledim; şimdiki durumunun oluşmasında belirleyici rol oynayan etkenler gizli değildir. Hackerlar, kültürlerini, rekabetin alacağı *şekiller* konusunda yaptıkları tercih kümesi ile belirlemişlerdir. Bu makalenin geri kalan kısmında bu rekabet şekillerini inceleyeceğiz.

8. Hackerlığın Zevki

Bu 'şöhret oyunu' analizini yaparken, güzel yazılım dizayn etme ve bunu çalışır kılmanın verdiği saf sanatsal hazzı gözardı etmek veya küçümsemek istemiyorum. Hepimiz bu hazzı tadar ve bununla motive oluruz. Bu hazzın yeterli motivasyon olmadığı kişiler zaten hiç hacker olmazlar, tıpkı müzik sevmeyenlerin bestekar olmadıkları gibi.

Öyleyse, belki salt zanaatkarlık hazzının ana motivasyon olduğu bir hacker davranış modelini incelemeliyiz. Böyle bir 'zanaatkar' modeli, hacker geleneğini hem zanaatkarlık için fırsatları artırmak, hem de ortaya çıkan eserin kalitesini yükseltmek için bir yöntem olarak açıklayabilmelidir. Böyle bir model, 'şöhret oyunu' modeli ile ilişkili midir ya da daha farklı sonuçlar verir mi?

Aslında hayır. 'Zanaatkarlık' modelini incelerken, hackerlığı bir hediye kültürü olmaya zorlayan problemlerle tekrar karşılaşırız. Eğer kalite için bir ölçüt yok ise, kalite nasıl yükseltilir? Eğer mal azlığı ekonomisi yok ise, başkalarının beğenisinden başka ne gibi bir ölçüt olabilir? Bütün zanaatkarlık kültürleri, zaman içerisinde bir şöhret oyunu vasıtası ile şekillenmeli sonucunu çıkarabiliriz. Ve bu yapıyı, ortaçağ ahiliklerinden başlayarak pek çok tarihi zanaatkar kültüründe görebiliriz.

Önemli bir konuda, 'zanaatkarlık' modeli, 'hediye kültürü' modelinden daha zayıf kalmaktadır — bu makaleye başlarken ortaya koyduğumuz ilişkiyi açıklayamamaktadır.

En son olarak da, 'zanaatkarlık' motivasyonu, psikolojik olarak şöhret oyunundan, inanmak istediğimiz kadar uzak olmayabilir. Bir çekmeceye kilitlenen ve bir daha kullanılmayan güzel yazılımınızı düşünün. Sonra da aynı yazılımın pek çok kişi tarafından etkin bir şekilde kullanıldığını. Hangisi size daha fazla haz verir?

Her halükarda zanaatkarlık modeline arada sırada göz atmaya devam edeceğiz. Pek çok hackera içgüdüsel olarak yakın gelmekte ve bazı bireysel davranışları yeteri kadar iyi açıklamaktadır [\[HT\]](#) (sayfa: 25).

Bu makalenin ilk halini Internet'te yayınladıktan sonra anonim bir okuyucu şöyle yazdı: "Şöhret için çalışmayabilirsiniz, fakat eğer işinizi iyi yaparsanız, şöhret, getirileri de olan gerçek bir ödeme şeklidir." Bu, önemli ve hassas bir noktadır. Şöhret getirileri, bir zanaatkar bundan haberdar olsa da, olmasa da çalışmaya devam etmektedir; dolayısıyla; bir hacker, kendi davranışını şöhret oyununun bir parçası olarak anlasa da, anlamasa da, davranışı bu oyun tarafından şekillenecektir.

Diğer bazı yazışmalar, hackerlıktan duyulan haz ve başkalarının beğenisini, Abraham Maslow'un yaygın bilinen, insan motivasyonu için 'değerler hiyerarşisi' modelinde [MH] (sayfa: 25) yaşam için gereken şeylerin üzerindeki seviyelerle alakalandırdı. Bu görüşe göre, hackerlık hazzı, daha alt seviyedeki ihtiyaçlar (fiziksel güvenlik, 'aidiyet' gibi) karşılanmadan kendini göstermeyecek bir kendini bulma, kendini aşma ihtiyacıdır. Dolayısıyla, şöhret oyunu, hackerlık hazzının bireyin ana motivasyonu haline gelecek bir sosyal ortam oluşturmakta kritik rol oynamaktadır.

9. Şöhretin Değişik Yüzleri

Her hediye kültüründe prestij sağlamanın neden önemli olduğuna dair farklı gerekçeler bulunur:

En başta ve en açık olarak, başkaları arasında iyi şöhrete sahip olmanın kendisi başlı başına bir ödüldür. Daha önce kısaca bahsedilen evrim sebepleri yüzünden bu şekilde hissederiz. (Pekçokları, prestij elde etme güdülerini, görülebilir bir denk kişiler kitlesi olmaksızın soyutlarlar, örneğin "şeref", "ahlaklılık", "dindarlık" vs.; bu, altta yatan mekanizmayı değiştirmez)

İkinci olarak, prestij, dikkat çekmek ve başkalarının işbirliğini sağlamak için iyi (ve saf bir hediye kültüründe *tek*) yoldur. Eğer birisi bonkörlüğü, zekası, adilliği, liderlik kabiliyeti ve başka iyi özellikleri ile tanınıyorsa, başkalarının onunla işbirliğine razı etmesi daha kolaydır.

Üçüncü olarak, eğer hediye kültürü, bir takas kültürü veya emir hiyerarşisi ile temas halinde veya iç içe ise, şöhret, bu kültürle yansiyabilir ve orada daha yüksek statü elde edilmesine yol açabilir.

Bu genel sebeplerin dışında, hacker kültürünün kendine has durumu, prestiji 'gerçek dünyada' yer alan bir hediye kültüründen daha da önemli kılmaktadır.

Ana 'özel durum', hediye edilen ürünlerin (veya başka bir gözle bakıldığında, hediye edilen enerji ve zamanın somutlaşmış hali) çok kompleks olmasıdır. Hediyelerin değerleri, materyal bir hediye veya takas ekonomisi parası ile kıyaslanamayacak kadar zor ölçülebilir. Objektif olarak değerli bir hediye değerli bir hediyeden ayırdetmek çok daha zordur. Dolayısıyla, hediye verenin statü elde etme girişiminin başarısı, kendisi ile denk olan başkalarının kritik değerlendirmesi ile alakalıdır.

Başka bir özel durum da, açık kaynak kod kültürünün nispeten saf olmasıdır. Çoğu hediye kültürü çeşitli tavizler vermiştir — ya takas ekonomileri ile lüks tüketim maddeleri ticareti için etkileşim ya da aile veya kabile gruplandırmaları ile emir hiyerarşisi ile etkileşim. Açık kaynak kod kültüründe bunların önemli karşılıkları bulunmamaktadır; dolayısıyla başkalarının değer vermesi ile oluşan şöhret haricinde statü kazanmak neredeyse imkansızdır.

10. Mülkiyet Hakları ve Şöhret Teşvikleri

Artık bahsettiğimiz analizleri, hacker mülkiyet geleneklerini izah eden tutarlı bir bütün haline getirebiliriz. Noosferi iskana açmanın getirisini artık biliyoruz; hackerların hediye kültüründe şöhret kazanma ve bunun getirdiği ikincil ödüller.

Bu noktadan, hackerlığın Locke mülkiyet geleneklerini, *şöhret teşviklerini maksimize etmenin* bir yolu olarak izah edebiliriz. Bu yol ile, verilen değer ait olduğu yere gitmesi güvence altına alınmaktadır.

Bu analiz ile birlikte, yukarıda gözlemlediğimiz üç tabu tamamen anlaşılır olmaktadır. Eğer başka birisi yapılan işe sahip çıkar veya bozarsa, işi yapanın şöhreti haksız yere darbe alır; tabular (ve bununla alakalı gelenekler) bunun

olmasını engellemeye çalışmaktadır. (Veya, daha pragmatik olarak bakıldığı zaman, hackerlar, aynı davranış kendilerine karşı yapıldığı zaman haklarından feragat etmemek için genellikle başkalarının işini çatalamaz veya korsan yama yayınlamazlar.)

- Projelerin çatalanması, çatal öncesi proje üzerinde çalışanların, ancak çatalanma sonrası oluşan her iki projede de aktif rol üstlenerek kontrol altında tutabilecekleri bir şöhret riskini getirmektedir. (Bu işlemi yapmak, pratikte ya çok karmaşık olacağı ya da çok zor olacağı için uygulanamaz.)
- Korsan yamalar (veya daha da kötüsü, korsan ikilik dosyalar) dağıtmak, proje sahiplerini haksız bir şöhret riskine uğratmaktadır. Resmi kod mükemmel dahi olsa, yamalardan kaynaklanan hatalar projenin sahiplerine mal edilecektir. (bakınız [\[RP\]](#) (sayfa: 25)).
- Birisinin adını tek taraflı olarak bir projeden çıkarmak, bu kültür ortamında, en ağır suçlardan birisidir. Mağdurun hediyesini çalmak ve hırsızın kendi hediyesi imiş gibi göstermek anlamına gelir.

Elbette, bir projeyi çatallamak veya korsan yama dağıtmak, aynı zamanda orijinal geliştirici gruba direkt bir öhret saldırısı manasına da gelmektedir. Eğer ben, sizin projenizi çatalar veya korsan bir yama yayınlarsam, "[projeyi benim götürdüğüm yere götürmemek ile] yanlış bir karar aldınız" diyorum. Benim çatalanmış projemi kullanan herkes, bu görüşü destekler. Fakat, bu, kendi başına, şiddetli olmakla beraber geçerli bir saldırı yöntemi olurdu: denklemin değerlendirmesinin en keskin tarafı. Dolayısıyla bu açıklama, tabuları güçlendirmekle beraber, tek başına açıklayamaz.

Bu üç tabu davranışın hepsi de mağdur(lara) lokalize zarar vermekle beraber, açık kaynak kod toplumunun tamamına da global zarar vermektedir. Potansiyel katkı sağlayıcıların, hediyelerin/pozitif davranışların ödüllendirileceği beklentisini azaltarak, toplumun geneline zarar vermektedirler.

Bu üç tabunun ikisi için başka açıklamalar da getirilebileceğinden bahsetmekte yarar vardır.

İlk olarak, hackerlar genelde proje çatallamaya karşı duydukları antipatiyi, çatalanan projelerin aşağı yukarı birbirine paralel olarak gelişeceğini ve işi gereksiz yere çoğaltacakları ile açıklarlar. Aynı zamanda, çatalamanın geliştirici toplumu genelde ikiye böldüğünü ve her iki yavru projeye, ana projenin sahip olduğundan daha az beyin kaldığından bahsederler.

Bir okuyucum, bir çatal yavrusunun uzun vadede yaşayabilmesi için ciddi bir "pazar payı" gerektiğini yazdı. Bu, herkesin işbirliği yapması ve çatallamayı engellemesi için önemli bir etkidir, çünkü önceden kimin kaybedecek tarafta olacağını ve harcadığı emeğin ya tamamen kaybolmasını ya da tanınmaz hale geleceğini göreceğini kestirmek mümkün değildir.

Korsan yamaların sevilme nedeni olarak, genellikle hata takibini çok zorlaştırması ve proje sahiplerine, *kendi* hatalarını düzeltmek yeteri kadar iş iken, ek iş çıkarması gösterilmektedir.

Bu açıklamalarda doğruluk payı vardır ve mülkiyetin Locke geleneklerini sağlamlaştırmakta önemli rol oynamaktadırlar. Fakat, entellektüel olarak çekici de gelseler, bu tabuların çiğnendiği seyrek olaylarda neden bu kadar duygusal ve sahiplenici davranıldığını açıklayamamaktadırlar — bu duygusal tepki, yalnızca mağdurlar tarafından değil, çoğu zaman oldukça sert reaksiyon gösteren gözlemleyiciler tarafından da verilmektedir. Gözlemlenen bu davranışları yalnızca işin çoğalması ve gelişim sürecinin zorlaşması ile ilgili serin kanlı izahatlarla açıklamak mümkün değildir.

Üçüncü tabuyu ise, şöhret oyunu analizinden başka bir şeyle açıklamak zor görünmektedir. Bu tabunun "adil olmazdı" dan daha derin bir analizle pek incelenmemesi de, kendi başına bir anlam ifade etmektedir.

11. Ego Problemi

Makalenin başında, bir kültürün bilinç altından bilinmesi ile, bilinç üstü ideolojisi arasında genelde pek çok çelişki bulunduğundan bahsettim. Bunun önemli bir örneğini, standart lisanslarda yazarların tersine, Locke mülkiyet geleneklerinin takip edilmesi ile gördük.

Bu durumun başka ilginç bir örneğini, şöhret oyunu analizini hackerlarla tartışırken gördüm. Pek çok hacker bu analizi reddetti ve kendi davranışlarının denkleri arasında şöhret edinme isteğinden, veya, o zamanlarda tedbirsizce isimlendirdiğim gibi 'ego tatmini'nden kaynaklandığını kabul etmek istemedi.

Bu, hacker kültürü hakkında önemli başka bir noktayı gündeme getirmektedir. Bilinçüstünde, egoya dayalı motivasyon ve egotizm'e güvenilmemekte, hatta tiksiniilmektedir. Kendini öne çıkarmak, toplumun geneli için faydalı dahi olsa, acımasızca eleştirilmektedir. O kadar ki, kültürün 'önemli şahsiyetleri' ve kabile yaşlıları yumuşak bir üslupla konuşmak zorunda ve statülerini koruyabilmek için her fırsatta kendileri ile dalga geçmelidir. Bu tutumun, neredeyse tamamen ego üzerine kurulu bir ödüllendirme yapısı ile nasıl bağdaştığı, izah aramaktadır.

Bu tutumun önemli bir kısmı, 'ego'ya karşı genelde negatif bir tavır alan Avro–Amerikan tutumundan kaynaklanmaktadır. Çoğu hackerın kültürel geçmişi, ego tatmini aramanın kötü (veya en azından olgun olmayan) bir davranış olduğunu; egonun en iyi ihtimalle yalnız prima–donna'larda kabul edilebilir bir egzantriklik ve çoğu zaman da ruhi patoloji belirtisi olduğunu öğretmektedir. Ego, ancak soyutlanarak ve gizlenerek "kendinden emin olmak", "denkleri tarafından beğenilmek", "profesyonellik", "başarıdan duyulan gurur" gibi biçimler aldığı zaman kabul edilebilmektedir.

Kültürel mirasımızın bu yönünün sağlıksız kökleri ve gerçekten 'saf' güdülere sahip olduğumuz konusunda (bütün psikoloji ve davranışın ispatlarına rağmen) kendi kendimizi kandırmanın yarattığı zarar konusunda bütün bir makale yazılabilir. Eğer Friedrich Wilhelm Nietzsche ve Ayn Rand bu işi zaten gayet güzel yapmamış ve 'altruizm'i göz ardı edilen kişisel çıkarlar halinde açıklamamış olsalardı yapabilirdim.

Fakat burada ahlak felsefesi veya psikolojisi yapmıyorum, dolayısıyla egonun kötü bir şey olduğuna inanmanın yalnızca küçük bir zararına işaret edeceğim: pek çok hacker için kendi kültürlerinin sosyal dinamiğini anlamayı zorlaştırmıştır!

Bu sorgulama ile işimiz tam olarak bitmedi. Görünür şekilde ego güdüsüyle yapılan davranışlara karşı o kadar kuvvetli bir tabu vardır ki, hackerlar için bunun önemli bir işleve sahip olması gerekir. Bu tabu pek çok diğer hediye kültüründe çok daha hafiftir (veya hiç yoktur), örneğin eğlence dünyası veya çok zenginler arasında!

12. Alçakgönüllülüğün Değeri

Hacker kültürünün ödül mekanizmaları için prestij edinmenin önemini gösterdikten sonra, neden bu olgunun yarı–gizli ve itiraf edilemez olduğunu anlamamız gereklidir.

Korsan kültürü ile karşılaştırma yaparsak, o kültürde statü elde etme davranışlarının açık ve ortada olduğunu görürüz. Korsanlar, "sıfırıncı gün warez" dağıtmak (orjinal yazılımın yayın tarihi ile aynı gün korsan, kırılmış sürümü çıkarmak) ile statü ararlar, fakat bunu nasıl başardıkları konusunda ağızları sıkıdır. Bu sihirbazlar oyunlarının içyüzünü göstermek ve vermek istemezler. Bunun sonucu olarak korsan kültürünün genel bilgi dağıtıcısı yavaş büyür.

Hacker toplumunda, kişinin aldığı tavır, yaptığı işten ibarettir. Çok sıkı bir meritokrasi işler (en iyi zanaat kazanır) ve kalitenin *tek başına* konuşması gerektiği konusunda ciddi bir anlayış hakimdir. En iyi övünme, "çalışan" ve iyi bir programcının kaliteli olduğunu görebildiği koddur. Dolayısıyla hacker kültürünün genel bilgi dağıtıcısı hızlı büyür.

Ego yoğun davranışlara karşı tabu, bu şekilde verimliliği artırmaktadır, fakat bu, ikinci derecede bir etkidir; burada esas korunan şey, toplumun denkler tarafından değerlendirme mekanizmasında kullandığı bilginin kalitesidir. Yani, övünmek veya kendisini önemli görmek, yaratıcı ve işbirlikçi davranış deneylerinden alınan bulgu sinyallerine gürültü kattığı için bastırılmaktadır.

Benzer sebeplerden dolayı, kodu eleştirmek yerine yazarı eleştirmek yapılmaz. Burada, ilginç bir hassasiyet bulunmaktadır; hackerlar ideolojik veya kişisel farklardan dolayı şiddetli tartışmalara girmekte kendilerini özgür hissederler, fakat bir hacker'ın başka birisinin teknik yeterliliğine saldırıda bulunması görülmez. (özel eleştiri bile nadirdir ve sakın bir tavırda yapılır). Hata ayıklama ve değerlendirme hep projeye karşı yapılır, kişiye karşı değil.

Bundan daha da öteye, geçmişte bulunan yazılım hataları otomatik olarak bir geliştiriciye mal edilmez; bir hatanın düzeltilmiş olması, ilk başta var olmasından daha önemli addedilir. Bir okuyucunun yazdığı gibi, 'Emacs hataları' düzeltilerek statü elde edilebilir, fakat 'Richard Stallman hataları' düzeltilerek değil — ve Stallman'ı *eski* ve artık düzeltilmiş Emacs hataları için eleştirmek son derece ayıp addedilir.

Bu davranış, akademik dünya ile ters düşmektedir. Akademik dünyada, başkaları tarafından yapılan hatalı işleri acımasızca eleştirmek, şöhrete kavuşmanın önemli bir öğesidir. Hacker kültüründe böyle davranışlar tabuya girer — o kadar ki, bu davranışların eksikliğini farketmedim bile, ta ki bu makale yayınlandıktan neredeyse bir yıl sonra bir okuyucu yazana kadar!

Yeterliğe karşı saldırının tabu olması (akademik dünya ile paylaşılmayan), o dünyada da geçerli olan övünmeye karşı tabudan daha da ilginçtir, çünkü akademi ve hackerlık arasında iletişim ve destek mekanizmalarında yer alan bir farkla alakalandırılabilir.

Hacker kültürünün hediye ortamı soyut, kültürün iletişim kanalları hissi nüansları iletmekte yetersiz ve üyelerin yüzyüze kontakta bulunması kaideden çok istisnadır. Bunlar, çoğu diğer hediye kültürüne kıyasla, gürültüye karşı daha az tolerans gösterilmesine neden olmakta ve yeterliğe karşı saldırının tabu olmasını açıklamakta oldukça başarılı olmaktadır. Hackerların yeterliğine karşı ciddi saldırıların kabul edilebilir olması, kültürün şöhret oyununu kabul edilemez şekilde bozar.

Gürültüye gösterilen bu duyarlılık, hacker toplumunun kabile ihtiyaçlarından beklenen alçak gönüllülüğü de açıklamaktadır. Gürültüye karşı tabunun etkin olabilmesi için, toplumun ileri gelenleri de övünmeden uzak olmalıdır. [DC] (sayfa: 25)

Eğer kişi, başarılı bir projenin yöneticisi olmak arzu ediyorsa, sakın bir üslup işe yaramaktadır; kişinin toplumu iyi karar verebildiğine inandırması gerekmektedir, çünkü bir proje yöneticisinin çoğu işi başkalarının kodları hakkında karar vermek olacaktır. Eğer birisi kendi kodunun kalitesi hakkında doğru bir karar veremiyorsa veya davranışları projenin getirisini kendi inhisarına alacağı izlenimini veriyorsa, onun yönettiği projeye kim kod bağışında bulunur? Potansiyel katılımcılar proje liderlerinin, objektif olarak gerektiği zaman “Evet, bu benim çözümümünden daha iyi çalışıyor, bunu kullanacağım” diyebilecek olgunluk ve alçak gönüllülükte olmasını isterler.

Alçak gönüllü davranmanın başka bir sebebi de, açık kaynak kod dünyasında bir projenin 'bittiği' izleniminin genelde verilmek istenmemesinden kaynaklanır. Böyle bir izlenim, potansiyel bir katılımcının kendine ihtiyaç duyulmadığını düşünmesine yol açar. Katılımcı çekebilmek ve bu kod gelişim modelinin kaldırıcı etkisini en fazla hale getirmenin yolu, yazılımın durumu hakkında alçak gönüllü olmaktır. Eğer övünmek kod yolu ile yapılıyor ve “X, y, z'yi yapamıyor, o kadar iyi olmasa gerek” deniyorsa, genelde x, y ve z'yi yapan yamalar hızla gelir.

En son olarak, kişisel gözlemlerim, bazı lider hacker'ların alçak gönüllü tutumlarının bir şahsiyet kültüne konu olmaktan duydukları (ve mesnetsiz olmayan) korkunun sonucu olmasıdır. Linus Torvalds ve Larry Wall böyle kaçınma davranışlarının açık ve çok sayıda örneğini göstermektedirler. Bir gün, Larry Wall ile bir akşam yemeğinde, kendisine şöyle takıldım: “Buradaki alfa hacker sensin — hangi restorana gideceğimize karar verebilirsin”. Görünür şekilde titredi. Ve haklıydı; paylaşılan değerlerini, liderlerinin şahsiyetlerinden ayıramamak pek çok toplumu yok etmiştir, Larry ve Linus bundan bihaber olamazlar. Öte yandan, pek çok hacker, eğer itiraf edebilseler, Larry ile aynı problemin sahibi olmaktan zevk duyardı.

13. Şöhret Oyunu Modelinin Global Sonuçları

Şöhret oyunu analizinin, hemen aşikar olmayan bazı sonuçları daha vardır. Bunların çoğu, başarılı bir proje kurmaktan elde edilen prestij, mevcut bir projede işbirliği yapmaktan elde edilenden daha fazla olmasına dayanır. Ayrıca, çarpıcı yenilik getiren projeler, var olan yazılımlara 'ben de varım' türü ufak eklemeler yapmaktan daha fazla başarı getirir. Diğer taraftan, yazardan başka kimsenin anlamadığı veya ihtiyaç hissetmediği yazılımlar şöhret oyununda hiç bir işe yaramaz ve mevcut bir projeye katılımı bulunarak dikkat çekmek genelde yeni bir proje başlatıp buna kullanıcı bulmaya çalışmaktan daha kolaydır. Son olarak, boş bir nişi doldurmak, başarılı bir proje ile rekabete girmekten çok daha kolaydır.

Dolayısıyla komşular (en yakın rakip projeler) arasında optimal bir uzaklık vardır. Çok yakına gelirse, ürün 'ben de varım'dan öteye gidemeyecek, sınırlı değerde, kötü bir hediye olacaktır (mevcut bir projeye katkıda bulunmak bundan daha başarılı olur). Çok uzağa gidilirse, kimse eforun değerini anlamayacak, önemini kavrayamayacak ve ürünü kullanamayacaktır (yine, kötü bir hediye). Bu, noosferde iskan için fiziksel serhaddi andıran bir yerleşim oluşturmaktadır — rastgele değil, difüzyon limitli bir fraktal gibi. Projeler, serhaddin yakınında yer alan işlevsel boşlukları doldurmak için doğarlar (yeniliğin çekimi için bakınız [\[NO\]](#) (sayfa: 25)).

Bazı çok başarılı projeler 'kategori katili' olurlar; kimse bu projelerin yakınında meskun olmak istemez, çünkü hackerların dikkatini bu projelerden çekmek çok zor olacaktır. Dolayısıyla başka halde kendi projelerini başlatacak olan bazı kimseler bu büyük, başarılı projelere katkıda bulunurlar. Klasik 'kategori katili' örneği GNU Emacs'dir; değişik sürümleri tamamen programlanabilen metin işlemci kategorisini öyle doldurur ki 1980'lerden bu yana kimse tek kişilik proje sürecinin ötesine geçebilen bir proje başlatamamıştır. Bunun yerine hackerlar Emacs kipleri geliştirir.

Global olarak bakıldığında, bu iki tandans (boşluk doldurma ve kategori katilleri), proje başlangıçlarında genel hatları ile tahmin edilebilen bir trend oluşturmıştır. 1970'lerde var olan açık kaynak kodlarının çoğu örnekler ve oyuncaklardı. 1980'lerde gelişim araçları ve Internet araçlarına kayış oldu. 1990'larda hareket işletim sistemlerine geçti. Her etapta, bir önceki etabın sağladığı problemler ve imkanlar tükenmeye yüz tuttuğu zaman yeni ve daha zor bir problem kümesine geçildi.

Bu trend, yakın gelecek için bazı çıkarımları doğurmaktadır. 1998'in başında, Linux, 'açık kaynak kodlu işletim sistemi' kategorisi için bir katil gibi görünmektedir — rakip olabilecek işletim sistemlerini geliştirebilecek kişiler Linux aygıt sürücülerini ve uzantıları geliştirmektedir. Kültürün açık kaynak kodlu olmasını düşündüğü pek çok alt seviye araç artık yazılmıştır. Geriye ne kaldı?

Uygulamalar. 2000 yılı yaklaştıkça, açık kaynak kodlu yazılımların kalan son bakir bölgeye doğru uzanacağını tahmin etmek zor olmamaktadır — teknik kökenli olmayan kullanıcılar için yazılımlar. Bu konuda erken bir gösterge, son on yılda ticari uygulamalar için *şart* addedilen GUI bazlı arayüz kullanan Photoshop benzeri resim işlemcisi [GIMP](#)^(B29)'dir. Bir diğer belirti ise [KDE](#)^(B30) ve [GNOME](#)^(B31) gibi uygulama araçkiti projelerine gösterilen yaygın ilgidir.

Bu makaleye yorum yazan bir okuyucu, iskan benzetmesinin, hackerların neden Microsoft'un Internet protokollerini önce karmaşılaştırmak, sonra da kapalı hale getirmek için uyguladığı "kucakla, genişlet" politikasına hiddet duyduklarını açıklamaya yaradığından bahsetmektedir. Hacker kültür çoğu kapalı kodlu yazılımla bir arada yaşayabilir; Adobe Photoshop'un varlığı GIMP'i çok daha az çekici kılmamaktadır. Fakat Microsoft, bir protokolü emtia olmaktan [\[HDJ\]](#) (sayfa: 26) çıkarmakta başarılı olduğu ve yalnızca Microsoft programcılarının bu protokol için yazılım geliştirebilmesini sağladığı zaman tekeline genişletmekle yalnızca müşterilerine zarar vermekle kalmamaktadır. Microsoft, bu hareketi ile hackerlara iskan edilecek ve işlenecek noosferin hem miktarını, hem de kalitesini azaltmaktadır. Hackerların Microsoft'un stratejisinden "protokol kirliliği" diye bahsetmeleri doğaldır; tıpkı ekinlerini sulamak için yararlandığı nehre zehir katan birine tepki veren çiftçinin reaksiyonunu göstermektedirler!

Son olarak, şöhret oyunu analizi, sık tekrarlanan şu kuralı açıklar: kendinize hacker dediğiniz zaman hacker olunmaz, *başka hackerlar* size hacker dediği zaman hacker olunur. Bu gözle bakıldığı zaman bir 'hacker', hem teknik kabiliyeti olduğunu, hem de şöhret oyununun nasıl oynandığını bildiğini (katkılarda bulunarak) göstermiş olan kişidir. Bu hacker olma değerlendirmesi, genellikle bilinç ve kültür özümsemesinin bir değerlendirmesidir ve yalnızca kültürün zaten içinde olanlar tarafından yapılabilir.

14. Ne Kadar İyi Bir Hediye?

Hacker kültürünün katkıları değerlendirme ve karşılığında denklemin saygısı ile ödüllendirme sürecinde tutarlı bazı olgular göze çarpmaktadır. Aşağıdaki kuralları gözlemlemek oldukça kolaydır:

1. Eğer iddia edildiği ve bu yönde beklentilerimin oluşmasına yol açıldığı kadar iyi çalışmaz ise, kötüdür — ne kadar orjinal ve zekice olduğu önemli değildir.

‘İddia etmek’ ve ‘beklenti oluşmasına yol açmak’ sözlerine dikkat edin. Bu kural, mükemmeliyet için bir talep değildir; beta ve deneysel yazılımın hataları olması beklenir. Bu kural, kullanıcının, projenin durumu ve geliştiricilerin hakkında söyledikleri şeylerden doğru olarak riskleri tahmin edebilmesi talebidir.

Bu kural, açık kaynak kod yazılımların uzun zaman beta sürecinde kalması ve geliştiriciler yazılımın pek çok kötü sürpriz içermediğinden iyice emin olana kadar 1.0 sürüm numarasına bile çıkmamasını getirir. Kapalı kaynak kod dünyasında, 1.0 Sürüm, “Eğer tedbirli iseniz buna dokunmayın” manasına gelir; açık kaynak kod dünyasında ise “Geliştiriciler bu kod için şöhretlerini ortaya koyarlar” manasına.

2. Noosferi büyüten işler, zaten işlenmekte olan bir parçayı yineleyen işlerden daha değerlidir.

Bu kuralın safça bir açıklaması: *Orjinal işler, mevcut yazılımların işlevlerini yineleyen ürünlerden daha iyidir.* olurdu. Fakat aslında iş o kadar basit değildir. Mevcut *kapalı* yazılımların işlevlerini yinelemek, eğer bu ürünü ortaya koymakla kapalı bir protokol veya biçim kırılıyor ve bu alan ilk olarak açılıyorsa, orjinal bir ürün kadar değerli sayılır.

Örneğin şu anda açık kaynak kod dünyasındaki en prestijli projelerden birisi Samba’dır — Unix makinalarının Microsoft’un kapalı SMB dosya paylaşım protokolü için istemci veya sunucu olmalarını sağlayan yazılım. Burada yapılan iş hemen hiç yaratıcı değildir; daha ziyade tersine çözümleme detaylarını tam ve doğru olarak yapabilmek söz konusudur. Yine de, Samba grubunun üyeleri, Microsoft’un pek çok kullanıcı kitlesini kendi protokollerine kilitlemek ve noosferin büyük bir parçasını kordon altına almak girişimini bertaraf etmelerinden dolayı kahraman addedilirler.

3. Ana bir dağıtım tarafından dağıtıma alınan ürün, alınmayan üründen daha değerlidir. Bütün ana dağıtımlarda yer alan bir ürün en prestijlisidir.

Ana dağıtımlardan kastedilen yalnızca Red Hat, Debian, Caldera ve SuSE gibi büyük Linux dağıtımları değil, kendi başlarına şöhret sahibi olan ve bir kalite belgelemesi addedilen diğer dağıtıcılardır — örneğin BSD dağıtımları veya Free Software Foundation’un kaynak kod koleksiyonu.

4. Kullanmak, en iyi övme biçimidir ve kategori katilleri, ‘ben de varım’lardan daha iyidir.

Denklerin muhakemesi süreci başkalarının muhakeme kabiliyetine güvenmeyi gerektirir. Bu, kimsenin mümkün olan bütün alternatifleri teker teker denemeye vakti olmadığından dolayı elzemdir. Dolayısıyla pek çok kişi tarafından kullanılan ürünler, bir kaç kişi tarafından kullanılan ürünlerden daha üstün sayılır.

Dolayısıyla kimsenin alternatiflerini kullanmak istemeyeceği kadar iyi ürün çıkarmak büyük bir prestij kazandırır. Denklerin en büyük saygısı, bütün dağıtımlar tarafından dağıtılan, son derece popüler, kategori katili orjinal eserlere verilir. Bunu bir defadan fazla başarmış olan kişilere yarı ciddi olarak ‘yarı-tanrı’ denir.

5. Zor, can sıkıcı işlere tutarlı bir şekilde devam etmek (hata ayıklamak, belge hazırlamak gibi), kolay ve zevkli kod oyunları ile uğraşmaktan daha fazla takdire layık görülür.

Bu norm, topluluğun normalde yapmak istemeyeceği işlerin yapılmasını sağlamak için ödüllendirme uygulaması mekanizmasıdır. Belli bir noktaya kadar aşağıdaki kural ile çalışmaktadır:

6. Çok basit olmayan işlev eklemeleri, düşük seviyeli yamalar ve hata ayıklamaktan daha iyidir.

Bu kuralın işleme biçimi: Eğer hata takdire şayan gariplikte ve zorlukta olup, hatayı bulmak ve çözümlmek işleminin kendisi bir zeka ve az rastlanan kabiliyet becerisi değil ise, bir işlev eklemek, bir hata ayıklamaktan daha değerlidir. Ama bu davranışlar uzun zaman devam ettirildiği zaman, uzun zaman boyunca tutarlı bir şekilde

normal ve kolay hataları ayıklayan birisi, aynı zamanı kolay işlevler eklemekle geçirmiş birisinden daha üstün addedilir.

Bir okuyucum bu kuralların birbirleriyle ilginç şekilde etkileşimde bulunduğu ve her zaman en fazla verimi ödülendirmede olduğu gözleminde bulundu. Bir hackera, tamamen kendi ürünü bir araç için mi, yoksa başka birisinin aracına bir ekleme için mi daha fazla şöhret kazanacağını sorun. Alacağınız “yeni araç” cevabı kesin olacaktır. Fakat:

(a) yalnızca görünmez şekilde günde bir kaç sefer işletim sistemi tarafından kullanılan, fakat kategori katili olan yepyeni bir araç

veya

(b) mevcut bir araca, kategori katili olmayan, çok ilginç de olmayan fakat çok sayıda kullanıcı tarafından her gün kullanılan ve her gün kullanıldığı farkedilen bir kaç ekleme

için aynı soruyu sorun ve cevabın (a) olarak verilmesinden önce hackerın yaşadığı tereddüdü seyredin. Bu alternatifler yaklaşık aynı değere sahiptir.

Aynı okuyucu bu soru için şu açıklamalarda bulundu: “(a) şıkkı fetchmail’dir; (b) şıkkı ise yapmış olduğunuz pek çok Emacs eklentisidir, örneğin vc.el ve gud.el.” Ve haklıydı, her ne kadar ikincisi zaman içerisinde daha fazla kullanılmış olsa bile, kamu önünde, ağırlıklı olarak ‘fetchmail yazarı’ olarak biliniyorum, ‘bir sürü Emacs kipi yazarı’ olarak değil.

Burada işleyen mekanizma, yeni bir ‘marka kimliği’ne sahip bir eserin mevcut bir ‘marka’ya katılan eklemelerden daha fazla farkedilmesi olabilir. Bu kuralların daha berrak hale getirilmesi ve hacker kültürünün değerlendirme sistemi hakkında ne söylediklerinin anlaşılması daha detaylı bir inceleme konusu olabilir.

15. Noosferin Özellikleri ve Mekan Sahiplenmesi

Locke mülkiyet geleneklerinin sebepleri ve sonuçlarını anlamak için onlara bir açıdan daha bakmak yararlı olacaktır; hayvanlar aleminde özellikle mekan sahiplenme alışkanlıkları.

Mülk, aynı cins arasında şiddeti azaltmak için evrimle ortaya çıkmış olan hayvani mekan sahiplenme güdüsünün soyutlanmış halidir. Bir kurt, mekanının sınırlarını işaretleyerek ve başkalarının sınırlarını ihlal etmeyerek kendisini zayıflatacak veya öldürecek ve çoğalmasına olumsuz etkide bulunacak bir kavgaya girme ihtimalini azaltır. Benzer şekilde, insan toplumlarında mülk, insanlararası çatışmaları, araya barışçıl davranışlarla şiddeti kesin çizgilerle ayıran hudutlar koyarak engeller.

Bazı kesimlerde mülkü tamamen sosyal bir alışkanlık olarak niteleme modası vardır, fakat bu tamamen yanlıştır. Bir köpek sahibi olmuş olan ve sahibinin mülküne yaklaşan yabancıları görünce havlamasını gözlemlemiş olan herhangi birisi, hayvan mekan sahiplenmesi ile insan mülkiyeti arasındaki ilişkiyi birebir görmüştür. Evcilleşmiş kurt kuzenlerimiz, mülkün sosyal bir alışkanlık olmadığını, şiddeti engellemek için evrimle ortaya çıkmış, kritik derecede önemli bir mekanizma olduğunu içgüdüsel olarak bilirler. (bu, onları pek çok siyasi teorisyenden daha akıllı yapar.)

Mülk sahiplenme (tıpkı mekan işaretleme gibi), aksiyon gösterilen bir işlemdir, hangi hudutların savunulacağını belirler. Mülkiyet haklarının toplum tarafından desteklenmesi, işbirlikçi davranışların maksimize edilmesi ve sürtüşmelerin en aza indirilmesi için bir yoldur. Bu gerçekler, “mülk sahiplenme”nin bir çitten, bir köpeğin havlamasından daha soyut olduğu, örneğin bir README dosyasında bir proje liderinin adından ibaret olduğu durumlarda bile geçerlidir. Bu, mekan sahiplenmenin soyutlanmış bir halidir (ve diğer mülkiyet biçimlerinde olduğu gibi), sürtüşmeleri engellemek için evrimle oluşmuş mekan sahiplenme içgüdülerine dayanır.

Bu analiz ilk başta fazla soyut ve hacker davranışlarını izahtan uzak görünebilir. Fakat, önemli bazı çıkarımları bulunmaktadır. Bir tanesi web sitelerinin popülerliğini ve bir web sitesine sahip olan açık kaynak kod projelerinin neden diğerlerine nazaran çok daha ‘gerçek’ göründüğünü anlatmaktadır.

Objektif olarak bakıldığında, bu olguyu izah zordur. Ufak bir yazılımı ortaya çıkarmak ve güncel tutmak için gereken efor ile karşılaştırıldığında, bir web sitesi kolaydır, dolayısıyla bir web sitesini bir gerçeklik göstergesi veya özel bir efor olarak saymak zordur.

Webin işlevsel özellikleri de bu olguyu açıklamakta yeterli olmamaktadır. Bir web sayfasının iletişim görevi, bir FTP sitesi, bir posta listesi ve USENET yayınları tarafından rahatlıkla üstlenilebilir. Hatta, bir projenin rutin iletişiminin bir posta listesi ya da haber grubu yerine web üzerinden yapılmasına pek rastlanmaz. Peki, o zaman web siteleri neden proje mekanları olarak popülerdir?

‘Ev sayfası’ deyiminde yer alan metaforun kendisi önemli bir ipucu vermektedir. Bir açık kaynak kod projesi kurmak noosferde bir mülk tescili olsa bile (ve o şekilde kabul edildiği halde), psikolojik açıdan zayıf kalmaktadır. Ne de olsa yazılımların doğal bir mekanı yoktur ve kolaylıkla yeniden oluşturulabilirler. ‘Mekan’ ve ‘mülk’ içgüdülerimize ancak çaba gösterilerek uydurulabilirler.

Bir proje ev sayfası, yazılım uzayında iskan tescilini, daha somut olan web dünyasında bir ‘ev’ mekanı halinde somutlaştırmaktadır. Noosferden ‘siber–uzaya’ inmek bizi gerçek dünyanın çitleri ve köpek havlamalarına ulaştırmamaktadır, fakat soyut mülkiyet iddiasını bizim içgüdüsel mekan bilincimizle daha sıkıca ilişkilendirmektedir. Bu nedenle de web sitesi sahibi olan projeler daha ‘gerçek’ görünmektedir.

Bu nokta, linkler ve iyi arama motorlarının varlığı ile daha da kuvvetlendirilmektedir. Web sitesi sahibi bir proje, noosferde civarı keşfe çıkan birisi tarafından daha rahatlıkla bulunabilir; başkaları ona link verecektir, arama motorları onu bulacaktır. Dolayısıyla bir web sayfası daha iyi bir reklam, daha efektif bir işaretleme mekanizması ve mülk üzerinde daha belirgin bir hak iddiasıdır.

Bu analiz, açık kaynak kod kültüründe çatışmaları çözümleme mekanizmalarına daha dikkatle bakmamızı getirmektedir. Şöhretin arttırılmasına yönelik teşviklerin yanı sıra, mülkiyet geleneklerinin çatışmayı engelleyici ve çözümleyici rolü de olması gerektiğini göstermektedir.

16. Çatışma Sebepleri

Açık kaynak kod etrafında yaşanan çatışmalarda dört ana tür sayabiliriz:

- Bir proje hakkında kalıcı kararları kim verebilir?
- Kim, ne için suçlanır veya hakkı teslim edilir?
- Çabanın tekrarlanması ve hata ayıklamanın korsan sürümler tarafından zorlaştırılması nasıl en aza indirgenir?
- Teknik olarak bakıldığı zaman Doğru Olan nedir?

“Doğru Olan Nedir” sorusuna tekrar bakarsak, kaybolduğunu görürüz. Bu tür her sorunun ya herkes tarafından kabul edilebilir objektif bir cevabı vardır ya da yoktur. Eğer varsa, herkes kazanır. Eğer yoksa, “kim karar verecek” sorusuna dönüşür.

Bir çatışma çözümleme teorisinin bir proje için cevaplaması gereken üç problem (A) dizayn kararlarında kimin belirleyici olduğu (B) hangi katılımcıların, hangi katkılarına nasıl kredi verildiği, (C) bir proje grubu ve ürününün değişik kollara ayrılmasının nasıl engellendiği olmaktadır.

(A) ve (C) problemlerinin çözümünde mülkiyet geleneklerinin oynadığı rol aşikardır. Gelenek, proje sahiplerinin belirleyici kararları aldığını söyler. Daha önce de geleneğin mülkiyetin çatallanma yolu ile zayıflatılmasına karşı geldiğini göstermiştik.

Bu geleneklerin şöhret oyunu unutulsa ve hacker kültürü tamamen bir zanaatkarlık modeli doğrultusunda incelense dahi anlamlı kaldığını gözlemlemekte fayda vardır. Bu görüşe göre bu gelenekler şöhret ödülleri

paylaşılarak azalmasından daha çok zanaatkarın kendi vizyonunu kendi seçtiği yöneme göre uygulama hakkı ile ilgilidir.

Fakat zanaatkarlık modeli (B) problemi ile ilgili bir çözüm getirememektedir — kim, ne için kredi alacaktır (çünkü, şöhrat oyunu ile hiç ilgilenmeyen bir zanaatkar için buna aldırış etmek gereksizdir). Bunları incelemek için Locke teorisini bir adım daha ileriye götürmek ve çatışmaları ve proje *içinde* ve projeler *arasında* mülkiyet haklarının uygulamalarına bakmak gerekmektedir.

17. Proje Yapıları ve Mülkiyet

En basit durum, bir projenin tek bir sahibi/lideri olduğu durumdur. Bu durumda bir çatışma olması ihtimali yoktur. Sahip, bütün kararları verir, bütün krediyi alır veya bütün suçlamaların muhatabı olur. Tek çatışma mevzuu, yerine kimin geçeceğine dair karar verilmesi esnasında olabilir — eğer eski sahip kaybolur veya ilgisini kaybederse yerine kimin geçeceği. Topluluk, (C) problemi uyarınca, ayrıca çatallanmayı önlemek arzusundadır. Bu arzular, sahip/liderin, artık proje liderliğini yerine getiremediği durumlarda kamu önünde yerine geçecek olana bayrak teslimi yapmasına yönelik bir kültürel norm doğurur.

Bir aşama daha kompleks olan durum, tek bir ‘iyiniyetli diktatör’ önderliğinde çalışan çok sayıda proje yöneticisi olmasıdır. Alışkanlıklar grup projelerinin bu şekilde organize olmasına yöneliktir; bu yapı Linux çekirdeği veya Emacs kadar büyük projeler için de kullanılmakta ve ‘karar veren kim’ problemini diğer alternatiflere göre bariz bir kötülüğü olmayan bir şekilde çözümlenmektedir.

Genelde, iyi niyetli diktatör yapıları tek sahipli yapılardan, kurucu lider yeni katılımcılar çektikçe doğal evrimle oluşur. İlk sahip, diktatör olarak kalsa bile projenin hangi kısımları için kime kredi verileceği konusunda yeni tartışma noktaları oluşabilir.

Bu durumda, gelenekler sahip/diktatörün katılımcılara adilane bir şekilde kredi vermesini (örneğin README dosyaları veya proje tarihçesinde) gerekli kılar. Locke mülkiyet modeline göre bu, bir projeye katkıda bulunarak o projenin şöhrat getirisine (pozitif veya negatif) ortak olunma hakkının elde edilmesidir.

Bu mantığı devam ettirirsek, ‘iyiniyetli diktatör’ün aslında kayıtsız şartsız olarak projesinin sahibi olmadığını görürüz. Kalıcı kararlar alma hakkı olsa dahi, başkalarının katkısı karşılığında toplam şöhrat getirisini katılımcıları ile paylaşmaktadır. Bir çiftlikte ‘yarıcı’ olmak benzetmesinden neredeyse kaçınılamaz, fakat burada katılımcının ismi kredi listesinde yer almaya devam etmekte ve katılımcı artık aktif olmasa bile ‘getiri’ sağlamaya devam etmektedir.

İyiniyetli diktatör projeleri daha fazla katılımcı eklemeye devam ettikçe iki sınıf katılımcı oluşturmaya başlarlar; normal katılımcılar ve eş–geliştiriciler. Eş–geliştirici olmaya giden tipik bir yol, projenin önemli bir alt sistemi için sorumluluğu almaktır. Bir diğeri, çok sayıda hatayı bularak ve ayıklayarak ‘yüksek hata ayıklayıcı’ olmaktır. Bu ve benzer şekillerde oluşan eş–geliştiriciler, projeye önemli ve devamlı bir zaman yatırımında bulunan kişilerdir.

Altsistem–sahibi rolü analizimiz için önemlidir ve daha dikkatle incelenmelidir. Hackerlar, ‘yetki, sorumluluğu takip eder’ demekten hoşlanırlar. Bir altsistemin bakım sorumluluğunu üstlenen bir eş–geliştirici, genellikle hem o altsistemin iç dizaynı ile ilgili hem de projenin geri kalan kısmı ile iletişim için kullanacağı arayüzler ile ilgili kararları, yalnızca (proje mimarı sıfatı ile) proje liderinin düzeltmelerini uygulamak ve kabul etmek koşulu ile verebilir. Bu kuralın, pratikte, bir proje içerisinde Locke modeline uygun ayrılmış mekanlar oluşturduğunu ve bütün diğer mekan sınırları ile aynı çatışma engelleyici işlevi yerine getirdiğini gözlemliyoruz.

Gelenek olarak eş–geliştiricileri olan bir ‘diktatör’ ya da proje liderinin önemli kararlarda eş–geliştiricilerinin fikrini alması beklenir. Bu, özellikle alınacak karar bir eş–geliştiricinin ‘sahip’ olduğu (yani zaman ayırdığı ve sorumluluk üstlendiği) bir altsistemi ilgilendiriyorsa titizlikle uygulanır. Akıllı bir lider, projenin iç mekan sınırlarının işlevini kavrar ve altsistem sahiplerinin almış olduğu kararlara çok gerekmedikçe karışmaz veya değiştirmez.

Bazı çok büyük projeler ‘iyi niyetli diktatör’ modelini tamamen bırakırlar. Bunu uygulamanın bir yolu, eş–geliştiricileri oy hakkı bulunan bir komite haline getirmektir (Apache’de olduğu gibi). Bir başkası, kıdemli eş–

geliştiriciler arasında sırayla kontrol devrinin yapıldığı dönüşümlü diktatörlüktür; Perl geliştiricileri kendilerini böyle yapılandırmışlardır.

Böyle karmaşık yapılar genel olarak dengesiz ve zor kabul edilir. Bu düşüncenin genelde bilinen 'komite yolu ile dizayn' güçlüklerinden ve komitelerin kendilerinden kaynaklandığı aşıkardır; bu problemler hacker kültürü tarafından yakından bilinir. Fakat, hackerların komite veya dönüşüm yapılarından duyduğu rahatsızlığın bir kısmının, hackerların daha basit yapılar üzerinde değerlendirme yaparken kullandıkları bilinç altı Locke modeline bu yapıları uydurmakta yaşadığı zorluktan kaynaklandığını düşünüyorum. Bu karmaşık yapılarda kontrol amaçlı mülkiyetin ve şöhret getirilerinin sahipliğinin muhasebesini yapmak güçtür. İç sınırların ne olduğunu görmek zordur ve dolayısıyla eğer grup normalin üzerinde bir armoni ve güven içerisinde değil ise çatışmayı engellemek zordur.

18. Çatışma ve Çatışma Çözülmesi

Projelerin içinde alınan rollerin karmaşıklığının dizayn yetkisi ve kısmi mülkiyet hakları ile ifade edildiğini gözlemledik. Bu metod, getirileri dağıtmanın verimli bir yolu olsa da, proje liderinin yetkilerini sulandırmaktadır — daha da önemlisi liderin potansiyel çatışmaları bastırma yetkisini azaltmaktadır.

İlk bakışta dizayn hususunda teknik tartışmaların iç çatışma konusu olmak için başlıca sebep olacağı düşünülse dahi, gerçekte bu hususta çatışmalara çok az rastlanır. Bu çatışmalar yetkinin sorumluluğu takip ettiği kuralına dayanarak genelde kolayca çözülür.

Çatışma çözümlemenin başka bir yolu kıdemdir — eğer iki geliştirici veya iki geliştirici grup bir anlaşmazlık içerisinde olur, anlaşmazlık objektif olarak çözülemez ve anlaşmazlık konusunun mülkiyeti ikisinde de değil ise, projenin geneline en fazla katkıda bulunmuş olan taraf kazanır (yani, projenin genelinde en fazla mülkiyet hakkı bulunan taraf).

(Eşit olarak, daha az katkısı bulunan taraf kaybeder. Enteresan olan, bu kuralın pek çok veritabanının kilitlenmeyi engellemek için kullandığı kural olmasıdır. Veritabanlarında iki iplik aynı kaynak için kilitlendiği zaman mevcut işlemde en az vakit harcamış olan iplik kilitlenme mağduru olarak seçilir ve yok edilir. Bu işlem genelde en uzun işleme vakti alan işlemi — yani en kıdemliyi seçer.)

Bu kurallar çıkan çoğu anlaşmazlıkları çözmek için yeterlidir. Bu kurallarla bir sonuca ulaşılamazsa, proje liderinin kararı genelde yeterli olur. Bu iki filtrede de çözülmemeyen anlaşmazlıklara pek sık rastlanmaz.

Anlaşmazlıklar, bu iki kural ("yetki sorumluluğu takip eder" ve "kıdem kazanır") farklı yönlerle işaret etmez ve proje liderinin yetkisi zayıf veya yok olmaz ise genelde ciddi bir hal almaz. Bu tarz bir anlaşmazlık için en yaygın örnek, proje liderinin yok olmasından sonra yaşanabilecek bir devir teslim kavgasıdır. Bu şekilde bir kavga içerisinde bulundum. Çirkin, acı veren, uzun ve yalnızca bütün taraflar kontrolü dışarıdan birisine verecek kadar yorulunca biten bir kavga idi ve bir daha bu şekilde bir şeyin içerisinde bulunmamak için dua ediyorum.

Günün sonunda, bütün bu çatışma çözümleyici mekanizmalar genel hacker topluluğunun bu kurallara uyma isteğine dayanırlar. Mevcut olan tek yaptırım mekanizması ciddi tartışmalara girmek veya yok saymaktır — geleneğe karşı gelenlere kamu önünde tepki verilmesi ve daha sonra kendileri ile işbirliğine girmenin reddedilmesi.

19. Kültüre Giriş Mekanizmaları ve Akademi ile Bağlar

Bu makalenin ilk sürümlerinden birisi şu araştırma sorusunu sormuştu: Topluluk, üyelerine gelenekleri ile ilgili nasıl bilgi ve eğitim verir? Gelenekler yarı bilinçli bir şekilde kendi başına anlaşılabilir ve uygulanabilir halde midir, başkalarının davranışları örnek alınarak mı öğrenilir, açık açık öğretilerek mi edinilir?

Açık açık öğretmeye, bugüne kadar kültürün normlarını izah eden açıklamalar pek mevcut olmadığından dolayı pek sık rastlanmaz.

Pek çok norm, davranış örnekleme ile öğretilir. Basit bir örnek, her yazılım dağıtımının içerisinde README veya README adında bir dosya bulunması ve dağıtımı incelerken gerekecek ilk bilgilerin burada bulunması alışkanlığıdır. Bu alışkanlık 1980'lerin başından beri mevcut ve yaygındır, hatta bazen yazıda bahsedildiği bile olmuştur. Fakat bu alışkanlık genelde pek çok dağıtıma bakılarak elde edilir.

Bazı hacker gelenekleri ise, şöhret oyununun temel (belki de bilinçaltı) bilgisi edinildikten sonra kendi kendine uygulanabilir şekildedir. Pek çok hackera bu makalede bahsedilen üç tabunun öğretilmesi gerekmemiştir veya bu tabuların aşikar mı olduğu yoksa öğrenilmesi mi gerektiği sorusuna aşikar olduğu cevabını verirler. Bu olgu daha yakından incelenmelidir — ve belki cevabını hackerların kültür hakkında nasıl bilgi edindikleri konusunu incelerken bulabiliriz.

Pek çok kültür, kültüre giriş mekanizması olarak gizli ipuçları (veya dini/mistik anlayışta 'sırlar') kullanır. Bu sırlar yabancılara açıklanmaz fakat yeni başlayanların kendi kendisine bulması beklenir. Kabul edilebilmek için katılımcının sırrı anladığını ve bunu kültürün kabul ettiği şekilde öğrendiğini göstermesi gerekir.

Hacker kültürü bu tarz ipucu ve testleri yoğunlukla ve bilinçli olarak kullanır. Bu süreci en az üç seviyede görebiliriz:

- Parola benzeri spesifik sırlar. Bir örnek olarak alt.sysadmin.recovery USENET haber grubu gösterilebilir. Bu haber grubuna haber yazabilmek için belli bir sır vardır, bu sırrı bilmeksizin haber yollayamazsınız ve sırrı bilmek, haber yollamaya yetkin olduğunuzun göstergesidir. Bu grubun üyeleri bu sırrı ifşa etmeye karşı kuvvetli bir tabuya sahiptirler.
- Bazı teknik sırları haiz olabilmek için el verilmesi gereği. Bir katılımcının değerli bir hediye verebilmesi için oldukça yüklü miktarda teknik bilgiye (örn: en azından ana bilgisayar dillerinin birisinin bilinmesi) sahip olması gerekir. Bu gereklilik, gizli sırların küçük ölçekte yaptığını büyük ölçekte yapar ve kültür içinde etkin olabilmek için gereken kalite standardını belirler (örn. soyut düşünme yetisi, sebat ve esnek zeka).
- Sosyal bağlam sırları. Kişi, kültüre belirli projelere katılarak girer. Her proje, potansiyel katılımcının etkin olabilmesi için teknik ve sosyal yapısını anlaması gerekeceği canlı bir hacker sosyal grubudur. (Somut olarak, bu genelde projenin Web sitesini ve/veya eposta arşivlerini okuyarak yapılır.) Yeni katılımcı, bu proje grupları yolu ile kıdemli hackerların davranış örnekleme gözetimlerini.

Bu sırları edinme süreci sonunda, potansiyel hacker, (bir müddet sonra) üç tabuyu ve diğer gelenekleri 'aşikar' kılan bağlamsal bilgiyi edinir.

Hacker hediye kültürünün kendisinin kendi ana sırrını oluşturduğu iddia edilebilir. Kültüre kabul işlemi (somut olarak: kimse sizden hacker olarak bahsetmez) şöhret oyununun içselleştirilmiş bir anlayışının ve bunun getirdiği gelenek, tabu ve kullanımların edinildiği gösterilmeden yapılmaz. Fakat bu herşey için geçerlidir; bütün kültürler kendilerine katılanlardan kendilerini anlamasını isterler. Dahası, hacker kültürü kendi iç mantığı ve adetlerinin gizli tutulması konusunda bir arzu göstermemektedir — en azından bunları ifşa ettiği için kimse eleştirilmemiştir!

Bu makaleye cevap yazan pek çok kişi hacker mülkiyet geleneklerinin akademik dünya, özellikle de bilimsel araştırma dünyası uygulamalarıyla benzerlik taşıdığı (ve onlardan kaynaklanmış olabileceği) gözleminde bulunmuştur. Bu araştırma toplumu potansiyel olarak verim getirebilecek olan fikirlerden oluşan bölgeyi araştırmakta benzer problemler yaşar ve bu probleme getirdiği çözümler özellikle denklemlerin değerlendirilmesi ve şöhret konularında önemli benzerlikler taşır.

Pek çok hackerın akademik dünya ile formatif yıllarında etkileşimi olduğu (üniversite yıllarında hackerlığa başlanmasına sık rastlanır) gözönüne alınırsa, akademik dünyanın hacker kültürü ile uygulamada benzerlikler arzemesi bu geleneklerin nasıl uygulandığını anlamak konusunda incelemeye değer bir noktadır.

Hacker 'hediye kültürü' ile benzerliklere akademik dünyada sık rastlanır. Bir araştırmacı tenür sahibi olduktan sonra yaşamsal meseleler konusunda düşünme ihtiyacı ortadan kalkar. (Tenür kavramının kendisinin, "filozofların" esas olarak vakitlerini araştırmaya hasredebilecek kaynakları olan zengin asiller olduğu daha eski bir

hediye kültürüne dayanması muhtemeldir.) Yaşam kavgasının olmadığı bir durumda şöhret artırımı belirleyici hedef haline gelir ve akademik yayınlar ve diğer ortamlar vasıtası ile yeni fikirlerin ve araştırmanın paylaşılmasını teşvik eder. Bu olgu objektif açıdan işlevseldir çünkü bilimsel araştırma, tıpkı hacker kültürü gibi, 'devlerin omuzlarında yükselmeye' dayanır ve temel prensiplerin tekrar tekrar yeniden bulunmaması gerektiğini savunur.

Bazıları hacker geleneklerinin, araştırma toplumunun geleneklerinin bir yansımasından ibaret olduğunu savunmuş ve bu geleneklerin (genelde) bireysel hackerlar tarafından araştırma toplumunda edinildiğini söylemişlerdir. Bu, sadece hacker geleneklerinin zeki lise öğrencileri tarafından bile öğrenilebildiğini gözlemlemekle ispatlanabileceği gibi, bir abartma olsa gerektir!

20. Hediyein Takastan Üstün Gelmesi

Burada daha enteresan bir alternatif mevcuttur. Akademi ve hacker kültürünün ortak davranış biçimlerinin olmasını genetik bağlarına değil, yapmaya çalıştıkları şeyler için doğa kanunları ve insan içgüdüleri çerçevesinde en optimal yapıları evrimle elde etmelerine bağlıyorum. Tarihin verdiği hüküm, serbest pazar kapitalizminin ekonomik verimlilik için global olarak optimal bir yol olduğu yönündedir, belki de benzer bir şekilde şöhret oyunu hediye kültürü de yüksek kaliteli yaratıcı işler için global olarak optimaldir.

Bu teori için destek, sanat ve ödül arasındaki ilişkiyi inceleyen çok sayıda psikolojik etüdden gelmektedir. [GNU] (sayfa: 26). Bu etüdler, layık olduklarından daha az ilgi çekmiştir, belki de bunun sebebi onları savunanların bu etdlere gereğinden fazla önem atfederek genel anlamda serbest pazar ve fikri mülkiyete yönelik saldırılar için kullanmalarındandır. Bu etüdlerin sonuçları bazı tür mal azlığı ekonomisi ödülleri programcılar gibi yaratıcı çalışanların verimliliğini, tahminlerin aksine azalttığını göstermektedir.

Brandeis Üniversitesi'nden psikolog Theresa Amabile, 1984 tarihli motivasyon ve ödül konulu bir çalışmanın sonuçlarını özetlerken şunu demiştir: "Belki de sipariş üzerine yapılan eserler genel olarak salt kendi istekleri doğrultusunda yapılan eserlerden daha az yaratıcı olacaktır." Amabile şöyle devam etmektedir: "Çalışma ne kadar kompleks ise, harici ödüllerin varlığı o kadar zarar verici olmaktadır." Etüdün başka ilginç bir bulgusu da sabit maaşların motivasyonu azaltmadığı fakat parça başı ücretlendirme ve ikramiyelerin azalttığıdır.

Dolayısıyla hamburger pişiren veya kanal kazan kişilere performans ile alakandırılan ikramiye vermek ekonomik açıdan mantıklıdır, fakat büyük bir olasılıkla bir program üretim şirketinde maaşı performanstan ayırmak ve çalışanların kendi projelerini seçmelerine izin vermek daha iyi sonuç verecektir (bu iki trendi de açık kaynak dünyası nihai mantık sonucuna götürür). Bu sonuçlar, programcının performansını ödüllendirmenin ekonomik mantığının yalnızca programcının o işi herhangi bir ödül olmasa da yapacağı kadar motivasyona sahip olduğu durumda faydalı olduğunu göstermektedir!

Bu konudaki başka araştırmacılar meseleyi direkt olarak hackerları o kadar ilgilendiren otonomi ve yaratıcı kontrol üzerine getirmektedirler. Rochester Üniversitesi'nden psikoloji doçenti Richard Ryan, "Kişinin kendi kontrolüne sahip olduğu hissi azaldığı ölçüde kişinin yaratıcılığı da azalmaktadır" demektedir.

Genel olarak yapılacak olan herhangi bir şeyi, kendi başına bir hedef yerine başka bir şey için gerekli olan bir araç olarak tanımlamak motivasyon azalmasına sebep olmaktadır. Başkaları ile bir rekabetten galip çıkmak veya denkları tarafından saygı duyulmak bile iş için verilen ödül şeklinde görülürse motivasyon azaltmaktadır (bu bulgu, hackerların neden bu saygıyı direkt olarak aramadıkları veya bundan bahsetmedikleri konusunda aydınlatıcı olabilir).

Yönetim problemini daha da karmaşıklaştıran bir başka olgu da kontrol edici sözlü geri beslemenin neredeyse parça başı ücretlendirme kadar motivasyon azaltıcı olmasıdır. Ryan, şirket çalışanlarına "Güzel, yapman *gerektiği* gibi yapıyorsun" dediği zaman gayriresmi yollarda geri besleme alan çalışanlara kıyasla önemli derecede daha az motive olduklarını bulmuştur.

Bu bulgular karşısında yine de ödüllendirme politikası uygulamak iyi bir fikir olabilir, fakat ters sonuç doğurmaması için bunların herhangi bir şeye bağlı olmaması gerekmektedir. Ryan, "Bu ödülü sana eserinin

değerini anladığım için veriyorum” demek ile “Bu ödülü sana standartlarıma uygun performans gösterdiğin için veriyorum” demek arasında ciddi bir fark olduğunu gözlemlemiştir. Birincisi motivasyon azlığına yol açmaz, fakat ikincisi açmaktır.

Bu psikolojik gözlemlerden, bir grup açık kaynak kod geliştiricisinin (özellikle yaratıcılığın verimlilik artırıcı önemli bir faktör haline geldiği uzun vadede) aynı sayıda ve aynı yetenekte, fakat mal azlığı ekonomisi ödülleri ile motivasyonları azaltılan bir grup kapalı kaynak kod geliştiricisinden daha başarılı ve verimli olacağını çıkarabiliriz.

Bu çıkarım, *Katedral ve Pazar Yeri* makalesinde yer alan spekülasyonlardan birisini, biraz farklı bir açıdan çağrıştırmaktadır: Sonunda, endüstriyel/fabrika kipinde üretilen yazılımlar, kapitalizmin pek çok programcının mal–azlığı–sonrası bir hediye kültüründe yaşamasına imkan verecek bir refah düzeyi yaratmasından itibaren rekabeti kaybetmeye mahkumdur.

En yüksek yazılım verimliliği için reçete neredeyse bir Zen paradoksunu andırmaktadır; en verimli üretimi arzu ediyorsanız, programcıların üretmesini *sağlamaktan* vazgeçmelisiniz. Onların yaşam ihtiyaçlarını karşılamalı, kafalarını kendilerine bırakmalı ve tarih hedefleri vermekten vazgeçmelisiniz. Klasik bir yöneticiye bu şımarıkça ve kaybetmeye mahkum görünecektir — fakat şu anda açık kaynak kod dünyası rakibini *tam olarak* bu reçeteye duman etmektedir.

21. Sonuç: Gelenekten Yasaya

Açık kaynak kod yazılımlarının mülkiyet ve kontrolünü belirleyen gelenekleri inceledik. Bu geleneklerin Locke’nin toprak iskanı teorisi ile nasıl benzeştiğini gördük. Bunu, hacker kültürünü, üyelerinin prestij kazanmak için zaman, enerji ve yaratıcılıklarını dağıttığı bir ‘hediye kültürü’ olarak incelediğimiz analiz ile alakalandırdık. Kültürün içinde yer alan çatışma çözücü mekanizmaları inceledik.

Bundan sonraki soru "Bu niye gereklidir?" olacaktır. Hackerlar bu gelenekleri bilinçli bir analize tabi tutmaksızın geliştirdiler ve (şimdiye kadar) bilinçaltı güdülerini ile takip ettiler. Bu bilinçli analizin bize bir şey kazandırıp kazandırmadığı çok belirgin değildir — eğer bu tanımlamadan reçeteye gidemez ve bu geleneklerin işlerliğini daha iyileştirecek çıkarımlarda bulunamazsak.

Anglo–Amerikan töre geleneğindeki toprak iskanı teorisi ile hacker gelenekleri arasında yakın bir benzeşme bulduk. Tarihsel olarak [Miller], bu töreyi üreten Avrupa kabileleri çatışma çözümü mekanizmalarını yarı bilinçli, ifade edilmemiş gelenek sisteminden kabilenin ihtiyarları tarafından ezberlenen açık bir yasalar kümesine dönüştürerek — ve en sonunda bunları yazılı hale getirerek — iyileştirmişlerdir.

Belki nüfusumuz arttıkça ve bütün yeni üyelerin kültüre dahil edilmesi giderek zorlaştıkça hacker kültürünün benzer bir şey yapması gerekecektir — açık kod projeleri ile ilgili ortaya çıkabilecek olan çatışmaların çözülmesi için takip edilecek kuralların yazılı hale getirilmesi ve topluluğun kıdemli üyelerinin çatışmalarda arabuluculuk yapmasına yönelik bir arbitrasyon geleneği.

Bu makaledeki analiz böyle bir kurallar kümesinin anahatlarının ne olabileceğini göstermekte ve bugüne kadar ima edilenleri ifşa etmektedir. Böyle kurallar tepeden getirilemez; bireysel projelerin kurucuları veya sahipleri tarafından kabul edilmesi ve uygulanması gerekir. Böyle bir küme tamamen katı ve değişmez de olamaz, çünkü kültüre gelen baskılar zaman içerisinde değişecektir. Son olarak, böyle bir kurallar kümesinin işlerlik kazanabilmesi için hacker kabilesinin genel kabulünü kazanması gerekecektir.

Böyle bir kurallar kümesini oluşturmaya başladım, adına geçici olarak, yaşadığım küçük kasabaya atfen "Malvern Protokolü" diyorum. Eğer bu makaledeki genel analiz yeteri kadar yaygın kabul görülür ise Malvern Protokolünü çatışma çözümü için örnek bir kurallar kümesi olarak yayınlayacağım. Bu kuralları eleştirmek veya üzerinde çalışmak isteyen veya bunun iyi bir fikir olup olmadığı konusunda fikir belirtmek isteyenlerin benimle eposta <esr (at) thyrsus.com> yolu ile temasa geçmesini rica ediyorum.

22. Araştırma Soruları

Kültürün (ve benim), iyiniyetli diktatör modelini uygulamayan büyük projeler hakkındaki bilgisi zayıftır. Bu modelin dışında kalan çoğu proje başarısız olmaktadır. Bir kaç tanesi ise son derece değerli ve önemli olmayı başarmıştır (Perl, Apache, KDE). Kimse, aradaki farkın neden olduğunu bilmemektedir. Genellikle, bu tarz her projenin *nevi şahsına münhasır* olduğuna ve kendi üyelerinin şahsiyeti ile alakalı olarak ya başarısız ya da başarılı olduğuna dair, pek de berrak olmayan bir görüş hakimdir. Bu doğru mudur yoksa bir grubun takip edebileceği uygulanabilir stratejiler mevcut mudur?

23. Notlar

[N] ‘Noosfer’, felsefede az bilinen bir terimdir.

Daha detaylı olarak “İnsan düşüncesinin küresi” manasına gelen bu terim eski Yunanca’dan, akıl, ruh veya nefes manasına gelen ‘nous’ sözcüğünden, E. LeRoy tarafından türetilmiş ve ilk olarak *Les origines humaines et l’evolution de l’intelligence* (Paris 1928) içerisinde kullanılmıştır. Terime yaygınlık kazandıran kişiler Rus biyolog ve çığır açan ekoloğu Vladimir Ivanovich Vernadsky (1863–1945), sonra da Cizvit paleontolog/feylesof Pierre Teilhard de Chardin (1881–1955) olmuştur. Günümüzde terim, ağırlıklı olarak de Chardin’in gelecekte insan evriminin Tanrı ile birleşecek saf bir zeka halinde olacağına dair teorisi ile birlikte anılır.

[DF] David Friedman, modern ekonominin en sarıh ve kolay okunur düşünürlerinden birisidir. Yazmış olduğu fikri mülkiyet kanununun tarih ve mantığı ile ilgili [eser](#)^(B36) bu konulara ilgi duyan herkesin bir başlangıç noktası olarak tavsiye edilir.

[SP] Linux ve BSD dünyaları ile ilgili ilginç bir fark, Linux çekirdeğinin (ve onunla beraber çalışan OS araçlarının) hiç çatallanmamış olması, buna karşın BSD’lerin en az üç defa çatallanmasıdır. Bunu ilginç kılan şey ise BSD gruplarının sosyal yapısının merkezi olması ve özellikle yetkinin sınırlarının açık ve seçik olarak ortaya konarak çatallanmanın engellenmesi için tasarlanmış olması, buna karşın amorf ve merkezi olmayan Linux topluluğunun hiç bir zaman böyle tedbirler almamış olmasıdır. Genelde geliştirmeyi herkese açan projelerin çatallanma riski *en aza* inmektedir!

Henry Spencer <henry@spsystems.net> bir politik sistemin stabilitesinin sistemin politik sürecine giriş bariyerleri ile ters orantılı olduğunu söylemektedir. Analizinden burada bir alıntı yapmaya değer:

Genel olarak açık bir demokrasinin önemli bir kuvveti çoğu potansiyel ihtilalcinin, istedikleri sonuçlara sisteme saldırmak yerine sistem içerisinde çalışmakla daha kolay erişmeleridir. Bu kuvvet eğer kudret odakları birleşerek girişi zorlaştırırlar ve küçük, gayrimemnun grupların kendi hedeflerine yönelik *biraz* yol alındığını görmelerine imkan vermezlerse kolaylıkla kaybolmaktadır.

(Benzer bir prensip ekonomide de geçerlidir. Serbest pazarlar en sert rekabete sahiptir ve genelde en iyi ve en ucuz ürünleri üretirler. Bu sebepten dolayı, pazarda güçlü olan şirketlerin pazara girişi güçleştirmeleri kendi lehlerinedir — örneğin hükümete bilgisayarlara RFI testi getirmesi konusunda baskı kurmak veya büyük kaynaklara sahip olmaksızın uyulması mümkün olmayacak kadar karışık standartların oluşturulmasını sağlamak gibi. Pazara girişi en zor olan pazar segmentleri, ihtilalciler tarafından en şiddetle saldırılan bölgeler olur. Örneğin Internet ve Adalet Bakanlığı karşısında Bell System gibi.

Düşük giriş bariyerleri olan açık bir süreç ayrılmak yerine uzlaşmayı teşvik eder, çünkü ayrılmanın getireceği yüksek maliyete katlanılmaksızın sonuç alınabilir. Sonuçlar ayrılmakla elde edilen kadar parlak olmayabilir, fakat fiyatı daha düşük olacaktır ve çoğu kişi buna razı olacaktır. (İspanya hükümeti Franco’nun Basque karşıtı kanunlarını kaldırdığı ve Basque bölgelerine kendi okullarını ve sınırlı yerel otonomilerini verdiği zaman Basque Ayrılmacılarının çoğu kısa zamanda yok oldu. Yalnızca en sert Marxistler bunun yeterli olmadığını savundu.)

[RP] Korsan yamaların bazı incelikleri vardır. Bunları 'dostane' ve 'düşmanca' olarak ikiye ayırmak mümkündür. 'Dostane' bir yama projenin ana kaynak kodları ile proje liderinin kontrolü altında ileride bir gün birleştirilmek amacı ile yazılmıştır (bu birleşim hiç olmayabilir); 'düşmanca' bir yama, projeyi proje liderinin istemediği bir yöne çekmeye çalışır. Bazı projeler (özellikle de Linux çekirdeği) dostane yamalara oldukça sıcak yaklaşır ve hatta beta-test süreçlerinin bir parçası olarak bunları dağıtır. Düşmanca bir yama ise orijinal ile rekabet niyetini belirtir ve ciddi bir problemdir. Çok sayıda düşmanca yama bulunması genelde çatallanmaya götürür.

[LW] Michael Funk'a <mwfunk@uncc.campus.mci.net> korsan kültürü ile hacker kültürünü karşılaştırmanın ne kadar öğretici olduğunu gösterdiği için şükran borçluyum. Linus Walleij korsan kültürünün dinamiklerini benden farklı olarak inceleyen bir analiz yayınlamıştır (onları bir mal azlığı kültürü olarak alır): [A Comment on 'Warez D00dz' Culture](#)^(B37)

Kontrast kalıcı olmayabilir. Eski korsan Andrej Brandt <andy@pilgrim.cs.net.pl> korsan kültürünün ölmekte olduğunu ve en parlak üye ve liderlerinin açık kaynak kod kültürüne asimile olduklarını belirtmektedir. Bu görüşü destekleyen bağımsız bir bulgu, 1999 Temmuzunda kendisine 'Cult of the Dead Cow' diyen bir grubun Microsoft Windows güvenliğini bertaraf eden 'Back Orifice 2000' ürünlerini GPL altında yayınlaması olmuştur.

[HT] Evrimsel bir bakış açısıyla, zanaatkarın güdüsünün kendisi (tıpkı içselleşmiş ahlak gibi) aldatmanın getirdiği yüksek risk ve maliyetten kaynaklanabilir. Evrimci psikologlar insanların sosyal aldatmaları farketmek için geliştirdikleri beyin fonksiyonları olduğunu bulmuşlardır [BCT] ve atalarımızın kopya çekmeyi farkedendenler arasından seçilmiş olması da mantıklıdır. Dolayısıyla eğer avantaj veren fakat riskli veya maliyetli addedilen özellikler için şöret sahibi olmak isteniyorsa, bu özelliklere gerçekten sahip olmak, bunlara sahipmiş numarası yapmaktan daha iyi bir taktiktir. ("Dürüstlük en iyi politikadır")

Evrimsel psikologlar bu davranışın bar kavgaları gibi olayları açıkladığını iddia etmişlerdir. Genç erkeklerde 'sert' olmak şöreti hem sosyal hem de (bugünün feminist dünyasında bile) cinsel açıdan faydalıdır. 'Sertmiş' gibi yapmak son derece tehlikelidir; aldatmacanın anlaşılması, prestiji bu özelliğe hiç sahip olma iddiasında bulunulmaması halinden daha aşağıya getirir. Aldatmacanın maliyeti o kadar yüksektir ki, 'sertliği' içselleştirmek ve bir kavgada ciddi yara almak ihtimalini göze almak bundan daha iyidir. 'Dürüstlük' gibi daha kabul edilir özellikler için de benzer argümanlar getirilmiştir.

Yaratıcı çalışmanın meditasyon benzeri ana ödülü göz ardı edilmemelidir fakat zanaatkarın isteği en azından kısmen böyle bir iselleştirme sonucu (burada temel özellik 'dikkat isteyen iş yapma kabiliyeti' veya benzer bir şey olmalıdır) çıkmaktadır.

[MH] Maslow'un hiyerarşisi ve ilgili teorilerin kısa özeti Web'de bu adresten bulunabilir: [Maslow's Hierarchy of Needs](#)^(B39)

[DC] Fakat, liderlerden alçakgönüllülük talebinde bulunmak hediye veya bolluk kültürlerinin daha genel bir özelliği olabilir. David Christie <dc@netscape.com> Fiji'nin uç adalarına yapılmış bir yolculuktan bahsediyor: "Fiji köy şeflerinde aık kaynak proje liderlerine atfettiğiniz alçak gönüllü, yumuşak liderliği gördük. [...] Kendilerine çok saygı gösterilmesi ve Fiji'de olan bütün güce sahip olmalarına rağmen karşılaştığımız şefler gerçek bir alçak gönüllülük ve çoğu zaman azizleri anımsatan bir görev duygusu sergiliyorlardı. Bu, özellikle de şef olmanın nesilden nesile geçen bir görev olduğunu ve bir seçim veya popülerlik sonucu oluşmadığını düşünürsek ilginç. Şefler, seçilmeyip bu göreve doğmalarına rağmen bu davranışı sergilemek için bir şekilde kültürün kendisi tarafından eğitiliyorlar." Christie, Fiji şeflerinin bu davranış tarzını sergilemelerinin altında yatan sebebin işbirliğini sağlamadaki güçlükler olabileceği yorumunu getirmektedir. Bir şefin elinde kullanabileceği "büyük bir havuç veya büyük bir sopa" yoktur.

[NO] Gözlemlenebilir bir olgu, başarılı proje kuran kişilerin, başarılı bir projeye yardım ederek ve hata ayıklayarak eşit bir emek harcayan kişilerden daha fazla prestij sahibi olmalarıdır. Bu makalenin daha eski bir sürümü "Bu, kıyaslanabilir eforun rasyonel bir değerlendirmesi midir, yoksa burada varlığını ortaya koyduğumuz bilinçaltı mekan sahiplenme güdüsünün ikincil bir etkisi midir?" sorusunu sormakta idi. Bir kaç okuyucu benzer teorileri

ortaya koydular. Ryan Waldron <rew@erebor.com> tarafından gönderilen aşağıdaki analiz durumu açıklıkla ortaya koymaktadır:

Locke toprak teorisi bağlamında, yeni ve başarılı bir projeyi kuran, başkalarının iskan edebileceği yeni toprakları keşfetmiş veya açmış sayılır. Çoğu başarılı proje için azalan getiriler söz konusudur, dolayısıyla bir müddet sonra projeye katılımcılara verilen kredi o kadar yaygınlaşır ki katkıları ne kadar kaliteli olursa olsun sonradan gelen bir katılımcının edinebileceği şöhret olmaz.

Örneğin perl kodlarına ne kadar iyi bir katkıda bulunmalıyım ki katkılarımın karşılığında Larry, Tom, Randall ve diğerlerinin elde ettiği şöhretin pek azını dahi elde edebileyim?

Öte yandan eğer [başka birisi tarafından] yeni bir proje kurulursa ve ben bu projenin erken ve sık katılımcılarından birisi olursam, başarılı bir projenin yarattığı saygıdan pay alma ihtimalim erken katılımımdan dolayı artacaktır (bütün katılımların aynı değerde olduğu varsayılırsa). Bu olgu, Microsoft hisselerini erken alanlarla geç alanlar arasındaki ayrıma benzer. Herkes kar edebilir, ama erken alanlar daha çok kar eder. Dolayısıyla bir noktada yeni ve başarılı bir ilk halka arz, mevcut bir hisseyi almaktan daha fazla ilgimi çekecektir.

Ryan Waldron'un analojisi genişletilebilir. Projenin kurucusu başkaları için faydalı veya kabul edilebilir olup olmayacağı belli olmayan yeni bir fikri başkalarına satma misyonuna sahiptir. Dolayısıyla kurucu, birincil halka arza yakın bir risk alır (şöhretine gölge düşürme riski); kurucunun aldığı risk, denkler arasında belli bir kabul görmüş bir projeye yardım edenlerin aldığı riskten daha fazladır. Kurucunun getirisi de yardımcıları reel olarak daha fazla zaman ve emek sarfetseler dahi devamlı olacaktır. Bu da bir takas ekonomisindeki risk/getiri ilişkisi ile tamamen benzer durumdadır.

Başka okuyucular sinir sistemimizin farkları görmeye ayarlı olduğunu, değişmeyen durumları algılayamadığını söylediler. Dolayısıyla yeni bir projenin başlangıcının işaret ettiği devrimci değişiklik, sürekli küçük iyileştirmelerin kümülatif etkisinden daha büyük olarak algılanacaktır. Dolayısıyla binlerce başka katılımcının toplam iyileştirmesinin işletim sistemine olan etkisi bir kişinin yapabileceğini çok aşsa bile, Linus, Linux'un babası olarak bilinir.

[HD]'Emtialıktan çıkarmak' deyimini, Microsoft'un bu terimi açıkca müşterileri üzerinde sömürücü tekellerini devam ettirmek için kullandıkları uzun vadeli stratejileri için kullandığı [Halloween Belgeleri](#)^(B40)'ne bir atıftır.

[GNU] Free Software Foundation'un ana web sitesinde bu etüdlerin pek çoğunun sonucunu özetleyen [bir makale](#)^(B41) bulunmaktadır. Bu makalede yer alan alıntılar oradan yapılmıştır.

24. Teşekkür

Robert Lanphier'in <robla@real.com> egosuz davranış hakkındaki yorumlara pek çok katkısı oldu. Eric Kidd <eric.kidd@pobox.com> şahsiyet kültlerinin oluşmasını engellemekte alçakgönüllülüğe değer vermenin önemini vurguladı. Global etkiler hakkındaki bölüm, Daniel Burn'un <daniel@tsathoggua.lab.usyd.edu.au> yorumlarından ilham alınarak yazıldı. Mike Whitaker <mrw@entropic.co.uk> kültüre uyum sağlama bölümündeki ana argümanların ilhamını verdi. Chris Phoenix <cphoenix@best.com> hackerların başka hackerları aşağıya çekerek şöhret kazanamadıkları gözleminin önemini vurguladı.

Bu makalede yazılanlar için tek sorumlu benim ve bütün hatalar da bana aittir. Fakat yorum ve geri beslemeleri davet ettim ve bu makaleyi iyileştirmek için kullandım — bu sürecin süresiz devam etmesini bekliyorum.

A. Kaynakça

[Miller]

William Ian Miller -- *Bloodtaking and Peacemaking: Feud, Law, and Society in Saga Iceland* -- University of Chicago Press -- 1990 ISBN 0-226-52680-1 -- Bu ilginç etüd İzlanda törelerini inceler ve hem Locke mülkiyet teorisini aydınlatır hem de törelerin önce sözlü sonra da yazılı kanun oluş sürecini anlatır. --

[Mal]

Malaclypse the Younger -- *Principia Discordia, or How I Found Goddess and What I Did To Her When I Found Her* -- Loompanics -- ISBN 1-55950-040-9 -- Discordia hakkında aydınlatıcı bir sürü saçmalık. 'SNAFU prensibi'nde emir hiyerarşilerinin neden çok büyüyemediklerinin güzel bir analizi yer alıyor. Web'den okunabilir bir [HTML sürümü](#) ^(B42) mevcut. --

[BCT]

The adapted mind: Evolutionary psychology and the generation of culture. -- New York: 1992; Evrimci psikolojiye iyi bir giriş. Bazı makaleler burada bahsettiğimiz üç kültür kalıbı (emir/takas/hediye) ile ilgili ve bu kalıpların insan psikolojisine oldukça alt seviyelerde girmiş olduğuna işaret ediyor. -- Oxford University Press --

[MHG]

Michael K. Goldhaber -- *The Attention Economy and the Net* -- Bu makaleyi 1.7 sürümünü yayınladıktan sonra buldum. Bazı hatalar içeriyor (Goldhaber'in dikkat için iktisat açıklamasının geçerli olmayacağı iddiası daha yakın bir incelemeyi kaldırmaz) fakat Goldhaber'in yapılanma davranışında dikkat çekme güdüsü hakkında söyleyeceği komik ve derin şeyler mevcut. Bizim bahsettiğimiz prestij veya den-
klerin saygısı bu konuda dikkat çekmenin bir biçimi olarak düşünülebilir. --

[HH]

Hackerliğin Kısa Tarihçesi -- Hacker kültürünü [A Brief History Of Hackersdom](#) ^(B45) makalesinde özetledim. (Çevirisi: [Hackerliğin Kısa Tarihçesi](#) ^(B46)). Bu tarihi iyi anlatacak olan kitap daha yazılacak, muhtemelen benim tarafından değil. --

Notlar

- Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- 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.

^(B1) <http://www.catb.org/~esr/>

^(B2) <http://www.catb.org/~esr/writings/homesteading/homesteading/>

^(B3) <http://www.catb.org/~esr/writings/homesteading/homesteading.shtml>

^(B4) <http://www.arayan.com/da/yazi/esr/homesteading/>

^(B5) <http://www.arayan.com/da/yazi/esr/homesteading/homesteading.xml>

^(B6) <http://linux.okstation.com/raymond/homesteading/>

- (B7) <http://www.laisen.dk/opensource/homesteading.html>
-
- (B8) <http://www.linux-france.org/article/these/>
-
- (B9) http://www.phone-soft.com/raymondhomesteading/htn_g.0.html
-
- (B10) <http://www.apogeeonline.com/openpress/doc/homesteading.html>
-
- (B11) <http://www.post1.com/home/hiyoril3/freeware/noosphere.html>
-
- (B12) <http://www.geocities.com/jagem/noosfera.html>
-
- (B13) <http://www.opensource.org>
-
- (B14) <http://metalab.unc.edu/pub/Linux/welcome.html>
-
- (B15) <http://www.catb.org/~esr/writings/cathedral-bazaar>
-
- (B16) <http://www.opensource.org>
-
- (B29) <http://www.gimp.org>
-
- (B30) <http://www.kde.org>
-
- (B31) <http://www.gnome.org>
-
- (B36) http://www.best.com/~ddfr/Academic/Course_Pages/L_and_E_LS_98/Why_Is_Law/Why_Is_Law_Chapter_11.html
-
- (B37) http://www.df.lth.se/~triad/papers/Raymond_D00dz.html
-
- (B39) <http://www.valdosta.peachnet.edu/~whuitt/psy702/regsys/maslow.html>
-
- (B40) <http://www.opensource.org/halloween/>
-
- (B41) <http://www.gnu.org/philosophy/motivation.html>
-
- (B42) <http://www.cs.cmu.edu/~tilt/principia/>
-
- (B43) http://www.firstmonday.dk/issues/issue2_4/goldhaber
-
- (B44) [../howto/hacker-history.pdf](http://www.catb.org/~esr/faq/hacker-history.pdf)
-
- (B45) <http://www.catb.org/~esr/faq/hacker-hist.html>
-
- (B46) [../howto/hacker-history.pdf](http://www.catb.org/~esr/faq/hacker-history.pdf)
-

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

14 Şubat 2007