

penguence

Sayı 4 /Mart 2006

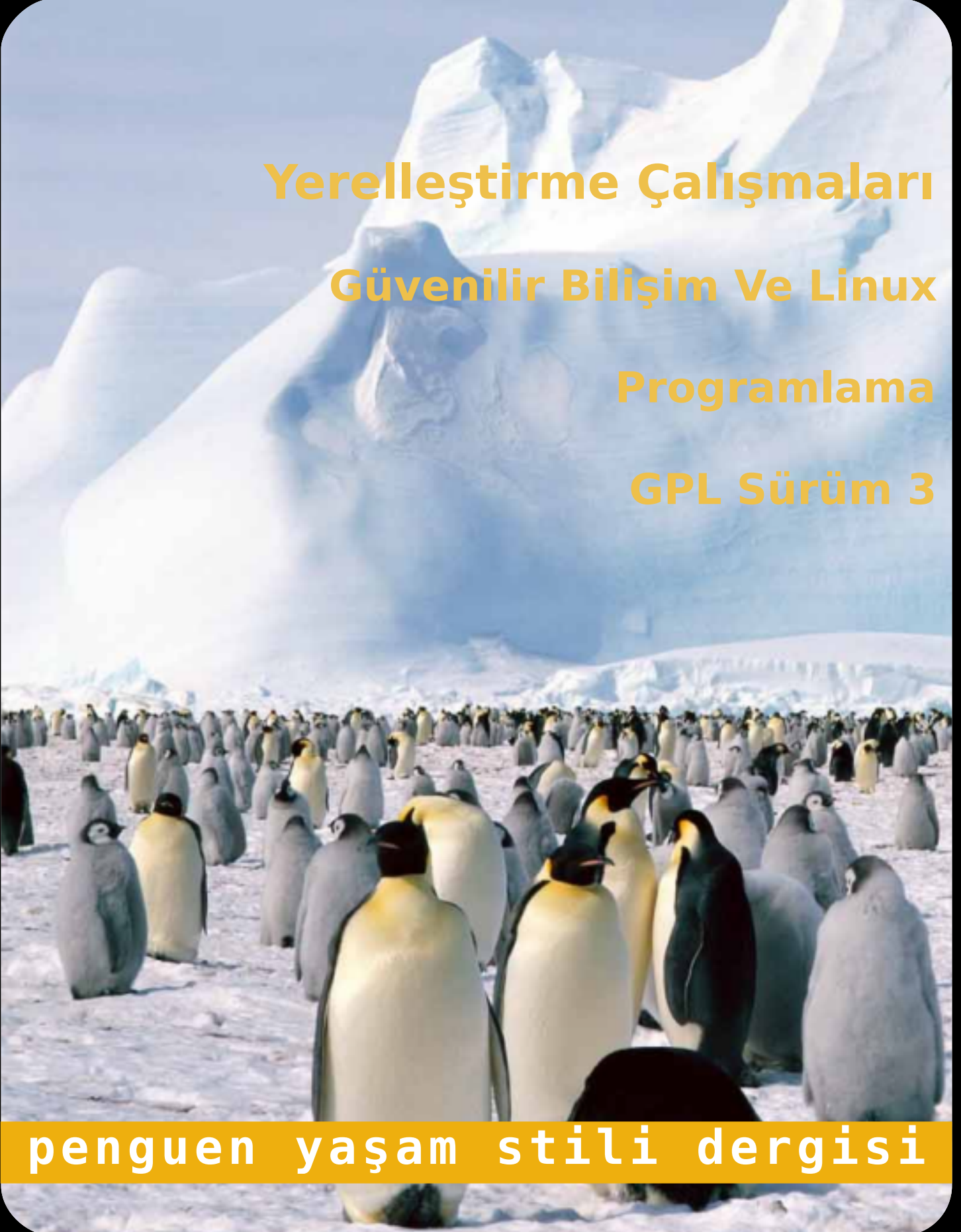


Yerelleştirme Çalışmaları

Güvenilir Bilişim Ve Linux

Programlama

GPL Sürüm 3



penguen yaşam stili dergisi

penguence

İçindekiler

Penguence'den

3

LKD'den

4

Şenlik 2006 , Genel Kurul /Onur Tolga Şehitoğlu

Gezgin Penguen

10

Open Days /Arda Çetin

Başarı Öyküleri

12

Karel Elektronikten Volkan Erinle söyleşi

Güvenilir Bilişim

18

Trusted Computing/Bora Güngören

İş Modelleri

25

Sahip Olmak Ya Da Olmamak /İzlem Gözükeleş

Dosya

27

Yerelleştirme /Deniz Akkuş

Masaüstü

44

Bilgi ağacı Programları ve Kullanılabilirlik Üzerine
Dr. Bekir Sonat

Müzik

48

Linux Sampler /Yrd. Doç.Server Acim

Programlama

50

Java Ve Özgür Yazılım /Bora Güngören

Programlama

53

Rubby /Hüseyin Gömleksizoğlu

Teknik

57

Open VPN ile VPN uygulamaları /Huzeyfe Önal

Gezgin Penguen

71

4. Tur /Arda Çetin

penguence

Penguence'den



Penguence

penguen yaşam sitili dergisi

Sayı 4 / Mart 2005

Linux Kullanıcıları Derneği (LKD)

Genel Yönetmen :Hakan UYGUN

Yazı İşleri : İzlem GÖZÜKELEŞ

Teknik İşler : Selen UĞUROĞLU

Grafik Tasarım : Onur ÜNAL

Lojistik Destek :Sefer KILIÇ

Üretim : Aysun KILIÇ

Bu dergi GNU FDL ile yayınlanmaktadır.



Penguence'den

Yeni bir sayı ile tekrar merhaba.

Dergimizin üretim sürecini olgunlaştırmaya, hedeflediğimiz düzenli çıkış periyotuna ulaşmaya çalışıyoruz. Bunun için yazarlarımız, çalışma grubumuz ve siz okurlarımızın katkılarına teşekkür ediyor, daha iyiye ulaşmak için desteğinizi istiyoruz.

Bu sayıda özgür yazılımın en önemli parçalarından biri olan Yerelleştirme çalışmalarını dosya konusu olarak seçtik. Bu konunun önemini vurgulamak, kullandığımız özgür yazılımların yerelleştirme yöntem, sorun ve ihtiyaçlarını duyurmak istedik. Yıllardır bu işi yapan ekipler ile görüştük, onların istek ve önerilerini sizlere sunuyoruz.

Geçen sayıdan bu yana bazı yeni köşeler ekledik. "Kim Kimdir?" köşesinde, özgür yazılımın simalarını tanıtmayı planlıyoruz. Okurlarımızın önemli bir kısmının sadece özgür yazılım kullanıcısı değil üreticisi de olduğundan yola çıkarak, Programlama köşesi başlattık. Bu köşede internet üzerinde bulabileceğiniz kod örneklerinden daha çok, uygulama dil ve teknolojileri ile ilgili teorik bilgiler bulunacaktır. Bu sayıda Java, Ruby dilleri var. Gelecek sayıda ise Mono, C++, Python, PHP gibi dil ve teknolojiler de yer alacak.

Önümüzdeki sayıda "Yazılım Lisansları" dosyası ile karşınızda olmak üzere...

Hakan Uygun

penguence

LKD ' den

Şenlik 2006, Genel Kurul

*Onur Tolga Şehitoğlu

Derneğimiz önümüzdeki aylarda 5. Linux ve Özgür Yazılım Şenliği'ni düzenleyecek. Bu sene ODTÜ Bilgi İşlem Dairesi'nin katılımıyla 11-14 Mayıs tarihlerinde ODTÜ KKM'de düzenleyeceğimiz şenlik için heyecanlı bir çalışma başladı bile.

Lisans ücreti karşılığında ne aldığımız ve yazılım dünyasında müşteri haklarının ahlaki ve hukuki boyutu çok uzun tartışılması gereken bir konu. Ama burada bizim vurgulamak istediğimiz slogan: "alternatifiniz var, hem de daha kaliteli!". İşletim sistemi, İnternet ve ofis uygulamaları gibi çok temel yazılımlara yüksek ücretler vererek sizi iki yılda bir aynı ücretleri ödemeye mahkum edenlerden sıkıldıysanız, sürekli virüs ve güvenlik tehditleriyle yaşamaktan bıktıysanız, bilgisayar başı lisans ücretinden başlayan ve kullanım başı ücret planları yapmakta olan şirketlere daha fazla esir olmak istemiyorsanız, 5 yıl boyunca oluşturduğunuz yaşamsal önemde belgelerin kullanamayacağınız kapalı bir kutu olarak elinizde kalmasını istemiyorsanız, sizi özgür ve

"Korsan değil Özgür Yazılım!"

Bu sene şenliğin bir teması olmasını düşündük ve geçtiğimiz sene içerisinde televizyonlarda bol bol gördüğümüz reklamlardan esinlenerek "Korsan değil Özgür Yazılım!" temasını belirledik. Bildiğiniz gibi bu reklamlarda daha çok halk arasındaki tabiriyle toplama bilgisayar satanlara çatılıyor ve bilgisayarlarla birlikte korsan yazılım verildiği ve tüketicinin kandırıldığı ima ediliyordu. Yine halen gördüğümüz bir reklamda aba altından sopa gösterilerek korsan tabir edilen yazılımın sizi hapse gönderebileceği söyleniyor. Alternatif olarak önerilen malum: gayet fahiş fiyatlarla kapalı kaynak kodlu ve çoğu standartı kapalı olan lisanslı yazılımlar satın almak. Ödenen ücretin karşılığı tüketici hakkı olarak ne alındığı da maalesef meçhul. Örneğin para vererek aldığınız bir işletim sistemi için 4-5 yıl sonra "ben artık desteklemiyorum" yanıtı alabiliyorsunuz. Tasarımdaki bir aptallık sonucu e-posta ile size bulaşan virüs yüzünden onbinlerce YTL zarara uğrayabilirsiniz ama lisans uzlaşmasında "tamam" dediğiniz bir madde bu olay yüzünden neredeyse sizi suçlu duruma düşürebilir.

açık kaynak kodlu yazılımları kullanmaya davet ediyoruz. Özgürce kullanabileceğiniz, paylaşabileceğiniz, yardım alabileceğiniz bir yazılım dünyası sizi bekliyor. Üstelik her şey yasal.

Şenlikle ilgili gelişmeler önümüzdeki günlerde yoğunlaşacak. Ayrıntılı bilgi için web sayfalarımızı ve eposta listelerimizi takip edebilirsiniz.

Şenlikle kısmen ilgili önemli bir konu da bu sene derneğimizin genel kurulunun yaklaşması. Son şenlikte olduğu gibi bu sene de genel kurulumuzu şenlik sırasında yapacağız. İki senede bir yapılan genel kurulda önerilen tüzük değişikliklerinin oylanmasıyla birlikte yeni yönetim kurulu da seçilecek.



Yönetim kurulu üyeliği derneğin önümüzdeki yıllardaki stratejilerinin belirlenmesinde etkin rol oynayabileceğiniz önemli bir konum.

Derneğin yürütme organı olan çalışma gruplarının yönlendirilmesi, eşgüdümü, idari kararlarda yönetim kurulu üyelerinin kararları rol oynuyor.

Kendinde yönetim kurulu üyesi olabilecek olgunluğu gören, diğer yönetim kurulu ve çalışma grubu üyeleriyle uyumlu çalışabilecek, üretken ve yapıcı olabilecek bütün üyelerimizi LKD'nin geleceğini şekillendirmek için yönetim kurulu üyeliğine aday olmaya davet ediyoruz.

Eğer kendinizi yönetim kurulu üyesi olmak için yeterli görmeyenleriniz varsa çalışma gruplarımızdan başlayarak birçok yerde birçok meslek ve yetenek grubunda gönüllülerin desteğine ihtiyacımız var. Dernek üyesi olsun olmasın herkesin Türkiye'de Linux ve özgür yazılımlar için yapabileceği birşey var.

Herkese Linux'lu ve özgür yazılımlı günler...

*LKD Yönetim kurulu üyesi

özgür yazılımın özgür dergileri...



<http://tectonic.magnumip.co.za/>

MySQL, Netfrastructure'ı satın aldı...

Son günlerde özgür veri tabanı yazılımları ile ilgili firmalar oldukça hareketli. InnoDB ve BerkleyDB'nin geliştirici firmalarını Oracle'ın satın almasıyla birlikte özellikle InnoDB motoru üzerinde çalışan MySQL'i zor günlerin beklediği düşünülmektedir.

Bu duruma çözüm oluşturmak amacıyla, MySQL AB, Interbase'in orijinal yazarı, Firebird veri tabanı destekçisi, Jim Starkey'in yeni firması Netfrastructure, Inc. satın aldı. Bu anlaşma ile birlikte Jim Starkey MySQL AB için tam zamanlı çalışmaya başladı.

<http://www.firebirdnews.org/?p=128>



OpenOffice 2.0.2 duyuruldu...

OpenOffice.org bu sürümü herkese önermektedir. Bu sürüm içinde birçok küçük hata ve bildirilen sorunlar giderilmiştir. Bazı yeni ayrıntılar da eklenmiştir.

Getirilen yenilikler;

- Almanca için bir imla denetim sözlüğü eklenmiştir.
- Birkaç dil için direk imla denetim entegrasyonu sağlanmıştır.
- Microsoft Word 2 ve Quattro Pro 6 dosyalarını aktarmak için aktarma süzgeci eklenmiştir.
- Mesaj birleştirme daha kolay hale getirilmiştir.
- KDE Adres Defteri ile entegre çalışma özelliği eklenmiştir.
- Linux kullanıcıları için yeni simge seçeneği eklenmiştir. (KDE ve GNOME)
- Bu sonuçlarla daha yetenekli ve kullanıcı dostu bir OpenOffice.org sürümü elde ettiğimizi düşünüyoruz.

http://www.openoffice.org.tr/index.php?option=com_content&task=view&id=67&Itemid=1



Çocuklara Bilgisayar ...

Çocuklara Bilgisayar projesi; kullanılmayan, bir kenarda duran, atıl durumda olan, "eskimiş teknoloji" olduğu için gözden düşen bilgisayarların toplanarak bunların elden geçirilmesini ve GNU/Linux kurularak ihtiyacı olan çocuklara iletilmesini amaçlayan bir projedir.

Bilgisayar ulaştırılacak çocuklara giriş seviyesinde temel bir eğitimin verilmesi ve ellerindeki bilgisayara dair soru/sorunlara hangi kanallar aracılığıyla çözüm bulabileceklerinin gösterilmesi de projenin temel amaçlarından biri olacaktır.

<http://www.cocuklarabilgisayar.org/>



Intel işlemcili Mac'ler...

Apple, Intel Core Duo işlemcili Mac'leri piyasaya sürmeye başladı. Hemen ardından da bu makineler üzerinde çalışan Linux duyuruları gelmeye başladı...

<http://arstechnica.com/reviews/hardware/macmini.ars/1>

http://www.mactel-linux.org.nyud.net:8090/wiki/Main_Page

Mozilla'nın Milyonları...

Mozzila Vakfı'nın ticari bir kuruluşu olan Mozilla Şirketi, Firefox tarayıcısının sağ üst köşesinde bulunan arama çubuğu nedeniyle Google'dan 72 milyon Dolar aldığını onayladı...

<http://www.crn.com/sections/breakingnews/dailyarchives.jhtml?articleId=181501810>



VoIP ve Linux...

Texas Instruments, üzerinde Linux gömülü, VoIP uygulamaları için kullanılacak işlemciler üreteceğini duyurdu..

<http://www.linuxdevices.com/news/NS6391318656.html>



Pardus Gönüllüleri

Pardus projesine katkıda bulunmak istiyor ve Linux ya da bilgisayar kullanımı konusunda farklı deneyimlere sahipseniz; ancak henüz Pardus topluluğunda kimseyi tanımiyorsanız, işte size ziyaret edebileceğiniz adres...

<http://www.uludag.org.tr>

python.org yenilendi...

Özgür yazılımın popüler araçlarından Python'un ana web sitesi ve logo'su yenilendi...

<http://python.org/>



pardus 1.0geldii...



www.pardus.org.tr

Gezgin Penguen Open Days'te!

Arda Çetin

Gezgin Penguen'iniz 24-25 Şubat'ta, İstanbul Bilgi Üniversitesi Bilgisayar Bilimleri tarafından organize edilen "Gelecek:Şimdi" temalı Özgür Yazılım ve Açık Kaynak Günleri 2006[1] etkinliğindeydi.

Dünyadan ve Türkiye'den açık kaynak ve özgür yazılım alanında uzman penguenlerin katılmasına rağmen etkinliğe geçen senelere oranla daha az bir ilgi olduğu gözlemlendi. Yine de gelen ziyaretçiler elleri dolu gittiler.

Debian GNU/Linux projesinin eski lideri Ian Murdock açık standartlar ve düz dünya hakkında konuşurken projenin yeni lideri olan Branden Robinson da Debian'in lider seçimi yönetim sistemi olan Condorcet hakkında bizi bilgilendirdi.

Türkiye'deki Linux ve açık kaynak etkinliklerinin vazgeçilmezi Linux Kullanıcıları Derneği Seminer Çalışma Grubu, Açık Kaynak Günleri 2006 için iki gün boyunca bir salonda Linux konulu seminerler düzenledi. Seminer konuları "Linux nedir" gibi başlangıç düzeyden trusted computing gibi ileri düzeye kadar her kesimi ilgilendirebilecek seminerlerdi.

Seminerleri kaçıranlar seminer notlarını LKD Seminer-CG web sitesindeki Seminer Notları[2] sayfasında bulabilirler.

Seminer ve paneller dışında etkinlikte çeşitli konularda atölyeler de vardı. Atölyelerdeki en büyük ilgi hiç kuşkusuz Pardus'a aitti.



Barış Metin , Pardus'un İç Yapısı

Ziyaretçileri programlar arasındaki zamanı değerlendirebilsin diye salon çevrelerinde bir çok stand onları bekliyordu. Park Yeri, Novell, IBM Türk, Hürriyet, tOSSad, Pardus ve LKD standları mevcuttu.



Stand Alanından Bir Kare

Her LKD standında olduğu gibi bu etkinlikte de çeşitli Linux dağıtım CD/DVD'leri, Linux tanıtım broşürleri, penguenli şapka ve tişörtlerle; Uludağ Ekibi de Pardus CD'leriyle ziyaretçilerin karşısındaydı.

Özgür Yazılım ve Açık Kaynak Günleri'nde her yıl olduğu gibi bu yıl da yurt dışından önemli konuklar/konuşmacılar vardı. Bunlar gnuLinEx projesinin başarı öyküsünü anlatmak için İspanya'dan gelen Manon van Leeuwen, son zamanlarda giderek yaygın bir kullanım sağlayan Creative Commons'tan Dragoslava Pefeva (Bulgaristan) ve Peter Steuer (Avusturya), bir özgür yazılım içerik yönetim sistemi projesi olan eZ Publish tanıtımı için Norveç'ten gelen Sandro Groganz ve Avrupa Birliği'nin 6. Çerçeve programı projesi olan tOSSad için Julia Velkova (Bulgaristan), Sandra Frings(Almanya) idi.

İstanbul Bilgi Üniversitesi Bilgisayar Bilimleri Bölümünden Boran Puhaloglu ve organizasyondaki herkese bize bir sene daha bilgilendirici, eğitici ve öğretici bir etkinlik hazırladıkları için teşekkür ederiz. Gelecek sene görüşmek üzere...

Etkinlik süresince çekilen fotoğraflar için:

<http://open.bilgi.edu.tr/freedays/photos>
<http://cekirdek.uludag.org.tr/~meren/photos/open-days-2006/>

Notlar :

- [1] <http://open.bilgi.edu.tr/freedays/>
- [2] <http://seminer.linux.org.tr>

penguence

LKD'nin ilk bayan teknik seminer konuşmacısı Pinar Yanardağ oldu

Necdet Yücel

1995'te düzenlenen Türkiye'de İnternet Konferansı ile birlikte seminerler vermeye başlayan Linux Kullanıcıları Derneği, 11 yıldır Linux ve özgür yazılımlarla ilgili seminerler organize ediyor. Bu süre içinde 100'ün üzerinde "erkek" konuşmacı özgür yazılımların anlatıldığı seminerlerde bilgilerini, deneyimlerini dinleyicilerle paylaştı. 24-25 Şubat tarihlerinde İstanbul Bilgi Üniversitesi'nde düzenlenen Özgür Yazılım ve Açık Kaynak Günleri'nde bu zincir sonunda kırıldı, dernek çatısında ilk kez bayan bir konuşmacı, Pinar Yanardağ, teknik içerikli bir seminer verdi.

LKD'nin düzenlediği etkinliklerde gönüllü olarak görev alan, panellere konuşmacı olarak katılan ve çok önemli görevler üstlenen bayanlar olmasına rağmen teknik seminer veren hiç olmamıştı. Val Henson tarafından yazılan "Kadınları Linux Kullanmaya NASIL Teşvik Edersiniz"[1] belgesinin de çevirmeni olan Pinar Yanardağ bir adım daha öne çıkarak bir ilki gerçekleştirdi.

Özgür Yazılım ve Açık Kaynak Günleri'nin ikinci gününde BS2 salonunda konuşan Pinar Yanardağ, seminerinde son yıllarda adından sıkça söz ettiren ve programlama dilleri arasında umut vaadedenden bir ivmeyle

büyüyen, açık kaynak kodlu Ruby Programlama Dili hakkında konuştu.

Seminerde Ruby Programlama Dili'nin yaratılış sürecinden ve yaratılış sebeplerinden bahsederek dilin tarihçesi ile ilgili bilgiler verdi. Ruby'nin temel aldığı "Özlülük", "Özgürlük ve Rahatlık" benzeri felsefelerini ve "En Az Sürpriz Prensibi" gibi prensiplerini anlattı. Ruby'nin göze çarpan önemli özellikleri ve "Nesneye yönelik programlama mantığı", "Yorumlanan diller" ve "Betikleme dilleri" gibi kavramları açıkladı. Ruby'nin ne tür uygulamalar için kullanılabileceğini ve kullanılamayacağını açıklayarak Ruby öğrenmeye istekli katılımcıları bilgilendirdi. Son olarak İnteraktif Ruby'den bahsederek kullanıcıların Ruby programlama dilini öğrenirken işine yarayacak ipuçları verdi.

[1]<<http://www.belgeler.org/howto/encourage-women-howto.html>>



pengvence

başarı öyküleri

Artık her sayıda özgür yazılımları kendi bünyesinde başarıyla kullanan bir kuruma yer vereceğiz. Telefon santralleri ile tanıdığımız Karel Elektronik ilk konuğumuz. Volkan Evrin yorulmadan sorularımızı yanıtladı :

Soru : Kurumunuzu tanımayan okuyucularımız için, Karel'in çalışma alanını kısaca anlatır mısınız?

Yanıt : Karel, 1986 yılında şef-sekreter tipi santral üretimi ile çalışmalarına başlamış ve özgün tasarımlarıyla, Türkiye'de elektronik telefon santral pazarını yaratmış %100 yerli bir firmadır. Bu pazara PBX ürünleri kazandırmış ve Türkiye'nin haberleşme alanında elektro-mekanik sistemlerden elektronik haberleşme sistemlerine geçişinde öncü olmuştur. Aynı zamanda bir elektronik sistem üreticisi olan Karel Elektronik, bu faaliyet alanıyla ilgili ağırlıklı olarak beyaz eşya elektroniği tasarımı ve üretimi konusunda çalışmaktadır.

Karel, ana faaliyeti olan telekomünikasyon konusunda çeşitli teknoloji, özellik ve kapasitelerde kablolu ve kablosuz haberleşme sistemleri geliştirmekte, üretmekte ve bu ürünleri yurtiçi ve yurtdışı pazarlarda satmakta ve satış sonrası hizmet organizasyonlarını gerçekleştirmektedir. Başlıca ürünler, üstyapı sınıfında küçük, orta ve büyük kapasitelerde olmak üzere özel telefon santralleri ve bunların çevre birimleri, altyapı sınıfında ise kırsal alan santralleri ve ara bağlantı santralleridir. Karel Elektronik üretimini yaptığı haberleşme ürünlerini yurtiçinde kendi yetkili satıcılarından oluşan dağıtım kanalı üzerinden, yurtdışında ise dağıtıcı firmalar üzerinden pazarlamaktadır.

Karel'in başarısı telekomünikasyon gibi büyük Ar-Ge gücüne ve yatırıma ihtiyaç duyulan, az

gelişmiş ülkelerin kullanıcı olarak bile bulunmadığı bir sektörde, gelişmiş ekonomilere sahip ülkelerin dünya devi üretici firmalarıyla rekabet ederek dünya ölçeğinde bir firma yaratması ile ölçülmektedir

Soru : Karel, Linux ile ilk ne zaman ve nasıl tanıştı?

Yanıt : 90'lı yılların ortalarında gelişim hızını arttıran Internet ve iletişim altyapıları, KAREL içindeki iletişim ve hizmet taleplerini de arttırmıştı. Bunlar için kendi kaynaklarımız ile hızlı, güvenli ve ekonomik çözümler için ilk denemelerimiz e-posta, dns gibi hizmetleri şirket içi sunucularda tutmak şeklinde oldu. Acemiliklerimiz sırasında en büyük yardımları da Linux Kullanıcıları Grubu'ndan ve e-posta listelerinden almıştık. Murat Koç'un, Mustafa Akgül'ün, Doruk Fişek'in ilk kurulum sancılarımız sırasındaki yardımları bizleri Linux üstünde çalışan hizmetlere de daha da çabuk ısındırmıştı. Gerçek anlamda 1997-1998 yıllarında çalışan ilk sunucularımızdan sonra hem kendimizi hem de kullandığımız sunucu sistemlerini geliştirdik.

Soru : Bugün Linux'u sunucularınızda hangi uygulamalarda kullanıyorsunuz?

Yanıt : KAREL içerisindeki tüm e-posta sunucuları (virüs ve spam korumaları ile beraber) Linux sunucular üzerinde çalışmaktadır. Yerel ağ içinde verdiğimiz bazı küçük servisler de (NFS, NIS, samba vs.) yine Linux sunucular üzerinde yürümektedir. Kendi adımıza en iddialı çalışmayı da ARGE merkezimizin yeni yerleşkesinde tüm ağ sunucularını Linux kurarak yaptık.

Windows ağ ve sunucu yönetimini Linux sunucular üzerine aktardık. Samba, Firewall, E-posta, dosya/dizin sunucularında Linux'u

KAREL

DHCP, NTP gibi uygulamaları hizmete sunmak istiyoruz. Yeni kurduğumuz ARGE yazılımının kaynakları, yönetim ve sürüm takip uygulamasını da yine Linux üzerinde çalıştırmaya başladık. KAREL için en önemli hedeflerden biri de halen şirket bünyesinde çalışan Oracle ERP uygulamasını, performans ve güvenlik gerekçeleri ile Linux üzerine taşımak olacaktır.

Soru : Masaüstünde Linux kullanımına nasıl bakıyorsunuz?

Yanıt : Masaüstü kullanıcılarımızın neredeyse tamamına yakını Windows işletim sistemlerini kullanıyor. Burada geçiş veya değişim sunucu hizmetlerindeki kadar kolay olmuyor maalesef. Çünkü, gerek kullanıcıların yaygın alışkanlıkları ve geçmiş çalışmaları, gerekse kullanılan uygulama ve yazılımların sadece Windows işletim sistemleri ile uyumlu olması bu konularda bize fazla şans tanımıyor. Fakat, işletim sistemi dışında ofis paketleri, İnternet uygulamaları, sıkıştırma yazılımları vs. gibi pek çok türde açık kaynak kodlu ürünleri yaygınlaştırmaya çalışıyoruz. Gerek lisans maliyetlerini en aza indirebilmek, gerekse orta vadede normal kullanıcılarımızı Linux masaüstüne taşımak adına bunu gerekli görüyor ve ısrarla takip etmeye çalışıyoruz.

Soru : Bu konuda son kullanıcılardan herhangi bir direnç ya da olumsuz tavır görüyor musunuz?

Yanıt : Bazen evet, bazen hayır. Kullandığı işletim sistemine ya da uygulamaya hakim, seven ve iyi kullanan çalışanlarımızın var. Bu kişilere seçenekler sunduğumuzda kimi merak ve heyecanla kabul edip denemelere katılıyor ve aktif kullanıma hemen geçiyor. Kimi ise pek oralı olmuyor. Örneğin OpenOffice. Firma genelinde yaygınlaştırma çalışmalarını ısrarla yürütüyoruz. Daha önce fazlaca Microsoft ofis paketini kullanmamış kişiler kesinlikle hemen uyum sağlıyorlar ve kullanıyorlar. Bir kısım çalışanlarımız ise, elindeki paketi kullanmakta ısrarcı oluyor. Bizim hedefimiz, kullanılan uygulama yazılımını her yönüyle değerlendirilip (bütünlük, uyumluluk, maliyet, verimlilik ve

süreklilik) gerçek çözümü kullanıcılarımıza sunmak. Onları ikna edebilmek için önce kendimiz kullanıyoruz ve örneklerini gösteriyoruz, sonra konuşarak seçenekleri tanıtıyoruz ve en sonunda da kullanıma geçiş sürecinde elimizden geldiğince destek oluyoruz. Bu süreç sonunda yeterli altyapıyı ve bilinçlendirmeyi oluşturduktan sonra, firma kültürüne ve şirket çalışmalarına kullanılmasını istediğimiz işletim sistemlerini, kullanıcı hizmetlerini ve uygulama yazılımlarını tam olarak yerleştirmeyi düşünmekteyiz.

Soru : Ofis içi uygulamaları dışında kendi ürettiğiniz santral ve benzeri cihazlarda Linux kullanmayı düşünüyor musunuz? Müşterilerinizden bu yönde bir talep var mı?

Yanıt : Aslında düşünmekten öte kullanıyoruz da! Şu an da ürün olarak da çıkmış DS10 santrallerimizde gömülü bir Linux çalışmakta. Ayrıca, yeni nesil santrallerimizde de Linux tabanlı işletim sistemleri kullanılmaya başlandı. Son kullanıcıya yönelik uygulamalarda da işletim sistemi bağımsız seçenekler daha çok talep edilir hale geldiği için yeni arayüzler bu yönde geliştirilmektedir. Ayrıca, gelişen teknoloji ve PBX sistemlerin artık ağ servisleri ile beraber çözüm üretebilir hale gelmesi nedeni ile sunucu ve veritabanı hizmetlerini de santral sistemleri ile bütünleşik hale getirmek gerektiğine inanıyoruz. Bu alanda da en uygulanabilir çözümler genelde Linux ve Açık Kaynak kodlu sistemler ile mümkün olacaktır diye düşünüyoruz.

Soru : Kurum bünyesinde uygun bulduğunuz hemen her alanda kapalı kaynak kodlu yazılımları özgür yazılımlarla yenilemeye çalışıyor gözüküyorsunuz. Karel neden özgür yazılımı tercih ediyor?

Yanıt : Sunucu sistemleri açısından bütünlük, lisans bağımsızlığı ve en önemlisi güvenlik! Bu başlıkları Microsoft ya da başka bir marka ile de sağlayabilirsiniz, ama çok emek ve çok para harcamak gerekiyor. Ve elde ettiğiniz sonuçların devam edebilmesi her zaman garantili olmuyor, çünkü sistem kapalı kaynak

KARELARGE

kodlu ve bir ticari firmaya bağımlı! Görerek ve deneyerek kurduğumuz Linux sunucu servislerinde kısa vadede yoğun emek harcasak bile oluşturduğumuz yapının sağlam ve güvenli olması her zaman içimizin rahat olmasını sağlıyor. Bunun dışında kullandığımız son kullanıcı uygulamalarının ve hizmetlerinin işletim sistemi bağımsız olması gerektiğine inanıyoruz. Zira, kullanıcılarımızın bugün ve gelecekte farklı işletim sistemlerinde çalışmaları mutlaka olacaktır. Bunu tek bir işletim sistemine indirgemeye çalışmak pek mümkün değil. Bu nedenle firma genelinde kullanılan uygulamaların, yazılımların ve şirket çalışmalarının özgür yazılımlarda çalışıyor olması bizim maliyetlerimizi azaltmakla kalmayacak, işletim sistemi seçme özgürlüğümüzü, verimliliğimizi ve orta-uzun vadede başarımızı da arttıracaktır diye düşünüyoruz.

Soru : Son olarak, Linux kullanmayı düşünen diğer kurumlar için önerileriniz olur mu?

Yanıt : Öncelikle küçük ya da büyük herhangi bir firmanın ya da kuruluşun Linux ve açık kaynak kodlu uygulamaları kendi bünyesinde kullanabilmesi için iyi düşünülmüş bir yol haritasına ihtiyacı olacaktır. Erken ya da geç atılacak adımlar, yarardan çok zarar verebilir.

Geçiş sürecinde önce en kolay olan sunucu uygulamalarını kullanmalarını, daha sonra işletim sistemi bağımsız uygulamaları yaygınlaştırmalarını ve en sonunda da gereken yerlerde masaüstü işletim sistemlerini değiştirmelerini öneririm. Bunları yaparken 3 önemli etken vardır. Bilgi İşlem birimleri, Üst Yönetim ve son kullanıcılar: Bilgi İşlem birimleri bunları yapabilecek bilgi ve beceriye sahip olmalı ya da bu konularda destek alabilmeli. Üst Yönetim, başından sonuna kadar sürece destek olmalı ve sahiplenmeli. Son kullanıcılar da değişime ve yeniliklere hazır olmaları konusunda bilinçlendirilmeli ve eğitilmeli. Biz KAREL olarak bu konuda epeyce yol aldık, ama daha bitirmedik. Üst Yönetim'in de desteği ile başarılı çalışmalar yaptık ve yapmaya da devam ediyoruz. Tüm sistemlerimizin Linux ve Açık Kaynak kodlu yapıda çalışması kısa ve orta vadede tamamen mümkün olmayabilir, ama kazandığımız yeni yetenekler, beceriler ve seçenek çeşitliliği bize daha sağlam sistemler üzerinde daha verimli çalışma olanakları verecek diye düşünüyoruz.



Çocuklara Bilgisayar Projesi



Çocuklar için bir milyon bilgisayarı yeniden kullanıyoruz.

www.cocuklarabilgisayar.com

GPL'in Yeni Versiyonunda Değişenler

Derleyen : İzlem Gözükeleş

GNU (GPL) Genel Kamu Lisansı, 15 yılda beklentilerin çok ötesinde bir başarı elde etti. Dünya genelinde kullanıcılar/geliştiriciler arasında bir güven ve işbirliği ortamı yarattı. Temel ilke, kullanıcıların hakkına saygı duymak, yazılım alanının da dışına taşı; yaratıcı ve bilimsel girişimlerde de kullanılmaya başlandı.

GPL'in başarısının ardında yatan, yalnız ya da birlikte çalışan kullanıcıların üretme özgürlüğünü güvence altına almasıydı. Ancak değişen teknolojik ve hukuksal koşullar bazı düzeltmeleri ve yenilemeleri gerekli kıldı. İşte, GPLv3 süreci bu kaygılarla başlatıldı. GPLv3 herkesin katılımına açık bir süreç olarak şekillendi. Sürecin gelişimi ve GPLv3 hakkında ayrıntılı bir bilgiye Özgür Yazılım Vakfı'nın(Free Software Foundation-FSF) web sitesinden ulaşabilirsiniz [1]. Bu yazıda ise GPLv2 ve GPLv3 arasındaki temel ayrımlara değinilecek ve GPLv3'teki tartışmalı noktalar (Linus Torvalds'ın yeni GPL versiyonuna itirazı gibi) başka bir yazıya ertelenecek.

Yazılım Patentleri

Bilgisayar teknolojisi 1991'den beri hızla değişiyor olsa da GPL'in yeniden tasarlanmasının arkasındaki birincil neden bu değildi. Son 15 yıldaki teknolojik gelişmeler kullanıcılara yeni özgürlükler sunarken yeni tehditlerin de kapısını araladı. GPL'in ana amacının kullanıcıların özgürlüğünün sürekliliğini sağlamak olduğunu, şu ya da bu teknolojiden bağımsız olduğunu düşünürsek, GPL'in yeniden tasarlanmasını gerektirecek köklü bir teknolojik değişiklik yaşanmadı.

Bu bağlamda, yeni bir GPL'i ateşleyen birincil nedenler teknolojik değişimden değil, hukuksal değişimlerden kaynaklandı. Bu değişimlerden ilki yazılım patentleri alanında yaşandı. Yazılım patentleri özgür yazılım projelerini tehdit ettiği gibi sahipli yazılımları da tehdit ediyor. Yazılım patentlerine göre, herhangi bir program, patent hakkına sahip olmaktan başka bir özelliği olmayan biri tarafından çok rahat ortadan kaldırılabilir.

Aslında GPL yazarları 1991 yılında patentlerin yol açabileceği sorunların farkında olan az sayıdaki insanlar arasındaydı. O yıllarda problem sadece bir ülkeyle, ABD ile, sınırlı görünüyordu. Ancak birçok ülke farklı boyutlarda da olsa ABD'nin yazılımın patentlenmesi konusundaki yönergelerini

GPL yazarları 1991 yılında patentlerin yol açabileceği sorunların farkında olan az sayıdaki insanlar arasındaydı.

izledi. Dolayısıyla patent tehdidi ve adaletsizlik giderek büyümeye başladı. GPLv2'de patentlere yönelik bir madde de vardı ve o günlerde patent tehlikesine karşı tek dayanak noktasıydı. Fakat giderek yoğunlaşan patent tartışmaları GPL'i patentler karşısında yetersiz duruma getirdi.

Dolayısıyla, GPLv3'ün odaklandığı en önemli nokta patentler. GPLv3, bir çalışmadaki patent haklarının lisans sahibi tarafından başışına olanak veriyor. Ayrıca lisans sahibinin bu başışından vazgeçmesi ve türetilmiş çalışmalarda patent hakkı talep etmesine karşı da önlemler almaya çalışılıyor. Ancak lisans sahibi, patentin kullanımına dair belirli sınırlamalar getirebiliyor.



Sayısal Hak(sızlık)lar Yönetimi

Patentler dışında GPLv3'ün ilgilendiği bir diğer husus ise Sayısal Hak(sızlık)lar Yönetimi (Digital Rights/Restrictions Management – DRM). FSF'ye göre DRM, kullanıcı haklarına tamamen aykırı. Ancak FSF, DRM'e karşı yapabileceklerinin sınırlı olduğunu da farkında. DRM hakkındaki bölüm FSF'nin kaygıları ve aldığı önlemlerle ilgili; özellikle GPL'li programların DRM teknolojisi kapsamında kullanımına karşı (kopyalama koruma teknolojilerinden daha gelişmiş koruma teknolojilerine kadar.). Yeni lisans, ilgili yazılımla teslim edilen kaynak kodunun, şifreleme, şifre çözme ve yetkilendirme ile ilgili kaynak kodlarının da verilmesini gerekli kılıyor.

Türetilmiş Çalışmalar

GPLv3 türetilmiş programlar hakkında da yeni düzenlemeler/açıklıklar getiriyor.

Lisans Uyumluluğu

GPL'in yanında çok sayıda açık kaynak kod lisansı mevcut. GPL'in yeni versiyonu, GPL'in bu lisanslarla ilişkisi ve uyumuna dair açıklamalar içeriyor. Aslında GPLv2'de bu konuyla ilgili hususlar içeriyordu. Fakat GPLv3 lisansların birlikte kullanılabilirliğine dair yeni açıklamalar getiriyor.

Dağıtımlar

GPLv3'te, dağıtıma yönelik yeni açıklıklar ve değişiklikler de mevcut.

Ek Lisans Hükümleri

GPLv3'e göre lisans sahibi GPL'e ek yasak hükümler getirebilir. Ancak bu ek hükümlerde GPL'in geneli ile uyumluluk temel şart. Ayrıca lisans sahibi (yine GPL ile çalışmamak kaydıyla) kullanıcılara ek seçenekler sunabilir.

Uluslararası Uygulama

GPLv3'te lisansın uluslararası uygulanabilirliğine dair yeni düzenlemeler yapılıyor.

Sonuç

Yukarıdaki bilgiler, GPLv2 ve GPLv3 arasındaki temel farklılıklar. Yazının başından da belirtildiği gibi GPLv3 süreci herkesin katılımına açık bir çalışma. Yeni lisansın oluşturulması amacıyla farklı kurullar oluşturulmuş. Uluslararası konulardaki sınırlamalarda olduğu gibi bazı bölümlerde boşluklar var ve farklı görüşlere açık.

Ocak ayında ilk taslağı yayınlanan GPLv3'ün nasıl oluşturulacağını önümüzdeki günlerde göreceğiz. FSF sitesinde belirtilen takvime göre süreç aşağıdaki şekilde planlanıyor:
Haziran, 2006: ikinci tartışma taslağı
Eylül, 2006: GPLv3'ün olası en erken yayımı
Ekim, 2006: Olası üçüncü tartışma taslağı
Mart, 2007: GPLv3'ün yayımı için en geç tarih

Referanslar:

[1] <http://gplv3.fsf.org/>



penguence

güvenilir bilişim

Güvenilir Bilişim (Trusted Computing) ve Linux

Bora Güngören

“Güven” (trust) sözcüğü bir çok anlama gelen ve bu anlamları genelde esnek olarak kullandığımız sözcüklerden birisi. Güvenilir bilişim de güven ilişkilerinin sağlanması üzerine kurgulanan yeni ve çok önemli bir kavram olarak önümüzdeki birkaç yıl içerisinde çok sık duyacağımız bir şey.

Sözcüğün doğru kullanımını sağlamak için gerçek yaşamdan bir örnek vermek doğru olabilir. Resmi bir kurumda, kendinizle ilgili bir başvuru yaptığınız zaman kimliğinizi ispatlamanızı isteyebilirler. Bu ispat için resmi bir kurum tarafından onaylanan bir kimlik belgenizi, örneğin ehliyetinizi görmek isteyebilirler. Burada size güvenemedikleri için, sizin kimliğinizi kanıtlayan üçüncü bir şeye, yani kimliğe güveniliyor. Aslında o kurumla sürekli iş ilişkiniz olsa, gün aşırı gidip gelseniz, size güvenebilirler.

Uzun lafın kisası, üçüncü bir şahısla olan “güven ilişkiniz” (trust relationship) üç türlü olur. Ya ona güvenirsiniz, ya ona güvenemeseniz bile, güveneceğiniz bir yer ona güveniyordur, ya da hiçbir şekilde güvenemezsiniz. Aslına bakarsanız, tüm ticaret sistemi bu üçlü güven modeli üzerine kuruludur. Aracı kuruluşlar, noterler, bankalar, finans kuruluşları, sigorta sistemi hep güven duyamadığımız üçüncü kişilerle olan ilişkilerimizi kurmamıza yardımcı olur.

Ancak bu kuruluşların kendilerine duyduğumuz güvenin de sorgulanması söz konusu olabilir. Örneğin bankalara ne kadar

güvenebilirsiniz? Ya da size gösterilen bir kimlik belgesinin sahte olmadığı ne kadar güvenilirdir? Bu durumda bu konularda ek bir güvence veren ve kendisinin güvenilirliği sorgulanmayan bir kişiye gerek duyarız. Bu kişi, gündelik yaşamımızda “devlet” kavramıdır. Teknik açıdan bakarsak, her koşulda güvenilen bu kişiye “güvenin kökeni” (root of trust) adı verilir.

Bilişim İçin Güven Kavramı

Bilişim sektörü giderek artan ve saydamlaşan kullanımı ile yaşamımızdaki en temel gereksinimlerimizi karşılamamızda yardımcı olmaya başlamış durumda.

Bu görünmez sistemlerin alt yapıları tek ve dev sistemlerden giderek karmaşık birlikteliklere doğru ilerlemektedir. Kişisel bilgisayarların donanımları binlerce üretici, tipik bir kurumsal yazılım ise onlarca üretici tarafından ortaya konan bileşenlerden oluşur.

Bilişim sektörü giderek artan ve saydamlaşan kullanımı ile yaşamımızdaki en temel gereksinimlerimizi karşılamamızda yardımcı olmaya başlamış durumda.

Bileşenlerin arasında iletişim kurulması işlevselliğin sağlanması için zorunludur. Bu nedenle bütün sisteme güvenebilmemiz için aslında sistemin içinde gerekli güven ilişkilerinin sağlanması gerekir. Matematikle ilgisi olan okuyucuların bu tür bir ilişkinin yönleri olan bir çizge (directed graph) olarak ifade edilebileceğini zaten tahmin edecektir.

Eğer sistemdeki her bileşenin iletişimde olduğu tüm bileşenlere güvenmesi için bir yol bulabilirsek sistemin bütününe güvenebiliriz. Bu doğrulamanın yapılması kendi başına önemli bir tasarım sorunudur. Bizim güvenilir Bilişim (Trusted Computing, kısaca TC) dediğimiz şey de aslında bu doğrulama sorunu için getirilen bir çözüm yaklaşımından ibarettir. Başka yaklaşımlar da olabilir. Veya

insanların bu yaklaşımdan anladıkları, bu yaklaşımı uygulama planları, bu planlar içindeki teknik detaylar farklı olabilir.

Güveni Neler Oluşturur

Bir bileşenin diğerine güvenebilmesi için dört temel koşulun sağlanması gerekir. Şimdi bu dört koşulu kısaca inceleyelim.

Bunların birincisi karşılıklı iletişimde veya saklanan bilgilerde gizliliğin (confidentiality) sağlanmasıdır. Gizlilik ile kastettiğimiz şey, üçüncü kişiler bilgiye erişse bile ondan bir anlam çıkarmalarının imkansız olmasıdır. Verilerin şifrelenmesi bunun için en temel gereksinimdir. İnternet bankacılığında kullandığımız, e-postalarımızı POP istemcimize çekerken önerilen SSL, TLS gibi teknolojiler gizliliğin sağlanması için geliştirilmiştir. Gizlilik tek başına faydalı olsa da yanında başka bir şey daha ister. Bütünlüğün (integrity) sağlanması, kısaca bir şeyi “bıraktığınız gibi bulmanız” anlamına gelir. Üçüncü kişilerin size ait bilgileri saklandıkları yerde ele geçirip değiştirmediklerinden emin olursanız, saklama ortamına güvenebilirsiniz. Aynı prensibi iletişime uygularsanız, bu sefer de giden ve gelen bilginin araya giren üçüncü kişilerce tahrifata uğramadığından emin olursunuz. Gerçek dünyada karşılaştığımız “kapalı zarf” uygulaması (zarfın kapatıldıktan sonra mühürlenmesi) aslında bütünlüğün sağlanmasından ibarettir.

Güvenin üçüncü koşulu, bulunurluk (availability) olarak adlandırılır. Bir bileşen gerek duyduğunuz zaman size gereken hizmeti sağlayamazsa ona güvenemezsiniz. Bulunurluğun bir şekilde garantilenmesi gerekir. Bu garanti koşulsuz olmak zorunda değildir; bazı koşullarla birlikte de gelebilir. Telekomünikasyon sektöründe kullanılan “hizmet niteliği” (QoS) sözleşmeleri aslında güvenin oluşması için bulunurluğu tanımlar. Güvenin son koşulu, kurtarılabirlik (recoverability) olarak adlandırılır. Olağanüstü durumlarda, örneğin felaket anında, bir bileşenin kurtarılması gerekebilir. Bunun yapılması için mekanizmaları bulunmayan bir bileşene güvenemezsiniz.

Güvenilir Bilişim, bilgisayar sistemlerindeki her bir bileşen için bu dört kavramın sağlandığı bir güven modelini kurmayı hedefler.

Üst Üste Yığılan Güven Modeli

En temel ve en kolay uygulanan güven modeli, tıpkı bir zincir ya da üst üste yığılmış tabaklara benzer. Herkes kendisinden öncekine güvenir. Bu modelin kendisinin doğrulanması oldukça kolaydır. Yani sistemin mimarisindeki hataları ya da zayıfları hemen tespit edebilirsiniz. Günümüzdeki Güvenilir Bilişim modeli de bu tür bir güven modeli ile çalışır. Daha karmaşık, örneğin dağıtık modeller ise hala araştırma konusudur.

Bu modelin çalışması için bilgisayarın açılışından itibaren güven ilişkilerinin doğrulanması ve doğrulama yapılamayan yerde sorunun bildirilmesi, bir olasılıkla da sistemin durdurulması gerekir. O zaman ilk güven ilişkisi ne olmalıdır ve bu ilişkiyi kime güvenerek doğrularız?

İlk güven ilişkimiz bilgisayarımızdaki donanımların kendilerine güvenmemizdir. Çünkü donanımların içinde de belli yazılımlar bulunur. Ayrıca veri saklayabilen veya veri iletebilen tüm benzeri donanımların casusluk amacı ile kullanılması mümkündür. Bir sabit disk arızasının bütün verilerinizi çalmak amacı ile tasarlanan bir saldırı olması dahi mümkündür.

Bu güven ilişkisini sağlayabilecek mekanizmanın bu donanımları kullanmayacak biçimde çalışması gerekir. Yani kendi içerisinde tam ve eksiksiz olmalıdır. Bu nedenle bu mekanizmanın bir diğer donanım bileşeni içinde durması planlanmıştır. Bu nedenle donanımsal güven kökeni (hardware root of trust) kullanmak zorunda kalırız.

Bu donanım var olan işlemcinizin içine gömülü olabileceği gibi (Transmeta Crusoe, Intel LaGrande ve AMD Pacifica mimarileri) ayrı bir entegre (Infineon, National ve Atmel örnekleri) de olabilir. Biz bu donanıma Trusted Platform Module (TPM) adını veriyoruz. Bu donanımın neler yapabileceğinden daha sonra bahsedeceğiz.



Daha sonra BIOS, İşletim Sistemi, Servisler ve Uygulama Yazılımları olarak üst üste yığılan bir model kendiliğinden takip eder. Her bileşen kendisinden daha alttaki bileşenlerin doğrulanmış olduğunu bildiği için onlara güvenir.

Donanımsal Güven Kökeni ve TCPA' in Kuruluşu

Bu tür bir mekanizmanın kurulması için donanım ve yazılım üreticilerinin bir araya gelmesi ve sistemin tasarımı yapmaları gerekir. Zamanla HP' den Fujitsu-Siemens' e bilgisayar üreticileri, Intel, AMD, National, Infineon ve Motorola gibi elektronik bileşen ve mikro işlemci üreticileri, Checkpoint' ten RSA' ya güvenlik şirketleri, Microsoft gibi yazılım şirketleri ve hatta PriceWaterHouseCoopers gibi danışmanlık şirketlerinden oluşan kalabalık bir örgütlenme oluştu. Bu örgütlenmeye ilk başta Trusted Computing Platform Alliance (TCPA) adı verildi. Şu anda Trusted Computing Group (TCG) olarak adlandırılıyor. Grubun aşağıdaki adreste bulunan web sitesine girerek firmalar hakkında bilgi edinebilirsiniz.

<https://www.trustedcomputinggroup.org/home/>

Grup TCPA olarak yapılandırıldığı sırada, bu mekanizmaların potansiyel uygulamalarından birisi olan sayısal hak yönetimi (DRM) konusunda Microsoft'un başını çektiği başka bir grup ABD' de bazı yasaların geçmesi için yoğun çaba sarf ediyordu. Bu yasaların insan hakları açısından önemli tepkiler çekerek yasalastığını belirtelim. Microsoft ve diğer ortakları, uygulamanın kolay olması açısından TC alt yapısının sağladığı olanaklardan yararlanacaklarını açıkladılar. Bu açıklama onlar için son derece sıradan bir teknik karardı.

Ancak özgürlük hedefleyen gruplar önemli bir yanlış anlamanın kurbanı oldular. Kötü amaçlı kullanılabilecek bir teknolojiye suç teknolojiye değil, onu kötü amaçlı kullananlarda olur. Ancak baskı grupları teknolojinin kendisine saldırdılar. Birdenbire TCPA kötü adamlar kulübü ve Trusted Computing sözcükleri kötü anlama sahip bir sözcük oldu. Bu gruplardan birisinin web sitesini aşağıdaki adresten inceleyebilirsiniz.

<http://www.againsttcpa.com/>

Halbuki Güvenilir Bilişim modeli, virüs, istenmeyen e-posta, solucan gibi genel güvenlik sorunlarını çözmekten başlayarak, geleneksel yazılım kırma tekniklerini olanaksız hale getirecek bir yapı sunmaktadır. Şimdi bu işin nasıl olduğunu inceleyelim.

Bir TPM Neler Yapabilir?

Güvenilir Bilişim için karşılıklı doğrulama mekanizmaları kriptografik tekniklerle sağlanır. Bu tekniklerin kendilerinin yazılım ya da donanım tabanlı olması mümkündür. TPM, en temel kriptografik işlemler için gerekli altyapıları güvenilir biçimde sağlar.

Kripto algoritmaları genelde rastgele sayılara dayanan matematiksel kuramlardan doğmuştur. Ancak normal şartlar altında bir bilgisayarın bir rastgele sayı üretmesi mümkün değildir. Ayrıca üretilen rastgele sayının güvenli biçimde teslim edilmesi de mümkün değildir. Yani birisi başka bir rastgele sayı sahibi olarak sizin rastgele sayınızı tahmin edebilir. Ancak TPM size güvenliği sağlanmış bir gerçek rastgele sayı kaynağı (true random number generator - TRNG) verir. Bu yazılım tabanlı kriptografik sistemlerin hem performansını hem de güvenliğini ciddi anlamda artırır.

Bütünlüğün korunması işlemi özet (hash) algoritması kullanımı ile yapılır. İnternet' ten büyük dosyalar çektikten sonra dosyanın hatasız çekildiğini kontrol etmek için kullandığınız MD5 ve SHA-1 toplamaları aslında bu özet algoritmalarının sonuçlarıdır. Bir TPM, bize bu algoritmaların güvenilir donanım uyarlamalarını sağlar. Böylece bu teknikleri kullanan yazılımlar hızlanır ve bu işlemin kendisi güvenilir hale gelir. Bir çok TC alt yapısı, sertifikaları ve açık anahtar altyapısını (PKI) kullanır. Aştığımız SSL benzeri güvenlik mekanizmaları da bu altyapıyı kullanmaktadır. Bu altyapının temelinde bir çift anahtarın yaratılması yatmaktadır. Bu anahtarların düzgün ve tahmin edilemez biçimde yaratılması önemlidir. Bilgisayarında bulunan bir TPM, son derece güvenli biçimde, bize 2048 bit uzunluğuna kadar simetrik yada asimetrik anahtarları üretebilir. Bu sayede gerektiği zaman sadece o işlemde kullanmak yada saklamak üzere anahtar üreten çok çeşitli yazılımlar bu işleme güvenebilir.

Dikkat etmişseniz, bir TPM zaten var olan güvenlik tekniklerinin alt yapılarını, Güvenilir Bilişim'deki tanımı ile güvenebileceğiniz bir şekilde sunmaktadır. Bu sayede güvenlik yazılımlarının becerileri artacağı gibi aynı zamanda "ben kırıldım, artık bana güvenemezsin" diyen yazılımlar üretmek de mümkün olacaktır.

Bütün bunların temelinde yazılım bileşenlerinin kendi kimliklerini sertifikalarla ve açık anahtar altyapısı ile kanıtlaması yatacaktır. Bu sistemin önemli bir ihtiyacı bir anahtar deposudur. Bir TPM aynı zamanda küçük bir güvenilir anahtar deposu da sağlar. Bazı tür anahtarlar asla TPM' den dışarı çıkmaz. Anahtarlarla yapılacak işlem TPM' in içerisinde gerçekleşir.

Peki TPM' in kendisi nasıl güvenlikte olacaktır? TPM entegreleri askeri entegrelerde gördüğümüz üretim sürecine ve tasarımlara benzer biçimde üretilir. Yani TPM entegresini fiziksel olarak kurcalamaya kalkarsanız kendisini imha edebilir. Bu sırada içerisinde sakladığı önemli bilgileri (örneğin anahtarları) da standartlara uygun biçimde imha eder. Buna kurcalamaya karşı güvenli (tamper proof) olmak adını veriyoruz.

Bu sayede bir bilgisayarı çalan bir saldırgan TPM' i özel bir cihaza bağlayarak kırmayı denese dahi başarısız olur. Bu işi yapabilseydi, TPM' i kırabildiği için tüm güven mekanizmasının sahibi olurdu.

Güvenilir Bir İşletim Sistemi (Trusted OS)

Donanım platformunu güvenilir kıldıktan sonraki hedefimizi BIOS' u güvenli kılmaktır. BIOS göreceli olarak durağan bir yapıda olacağı için bu işin kolayca yapılacağını öngörürsek bir sonraki basamağa çıkmamız gerekmektedir. Yani işletim sisteminin kendisinin güvenilir olması gerekir.

Güvenilir işletim sisteminin geri kalan özellikleri işletim sisteminin görevi olan kaynak yönetiminin değişik yönlerinin güvenilir kılınmasıdır.

İşletim sisteminin güvenilir olması için belli başlı işlevselliklerinin güvenilir olması gerekir. Bunların en belirgin olanı işletim sisteminin güvenilir biçimde başlatılması (trusted booting) özelliğidir. İşletim sistemi yüklenirken kendisinin bütünlüğünün korunmuş olduğunu MD5 yada benzeri bir özet tekniği ile doğrulanmalıdır. Bu doğrulama geçilmezse, işletim sistemi yüklenmeyecektir.

Ancak bir işletim sistemi sürekli olarak güncellemeler ve yamalarla değişikliği

uğrayacağı için güvenilir başlatma o kadar da kolay kazanılan bir beceri olmaz. İşletim sisteminin kendi durumunu yamalarla birlikte değerlendirmesi gerekir. Daha önceden güvenilir olan işletim sisteminin üzerine eklenen yamayla birlikte güvenilir sayılması için yamanın kaynağının güvenilir olması gerekir. Bu da yine kriptografik tekniklerle kimlik doğrulama gerektirmektedir. Yama indirdiğiniz her yerin ve her yamanın birileri tarafından sayısal imzalarla donatılması gerektiğini düşünün.

Elbette bu tür becerilerin işletim sistemine nasıl ekleneceği işletim sisteminin tasarımına da oldukça bağlıdır. Belli oranda modüler bir yapısı olan bir işletim sistemi göreceli olarak daha kolay güvenilir hale getirilebilir.

Güvenilir işletim sisteminin geri kalan özellikleri işletim sisteminin görevi olan kaynak yönetiminin değişik yönlerinin güvenilir kılınmasıdır.

Örneğin uygulamanız bilgisayarınızdaki fiziksel bellekten daha çoğunu kullanır çünkü işletim sistemi diskinizde sayfa dosyası (page file) adı verilen bir dosyayı bellek gibi kullanır. Veriler bu dosya ile bellek arasında gider gelir. Bu işleme sayfaya girmek / çıkmak adı verilir. Peki bu dosyaya herkes okuyup yazabilir mi? Ya da bir uygulamanın verilerinin sayfa dosyasında nerede olduğunu bulabilir miyiz? Bulacak olursak o verileri okuyup bilgi almaya hakkımız olmalı mıdır? Güvenilir bir işletim sisteminde bu dosya korunur. En azından içeriğinin değişmediğinden emin olmamız için bütünlük denetimi yapılır, belki de dosya, içeriğinin okunmasını engellemek için şifreli saklanmalıdır.

Benzeri bir güvenilirlik gereksinimi de iki uygulamanın iletişim kurmasında ortaya çıkar. Uygulamalar karşılıklı güven duymak zorunda değildir. Bu nedenle işletim sisteminin bu uygulamaların güvenilir olup olmadıkları konusunda garanti vermesi gerekir. Ayrıca uygulamalar arasındaki iletişimin de güvenliğe alınması ve kriptografik olarak yürütmesi gerekir. Bütün bunlar çok yoğun bir açık anahtar (public key) kullanımı demektir. Bu da işletim sisteminin kendisine ait yerel bir sertifika otoritesinin (local certificate authority – local CA) bulunması demektir.

Buna ek olarak her bir uygulama kendisine ait olan bellek ve benzeri kaynakların sadece kendisi tarafından kullanılacağından emin

olmalıdır. Bu da işletim sisteminin her bir uygulamayı güvenli bir alanda (secure compartment) çalıştırması demektir. Ancak bunu çok görevlilik (multi tasking) ile karıştırmamak gerekir. Çok görevlilik aynı anda birden fazla uygulama için birden fazla süreç uzayı açılmasına olanak verir ancak bir uygulamanın bir diğerinin alanına erişmesi için düşük seviye yolları da açık bırakır. Güvenli alan mekanizmasında ise alan kriptografik olarak korunur.

Güvenilir işletim sisteminin ayrıca grafik kullanıcı arabirim (GUI) anlamında güvenli olması gerekir. Bugün İnternet bankacılığının korkulu rüyası olan ve klavyede yazdığınız karakterleri kaydeden “keylogger” programlarının engellenmesi için grafik kullanıcı arabirimini oluşturan bileşenlerin (düğmeler, araç çubukları, vs.) dahi güvenilir biçimde çalışması gerekir.

Gördüğünüz gibi güvenilir bir işletim sisteminin özelliklerinin sadece küçük bir alt kümesi bile son derece zor uygulama projelerini getirmektedir.

Microsoft'un Güvenilir Bilişim Vizyonu

Microsoft bir TCPA üyesi olarak kendi vizyonunu belli başlı güvenilir işletim sistemi özellikleri ve sayısal hak yönetimi üzerine kurmuştur. Önce Longhorn, sonra Vista olarak adlandırılan ancak bir türlü piyasaya çıkamayan yeni nesil işletim sistemlerinin bazı temel güvenilir işletim sistemi becerilerine sahip olacağını belirtmektedirler. Örneğin, 2007 yılı içerisinde Vista (veya beta sürümünü) güvenilir yükleme özelliğini kazanacaktır. Daha sonraki (2009-2010 yıllarında çıkacak) işletim sistemlerinin ise daha çok özellik destekleyeceğini tahmin edebiliriz.

Bununla birlikte yaklaşımlarının tepki çekmesine neden olan bir çok sorunları da bulunmaktadır. Bunların en önemlisi, Microsoft'un “güvenilir yama” listesine ek olarak kendisine ait bir “güvenilir yazılım” listesi tutmak istemesidir. Bu işletim sisteminize yükleyeceğiniz yazılımlar için Microsoft'tan onay almanız anlamına gelebilir. Elbette kendinize ait “güvenilir yazılım” listesi oluşturmanız mümkündür ancak kullanıcıların



büyük çoğunluğunun bu teknik beceriye sahip olmayacağını düşünürsek, bu girişim pratikte Microsoft'un dünya yazılım sektörünü yönetmesi anlamına gelebilir.

Microsoft bir yazılım şirketini “güvenilir yazılım” listesine başvuru sürecinde oyalayarak ona çok ciddi zaman kaybettirebilir, hatta hiç kabul etmeyerek şirketi pazardan silebilir. Microsoft'un ticari geçmişinin küçük yazılım şirketlerini yutarak büyümek üzerine kurulu olduğunu düşünürsek bu tür bir mekanizmayı Microsoft'un işletmesi haklı şüpheler uyandıracaktır.

Buna ek olarak açık kaynak kodlu yazılımların bu başvuruyu yapacak makamları olmayabilir. Bir şahsın yazdığı yazılım için bu başvuruyu takip etmesi çok zor olacaktır. Bu da açık kaynak kodlu yazılımların Windows sürümleri üzerinde kullanımını ciddi anlamda zaafa uğratacaktır.

Bununla birlikte piyasaya ilk çıkacak olan güvenilir işletim sisteminin gelişmiş özellikleri nedeni ile pazara hakim bir konuma geçeceği de kesindir. Peki Microsoft'un bu “ilk hamle” (first mover) avantajını kazanması olası mıdır? Benim kişisel görüşüm bunun pek de olası olmadığı yönünde. Microsoft'un işletim sistemi tasarımları pek çok çağdaş özellikler içermekle birlikte, önemli miktarda kod yığını da eski tasarımlarından miras alınmıştır. Bu da işletim sisteminin var olan yapısının güvenilir hale getirilmesinde önemli bir sorun çıkarmakta.

Bunun en iyi örneğini çekirdeğin sanallaştırılması (kernel virtualization) ile soyutlanması tekniğinden görüyoruz. Bu teknik terim ne demektir? Çekirdeğin bu şekilde soyutlanması işletim sisteminin değişik çekirdeklerle çalışabilmesi, hatta aynı anda birden fazla çekirdeğin yüklenebilmesi demektir. Hatta aynı anda çalışan bu çekirdeklere değişik uygulamaları

atayabilirsiniz. Bir çekirdeğin sorun yaşaması, örneğin kilitlenen bir uygulama diğerini etkilemez. Bu sayede oyunlarınızı ve iş uygulamalarınızı hiç korkmadan aynı anda çalıştırabilirsiniz. Hatta teoride aynı anda hem Linux hem Windows açık olabilir.

Microsoft belirli teknik tercihler nedeni ile bir süredir bu tekniği uygulamak istemiyordu. Ancak sonradan güvenilir bir işletim sistemi için bunun gerekli olduğuna ikna oldu. Bu nedenle

kaybettikleri 3 yıla yakın süreden sonra Linux dünyasında şu anda hazır ve çalışan bu teknolojiyi “geliştirmeye” karar verdiler.

Peki Linux?

Linux'un Güvenilir Bilişim alanında son derece önemli avantajları var. Örneğin XEN adındaki çekirdek soyutlama mekanizması, Fedora Core 4 ve Suse 10.0 ile birlikte gelmekte. Kurulum sırasında XEN kurmak istediğinizi belirtirseniz, çekirdeğiniz ile diğer bileşenlerin arasında XEN adındaki soyutlama katmanı yer alıyor. Şimdilik yönetilmesi biraz karışık olsa da çalışan ve sorunsuz çalıştığını bildiğimiz bir teknolojinin olması bize güven vermektedir.

Linux'un işletim sistemi tasarımının bir çok özgür yazılımda olduğu gibi son derece modüler olması da bu alandaki teknik sorunları tek ve büyük bir sorun yerine çok sayıda küçük sorundan oluşan bir gruba dönüştürmektedir.

Bununla birlikte Linux'ta Güvenilir Bilişim uygulamasının nasıl olacağı önemli birkaç soruna da sahiptir. Bunların ilki TCPA ve Microsoft'un DRM odaklı yaklaşımlarına karşı oluşan tepkinin Linux toplumunda aldığı destek nedeni ile Linux'ta bu tekniklerin olmasını “istemeyen” toplumdur.

Örneğin GNU projesini başlatan Richard Stallman, bu özelliklere sahip bir Linux'un GPL sayılmayacağını belirtmiştir. Aslında GPL' in 3 numaralı yeni sürümü DRM yazılımlarının GPL olamayacağını açıkça belirtmektedir. Ancak elbette GPL' in 2 numaralı eski sürümünü kullanmakta sakınca yok. Linux çekirdeğinin ve Linux'un isim babası Linus Torvalds ise, Güvenilir Bilişim taraftarıdır ve gerekirse çekirdeğe patentli kodların bile girmesine izin vereceğini açıklamıştır. Aslında patent konusunda sorun yoktur çünkü IBM başta olmak üzere bir çok firma, patentleri için Linux çekirdeğine sınırsız ve koşulsuz lisans verdiklerini açıklamaya başlamıştır. Güvenilir Bilişim konusundaki önemli patentler de bu kapsamdadır.

Şu an için önümüze baktığımızda, çekirdek sanallaştırılmasının neredeyse tamamlanmasının ardından gelecek olan şey Linux işletim sisteminin kritik bileşenlerinin ve ardından temel bazı servislerin Güvenilir Bilişim altyapılarını kullanması olacaktır.

Bu konuda çok çeşitli ve biri birinden bağımsız çalışmalar sürmektedir.

Suse, Debian ve Gentoo dağıtımlarının güvenilir birer sürümleri için çalışmalar vardır. Ayrıca IBM ve HP de bu sistemlerin kendilerinin sattığı ve zaten TPM bulunduran 20 milyondan fazla kişisel bilgisayarda ve ileride satacakları diğer bilgisayarlarda çalışması için önemli çaba göstermektedir.

Open Trusted Computing Projesi

Açık kaynak kodlu bir Güvenilir Bilişim uyarlamasını hedefleyen projelerden birisi ve belki de en büyüğü Avrupa Birliği tarafından desteklenen ve 2005 Kasım ayı içinde başlayan Open Trusted Computing projesi. HP, IBM, AMD ve TPM üreticisi Infineon gibi önemli şirketlere ek olarak Suse'nin de yer aldığı

Linux'un işletim sistemi tasarımının bir çok özgür yazılımda olduğu gibi son derece modüler olması da bu alandaki teknik sorunları tek ve büyük bir sorun yerine çok sayıda küçük sorundan oluşan bir gruba dönüştürmektedir.

projenin içerisinde Türkiye'den TÜBİTAK UEKAE ve Portakal Teknoloji bulunuyor.

Projenin amacı 2008 sonuna kadar açık kaynak kodlu bir Güvenilir (Trusted) Linux'u çıkarmak. Projenin içerisindeki 23 ortağın her birinin yapacakları farklı farklı. Örneğin Java güvenlik API' lerinin Güvenilir Bilişim alt yapısını kullanacak şekilde yeniden yapılandırılmasından, sayısal hak yönetimi için daha farklı tekniklerin hazırlanmasına kadar bu yazıda saymadığımız ancak gerekli olduğu tartışılmaz bir çok bileşen de üretilecek.

Proje şu anda gereksinim belirleme sürecinde. Linux üzerindeki çok sayıda değişik bileşenin ortak kullanabileceği alt yapıların olacağını öngörürsek, bu gereksinim belirleme sürecinin son derece yoğun bir çalışma olduğunu kabul etmemiz gerekli. Gereksinim belirleme sürecinin sonunda Güvenilir Linux için bir spesifikasyon yazılmış olacak. Daha sonra tüm ortaklar bu spesifikasyona uyarak kendi kısımlarını geliştirmeye başlayacak.

Proje sadece yazılım geliştirmeye odaklı değil. Aynı zamanda üretilen GPL yazılımların ve alt yapıların nasıl kullanılacağını anlatan çok kapsamlı eğitim materyalleri de oluşturulacak. Portakal Teknoloji'nin liderliğini üstlendiği bu kısım bu konuda dünyada yazılan ilk pratiğe odaklı materyal olarak öncü görev üstlenecek. Elbette geliştirilen eğitim materyali de açık lisanslı.

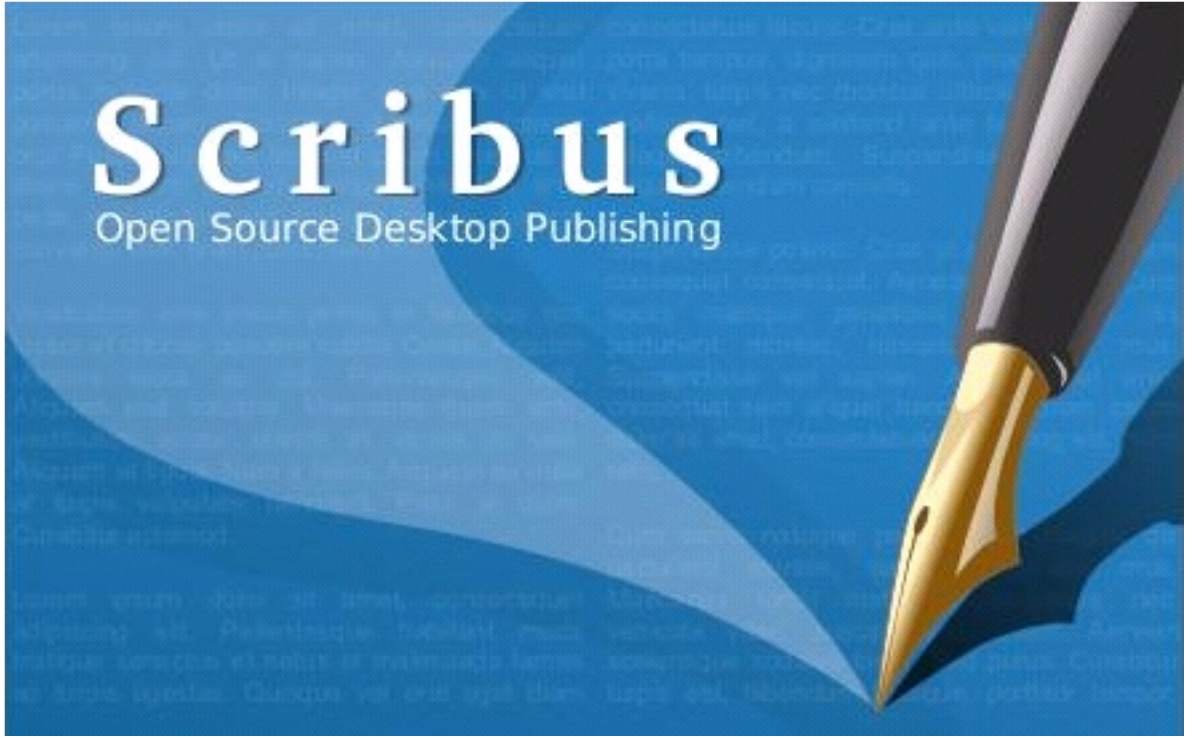
Sonuç

Güvenilir Bilişim, önümüzdeki yıllar içerisinde kaçınılmaz biçimde kullanmaya başlayacağımız bir teknolojiler bütünüdür. Kişilerin ve kurumların istediği güvenilir bilişim uyarlamasını seçmesi ve bu uyarlamanın hangi işlemlerde kullanılacağını

ayarlayabilmesi gerekmektedir. Bu teknolojilerin açık kaynak kodlu ve kişisel özgürlüğe gereken değeri verecek uyarlamalarının olması bu nedenle zorunludur.

Bildirim:

Open Trusted Computing projesi 027635 sözleşme numarası ile Avrupa Komisyonu tarafından desteklenmektedir. Bu yayında sunulan görüşler, proje ortaklarından Portakal Teknoloji ve yazının yazarlarının kendi görüşleri olarak nitelenmelidir. Yazı, diğer proje ortakları ve Avrupa Komisyonu'nun görüşlerini yansıtmak zorunda değildir. Proje hakkında detaylı bilgi edinmek isteyenler, <https://www.trustedcomputinggroup.org/home/> adresindeki proje web sitesini ziyaret



Özgür Masaüstü Yayıncılığın Özgür Aracı

Penguence'nin bu sayısı Pardus 1.0 altında Scribus 1.3.2 ile hazırlanmıştır

penguence

iş modelleri

Sahip Olmak ya da Olmamak

İzlem Gözükeleş

Hardin [1] "Ortaklaşanın Trajedisi" adlı makalesinde bireysel çıkarlar ve ortaklaşa kullanılan mallar arasındaki karşıtlığı tartışır. Sorunun temelinde, bireylerin kamusal malların kullanımı sırasında eylemlerinin sonuçlarını bütünsel olarak değerlendir(e)memesi yatmaktadır. Hardin, tüm çobanların kullanımına açık olan otlak örneğini verir. Bir çoban, rasyonel biri olarak, sürüsüne bir hayvan daha eklemek ister. Ancak otlağa bir hayvan daha eklemenin iki sonucu olacaktır. Birinci olarak, bu eylemden kazanç sağlayan yegane kişi, ekleme işlemi yapan çobandır. İkinci sonuç ise, her eklenen yeni hayvanla, otlağın azalacağıdır. Fakat bu durumda zarar, tüm çobanlarca paylaşılabilecektir. Dolayısıyla, rasyonel çobanımız, kısa bir kar/zarar muhasebesinden sonra sürüsüne bir hayvan daha ekleyecek ve onu diğer rasyonel çobanlar takip edecektir. Sonuç olarak, otlak giderek tükenecek, otlak ve otlağı kullananlar için bir trajedi kaçınılmaz olacaktır.

Hardin makalesinde trajedinin sonlandırılması için bir takım öneriler sunar. Gerek trajedinin kendisi gerekse de önerilen çözümler tartışmalı olsa da Hardin'in temel öncülü, kaynakların kıtlığı, açık kaynak kod (AKK) iş modellerinin kavranması açısından özseidir. Şöyle ki, eğer otlak sınırlı olmasaydı, sonuç yine bir trajedi olur muydu? Hatta daha da ileri gitmeyi göze alarak, otlağa eklenen bir koyunun, otlağı eksiltmediği, ama çoğalttığı gibi bir durumu düşünelim. Herhalde tüm çobanların yüzünün güldüğü bu durumu hiç kimse trajedi olarak adlandıramayacaktı.

İşte bu tezatlıklar yüzünden Smith ve Kollock

[2], GNU/Linux'u "imkansız kamu malı" olarak adlandırmaktadır. GNU/Linux zorunlu sahipsiz ve rakipsizdir. Zorunlu sahipsizdir, çünkü Hardin'in trajedisinin senaryosunda olduğu gibi yazılımı, başkalarının kullanımına engel olmadan sınırsız sayıda kişi kullanabilir. Örneğin herhangi bir kullanıcı GNU/Linux'u Internet'ten indirip kullanabilir. Hatta kaynak koduna sahip olduğundan ona yeni özellikler ekleyebilir ve daha da ileri giderek yeni yazılımın adını Alix ya da Ayşex diye değiştirebilir. Rakipsizdir, çünkü neredeyse sıfır maliyetle (sadece boş cd parası) GNU/Linux'un bir kopyası yaratılabilir. Bir diğer deyişle, GNU/Linux'un çoğaltılmasında kayda değer bir marjinal maliyet yoktur.

"Sahipli yazılım ise adından anlaşılacağı üzere herkesin kullanımına açık değildir; satan şirkete kullanım hakkı için lisans parası ödemeniz gerekir. "

Ancak çoğaltmada sıfır maliyet sahipli yazılımlar için de geçerlidir .

Hardin'in örneğine dönersek, sahipli yazılımda otlağa eklenen yeni bir hayvan otlağı azaltmamakta ya da sınırlı oranda çoğaltmaktadır. AKK'de ise her yeni hayvan (*) otlağın bir geliştiricisi/çoğaltanı olmaktadır.

Fakat AKK'den nasıl para kazanılacağı sorusu hâlâ yanıtsızdır. Dikkat edersek zorunlu sahipsizlik, herkesin AKK yazılımlara sahip olabileceği anlamına gelir. Dolayısıyla, ben A yazılımına sahipsem, benim müşterilerim de A yazılımına sahip olabilir. Peki ben bu durumda nasıl para kazanabilirim? Ali'nin malını Ali'ye satmak üzerine kurulu bir iş modeli ne kadar akılcıdır?

Sorulara yanıt vermeden önce iş (business)'in tanımını yapmak gerekir. İş, her şeyden önce, firma ve müşteri arasındaki bir ilişkidir.

Bu ilişkinin temelinde müşteriye bir değer sağlamak ve karşılığında para kazanmak vardır. Bu nedenle, yazılımın sahipli ya da AKK

olması, işin sürdürülebilirliği açısından birincil derecede önemli değildir. Dolayısıyla iş dünyasında rekabet de müşteriye rakiplerden daha fazla değer sağlayabilmek ile ilgilidir.

Fakat, kaynak kodu, iş modelinin sahipli ya da açık kaynak üzerine kurulu olmasına göre farklı işlevler üstlenir. Örneğin, sahipli yazılımdan para kazanan firmalarda kaynak kodu, hiç kimseyle paylaşılmaması gereken hazine niteliğindedir ve firmalarca gizlenir. Böylece firma, kendisinin sahip olduğu ama diğerlerinin sahip ol(a)madığı kaynak kodu yardımıyla müşteriye rakiplerinden daha çok değer sağlayabilir. Ancak AKK iş modellerinde kaynak kodu herkesten saklanan bir hazine olarak görülmez. Aksine, bu iş modellerinde kaynak kodu paylaşımına açık, sürekli biriken bilgi biçimini alır. Sonuçta, AKK firmasının müşterisine sağladığı değer süreç içinde artar. AKK yazılımların daha güvenli olduğu görüşünün arkasında yatan da yine bu özelliğidir. AKK, paylaşıldıkça, çok sayıda göz tarafından incelendikçe hatalarından (bug) arınır.

Şimdi sorumuza dönebiliriz: AKK üzerine kurulu bir firma, A yazılımına sahipse, AKK'nin doğası gereği bu yazılıma sahip olabilen bir müşteriye nasıl bir değer sağlayıp para kazanabilir?

Aslında sorunun yanıtı "sahip olmak" kavramında yatmaktadır. GNU/Linux'un zorunlu sahihsiz olması onun koşulsuz herkesin olduğu anlamına gelir mi? Şöyle ki, Linux'un kaynak koduna sahip olmam beni Linus Torvalds kadar Linux'un sahibi yapar mı? Projeyi başlatan kişi olmaktan bahsetmiyorum. Şu soru daha açıklayıcı olacaktır: Linux-Apache-Mysql-Php (LAMP) ile çok sayıda uygulama geliştirmiş ve LAMP'ın birçok özelliğini bilenle LAMP'i sadece bilgisayarında bulunduran ona aynı ölçüde sahip midir?

Bu bağlamda sahip olmak soyut olarak değil somut olarak ele alınmalıdır. Bir malı ya da yazılım bağlamında konuşursak uygulamayı, kullandıkça, hatta onu ihtiyaçlarınıza göre uyarlayabildikçe ona sahip olursunuz.

Penguence'nin bu sayısının da dosya konusu olan yerelleştirme (ya da uluslararasılaştırma) de soyut olarak herkesin olanın somut olarak bizim olmasıdır. KDE, GNOME ya da DEBIAN vardılar; ancak yerelleştirme çalışmalarıyla Türkiyeli kullanıcıların olmuşlardır.

Dolayısıyla, bir AKK firması A yazılımını müşterisine satarken,

- i.A yazılımını müşterinin ihtiyaçlara göre değiştirerek,
 - ii.A yazılımına yeni özellikler ekleyerek,
 - iii.Farklı yazılımları A yazılımı ile entegre ederek,
 - iv.A yazılımın kullanımı için eğitim vererek,
 - v.Olası sorunlarda ona destek vererek,
- müşteriyi somut olarak A yazılımın sahibi yapar. Özellikle, birinci ve ikinci maddeler, Özgür Yazılım lisanslı (GPL) yazılımlarda firmaca yapılan katkının tekrar kamuya dönmesiyle yazılımın değerini arttırır.

Özetlersek, sahipli yazılım firmaları, sahip oldukları bilgiyi gizli tutarak müşteriye rakiplerinden daha fazla değer sağlamayı amaçlamaktadırlar. AKK firmalarının iş modeli ise soyut olarak herkesin olanı müşteri için somut hale getirmek üzerine kuruludur. Peki bu iki modelin birbirine göre avantajları ve dezavantajları nelerdir? Firma için hangisi, hangi koşullarda daha karlıdır ? (**)

Bu soruların yanıtlarını da sonraki sayı da vereceğiz.

Notlar:

- * Belkide burada koyun yerine öküz demek daha yerinde olacaktı (bkz. <http://en.wikipedia.org/wiki/Wildebeest>)
- * Toplumsal açıdan Özgür Yazılım daha tercih edilendir. Ancak bu köşenin konusu AKK'den nasıl para kazanılacağıdır.

Kaynaklar:

- [1] Hardin, G. (1968). The tragedy of the commons. Science New Series, Vol. 162, No. 3859,1243-1248.
- [2] Smith, M. A. & Kollock, P. (1999). Communities in Cyberspace (p. 230). London: Routledge.
- [3] Weber, S. (2005). The Success of Open Source. Harvard: Harvard University Press.



Penguence olarak yerelleştirme çalışmalarının Türkiye'de Özgür Yazılım adına yapılan en önemli etkinliklerden olduğunu düşünüyor ve bu nedenle de dosya olarak işlemeyi seçtik. Dosyamız Deniz Akkuş'un yerelleştirme çalışmalarının neler olduğunu açıklayan yazısı ile başlıyor. Ardından bu çalışmaları gerçekleştiren ekipler ile yaptığımız röportajlar ile sürüyor. Sorularımızı cevaplayan, emeği geçen herkese çok teşekkürler.

Sorularımızı Belgeler.orG adına **Nilgün Belma Bugüner**, Gnome Türkiye adına **Barış Çiçek**, Debian Türkiye adına **Recai Oktaş**, OpenOffice Türkiye adına **Rıdvan Can**, Mandriva Tr adına **Ömer Fadıl Usta**, Mozilla Tr adına **Ahmet Serkan Tıratacı**, Gnu-tr adına **Deniz Akkuş**, Pardus adına **Barış Metin**, KDE Türkiye adına **Adem Alp Yıldız** cevapladılar.

Yerelleştirme

Deniz Akkuş

Özgür yazılımlar, ve özellikle Linux işletim sistemi, ağ ortamında hayat buldular ve ağ ortamında gelişmeye devam ettiler. 1984 senesinde Richard Stallman'ın başlattığı Özgür Yazılım Hareketi etrafında çeşitli özgür yazılımlar gelişmeye başladı. Bu yazılımların ortak özelliği, bütün kaynak kodlarının ağ ortamında serbestçe dolaşıyor olması ve geliştiricilerin de ağ ortamında örgütlenmeleri idi. 1991 senesinde Finlandiyalı bir öğrenci olan Linus Torvalds, işletim sisteminin eksik olan parçası olan Linux çekirdeğini yazdı.

Bugün, "Linux işletim sistemi" dediğimiz zaman tamamen özgür yazılımlardan oluşan ve Linux çekirdeğini kullanan işletim sistemlerini kastediyoruz. Bu sistemler ücretsiz olarak Internet'ten alınabiliyor ve imece usulü ile geliştiriliyor.

Linux'un kaynak kodlarının açık olması, Türkiye için stratejik bir önem arz etmektedir. Bilişim teknolojilerinde, kapalı kaynak kodlu modellerde, az sayıda geliştirici haricinde kalan teknik personel sadece satış ve satış sonrası destek konumunda teknisyen statüsünde çalışırlar. Örneğin bir veritabanı yazılımının Türkiye ofisi, veritabanı yazılımının

nasıl işlediğini göremez. Bunda değişiklik yapamaz. Veritabanını paketli bir ürün olarak kabul edip, müşteri satış ve destek hizmetlerini yerine getirir. Yani veritabanının Türkiye'de satılıyor olması, Türkiye'ye veritabanı teknolojisinin aktarıldığı anlamına gelmez. Teknoloji, genellikle Kaliforniya'da olan geliştirme merkezinde saklıdır.

Linux bu kısır döngüyü kırmak ve ülkemize teknoloji aktarımını sağlamak konusunda stratejik bir imkan sunuyor. Veritabanı örneğine geri dönersek, Linux üzerinde mevcut olan veritabanlarında, bütün detayları ile kodları inceleyip, üzerinde değişiklik yapmak mümkün. Bu konuda yapılan tartışmaları değerlendirip çeşitli dizayn seçeneklerinin neden kullanıldığını görmek imkanı mevcut. Kısacası, açık kaynaklı kodlar elinizde olduğu müddetçe, teknisyen statüsünden mühendis statüsüne geçiş imkanınız var. Tabii Türkiye açısından büyük bir problem var: Bütün bu bahsettiğimiz bilgi dağı, İngilizce. Yerelleştirme çalışmaları, bu bilgi dağını Türkçe'leştirmek için yapılıyor.

Yerelleştirme dediğimiz zaman genelde bunu Türkçe arayüz manasında kullanıyoruz. Aslında, yerelleştirme, bir dilin bilgisayar ortamında desteklenmesi için gereken çalışmaların tümünü kapsar. Ve Linux'un imece usulü ile geliştiğini göz önüne alırsak bu şu demektir: Ulusal klavyenizin Linux üzerinde çalışmasını istiyorsanız, bunu yazmak görevi size aittir. "Bu yapılsın" diyebileceğiniz bir merci yoktur.

Bu bakımdan, Linux'un yerelleştirilmesi için gereken bütün işlemlerin üzerinden geçmekte ve yoğunluğunun çok olmasından dolayı genelde yalnızca yazılım çevirileri olarak

algılanan yerelleştirme konusunun neleri kapsadığını incelemekte yarar var:

1. Karakter kümesi:

Karakter kümeniz sistem tarafından tanınmaz ise, ciddi bir problem yaşarsınız. Bu teknik bir problem, fakat geçmişte ı, ğ, ş harfleri yerine boş kutular veya bir takım başka sembollerle karşılaşmış iseniz, sorun bundan kaynaklanır.

2. Klavye:

Günümüz klavyeleri çeşitli tuş kombinezonları ile tuş başına 3 4 ayrı sembol basıyor. Bunların o dilin klavye yapısına göre tanımlı olması gerekli. Türkçe özelinde, gerek Q, gerek F klavyenin seçenekler arasında mevcut olması gerekli.

3. Yazı Tipleri:

Yazı tipi, gene geçmişte ş ı, ğ harflerini görmemize engel teşkil eden başka bir sorun. Her yazı tipi her karakter kümesi için mevcut olmayabiliyor. Dolayısıyla dil desteğinde, sistemin karakter kümesini tanıması yeterli değil, o dili doğru şekilde ekranda gösterebilmek için, yazı tiplerinin de mevcut ve o dilin harflerinin tamamını içerir olması gerekli.

4. Yerel Bilgileri:

Yerel bilgilerinden dile özgü biçimleri anlıyoruz. Ondalık işareti nokta mıdır, virgül mü? Kısa tarih yazımlarında önce ay mı yazılır, yoksa gün mü? Saatler 12 saat biçimine göre mi, 24 saat biçimine göre mi ayarlanır? Alfabe sıralaması nedir? Eğer geçmişte İngilizce alfabede bulunmayan harflerin en sona sıralanması gibi problemlerle karşılaştı iseniz, sorun bundan kaynaklanır.

5. Yerelleşmiş Arayüz:

Arayüzün Türkçe olması, yerelleştirme denince ilk anladığımız şey. Bundan, yazılımın iletilerini Türkçe vermesini ve kullanıcıdan gelen iletileri de Türkçe kabul etmesini anlıyoruz. Buna örnek olarak "Evet/Hayır?" diye soran bir yazılımın cevap olarak "Yes" veya "No" dizgelerini kabul etmesi yerine "Evet" veya "Hayır" dizgelerini kabul etmesini sıralayabiliriz. Bir yazılımın arayüzünün



yerelleştirilmesi için bunların teker teker Türkçeleştirilmesi gerekir.

6. Yerelleşmiş Belgeler:

Çoğu ücretli yazılım dahi, yerelleştirme kapsamına yardım dökümanlarını almaz.

Arayüz Türkçe'dir, fakat herhangi bir yardım dökümanına başvurduğunuz zaman İngilizce olduğunu görürsünüz. Halbuki kullanıcının etkin kılınması ve yazılımın tüm özelliklerini öğrenebilmesi için yardım dökümanları anlayabileceği bir dilde olmalıdır. Başarılı bir yerelleştirme, arayüz kadar yardım dökümanlarının da Türkçe'leştirilmesini kapsar.

7. Dile/Ülkeye Özgü Bilgiler:

En son olarak da dile ve ülkeye özgü bilgilerle yazılımın yerelleştirilmesini ele almak gerekir. Bu bir takvim yazılımı ise, 19 Mayıs'ın tatil olduğunu göstermelidir. Bir muhasebe yazılımı, Türk mevzuatını kapsamalıdır. Bankacılık yazılımı BSMV kesebilmelidir.

Linux Yerelleştirme Çalışmalarında Mevcut Durum:

Karakter kümesi:

Karakter kümesi desteği konusunda, çabalar bireysel. Bütün dilleri kapsayan tek bir karakter kümesi olan Unicode'a geçiş sayesinde bu konudaki problemler azaldı, fakat halen çeşitli yazılımların Türkçe arayüzlerde problem yaşadığını görebiliyoruz. Bu olduğu zaman genelde tespit eden kişinin bireysel çabaları ile programa değişiklik yapılıyor.

Klavye:

Klavye konusunda son bir kaç yıldır epey mesafe kat edildi. Bundan daha önce standart bir Linux sisteminden çalışan bir Türkçe klavye çıkmazdı. İnternet'te olan çeşitli ara çözümleri bulur ve uygulardınız. Bir yerelleştirmeci arkadaşımızın çabaları sayesinde bu problem giderildi ve çalışan Türkçe klavyeler standart dağıtımların içerisine girdi.

Yazı Tipleri:

Batı Avrupa dilleri ile kıyaslandığı zaman Türkçe için çok daha az sayıda yazı tipi mevcut. Bu, Linux'a özgü bir problem değil,

çünkü yazı tipi teknolojileri işletim sistemlerinden oldukça bağımsız.

Fakat, yazı tipi tasarımının hem bilgisayar mühendisliği hem de grafik beceri gerektiren bir iş olması, bu konuda hobi seviyesinde çabaların olmasını zorlaştırıyor. Türk kaynaklı bir yazı tipi çalışması yok.

Yerel Bilgileri:

Türkçe yerel, çeşitli bireylerin çabaları ile olgunlaştı. Buna rağmen zaman zaman çeşitli problemler yaşanabiliyor ve gene bireysel çabalar ile düzeltiliyor.

Yerelleşmiş Belgeler:

Yardım dökümanları, arayüze kıyasla daha geri durumda, fakat kimi çeviri grupları tarafından ele alınıyor. Ayrıca, bir yazılıma bağlı olmaksızın Internet'te bulabileceğiniz, hemen her detay seviyesinde, bir şeylerin nasıl işlediği ile ilgili bilgiler içeren "how-to" ("nasıl") belgelerinin Türkçe'leşmesi konusunda mevcut olan açık, belgeler.org tarafından karşılanıyor.

Dile/Ülkeye Özgü Bilgiler:

Bu konuda henüz çok gerideyiz. Bireysel bazı çabalar mevcut, fakat problem uzayının tümünü hedefleyen kapsamlı çalışmalar henüz yok.

Yerelleşmiş Arayüz:

En fazla ve tutarlı çaba, arayüz yerelleştirmelerinde mevcut. Gnome Türkiye, KDE Türkiye, GNU-TR, Mozilla Türkiye, Türkçe OpenOffice.org, PostgreSQL Türkçe gibi gruplar çeşitli yazılımların ilk Türkçe'leştirmesini ve daha sonra güncel tutulması görevlerini üstlenmiş durumda.

Bu çeviri grupları tamamen gönüllüler tarafından, Internet üzerinde organize olan topluluklar. Birbirlerini yüzyüze çoğu zaman hiç görmüyorlar. Aralarında hukuki herhangi bir bağ yok. Katılımcılar tamamen kendi istekleri üzerine bir araya geliyorlar, organize oluyorlar, görev paylaşımı yapıyorlar ve üzerlerine aldıkları görevleri, kimsenin görmeyeceği bir dosyada "çeviriyi yapan falanca şahıstır" ibaresinden başka bir getiri elde etmeksizin yüzbinlerce satırlık çeviriler yapıyorlar.

Gelecek:

Linux'un önemi, sadece bedava olmasından öte, teknoloji transferini mümkün kılması ve bireysel sorumluluk almak ve gönüllü olmak alışkanlıkları üzerine kurulu bir imece kültürü getirmesi.

Bütün farklı yönleri ile yerelleştirme, Linux dünyasını Türklere açabilmenin tek yolu. Bu konu üzerinde zaman içerisinde bir hayli mesafe katedilmiş olmasına rağmen yapacak çok şey olmaya devam ediyor.



Mozilla - Tr

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi? Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Elbette ki yeterli değil. Kendi ana dillerinde insanların bilgiye erişmesi ve öğrenebilmesi çok daha avantajlı ve daha hızlı oluyor.İnsanlara bu imkanı vermek lazım.

Çalışmaların başarılı olabilmesi için, mesela üniversitelerde Edebiyat dillerinde okuyan öğrencilerin ve öğretmenlerin de katılması ile çevirilerin çok daha kaliteli bir şekilde olacağına inanıyorum. Çünkü şu anda çevirileri yapan arkadaşların, ki bizlerde dahil, çoğu yerde hata yapıyor, deyim yerinde olursa, bazı noktalarda çeviremediğimiz yerleri, cümlelerin gelişine göre ve anlamı bozmadan çevirmeye çalışıyoruz. Bu arada istemeden de olsa cümleleri bazen katlediyor veya yeni kelimeler türetiyoruz. Bu türetilen kelimelerin artık Türkçe'de yer tutması biraz zaman alıyor. Belki bu tür kollarda okuyan arkadaşlar da katılırsa çalışmalara, çok daha kaliteli çeviriler yapılabilir.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : LKD'nin bence Yerelleştirmeye bir faydası olabileceğini sanmıyorum. Sonuçta bu yerelleştirmeyi yapacak olanlar kendi gruplarını kurmuş, öyle ya da böyle yollarını tutmuş ve ellerinden geldiğince çevirileri tamamlamaya çalışıyorlar.İllaki LKD ye bir görev atamak gerekiyorsa, bence bunu, kalite kontrolü yaparak yapabilir.

LKD çerçevesinde kurulacak bir çalışma grubu, mesela çevirileri aktif bir şekilde test edip, bulduğu hataları not alarak yerelleştirme gruplarından neden göstererek düzeltilmesini isteyebilir. Bence bu tür bir rol LKD'ye daha güzel yakışır.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Kim ne derse desin, özgür yazılım için çeviri yapan arkadaşlar bu işi gönül verdikleri için ve hiç bir beklenti içerisinde olmadan yapıyorlar. Yapılan iş gerçek anlamda ne kadar profesyonel bir çalışma olmasa da, ben yapabileceklerinin EN İYİSİNİ yaptıklarına inanıyorum. Sonuçta bu işi yapanların bir çoğu ne Türkçe öğretmeni ne de profesyonel olarak bir çevirmen iş kolunda çalışıyor.Bu sebepten dolayı, bir çok çevirilerde, esasında aynı olması gereken bir çeviri veya bir program, farklı sürümlerden farklı çeviriler ile çıkıyor karşımıza. Bence temel alınması gereken bir sözlük gerek bize. Herkes bu sözlüğü kullanabilmeli ve aynı kelime farklı programlarda farklı olarak kullanılmamalı. Bu da sorun çözmede çok büyük dezavantaja sebep oluyor. Mesela birisine ICQ dan KDE'deki sorunu çözebilmesinde yardımcı olduğunuzu farz edelim. Menülerde siz bir yeri tarif ediyorsunuz, karşı taraftakinin kullandığı KDE sürümünde anlatmak istediğiniz kelimeler yok veya mevcut değil veya başka bir şekilde çevrilmiş.

OpenOffice Türkiye

Soru : OpenOffice Türkiye ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : OpenOffice.org projesinin başından beri ana projenin Türkiye ayağını götürmeye çalışan bir ekibiz. Şu ana kadar çıkan ana sürümlere ve ara sürümlere %100'e yakın arayüz yerelleştirme desteğini başarıyla verdik. Bunun yanında OpenOffice.org'un Türkçe'ye daha uyumlu olması için bazı çalışmalarımız oldu. Zemberek'le beraber OpenOffice.org imla denetim aracına kavuştu.

Otomatik düzeltme seçeneklerinin ve şablonları Türkçe'ye uyarlanması yapıldı. Sitemiz üzerinden ve bazı organizasyonlarla tanıtım ve belgeleme çalışmalarımız devam ediyor. AB 2006'da 1000 kadar OpenOffice.org CD'sini ücretsiz dağıttık. Üyelerimiz 20'den fazla seminerde tüm Türkiye çapında OpenOffice.org'u anlattı.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : Aslında bu konuda net sayı vermek biraz zor. Zaman zaman ekibe dahil olan ve sonra ayrılan birçok kişi oldu. Ama projenin başında bu işin içinde sürekli olan 5-6 kişi var diyebiliriz.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Bu kişiler genellikle bir süre OpenOffice.org kullanan ve memnun kalan insanlar diyebiliriz. Üniversiteye giden ya da üniversite mezunu, dilini iyi kullanan, internet ortamında kolayca iletişim kuran, zamanının bir kısmını bilgisayarda geçirmeye yatkın kişiler, OpenOffice.org yerelleştirme grubunun genelini oluşturuyor.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Ekibe katılmak isteyen bir kişinin kendini öncelikle kendini tanıtmayı ve neden katılmak istediğini öğrenmek istiyoruz. Şunu da belirtmekte yarar bu işin sırf yerelleştirme ayağı olmadığı söylemeliyiz. Bu yüzden diğer alanlarda da bu projeye katkıda da bulunmak isteyen herkese açığız.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Özgür yazılım dünyasında oldukça iyi bir yere sahip olduğumuzu söyleyebilirim. Bu konuda atılan her türlü adım, özgür ve açık kaynak kodlu yazılımların benimsenme sürecini artıracaktır. Bu alanda yapılan çalışmalar 1997 yılına kadar gidiyor, dolayısıyla 9 yıllık bir bilgi birikimimiz var ve geriye dönüp baktığımızda gerçekten önemli bir engeli aşmış ve Linux'u "Türkçe" kullanıyoruz dedirtecek kadar muazzam bir çalışma yapmışız.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Öncelikle başarının ölçütünü belirlemek gerekiyor. Yüzdesel olarak çok yerelleştirme yapmayı bir hedef olarak görüyorsak, daha fazla sayıda kişinin çalışması yeterli olur. Ama ölçütünüz kaliteli ve sürdürülebilir yerelleştirme ise olaya daha farklı yaklaşmamız gerekir. Öncelikle çalışma yapanların sürekliliğini sağlamak gerekiyor. Daha sonra çalışma yapan arkadaşların üzerinde çalıştığı programı az veya çok kullanması gerekir. Tabi bilişim dünyasındaki terminolojiye de biraz yatkın olmak gerekir. Biraz da çalışma yapanların çeşitli uzmanlık alanlarında gelmesi işleri daha da kolaylaştırıyor. Özellikle bu belli bir alan için yazılan programlarda çok işe yarıyor. Mesela bir nota programının bir müzisyen tarafından yerelleştirilmesi gibi.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : İkisi de yazılım olduğuna göre arada fark olmaması gerekir. Ama özgür yazılım dünyasının kendine has yapısı bu işi de farklı kılıyor. Bunu şöyle açıklayabiliriz. Yazılım geliştirme ortamı ve gerektiği avantajlar yazılımı yazan kişiye kolay bir şekilde çoklu dil desteği vermesini sağlıyor. Bu yapıyı oluşturanlar tabi ki yerelleştirme yapmaya yarayan programları da yapmayı unutmamışlar. Bu da herkese kolaylık sağlıyor. Bir çok karmaşık aşamayla uğraşmıyorsunuz.

Mandriva - Tr

Soru : Mandriva Tr ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Mandriva Linux Türkçeleştirme grubu olarak 1998 yılında baştan organize olduk. Ve aynı yıl içinde (yanlış hatırlamıyorsam 9.0 sürümünde) %38 civarından olan Türkçeleştirme yüzdemizi değerli grup arkadaşlarımla %92 civarına kadar yükselttik. (9.0 %91 gibi bir çeviri oranı ile yayınlandı) şahsen Mandrake (yeni ismiyle Mandriva) Linux dağıtımının Türkiyede bu derece tutulmasının ve tercih sebebi olmasının mevcut diğer dağıtımlara oranla biraz daha iyi derecedeki türkçeleştirme çalışmaları olduğu inanışındayım. Tabi bu çalışmalarda Yalnız olmadığımızı sizlerde bilmektesiniz. Bir Linux dağıtımında KDE GNOME ve GNU Programlarının kendi ayrı çeviri grupları mevcuttur. Yani KDE Türkçeleştirme takımının yaptığı bir çeviri Tüm Linux dağıtımlarının Türkçeleştirme oranını artırırken Mandriva Linux'un Türkçeleştirmelerini yapmakta olan bizlerin çevirileri ise sadece Mandriva Linuxta bulunan userdrake , urpmi , rpmdrake gibi Mandrivaya özgü kısımların Türkçeleştirme oranını yükseltmektedirler.

Bir dağıtımın %100 Türkçe olan bilmesi gerçekten zordur. Çünkü bunun için biraz önce de değindiğim gibi GNU , KDE , GNOME , O dağıtıma özel çeviri grubunun gerçekten iyi çalışması gerekmektedir.

Son 2 yıldır benim başıma gelen bir felaket (ailesel) yüzünden çalışmalar durma seviyesine geldi bundan dolayı çeviri oranlarımız görünür bir şekilde düşüşe geçti. Yeni arkadaşların bizlere katılması ile tekrar eski günlerimize en kısa sürede döneceğimize eminim.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : Mandriva Linux çeviri grubumuz şu anda 10 kişilik bir ekibi mevcut ama maalesef 4 5 kişilik aktif bir katılım mevcut.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Ben dahil çeviri yapan birçok arkadaşımız Üniversite öğrencisi konumunda.Ve tabiki Linux özellikle Mandriva Linux dağıtımını kullanan arkadaşlar.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Yerelleştirme çalışmalarına katılmak isteyen arkadaşlarda aradığımız en önemli özellik Linux işletim sistemi üzerinde çalışabileceği bir makinesi olması ve yeterli düzeyde teknik İngilizce bilgisine sahip olması.

Sizlerinde kabul edeceğiniz gibi çeviri bir kitabı okumak nasıl zevk vermiyorsa bizlerde böyle zevksiz bir dağıtım oluşturmamak için birebir çeviri yapmamaya özen gösteriyoruz. Yani çeviriyi Türk milletinin daha iyi anlayabileceği şekilde düzenlemeler yapmaktayız. Bu nedenden ötürü en azından çeviriyi yaptığı paketin kullanım alanıyla yakından bilgili arkadaşları tercih ediyoruz. Demek istediğim mesela grafik işleme programları ile ilgili arkadaşlara bu programlarla haşır neşir arkadaşları görevlendiriyoruz bu sayede o çevirinin anlaşılabilirliğini artırıyoruz.

Çeviriler için genel olarak Kbabel ismi verilen KDE programını kullanmaktayız. Çeviri Grubuna katılacak arkadaşımız Linux veya Hiçbir şekilde Kbabel programı hakkında hiçbir bilgiye sahip olmasa bile yarım saatlik bir eğitimle bu programın ve çevirilerin nasıl yapılacağı konusunda bilgi sahibi olabilmektedir.

http://www.kde.org.tr/index.php?option=com_content&task=view&id=68&Itemid=84)

Nasıl çeviri yapılacağını öğrenen bir arkadaşımız benim ile omerusta@gmail.com adresinden irtibata geçmekte ve kendisine ufak bir alıştırma/test dosyası yollamaktayım. Sonrasında bu arkadaşları çeviri grubu ortak e posta listemize üye etmekteyiz.

<http://www.mandrivalinux.com/l10n/tr.php3> adresinden çeviri yapmayı düşündüğü dosyanın ismini bu e-posta listemizde belirterek çifte dikiş gitmememizi sağlamak ve çevirisi tamamlanan dosyayı yine bana omerusta@gmail.com adresinden geri yollamaktadır.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Türkçeleştirme gruplarının işi gerçekten zor. Bunda her şeyden önce bu işin bir hobi olması ve kişinin maddi bir kazanç gibi bir geliri olmadığı için mevcut grubun üye sayısının sabit tutulamamasının büyük payı var. Türkiyede gerçekten bu konuya gerçekten ciddiyle yaklaşan ve amatör ama profesyonel bir çaba ile Türkçeleştirmenin bir ucundan tutmak için uğraşan onlarca arkadaşımız mevcut.

Yeterlilik konusu biraz zor cevap verilebilecek bir soru benim için. Burada yeterli değil demem durumunda mevcut arkadaşların gayretini boşlamış olurum ama yeterli demem durumunda ise bir gerçeği gizlemiş olurum.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Yerelleştirme çalışmalarımızda özellikle yeni olan arkadaşların belirli bir tecrübeye ulaşana kadar belirli bir zaman geçmekte. Bu arkadaşların çevirileri aramızdan belirlediğimiz bir arkadaşımız (genelde o paketin en son çevirmeni) tarafından bir kere daha üstünden geçilip hatalı veya değiştirilmesi gereken yerleri bu yeni arkadaşımıza bildiriyor bu şekilde yeni gelen arkadaşımız hem kendisini geliştiriyor hemde daha oturmuş bir çeviriye ulaşmamız sağlanmış oluyor.

Yerelleştirme çalışmalarında daha başarılı olunması için ortak bir bilişim sözlüğü kullanmaktayız. (http://www.kde.org.tr/index.php?option=com_glossary&func=display&letter=All&Itemid=72&catid=69&page=1) bu sayede programdan programa farklılaşan çevirilerden kurtulmuş oluyoruz.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : LKD gerçekten bizlere sonuna kadar destek veren ender gruplardan biri.

Linux Kullanıcıları Derneği gerek seminerlerde gerekse fuarlarda ayarladıkları form ve bildirilerle bizleri çeviri yapmak isteyen veya programlama bilmese bile bir şekilde Linux'un gelişiminde rol oynamak isteyen kişilerle bizleri buluşturmaktadır.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Özgür yazılımlardaki en büyük avantajımız çevirisini yaptığımız dosyanın içindeki bir çevrilmemiş metnin nerede ve ne amaçla kullanıldığını o projenin kaynak kodlarına bakarak veya programcılar ile hızlıca iletişim kullanarak öğrenebilmemizdir. Dezavantaj olarak gösterebileceğimiz bir tek maddi bir kazancınızın bulunmaması olabilir.

Özgür olmayan projelerin çevirilerinde karşılaştığım en büyük dezavantajda aynı konu ile ilişkili. Herhalde çeviriyi yaptığınız projenin lisanslı bir şeklinin size hediye edilmesi avantajlarından sayılabilir.

Debian - Tr

Soru : Debian Tr ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Debian Türkçe yerelleştirme grubu (debian-10n-tr) Debian dağıtımına eksiksiz Türkçe desteği kazandırmayı hedefleyen bir proje grubudur. Projenin genel gayesi Türkçe-NASIL belgesinde de ifade edildiği gibi Debian kurulu sisteminizle Türkçe iletişim kurmanızı sağlamaktır. Bu hedef daha üst bir düzlemde GNU/Linux'un Türkçe desteğiyle ilintili olduğundan yürütülmesi zorunlu faaliyetler sayı ve hacim itibarıyla çok geniş bir dağılım gösterecektir. Bu yüzden debian-10n-tr grubu olarak proje kapsamında sorumlu olduğumuzu düşündüğümüz alanlar daha ziyade Debian'a özgü olması yönüyle belirlenmektedir. Debian yerelleştirme çalışmalarının amaç ve kapsamı aşağıdaki genel ağ sayfasında ayrıntılı şekilde açıklanmıştır: <http://10n-turkish.alioth.debian.org/about.html>

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : Katkı sağlayanların listelendiği sayfaya[1] göre şu ana dek 11 kişi katkıda bulunmuş. Bu sayfa bir süredir yenilenmediğinden bu rakamı 15 civarı olarak verebiliriz.
[1]<http://10nturkish.alioth.debian.org/contrib.html>

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Önemli bir kısmı üniversitelerde okuyan öğrencilerden oluşuyor. Bir miktar bilişim profesyoneli ve üniversite öğretim görevlisi de mevcut.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Debian yerelleştirme çalışmaları bir e-posta listesi üzerinden yürütülüyor: <http://lists.debian.org/debian-10n-turkish>

Bu listeye üye olarak bizimle iletişim kurmaları yeterli. Daha ayrıntılı bilgi için aşağıdaki sayfaya bakılabilir: <http://lists.debian.org/debian-10n-turkish>

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Türkiye'deki yerelleştirme çalışmalarının durumunu nicelik açısından tatmin edici buluyorum. Çevirilerin kalitesini arttırmamız lâzım (bk. bir sonraki cevap). Yerelleştirme grupları son yıllarda çok güzel organize olmaya başladı. Bu gelişim sürecinde çeviri kalitesi de zamanla artacaktır diye umuyorum.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Proje grubu içinde ve proje grupları arasında koordinasyon temel şart. Özellikle gruplar arasındaki koordinasyonu (meselâ ortak bir e-posta listesi kullanarak) arttırmamız lâzım. Bu koordinasyon şimdilik bir kaç projeye birden katkı sağlayan üyeler vasıtasıyla yürüyor görünümünde. Eksikliğini gözlediğim bir diğer nokta tamamlanmış çevirilerin de tıpkı bir programın gelişiminde olmazsa olmaz nevinden yapılan kod temizliğine benzer bir şekilde iyileştirilmeye çalışılmaması. Kod bakımıyla benzeştirecek olursak bazı ilginç örnekler verilebilir. Meselâ "refactoring" ile, program davranışı değişmeksizin, aynı davranış daha doğru, daha temiz bir mantıkla yeniden kodlanır. Buna benzer şekilde çevirilerde (bir tür "translation refactoring" :-)) yapılarak aynı anlam daha temiz, daha az çeviri kokan bir üslupla ve kelime seçimleriyle ifade edilebilir. Bu tür çeviri bakımlarının sürekli ve çok agresif şekilde yapılması gerektiği inancındayım. Ortak bir koordinasyon listesinin (veya ortak bir Türkçe çeviri hata veritabanının) bu kalite noktasında pratik bir yararı da olacaktır. Meselâ bir çeviride hatalı bir nokta tespit ettiğimde birkaç adımlık bir prosedür uygulamak zorundayım. "Son çevirmeni bul", olmadı "ilgili grubu bul" gibi. Az çok işlerin nasıl yürüdüğünü bilen birisi olarak benim için bile yorucu olabiliyor

bu. Ortak bir liste olması halinde, en azından listeye üye proje koordinatörleri sayesinde, basitçe ortak listeye irtibata geçmek yeterli olabilirdi.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : Yukarıda bahsettiğim proje grupları arasındaki koordinasyon için LKD'nin uygun bir platform oluşturduğunu düşünüyorum.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Şöyle bir etrafınıza bakın. Yaygınlık kazanmış özgür olmayan bir işletim sistemi ve ona eşlik eden ofis yazılımı dışında yerelleştirmesi mevcut kaç tane yazılım var? Özgür yazılım aynı zamanda "seçenek zenginliği veya özgürlüğü" demektir. Bu açıdan bakıldığında özgür yazılımın açık ara önde olduğunu görüyorum, kendi dilimizde kullanabileceğimiz irili ufaklı o kadar çok özgür

Pardus

Soru : Pardus ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Pardus'un bir ürünü olduğu Uludağ Projesi'ne başladığımız günden itibaren özgür yazılım gelişim metodolojisi ile çalışmaya karar vermiştik. Bu bağlamda "Her işi biz yapacağız!" fikri yerine, "Özgür yazılım topluluğu ile çalışacağız!" fikrini benimsedik. Pardus üzerindeki özgür yazılımların yerelleştirme çalışmaları, yazılımların Türkçe çeviri/yerelleştirme grupları tarafından yapılıyor. Biz Pardus geliştiricileri olarak bu yerelleştirme grupları ile etkileşimli olarak çalışıyoruz. Yazılımların Türkçe kullanımındaki sorunlarını tespit ederek bunları çözmeye çalışıyoruz. Tabi, Pardus geliştiricilerinden bazıları çeviri çalışmalarına doğrudan yardımcı oluyor. Bunların yanında, Türkçe çevirilerine destek olabilmek için ortak sözlük projesi başattık. sozluk@jabber.uludag.org.tr Jabber adresinden de yayın yapan sözlüğün bu konuda geliştirilebilecek bir adım olduğunu düşünüyoruz.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : Pardus geliştiricilerinin çoğu, yalnızca doğrusal problem alanları üzerinde çalışmıyor. Bunun yerine, farklı zamanlarda, pek çok farklı alanda çalışıyorlar. Dolayısı ile, Pardus özelinde, yerelleştirme çalışmaları ve Türkçe problemleri konusunda yalnızca bir kaç geliştiricinin ismini vermek doğru olmayacaktır.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Uludağ Projesi çalışanı Pardus geliştiricileri, uzun zamandır özgür yazılımlar üzerinde çalışan yazılım geliştiricileri. Gönüllü olarak projeye destek veren geliştiricilerin kimi de öyle...Söylediğim gibi, çoğu geliştiricinin yerelleştirme çalışmalarında da aktif olduğunu düşündüğümüzde profili "Türkiye'de, Türkçe özgür yazılımın geleceğine inanan, özgür yazılım kullanımını destekleyen gönüllü geliştiriciler" kadar genel tutabiliriz.

Soru: Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Pardus özelinde bizim de bazı çeviri/yerelleştirme ihtiyaçlarımız oluyor. Pardus projesi aracılığı ile yerelleştirme çalışmalarına katılmak isteyenleri genellikle yerelleştirme gruplarına yönlendiriyoruz. Tabi, Pardus perspektifinden bakarak, Pardus ile öntanımlı olarak dağıtılan yazılımların Türkçe'ye çevirilmesi bizim için önemli oluyor. Fakat, pek çok durumda yalnızca yerelleştirme gruplarının listesini (<http://www.linux.org.tr/index.php?Pg=Yerellestirme>) işaret etmek ile yetiniyoruz.

Yanıt : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yerelleştirme çalışmaları büyük bir özveri gerektiriyor. Türkçe çevirileri konusunda çalışan yerelleştirme gruplarının da bu özveriyi gösterdiğini rahatlıkla söyleyebiliriz. Fakat daha fazla yerelleştiriciye, çevirilerin kalitesini ve bütünlüğünü kontrol edecek daha fazla insana ihtiyacımız var.Yalnızca bir defaya mahsus olarak, mevcut çevirilerin bir çeviri bürosu tarafından yapılmasının yüz binlerce YTL'ye mal olacağını biliyoruz. Bu işin sürekli olarak yapılması gereken bir iş olduğunu düşünürseniz maliyetin astronomik olduğu sonucuna rahatlıkla varabiliriz. Öte yandan, gönüllülük esasını da düşünerek, şu anda yerelleştirme grupları tarafından yapılan işlerin bizler için paha biçilemez olduğunu söyleyebiliriz.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Her şeyden önce, sürekliliğin esas olduğunu vurgulamak gerekiyor. Özgür yazılımlar her geçen gün yeni özellikler eklenerek geliyor. Gelişen yazılımların yerel desteklerinin de sıklıkla güncellenmesi ve kontrol edilmesi gerekiyor.Diğer bir önemli kriter ise bütünlüktür sanırım. Uygulamalarda kullanılan terimlerin bütün ve tutarlı bir dil ile çevirilmesi gerekiyor. Bu, tüm çeviri grupları tarafından kullanılacak ortak bir sözlüğün önemi de gösteriliyor.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : LKD'nin her zaman olduğu gibi özgür yazılımların yerelleştirme çalışmalarında da birleştirici ve destekleyici bir pozisyonda olması gerekiyor. Yerelleştirme çalışmalarının ihtiyaç duyduklarında her türlü destek için baş vurabilecekleri bir mecra... LKD'nin şu anki duruşu ile bu görevini yerine getirdiğini düşünüyorum.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Uygulamaları yoğun olarak kullanan ve kimi zaman geliştiren kişilerin çeviri çalışmalarında da aktif rol oynuyor olmaları özgür yazılım çevirilerinin bir avantajı. Çevirileri hatalarına çok hızlı bir şekilde müdahale edebiliyor olmak ise başka bir avantaj sanırım.Gönüllü yapılan çeviri çalışmalarının bütünlüğünün ve sürekliliğinin sağlanabilmesindeki zorluğu ise dezavantajları olarak listeyelebiliriz.Aynı cümlelerin karşıtlarını ise özgür olmayan (sahipli) yazılımlar için kurabiliriz.



KDE Türkiye

Soru : KDE Türkiye ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : KDE Türkiye, Türkçe konuşan yazan insanların buluşup sorunlarını giderebilecekleri bir platform olarak düşündük. Bunu yapmamızdaki asıl amaç yerelleştirme yaparken daha fazla geri bildirim almayı kolaylaştırmak ve çeviri yapan insan sayısını artırmaktı. Bu hedefimize şuanda ulaşabilmiş değiliz. Doğruyu söylemek gerekirse yakın zamanda da ulaşmak mümkün olmayacak ama umutlarımız asla yitmeyecek.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : KDE Türkiye'de yaklaşık olarak 20 kişi var. Fakat bu sayıyı sürekli olarak etkingörmek mümkün değil. Gönüllü bir proje olduğumuz için devamlı olarak çeviri yapan arkadaşımız yok. Haftalık olarak düşünürsek 3-4 arkadaşımız etkin olarak çalışıyor.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Linux ile yeni tanışmış olup yerelleştirmeyi yeterli görmeyip yardıma koşanlar ve bilgisayar mühendisliği ya da benzeri bölümlerde okuyan öğrenci arkadaşlarımız.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Yardım etmek isteyen kişi aşağıdaki sorulara evet cevabı verebiliyorsa, <http://www.kde.org.tr/mailman/listinfo/yerellestirme> adresinden yerelleştirme e-posta listesine üye olup kendini tanıtan bir e-posta göndererek işe başlamalı.

- * Akıcı ve anlamlı bir şekilde yazabiliyorsanız,
- * Daha önce çeviri yaptıysanız,
- * Herhangi bir dergi ya da makaleye yazı gönderdiyseniz, ya da teknik yazarlık yapmışsanız,
- * Kendi dilinizdeki teknik terimlerle aranıza iyi ise,
- * Linux'u evde ya da iş yerinde kullanıyorsanız,
- * KDE masaüstü ortamında çalışıyorsanız,
- * Linux ve İnternet terimlerine yatkınsanız,
- * Bu çalışma için ayda 4-5 saatinizi ayırabiliyorsanız,
- * Size ulaşabileceğimiz bir e-posta adresiniz varsa,

Bu e-postadan sonra kendisine deneme amaçlı bir dosya göndererek olası hataları engellemek amacıyla kendisini deniyoruz. Testten geçerse çevirilere başlıyoruz.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Tam olarak yeterli ya da yetersiz diyemeyeceğim. Kesin olan bir şey daha fazla insan, yani yardıma ihtiyacımız var. Çeviri ile iş bitmiyor. Çevirilerin anlamlı ya da yeterli olduğunun da kontrol edilmesi gerek.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Aslında daha fazla insana ihtiyaç var ama mevcut olan potansiyeli düzgün kullanmak daha kaliteli sonuçlar doğurur. Mesela haftalık yerelleştirme çalışmaları yapılabilir. Topluca yapılacak çalışmalar daha başarılı olacaktır.Örneğin; her hafta cuma akşamı 2 saat bir ortamda (IRC) toplanılarak çalışma yapılabilir.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : LKD, tüm yerelleştirme gruplarını bir araya toplayabilir. Tüm gruplarda Türkçe dilbilimcisine

ihtiyaç var. Her ne kadar çalışmaları ön kontrol ediyorsak da yeterli değil. Kalite kontrollerinin yapılabilmesi için Türkçe konusunda yardıma ihtiyaç var. Bu durumda LKD belki bu eksikliği giderebilir.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Özgür yazılımların yerelleştirmesini yapan arkadaşlarımızın hiç biri tam zamanlı bu işle uğraşmıyor. Adı üzerinde gönüllü olarak yapılan bir çalışma. Özgür olmayan yazılımlar ise tam zamanlı olarak bu iş üzerinde yoğunlaşan insanlar tarafından yapılıyor.

Gnome Türkiye

Soru : Gnome Türkiye ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Gnome Türkiye 2001 yılında GNOME projesi içerisindeki yazılımları Türkçeleştirmek için GNOME Translation Project altında kurulmuş bir topluluktur. 1,5 sene önce GNOME Türkiye, Türkiye'de GNOME'u yaygınlaştırmak ve Türk GNOME kullanıcılarının buluşma noktası olmayı hedefleyen bir çatı oluşturmayı hedefledi. Bugün GNOME Türkiye altında öncelikli olarak GNOME uygulamalarının arayüzlerinin , uygulama yardım ve belgelerinin, geliştirici belgelerinin Türkçeleştirilmesi, GNOME içinde olası Türkçe diline olan uyumsuzlukların giderilmesi, web sayfamız üzerinden GNOME ile ilgili haberlerin iletilmesi gibi faaliyetler gerçekleştirir. Ancak GNOME Türkiye kullanıcı güdümlü bir projedir. Örneğin Pardus projesi ile Pardus'un resmi olarak GNOME desteklememesi üzerine GNOME Türkiye katılımcıları Pardus için GNOME paketlerini ve uyumunu sağlamak için Gaplan (www.gaplan.org) projesini başlatmıştır. Bu gibi çeşitli yaklaşımlar kullanıcılar tarafından geldikçe GNOME Türkiye altında bu tip faaliyetlerde gerçekleşmektedir.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : GNOME Türkiye çalışmalarında kimin ne kadar iş yaptığına ya da yapmadığına bakmıyoruz. Bizim için en küçük katkının bile önemi çok büyük. Ancak kabaca her çeviri döneminde aktif olarak 10 ile 15 kişinin çalıştığını söyleyebiliriz. Bu rakam şimdiye kadar olan çevirileri içine aldığımızda oldukça büyüktür fakat daha önce de belirttiğim gibi GNOME Türkiye içinde kişilerden daha çok ürüne ve ürünün kalitesine önem veriyoruz.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Öncelikle hepsi Türkçe'yi kullanmayı çok iyi bilen eğitimli kişilerden oluşuyor. Çeviri takımındaki ve GNOME Türkiye'ye katkıda bulunanların bir çoğu bilişim sektörü içinde farklı şirketlerde çalışıyorlar. Bunun dışında yurt içinde ve yurt dışında öğrenim görmekte olan ve üniversitelerde öğretim görevlisi olan katılımcılar da var.

GNOME Türkçeleştirme çalışmaları rakamsal ve kalite yönünden her geçen gün daha da iyi noktaya ulaşıyor. Açık Kaynak Kodlu yazılımların genel özelliği olan herkesin katılım gösterebilmesi bu noktada genel profilde de sürekli bir yükseliş, konunun uzmanlarının katılımlarını da beraberinde getiriyor. Söz gelimi grafik ile alakalı bir konuda konunun uzmanı bir katılımcı hatayı dile getirebiliyor ve hata hızlıca giderilebiliyor. Bu hemen hemen her alanda karşılaştığımız bir durum.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Yerelleştirme çalışmaları herkese açık ortamlarda gerçekleşiyor. Bu açıdan katılım çoğu insanın sandığından daha kolay ancak önemli bir sorumluluk gerektiriyor. GNOME çevirilerine katılmak için öncelikle posta listemize üye olarak varlığınızı herkese duyurmanız gerekiyor. Daha sonra web sayfamızda yer alan belli başlı biraz teknik belgeleri okuyarak fikir sahibi olmanız

istiyoruz. Daha sonra katılmak isteyen arkadaşımızla bir modül seçerek ilk çevirisini etkileşimli bir şekilde yapmasını sağlıyoruz. Bu noktadan sonra GNOME çevirilerine katılmış olunuyor zaten. Çeviri çalışmalarına katılmak için bilgiler www.gnome.org.tr resmi adresimizde de detaylı bir şekilde yer alıyor.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Türkçe gibi 150 milyona yakın bir insanın konuştuğu bir dilden bahsettiğimiz için yeterlidir demek güç. Ancak her geçen gün daha iyiye gittiğini de söylemek gerekiyor.

Pardus gibi yerel dağıtımların da ortaya çıkmasıyla bu çalışmaların daha da hız kazanacağını düşünüyorum. Fakat gene de İnternet üzerinde ya da kütüphanelerimizde sadece Türkçe bilerek bilişim konusunda asgari bilgileri edinmek mümkün olana kadar da bu çalışmaların yeterli olduğunu söylemeyiz.

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Öncelikle bu işin önemini çok iyi idrak etmek gerekiyor. Türkçeleştirme çalışmaları esasında özellikle Açık Kaynaklı yazılımlar için yazılımların başka bir coğrafyada hayat bulması demek. Türkçe çevirisi olmayan bir GNOME masaüstünün Türkiye'de yaygın olarak kullanılması olası değil, hele ki rakipleri zaten Türkçe ise. Bu işin önemini gerçekten anladıktan sonra başarı da beraberinde gelir. Çeviri çalışmaları için belirli bir Ar-Ge ödeneğine, ya da donanım desteğine ihtiyacımız yok, tek gereken kaliteli insan gücü, bu noktada yapılacak en önemli şey çeviri çalışmalarının her ortamda dile getirilmesi, insanlara bu çalışmalara katılarak yapacakları katkının önemi anlatılması.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

Yanıt : LKD'nin çeviri çalışmalarının önemini anlamış olmasından dolayı zaten belirli bir rol aldığını düşünüyorum. Bizim listemize katılan arkadaşların çoğunun LKD listelerine de üye olduklarını biliyorum. Bu zaten bu çalışmalar için LKD'nin bilgilendirme rolünü üstlendiğini gösteriyor. Bunun dışında Yılın Penguenleri ödülünde yılın başarılı yerelleştirmecisi dalının bulunması da bu noktadaki çabanın en büyük göstergelerinden birisi.

Bunun dışında LKD bireysel olarak yapılması zor olan noktalarda rol oynamasını kurumsal kimlik olması sebebiyle daha da arttırmalı diye düşünüyorum. Bu biraz Linux'un ticari rakiplerine karşı bir kurumun durması gerekliliği ile alakalı. Mesela LKD özel bir araştırma şirketine Linux ile Diğer ticari yazılımların Türkçe desteklerini araştırarak bu sonuçları kamu oyuna sunabilir. Özetle işin biraz da pazarlama tarafında yer alması herkesin ortak amacı olan Linux'un yayılmasına büyük bir katkı sağlayacaktır.

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : En büyük avantajı çevirilerin de Özgür Yazılım olmasıdır herhalde. Özgür Yazılımın verdiği tüm artılardan çeviriler de faydalaniyor. Gerek hataların hızlı bir şekilde düzeltilmesi, gerek işin uzmanlarının maddiyattan ziyade manevi bir beklenti ile katkıda bulunması gibi.

Olaya yerelleştirme çalışması olarak baktığımızda ise, işin ticari boyutunun dışarıda kalması sebebiyle, özgür olmayan yazılımlarda kendi dillerini kullanmalarının hiç mümkün olamayacağı milletlerin, kendi dilleri ile bilgisayar kullanabilme imkanları olduğunu görüyoruz.

Bunun dışında görülebilecek tek dezavantaj özgür olmayan yazılımların çeviri kalitesi olabilir. Bazı özgür yazılımlarda kontrolsüz çeviriler kaçınılmaz olarak yer bulabiliyor. Her ne kadar özgür olmayan yazılımlarda bu konuda sorunsuz olmasa da GNOME, KDE, OpenOffice.org ve benzeri kurumsallaşmamış uygulamalarda çeviri sorunlarına rastlamak mümkün. Ancak Özgür Yazılım dünyasında büyük bir kesimde bahsettiğim gibi kurumsallaşmış çeviri ekipleri kendi iç kalite güvence sistemleri ile bu sorunun da üstesinden senelerdir geliyor.

belgeler.org

Soru : belgeler.org ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Görev Alanımız:

1. Nasıl belgelerinin dilimize çevrilmesi,
 2. Kılavuz ve Bilgi Sayfaları ile diğer uygulama yardım belgelerini dilimize çevrilmesi
 3. Linux kullanımını kolaylaştıracak yeni belgelerin hazırlanması,
 4. Yukarıda anılan konularla ilgili daha önce yapılmış çalışmaların toplanması,
 5. Tüm çalışmalarımızın sanalyöremizde yayınlanması,
 6. Linux Bilişim Terimleri Sözlüğünün geliştirilmesi ve sanalyöremizde yayınlanması
- Bu liste ilk günden beri başsayfamızda yeralır.

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : Katılım gönüllülük esasına dayandığından herkes potansiyel üyemizdir.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl? Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : İnisiyatif sahibi olmayan bir gönüllü düşünemiyorum.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Kendilerine saygıları bu kadarsa, yeterlidir...

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Genel başarı daha çok eğitimin hangi mecrada yürüdüğüne bağlı aslında; eğitim ulusçu mu yoksa küreselci mi? Ulusal Eğitim sistemimiz küresel köleler yetiştirmeye yönelikse dua etmekten başka yapacak birşey yoktur.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz ?

Yanıt : O ödül mekanizmalarını kaldırmakla başlayabilirsiniz. Bir avuç oy vermekten başka birşey yapmayan insan benim gönüllümden daha mı yüce? Üstelik bir kişiyi yücelttiğinizi sanırken bin gönüllüyü kırılıyorsunuz... Gönüllüler sadece kendileri ile yarışan insanlardır. Bu büyükenmelerinizi bir kenara kaldırırsanız, bu işler daha iyi yürür. Emek harcayanlara bir teşekkür edin yeter. Basit: adresine bir posta atın ve yerelleştirme çalışmalarına katkıda bulunduğu için teşekkür edin. Yılbaşı tebriği niyetine her birine yılda bir kere bir teşekkür iletisi atsanız o ödülde daha çok makbule geçer.

Hep (ayağınıza) gelsinler istersiniz, bir de gitmeyi denesiniz...

Soru : Özgür yazılımlar için yerelleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : İkincisinin çevirisinin nasıl yapıldığını bilmediğimden yorum yapamam.

Gnu-tr

Soru : Gnu-tr ve yaptığınız yerelleştirme çalışmalarını hakkında kısaca bilgi verir misiniz?

Yanıt : Gnu-tr (<http://gnu-tr.sourceforge.net>) Translation Project'i (<http://iro.umontreal.ca/translation/HTML>), Türkçe ayağı. Bu proje, ağırlıklı olarak GNU/FSF'in komut satırı yazılımları olmak üzere pek çok yazılıma yerelleştirme hizmeti veriyor

Soru : Bu çalışmalarda kaç kişi yer alıyor?

Yanıt : GNU-TR'de yer alabilmek için FSF'e, yaptığınız çeviriler üzerindeki telif haklarınızı FSF'e bağışladığınıza dair bir feragatname vermeniz gerekiyor. Bu feragatnameyi FSF'e göndermiş olan katılımcı adedimiz 65. Yaklaşık 20 kişi, bugüne kadar bir tercümeyi üstlenmiş ve bitirmiş. 3-4 kişi ise çok sayıda yazılımı devamlı olarak çeviriyor. Aslında GNU-TR için en uygun katılım biçimi, bir kişinin, tercihen sık kullandığı bir yazılımı üstlenip, onun çevirisini sürekli güncel tutması.

Soru : Yerelleştirme çalışmalarını yapan kişilerin genel profili nasıl?

Yanıt : Ağırlıklı olarak bilgisayar veya mühendislik dallarında öğrenci.

Soru : Yerelleştirme çalışmalarına katılmak isteyen biri ne yapmalı?

Yanıt : Tercihen kullandığı, önem verdiği bir yazılımın yerelleştirmesinden başlamalı. Bu, gerek çeviri kalitesini yükseltmek, gerekse kişinin kendi kullandığı bir şeye ekleme yapmasından daha büyük zevk almasından dolayı daha verimli bir başlangıç noktası. Yerelleştirmek istenen yazılımı tespit ettikten sonra o yazılımın yerelleştirmesini hangi grubun yaptığını araştırarak o gruba katılmak istendiği bildirilmeli ve o grubun başlangıç önerileri izlenmeli. Tabii yerelleştirme sadece yazılım çevirilerinden ibaret değil. Eğer çeviri yapmak istemiyor iseniz, takvim yazılımlarına Türk tatil günlerini eklemekten, yazıtipi güncellemeye kadar Türkiye'de kullanım ve Türkçe kullanım ile ilgili pek çok eksik üzerinde çalışmanız mümkün.

Soru : Sizce Türkiye'de yapılan yerelleştirme çalışmaları yeterli mi?

Yanıt : Yazılım çevirileri oldukça iyi bir noktada fakat unutmamak gereken bir şey var: Özellikle yazılım ve dökümantasyon çevirileri sürekli güncelleme ister. Bu bir maraton koşusudur. Bizim GNU-TR'de en fazla problem yaşadığımız konu da budur: Biz, katılımcının bir yazılımı bundan sonrası için üstlenmesini istiyoruz -- yani yazılımın her yeni sürümünde çevirinin elden geçirilerek eklemelerin yapılmasının sağlanması. Genelde devamlılıkta problem yaşıyoruz. Bu da gerek çeviri kalitesini, gerekse yapılmış işlerin düzeyini etkiliyor -- çeviri güncellenmediği zaman üç beş sürüm sonra yazılım tekrar hiç çevrilmemiş haline geri dönebiliyor.

Yine sevindirici bir nokta, artık klavye ve karakter kümesi tanımlarının Türkçe için problem olmaması.

Bunun haricinde kalan yerelleştirme seviyesi pek yeterli değil. Hangi takvim yazılımı size 19 Mayıs'ın tatil olduğunu gösteriyor? İstedığınız yazı tipinde "ğ" harfi mevcut mu? Türk mevzuatını destekleyen muhasebe yazılımı bulabiliyor musunuz?

Soru : Yerelleştirme çalışmalarının daha başarılı olması için neler yapılmalıdır?

Yanıt : Çeviriler kapsamında devamlılık, devamlılık, devamlılık. Bundan ömrünüzün çok önemli bir kısmını yerelleştirmeye ayırmayı kastetmiyorum. 200 iletlik bir yazılımı üstlenmeyi ve her altı ayda bir o yazılımın yeni sürümü çıktığı zaman gözden geçirerek eklenen 3 iletiyi vakitlice çevirerek göndermekten bahsediyorum. Ve bunu yıllar boyunca yapmaktan tabii....

Yine çevirilerde, dil kalitesi konusunda meraklı ve titiz olunmalıdır. Örneğin bugüne kadar pek çok

yazılımın çevirisini yapmama rağmen somut ve kullanılabilir çeviri önerisi/hata bildirimi almış değilim. Zaman zaman "ben İngilizce'sini daha iyi anlıyorum, ne bu böyle" gibilerinden yazanlar olur, fakat kimse "şu yazılımın falanca dizgesinde anlam kayması, cümle düşüklüğü var, doğrusu şu olacak" diye yazmış değildir. Bu, bütün kullanıcılara düşen bir görev.

Merak ve titizlik özgür yazılım kültürünün içselleştirmemiz gereken bir unsuru. Elinize yerleştireceğiniz takvim yazılımı geldiği zaman, yerleştirirken sadece 4 Temmuz, ABD'nin Bağımsızlık Günü diye çeviri yapmayacaksınız, Türk resmi tatilleri bunun içinde mi? diye bakacaksınız, yoksa nasıl ekleneceğini merak edeceksiniz, takvimin kullanıldığı iş günü/tatil günü kavramlarına doğru oturmasını sağlayacaksınız, acaba Kurban bayramı ve Ramazan bayramı tarihleri nasıl hesaplanıyor diye merak edeceksiniz, oradan güneş takvimine bağlı olmayan tarihlerin nasıl hesaplanacağına kayacaksınız... Alın size astronomi meraklısına uygun bir yerleştirme projesi: Kurban ve Ramazan bayramlarının tarihlerini **hesaplayarak** takvim yazılımlarına entegre etmek.

Soru : LKD'nin bu konuda nasıl bir rol almasının gerektiğini düşünüyorsunuz?

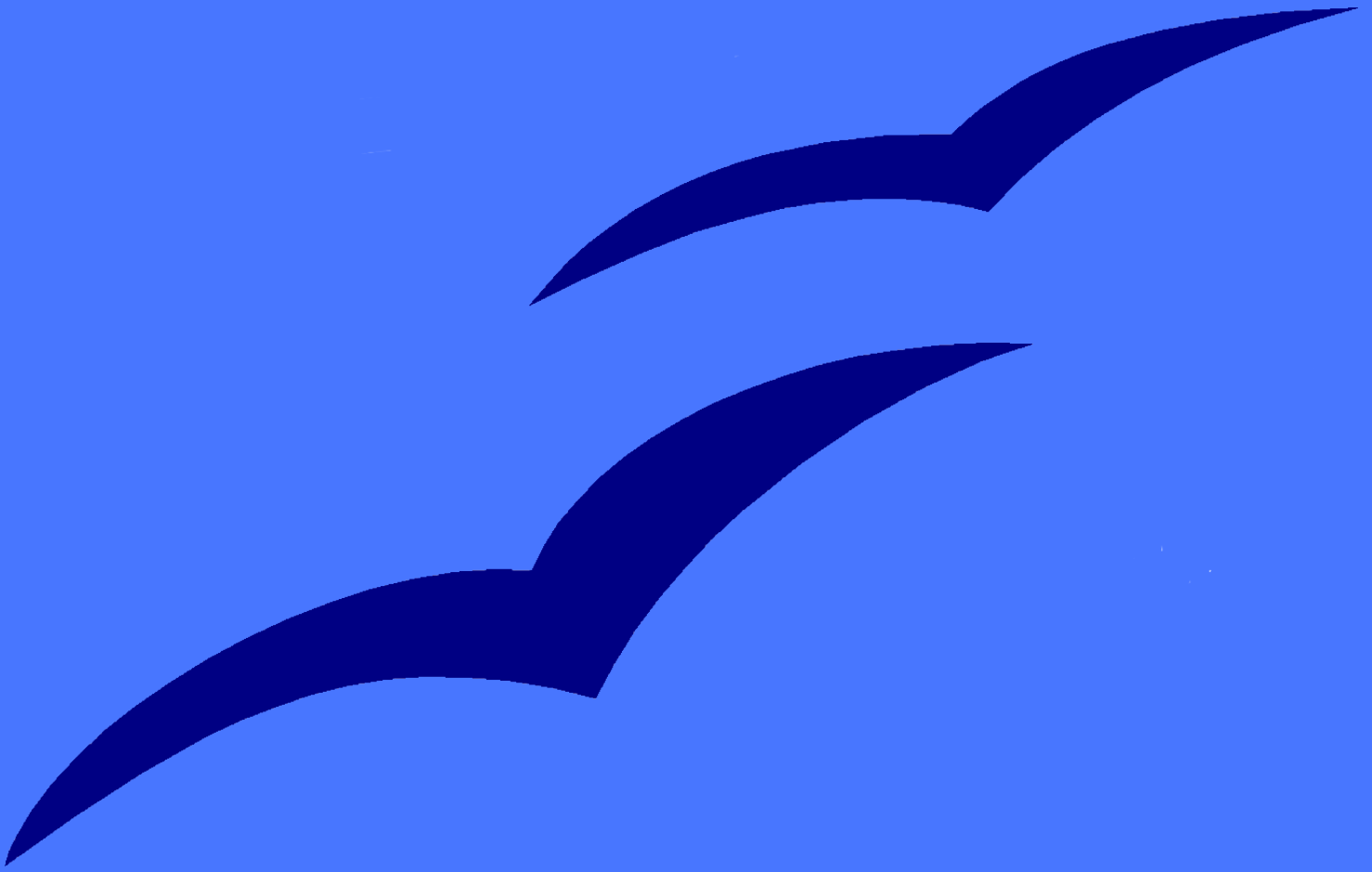
Yanıt : Destek ve bilinçlendirme.

Soru : Özgür yazılımlar için yerleştirme çalışmalarının, özgür olmayan yazılımlar ile karşılaştırıldığında avantaj ve dezavantajları nelerdir?

Yanıt : Kalite açısından bence pek fark yok, hatta özgür yazılım çevirileri daha kaliteli diyebilirim. Özgür olmayan bazı yazılımlarda karşılaştığım Türkçe neredeyse anlaşılamaz noktada idi. Özgür yazılımlarda, eğer mevcut durum birey olarak sizi rahatsız ediyor ise, bu konuda bir şey yapma imkanınız ve hatta sorumluluğunuz var. Kapalı kaynak kodlu dünyada birey olarak bir şey yapamazsınız, sadece kullanma veya kullanmama kararını verebilirsiniz.

Öte yandan, kapalı kaynak kodlu dünyada, büyük ve koordineli çalışılması gereken projelerde mesafe katetmek mümkün oluyor. Dolayısıyla halen daha Türk mevzuatını destekleyen özgür bir muhasebe yazılımı yok, fakat kapalı kaynak kodlu olanı var.





Açık Kaynak

Açık Standart

Açık Ofis



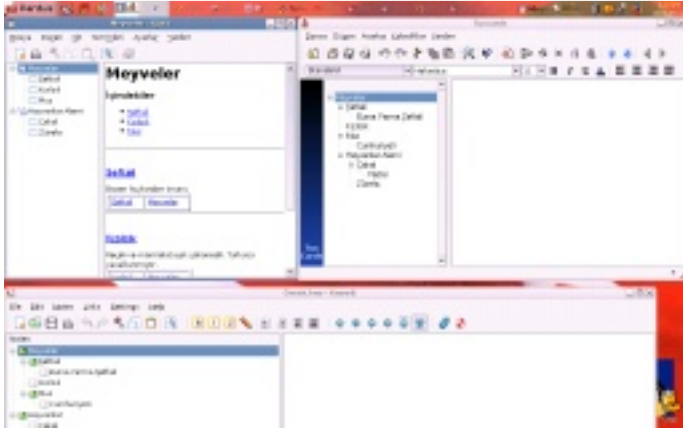
OpenOffice.org 2.0

BilgiAğacı programları ve Kullanılabilirlik üzerine

Dr. Bekir Sonat

Yazının amacı

Bu yazıda Bilgi Ağacı lakabını takabileceğimiz hiyerarşik bilgi saklama programlarını kısaca anımsatacağım. Ağırlıklı olarak Linux'da yaygınlık sırasına göre Kjets, Tuxcards, Know-it, TreeLine .. TreePad, Snippet, KeyNote, CueCards da anılmaya değer..[1]



Ama daha da çok insan ve bilgisayar etkileşimi üzerine geyik yapmaya kıskırtacağım, zira programlar fani, insan baki. Zaten ayakkabı gibiler, herkesin rahatça kullanabilmesi için bir müddet alıştırtıp, ayağınızın neresine ne kadar vurduğuna göre kararı siz vereceksiniz. Genelde kurulup – *kullanılmaları da hayli kolay programlar. Denemek için mazeretiniz kalmasın,-linkleri ekran görüntülerinin hemen yanında veya yazının en sonunda..*

Kullanılabilirlik Kavramı

Sakin bu zamana kadar bunlardan birini kullanmadığınızı veya onsuz yaşayamaz hale gelmediğinizi söylemeyin.

Kullanmadıysanız, zevk meselesi deyip geçmem sorarım: "Neden kullanmadınız ?". Yoksa siz bilgisayarı sadece fare ve joystick tıklamaları ile film ve oyun için mi kullanıyorsunuz? Yok canım, linux dergisi okuyan birisine bunu yakıştıramam.

Ha, eğer ki; "varlıklarını biliyorum hatta ara-sıra kullanıyorum ama.." derseniz, bu yazıyı sonuna kadar okuyup hatta anketi bile dolduracaklardansınız demektir.

"Onsuz yaşayamam" diyorsanız, "hadi canım abartmıyor musunuz biraz" derim. Eğer tam ayağınıza uyan ve her gereksiniminizi olgunlukla karşılayan birini bulduysanız, bir güzel kıskanırım sizi.. Hele burada bahsedileceklerden başka bir programdan bahsediyorsanız, lütfen bana bir e-posta atıp müjdeleyin ki size bir kasa bira ısmarlayayım.

Bu ufacak programlar benim için şu kadar önemli : Özellikle Linux kurduğum birine ilk olarak bu Bilgi-Ağacı uygulamalarından birini kurar ve kullanımını anlatırım. Hatta anlattıklarımın kısa notlarını bile hemen oracıkta o uygulamanın içine kaydederim. Sonra genelde doktor, öğretmen, avukat, yazar, öğrenci gibi mesleklerden birinde nasıl kullanılabileceğinin küçük bir örneğini gösteririm ve aylaaar sonra o bilgi ağacının (dosyasının) ne kadar büyüdüğünü görerek keyiflenirim. Evet, bağımlılık yaparlar.

Hatta kendi adıma "onsuz yaşayamam" diye abartabilirim. Birgün Masaüstüm açıldığında Tuxcards BilgiAğacım kendiliğinden yüklenmezse panik yaşayacağım kesindir. Çünkü parola korumalı dallarında, bozulan Palm'imden aktardığım banka - fatura numaralarım, üyeliklerim vs. vs. vardır. Tamam, kWallet gibi secure değil ama parola korumalı bir metin alanı benim için yeterli ve güvenli.

Kjets BilgiAğacım ise, on yıllardır, İnternet'ten

süzdüğüm linkler ve önemli yazılarla – ipuçları ile doludur tıka-basa.. Evet, düz metin kipinde.. Ne dosya formatları değişti, kaç dosyamı okuyamaz oldum ofislerde, kaç klasör dolusu kaydettiğim html sayfalar kayboldu disk taşınmaları ve code-page değişimleri sırasında.. Ama Kjots'da kategorize edip sakladığım klasörler dolusu önemli metin hala tek dosyada, sıkıştırdığımda tek diskete sığıyor, ya da webdeki bir postanın içinde emrime hazır ve nazır.

Tabii ki asıl kullanışlılık ve vazgeçilmezliğin sırrı; tuvalet kağıdı gibi upuzun düz metin dosyası formatında olması değil, bilgilerin hiyerarşik olarak yerleştirilmiş olarak saklanmış olmasında gizli. Zira, hala bilgisayar kullanarak hayatımızı kolaylaştırırken, hafızamızı ona devir-teslim ederken, fark etmeden, bir yandan, o kullanıma ait bilgi ile hafızamızı ve zihnimizi dolduruyoruz. Eğer, bu doldurma, kendi insani alışkanlıklarımıza ve birikimimize uygun oluyorsa, biz bilgisayarı kullanıyor oluyoruz; aksi takdirde bilgisayar bizi kullanıyor. İşte "kullanışlılık (usability)" denilen şey bu olsa gerek :)

Geyik'ten, bilim ve felsefeye geçecek olursak, bilgisayarların bilgi'leri saklama ve işleme yöntemleri ile insan beyninin yöntemlerini kabaca olsa da karşılaştırmak zorundayız. Bilgisayar, sakladığı bilginin **içeriği** ile ilgilenmez doğal olarak. O'na göre her şey bir dosyadır, o dosyaya ait fiziksel, biçimsel ve kullanışsal öznitelikleri ile erişilerek içindeki bilgi içeriğini size sunar. Her ofis dosyasını açmak istediğimizde, bizi, dizinlerde-ve-dosyalarda-*kelime*-aramak zulmünden kurtaran şey, bizim o dosyaları kaydederken çağrışımlı bir dizin yeri-dosya ismi belirlemiş ve kullanmış olmamızdır. Evet, kurumsal ya da kişisel projelerimizde sayıları sayılabilir olduğu sürece pek zorluk çekmiyorduk,-son zamanlara kadar. En fazlası "1998" "2. çeyrek" "bilanço" sözcükleri ile, nerede olduğunu bile anımsamadığımız küflü bir ofis dosyasını bulup getiriyor bilgisayar.

Ama özellikle internet hayatından sonra, çorabımızın arasına kadar bilgi taarruzu altında kaldık. Üstelik çoğu kez, bu bilgi okyanusunda, işimize yarayacak - önemli bilgi parçacığına rastlamak, keçiboynuzundan bal emmek için koca bir koçanı kemirmeye benziyor. İnsan belirli bir yaştan sonra, bazen Google'ın dost mu düşman mı olduğuna karar vermekte güçlük çekiyor doğrusu.. Üstelik bir yandan İnternet'te kuş gribi ile ilgili

bilginin peşinde sürüklenirken, konuyla ikinci-üçüncü derecede ilgili, hatta bazen de ilgisiz ama bu sefer de geçen yıl araştırdığınız bir konu ile ilgili bir sayfaya - belgeye - daha "kötüsü" bir paragraf'a rastladığınızda , tarayıcınızdaki yer-im'lerine eklemeye değer görmeyip (o kadar kolay link cehennemine dönüyor ki orası!) , "nöbette okurum" diye kaydedip geçtiğiniz o web sayfası (ve gereksiz bir sürü dizin ve simge çöplüğü) vesaire dosyalar ile, harddisk tam bir unutulmuş eşyalar ambarına dönüyor.

Daha kötüsü, daha kaydetmeye karar verdiğiniz o anda da bu akıbeti hatırlıyorsunuz. O anda bir cin çıkıp benden üç dilek istese, ben bir tane ile yetinirdim: "Şu web sayfasında fare ile boyayacağım yazı ve grafikleri hafızana al, link'i ile beraber "Tıbbi bilgiler" dizinine istediğim başlık ile kitaplığıma bir post-it gibi yapıştır! Üstelik görüntüsü bozulmasın, formatı yok olmasın. Üstelik bu klipler kütüphanem sadece bir tek dosyadan oluşsun. Üstelik lanetli disk hastalıklarına olabildiğince bağışık kalabilmek için düz metin dosyası biçiminde olsun" (Bu sefer dört dilek oldu değil mi!? Olsun, isteyen bir yüzü..). Hımmm, şimdi sizin ürettiğiniz clipboard cin'likleri ve bazı shareware program isimleri geliyor aklınıza değil mi? Beklemede kalın,-döneceğiz bu konuya yeniden..

Kısaca, insan zihni evrensel entropi kurallarından hiç de muaf değil çalışırken. Yani en az enerji sarf ederek, en çok ve en hızlı işlev görebileceği tembellik momentini yakalamaya çalışıyor sürekli. Bazen bunun için daha çok enerji sarf ettiği de oluyor,-sınavlar için kopya hazırlamakta olduğu gibi.. Ama beynin işleyişini en kolaylaştıran ve güçlendiren yetenek, bilgiler arasında sağladığı çağrışımlar ve ilişkilendirme-örgütlenme yeteneği.. Örneğin bir meslektaşınıza göstermek için, 3 yıl evvel kaydettiğiniz bir dosyayı, 150 megabaytlık metin klasörü içinde elinizle koymuş gibi bulabiliyorsanız, bilgisayar sizi değil, siz bilgisayarı kullanıyorsunuz demektir. İşte bilgileri **hiyerarşik** bir şekilde saklamaya olanak veren ve *yazımızın konusunu oluşturan uygulama programları bu yüzden önemliler. Ve bu yüzden ayağınıza pardon beyninize uydurduğunuz böyle bir programınız var ise ve onsuz yaşayamıyorsanız,-siz şanslısınız.. Hatta evliliğin tadı tuzu dedikleri türden uyuşmazlıklarınız olsa da..*

Özellik Listesi	kJots	Tuxcards	Know-It	TreeLine	TreePad	TreePad	KeyNote
Açık Kaynak Kodlu mu?	EVET	EVET	EVET	EVET	HAYIR	HAYIR	EVET
Alt dallar açarak	HAYIR	EVET	EVET	EVET	EVET	EVET	EVET
Bilgileri şifreleme	HAYIR	EVET	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR
RTF/HTML biçimli	HAYIR	AZ	HAYIR	HAYIR	AZ	İYİ	EVET
HTML biçimli çıktı (EVET	EVET	EVET	EVET	EVET	EVET	EVET
Bilgi saklama	XML	XML	XML	XML	-hjt	-hjt	-knt
Türkçe Çevirisi-Dil	EVET	EVET	HAYIR	HAYIR	HAYIR	HAYIR	HAYIR

* Tuxcards, hem öğretmenlerin hem ilköğretim çağındaki çocukların kullanabilmesi perspektifi ile Türkçeye çevrildi, ama henüz kullanılabilirlik testleri yapılmadı.

Hypertext doküman modellerinde (HTML, SGML, LaTeX vs.) , hem de ofis belgeleri içinde, hiyerarşik olmayan ilişkilendirme (Link – anchor) kullanılırken, Bilgi-Ağacı tarzındaki uygulamalarda, harici dosyalara yönlendirmeden ziyade ağaç dallarına iliştilmiş bilgi parçacıklarına – paragraflara ya da kopyalanmış ve kategorize edilmiş dosya içeriklerine köprüleme yapılıyor. Bu yönüyle kurumsal kullanımlardan çok kişisel kullanımlara daha eğilimli, mind-mapping tarzında çalışan programlar sayılabilirler.

Örneğin; Davaları ile ilgili notlarını, referanslarını biribiri ile mantıki ve kronolojik ilişki içinde derli-toplu tutmak isteyen bir avukat, ya da yazılarında – kitabında kullanacağı binlerce haber kupürünü ve kendi yorumlarını-notlarını tutmak isteyen ve Lyx kullanmaya bile üşenen bir araştırmacı-gazeteci için bundan iyisi Şam'da kayısı olur herhalde.

Hele bir menü tıklaması ile kurduğu bu hiyerarşik dosyayı web'de yayımlanabilir halde organize edilmiş bir şekilde export edebiliyorsa.. Zaten XML formatında bilgileri tutan Bilgi-Ağacı programlarını, mesela kJots ya da gJots'u, Tek sayfa içinde linkli paragraflar içeren TableOfContents (İçindekiler) -tarzı web sayfaları yapmak için kullanabilir, böylece sade ve etkili HTML kodu için devasa büyüklükteki Web Editörleri ile boğuşmak zorunda kalmazsınız.

Evet, bu aşamada insanoğlu veya insankızı istiyor ki; bilgi-ağacı üzerindeki bilgileri, kısıtlı

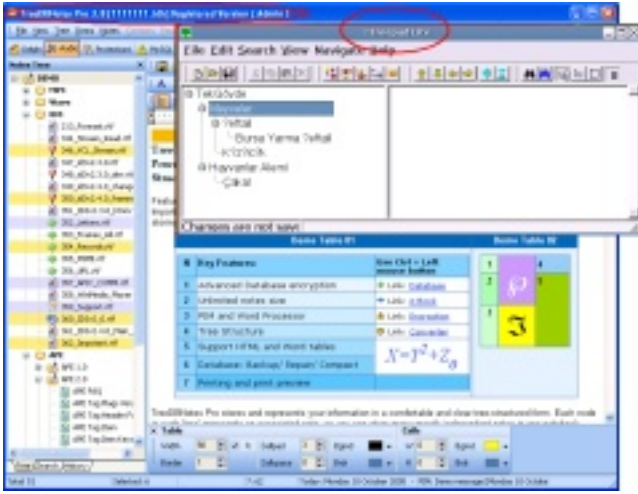
text ya da RTF ile değil, HTML ile WYSIWYG (Ne Görüyorsan Öyle) tarzında görüntülediği gibi düzenleyebiliyor olsak.. Bu neden bu kadar önemli; Çünkü, insan zihni, bilginin içeriği ile beraber şekli'ni de bir pattern olarak depoluyor ve biçimsel çağrışımlar – analogiler yolu ile bilgi içeriğine daha hızlı ve kolay erişebiliyor. Kendi tuttuğu notlardan çalışan üniversite öğrencilerinin daha kolay başarılı olmasının ardında yatan bu gibi sayfa kenarı karalamaları ve çizimleri olsa gerek.

Bu konuda en gelişkin gibi görünen TreePadPRO oluyor, ama onun bile HTML seti henüz çok kısıtlı. Oysa, Açık Kaynak Kod dünyasında bu özelliği bilgi-ağaçlarına eklemek o kadar kolay görünüyor ki, yıllık izninde python kursuna gidecek bir tıp doktoru bu işi becerebilir mi diye düşünmekten kendimi alıkoyamıyorum :)

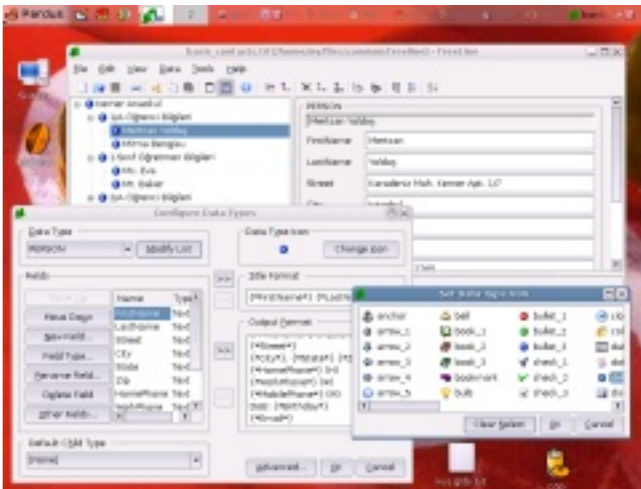
TreePad, eBook üretimi, Hypertext Yardım Dosyası üretimi, Web sitesi haline aktarım, fotoğraf albümü üretimi gibi beklentileri karşılamaya çalışıyor. Lite versiyonu Linux için ücretsiz, Pro versiyonu shareware kapsamında, Enterprise versiyonu hazırlanıyor.. Kurulum gerektirmiyor ve kendi başına çalışıyor, KDE veya Gnome farketmiyor. [2]

Windows ortamında çok büyük miktarlardaki veritabanını bile yönetmesiyle hayli rağbet gören program, Linux gezegeninde yaşayabilecek gibi görünmüyor. Çünkü henüz sadece 8859-1 code-page 'den başka dilleri destekleyemiyor ve kaynak kodu kapalı, Veritabanı formatı sadece kendine özgü. Ayrıca Linux'daki enterprise informatik – içerik yönetim gereksinimleri, tekil uygulama çözümleri ile değil de, endüstriyel standartlardaki yetkin – ölçeklenebilir-

eklemlenebilir araçlar ile yani daha akılcı ve kalıcı yaklaşımlar ile karşılaşma geleneğinde.. [3]

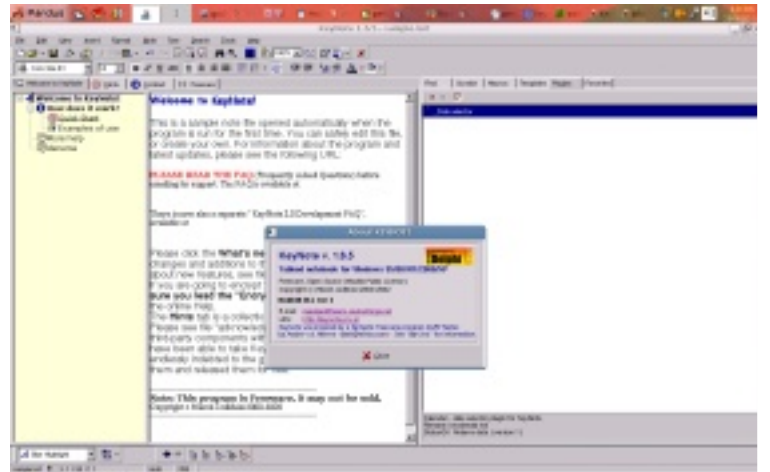


Daha çok veri-formatlı bilgi'ye gereksinim duyan örneğin doktorlar ve öğretmenler için, yine bu yukarıdaki programlar gibi çalışan ama ilaveten kullanıcının kendisinin tanımlayacağı veri-dosyası-başlığı sayesinde ağaçlarının dallarında veritabanı kayıtları tutan bir program olan TreeLine ise, bilgi-ağaçlarında düz-metin-arama, ya da regExp (Düzenli İfade) aramaları ile yetinmeyip, veri başlıklarını sorgulamak isteyen ileri kullanıcılar için eşsiz bir olanak sağlıyor. Her Linux'cu akademisyenin, literatür kaydı ve taraması için en az bir kez denemesi gereken bir program yani.. Başlangıçta alışması güç gibi geliyor ama zamanla işlevselliğini görünce birden her şey “kolaylaşıyor”.. Evet, ah şu nankör ve tembel insanoğlu diyorum, başka bir şey demiyorum şimdilik :) [4]



Aslında Gördüğüm en kapsamlı ve yetenekli – eklemlenebilir program, Marek Jedlinski (marekjed@users.sourceforge.net) tarafından yazılıp, Mozilla Public License (MPL) ile Açık kaynak kod ile dağıtılan, KeyNote 1.6.5 [http://keynote.sourceforge.net SourceForge homepage (needs a maintainer!)] .. İlk sürümünden itibaren ödül üstüne ödül kazanmış, ama 2.0 versiyonu arifesinde programcısının çok duygusal ve dürüst bir açıklama ile yüzüstü bırakıp gittiği bir Delphi programı..

Öyle ki, kullanıcılar, programı ve dağıtıldığı sitedeki forumu kullanmaya devam ettiği için, Marek bir türlü siteyi kapatamıyor. Bu sene foruma dönerek, windows kullanıcılarına, rakibi olan TreePad'i kullanmalarını önererek yeniden işinin başına döndü. Kadim Borland'ın bile Delphi'yi gözden çıkardığı şu günlerde insan garip şeylere bile alışıyor..

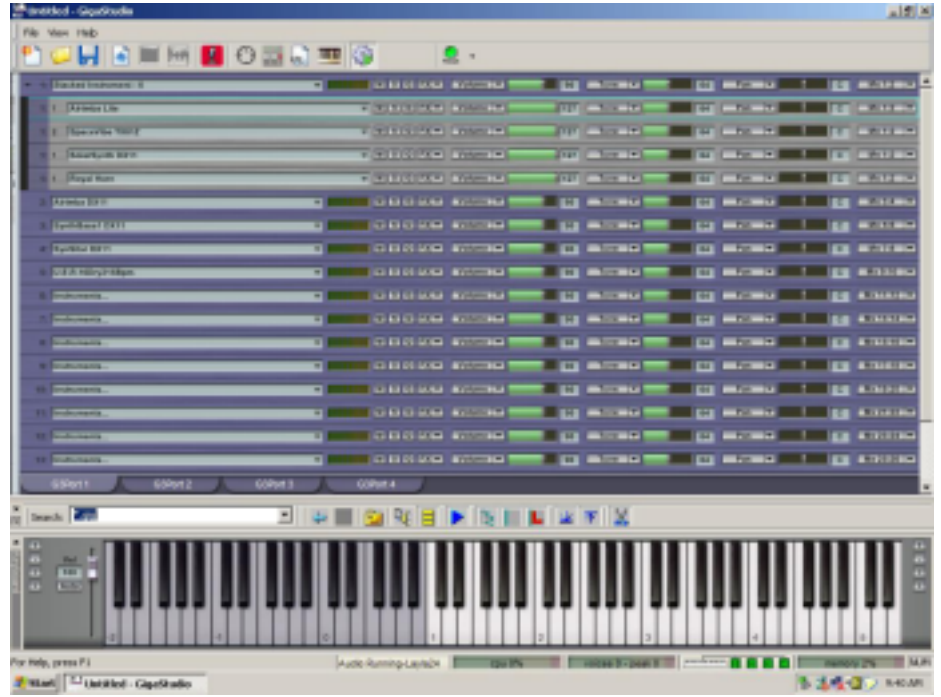


Notlar:

- [1] <http://knowit.sourceforge.net>
- [2] <http://www.treepad.com/linux/treepadlite/treepadpro>
- [3] <http://www.mytreedb.com>
- [4] <http://www.bellz.org/treeline>
- [5] <http://www.tranglos.com/free/keynote.html>

<http://www.synthmuseum.com/akai/akas100001.html>

Günümüzde bu "sampler" cihazları bilgisayar ortamında çalıştırılmaktadır. Günümüzde çok kullanılan "samplers" formatlarından biri de "Gigasampler" formatıdır. Yukarıda anlatılan gerçek seslerin örneklendiği ses kütüphanelerinin kullanılabilmesi için Windows veya Mac ortamınızda "GigaStudio" yazılımının yüklü olması gerekmektedir. Bu yazılım aracılığı ile senfonik orkestra seslerini veya farklı müzik türlerine uygun çalgıları bilgisayarınızın hard diskinde yükleyebilir ve bu sesleri oluşturacağınız müzik parçanızda bu yazılım aracılığı ile kullanabilirsiniz. Bu sistemi Hans ZİMMER, Trevor RABİN, Christopher YOUNG, Jeff RONA gibi film müziği bestecileri kullanmakta ve filmlerde duyduğumuz senfonik orkestra tınıları ile diğer farklı tınlar gerçek müzisyenlerden değil sanal çalgılardan (virtual instruments) elde edilmektedir. GİGASTUDIO ile her türlü detaya <http://www.tascamgiga.com> linkini ziyaret ederek ulaşabilir ve Mp3 formatındaki demoları dinleyebilirsiniz.



Gigastudio" Ekran Görüntüsü

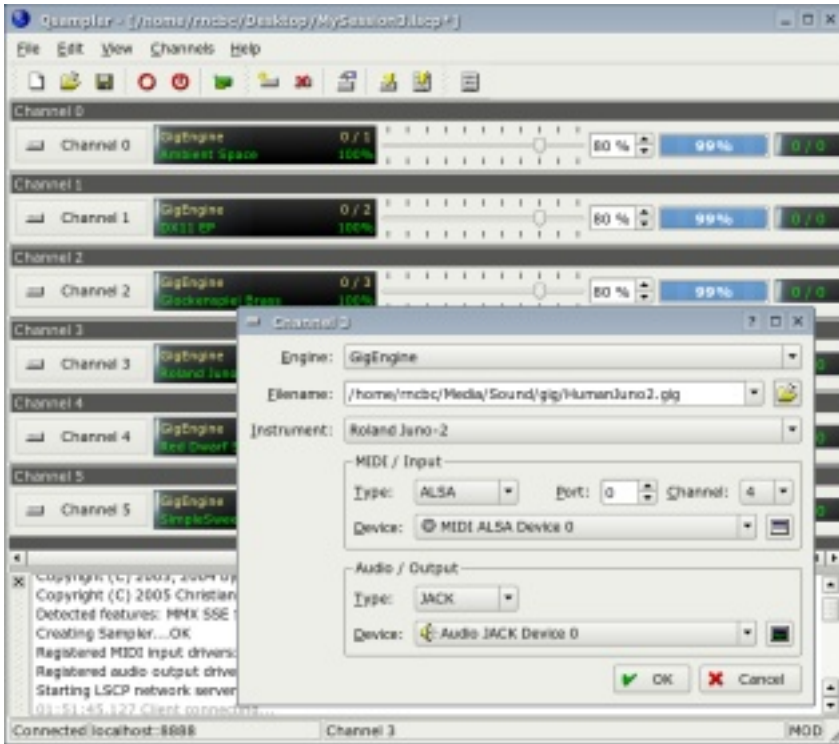
Linux Sampler sürmekte olan bir projedir. Bu yazılımın amacı, Windows/Mac ortamları için üretilmiş olan ve yaygın olarak kullanımda olan yazılım ürünlerinin seviyesine ulaşmış bir yazılım ortaya koymaktır. Bu projeye teknik olarak son şekli verilmemiş ve geliştirme çalışmaları devam etmektedir. Linux Sampler Web Sayfasının Ana Sayfasına tıkladığınızda yazılımın geliştirilmesi hakkındaki son gelişmeleri ve yazılım ile ilgili daha önceki gelişmeleri tarihsel bir dizin içinde izleyebilirsiniz.

Linux Sampler ile ilgili daha detaylı bilgiye ve Linux Sampler kullanılarak yapılmış parçaları .ogg formatında dinleyebilirsiniz. Bu formattaki parçaları dinlemek için Linux işletim sisteminde XMMS yüklü olmalıdır.

Linux Sampler Web sayfası: <http://www.linuxsampler.com>

Gelecek yazıda Linux için bir Nota Yazma programı olan Lilypond yazılımından bahsedeceğim.

Malatya – İnönü Üniversitesi
Güzel Sanatlar Fakültesi
Müzik Bölümü Öğretim Üyesi



Linux Sampler Ekran Görüntüsü

penguence

programlama

Java ve Özgür Yazılım

Bora Güngören

Java ve özgür yazılım köşesine hoş geldiniz. Bu sayıda ve takip eden sayılarda elimden geldiğince Java'nın özgür yazılım dünyasındaki yeri ve Java ile geliştirilen özgür yazılım projelerine yer vermeye çalışacağım. Zaman zaman Java kodları, zaman zaman ekonomik ve politik tartışmaların olduğu bir köşe olacak. Bu ilk yazıda ise Java'nın neden özgür yazılım geliştirmek için popüler bir dil olduğunu tartışacağız.

Her Şey Lisanstan Gelir

Bir yazılımı özgür olarak nitelememiz nedendir? Bu o yazılımın lisansı ile ilgili bir şeydir. Özgür yazılım olmanın gereksinimlerinden birisi kodlarının açık kalmasının garantilenmesidir. Yani sizin yazacağınız yazılımlar özgür yazılım bileşenleri kullanıyorsa, o bileşenlerin kodları açık kalmalıdır. Bu basit tanım çok değişik teknik sonuçlar doğurabilir.

Özgür yazılım bileşenlerini kullanmanın en kolay yolu kendi yazılımınızı da özgür yazılım olarak geliştirmek. Siz de ilgili lisansa tabi olduğunuz zaman önceki kodları almanızda sakınca oluşmaz.

Peki kendi yazdığınız yazılım özgür yazılım olmayacaksa yada farklı lisanslardan kodları (örneğin BSD ile GPL) entegre etmeniz gerekiyorsa ne yapacaksınız? Bu durumda hem kendi yazdığınız kodların kendi istediğiniz lisans modeline sahip olmasını sağlamak hem de alıntı yaptığınız kaynakların istediği lisans modellerine saygı göstermek zorundayız. Bu zor bir problem.

İşte bu ikinci durumda yazılım geliştiricilerin çok sevdikleri bazı prensipler ve teknikler yardıma koşuyor ve lisans modellerini daha rahat

kullanmamıza olanak veriyor.

Peki bu prensipler ve teknikler nasıl şeylerdir ve nasıl işlerler?

Her Yerde Bileşenler

Temel fikir yazılımın bileşenlerden oluşmasıdır. Eğer yazılımınız bileşenlerden oluşursa ve bu bileşenler arasındaki haberleşme belli bir biçimde sağlanırsa o zaman bileşenlerin kendi içlerindeki değişiklikler diğerlerini etkilemez. Örneğin bankaların mevzuatlarındaki değişiklikler bizim bankamatikten para çekmemizde bir değişiklik yaratmadı değil mi? Bu durumda banka ile olan ilişkimiz bu değişiklikten bağımsızdır deriz. Değişimden etkilenmiyor olmak, yani bağımsızlık fikri, önemli bir avantajı daha yanında getirir. Bir bileşen kolayca benzeri ile değiştirilebilir. Bu tür esnekliklerin olduğu durumda iki bileşenin yazılım kodları arasındaki ilişki teknik olarak başka türlü olur. Eğer yazdığınız yazılımın bir bileşeni diğerinden bağımsız ise, iki bileşen için farklı lisans modeli kullanmanız olanaklı hale gelir.

Bu fikir uzun yıllardır yazılımcılara yardım etmektedir. Eğer bağımsız bileşenler olacaklarsa, kendi yazdığımız kodların lisanslarını, yararlandığımız açık kaynaklı kodlardan ayrı

Eğer yazdığınız yazılımın bir bileşeni diğerinden bağımsız ise, iki bileşen için farklı lisans modeli kullanmanız olanaklı hale gelir.

tutabiliriz. Yazılım tasarımı ve kodlamasını bu şekilde yapacak olursak her şey kolaylaşacaktır. Ancak ne yazık ki bir satırlık yazı ile ifade edilen bu hedefi gerçekleştirmek zordur.

Kitaplık (library) olarak adlandırılan bileşenler için tasarlanan ve yıllardır piyasada LGPL lisansı altında bu fikrin bir uyarlamasıdır. LPGL'in GPL'den farklı olarak tasarlanması, bazı dillerde (örneğin C) bu şekilde net bir bağımsızlığın sağlanmasının oldukça zor olmasıdır. Çünkü C dili temel olarak bir çok kaynak kodunu bir araya getirip, bir tane çalışan kod üretmek üzere

tasarlanmıştır. Bu şekilde oluşturulan bir uygulamanın teknik avantajları vardır. Ancak lisanslar gibi hukuki konular söz konusu olunca bir sorun oluşur.

Java uygulamalarında dışarısı ile olan iletişimde bağımsız, kendi aralarında ise belli ölçüde bağımlı olan bileşenleri (sınıfları) birlikte barındırmak için paket (package) adı verilen bir mekanizma bulunur.

Ancak kendi tasarımı gereği bileşen tabanlı uygulamalar oluşturan Java dili ve platformu için bu tür bir sorun oluşmaz. Çünkü Java zaten her uygulamayı bir çok küçük parçadan (sınıflar) oluşacak biçimde üretir. Bir Java uygulaması çok sayıda minik dosyadan oluşur. Bu dosyaların her biri uygulamadaki küçük bir işlevselliği içerecektir. Dolayısı ile iyi tasarlanmış bir Java uygulamasında, bir kısımdaki değişiklikler, diğer kısımların büyük çoğunluğunda değişiklik gerektirmez. Sadece kendisinin çok yakın iletişimde bulunduğu, işleyişi hakkında bilgi sahibi olduğu bir kaç parçada değişikliğe yol açar.

Java uygulamalarında dışarısı ile olan iletişimde bağımsız, kendi aralarında ise belli ölçüde bağımlı olan bileşenleri (sınıfları) birlikte barındırmak için paket (package) adı verilen bir mekanizma bulunur. Bu durumda birileri uygun Java paketlerini GPL yada benzeri açık ve hatta kapalı lisanslarla dağıtsa bile, bu paketleri kullanan diğer paketler farklı lisanslarda olabilir. İşin en başında, Sun'ın sunduğu JDK içindeki 6 binden fazla sınıfın lisansları sizin yazacağınız uygulamanın lisansını bağlamaz. Bu da sizin bu sınıfları gönül rahatlığı ile kullanmanız demektir.

Sun tarafından sağlanan bu zengin kitaplık desteği (6 bin sınıfın kaynak kodları yaklaşık 40 MB düz metindir, satır sayısını tahmin edin) açık kaynağa herhangi bir engel koymadığı için Java platformunda açık kaynak kodlu yazılım geliştirmenin bir riski olmaz. Başka birisinden edineceğiniz açık kaynak kodlu (örneğin GPL) bir Java kitaplığının (yani paketinin) lisansı sizin kapalı yada daha farklı açık lisansta (örneğin Mozilla) bir yazılım geliştirmenize engel olmaz. Her bileşen kendi lisansına sahip olur.

Bu sayede Java ile geliştirilen açık kaynak kodlu projeleri doğru biçimde kullanarak bir çok ticari uygulama geliştirebilirsiniz.

Java Kurumsal Uygulamalarda Baskındır

Kurumsal uygulamaları ele aldığımız zaman, oldukça büyük ölçekli ve genellikle bazı teknik altyapıları kullanması zorunlu olan projelerle karşılaşırız. Java platformu sunduğu performans, güvenlik, çok dil desteği, entegrasyon gibi beceriler nedeni ile 1995'den bu yana kurumsal uygulamalarda baskın (dominant) bir konum elde etmiştir.

Kurumsal uygulamaların geliştirilmesi sırasında elde edilen deneyimi paylaşmak ve bunu açık kaynak kodlu alt yapılar sunarak

Java'nın bileşen tabanlı olması nedeni ile iyi tasarlanmış bir uygulamanın içindeki bazı bileşenler veya alt yapılar, kolayca diğerlerinden ayrı lisanslara (örneğin açık lisanslar) taşınabilir.

yapmak isteyen hatırı sayılır bir kitle de oluşmuştur. Bu kitlenin bir kısmı Sun, IBM gibi firmalar tarafından desteklenmiş, bazıları ise bağımsız çalışmıştır. Peki bu kişiler kodlarını açmak istedikleri alt yapı teknolojilerini, daha önce yaptıkları ancak kapalı kaynak kodlu olması gereken uygulamalardan nasıl sökecektir?

Java'nın bileşen tabanlı olması nedeni ile iyi tasarlanmış bir uygulamanın içindeki bazı bileşenler veya alt yapılar, kolayca diğerlerinden ayrı lisanslara (örneğin açık lisanslar) taşınabilir. Kurumsal Java alanında olan da budur. Firmalar yada kişiler kendi kullandıkları alt yapı teknolojilerinin yaygınlaşmasını sağlamak için (bundan çeşitli faydalar elde ederler) onları açık kaynak kodlu hale getirirler. Teknoloji açılınca diğer kişiler de ister açık ister kapalı lisanslardaki yazılımlarını bu teknolojilerle geliştirebilir. Bunda yasal bir engel olmaz.

İşte bu teknik ve hukuki esneklik sayesinde özgür yazılım Java altyapıları ve uygulamaları,

giderek yaygınlaşmaktadır. Çünkü aynı açık kaynak kodlu bileşenler, hem açık hem kapalı kaynak kodlu yazılımlarda kullanılmakta ve bu yaygın kullanım sayesinde daha iyiye doğru evrimleşmektedirler. Bu evrim, sadece açık kaynak kodlu yada sadece kapalı kaynak kodlu bir ortamda olabileceğinden çok daha hızlı olur.

Java Uygulamalarında Belgeleme Daha Kolay Sağlanır

Java geliştiricileri biraz dikkat ederlerse, kaynak kodları içine gömdükleri not satırlarından çok kapsamlı kod belgelemesi çıkartabilirler. Bu belgeleme tekniğine javadoc adı verilir. Javadoc sayesinde bileşenler arasındaki ilişkileri simgeleyen çapraz referansların da bulunduğu binlerce HTML sayfasını içeren bir dizin hiyerarşisi bir tek komutla oluşturulabilir. Kodlar güncellendikçe bu HTML belgelerinin güncellenmesi de otomatik olabilir. Asla unutmamak gereken bir prensip var: Belgeleme kolay üretilemezse, programcı onu hiç üretmez. Java'da belgeleme kolay üretilir, bu yüzden Java uygulamalarında daha çok belgeleme görürüz.

Bir özgür yazılımın kodları ile çalışmanız gerektiğinde bu tür belgeleme çok önemlidir. Java'da geliştirilen özgür yazılım kodlarının hatırı sayılır bir bölümü alt yapıya yöneliktir. Bu nedenle belgelemenin bulunması daha da önemlidir. Birlikte çalıştığımız şirketlerden birisindeki bir müşterim, Java uygulamalarının platform bağımsızlığı, güvenliği gibi özelliklerden son derece faydalansa da (kendisi e-imza uygulaması dahi geliştiriyor) "Java'daki en iyi şey çok fazla doküman ve örnek olması" diye özetliyor durumu.

Elbette ki bu genel bir ifade ama kendisini muhatap ettiğimiz alt yapıların hepsi açık kaynak kodlu. Yani onun gözünde "Java" aslında "açık kaynak kodlu Java alt yapıları" demek. Aslında bu algılama bir çok kişi için geçerli.

Genel Taşınabilirlik Avantajı

Geçtiğimiz günlerde Jakarta sitesine erişimin olmadığı (aslında İnternet erişiminde sorun yaşanan) bir anda bir Linux bilgisayara Tomcat kurmamız gerekti. Bilmeyenler için Tomcat, web üzerinden çalışan Java uygulamalarının durduğu, yine Java ile yazılmış bir sunucu ve Apache Vakfı tarafından geliştiriliyor.

Bir Windows kurulumu üzerindeki Borland JBuilder Enterprise Edition içinde bulunan Tomcat 5.5.9 sürümünü kopyaladık ve yerel ağ üzerinden Linux bilgisayarda olmasını istediğimiz yere yapıştırdık. Kurulum tamamlanmıştı. Bu olay iki durumu vurguluyor.

Java uygulamasını kurmak ve taşımak sadece dizinleri kopyalamaktan ibarettir. Windows ve Linux arası dönüşüm gerekmez. (Acaba Microsoft neden Java teknolojilerinden fazla hazzetmiyor?) Borland gibi ticari bir firma, binlerce dolara sattığı bir yazılımın içinde, gerek duyduğu özelliği standart bir kurulumla sahip özgür yazılım ile sağlamaktadır. Aslında Jbuilder'ın kendisi de Java ile yazılmış. (Acaba Borland bu sayede ne kadar tasarruf sağlamıştır?)

Özgür yazılım geliştiricilerinin önemli problemlerinden birisi, geleneksel geliştirme teknolojileri yüzünden yazılımlarının sadece Linux yada sadece Windows üzerinde çalışmasıdır. Open Office, Mozilla gibi özgür yazılımları birden çok işletim sistemini destekleyecek şekilde üretmek inanılmaz bir ek işgücü gerektirir. Ancak Java ile yazılan uygulamalar için bu dert olmaz. Uygulama her iki (yada daha fazla) işletim sisteminde kolayca kurulur ve çalışır. Bir işletim

Belgeleme kolay üretilemezse, programcı onu hiç üretmez. Java'da belgeleme kolay üretilir, bu yüzden Java uygulamalarında daha çok belgeleme görürüz.

sisteminde çalışırken bir sorun tespit edilirse, büyük olasılıkla bu her işletim sistemi için geçerlidir. Sorunu giderdiğinizde tüm işletim sistemleri için gidermiş olursunuz. Yada aslında uygulamada sorun yoktur, işletim sisteminin kendisinde bazı ilginçlikler bulunmaktadır.

Takip eden yazılarda hem Java'nın neden özgür yazılımlar için tercih edilir bir teknoloji olduğunu tartışmaya devam edeceğiz hem de çeşitli açık kaynak kodlu Java alt yapılarını inceleyeceğiz.

penguence

programlama

Ruby

Hüseyin Gömleksizoğlu

“Ruby bir programlama dili değildir. Ruby bir eğlence aracıdır. Temel tasarım sebepleri arasında programcıyı eğlendirmek olan başka bir programlama dili yoktur.”

Ruby özellikle çok kısa zamanda uygulama yazmak isteyenlere hitap eden bir dildir. Artık iş dünyası çok çok hızlı çalışan ama geliştirme süreci uzun olan çözümler yerine çok kısa sürede geliştirilen ve bir an önce kullanılabilecek çözümler peşinde. Ruby'nin esnek yapısı sayesinde çok karmaşık denilen programlar bile rahatlıkla yazılabildiği için NASA gibi büyük kurumlarda bile birçok alanda kullanılıyor.

Ruby açık kaynaklı, tamamen nesne-tabanlı, yorumlanan betik bir dildir. Ruby'de her şey bir nesnedir. Genelde programlama dillerinde kalıplaşmış olan yapıların dışına çıkarak nesne tabanlılığı sonuna kadar kullanır. Yazdığınız programlar üzerinde hiçbir değişiklik yapmadan farklı platformlarda çalışır.

Betik bir olması Ruby için bir dezavantaj gibi gözükse de kazandırdığı esneklik bakımında avantaj haline de gelebilir. Genel olarak betik dillerin yavaş olması gibi bir kanı vardır. Evet Ruby derlenen dillere göre yavaş bir dildir, ama hıza ihtiyacınız olan yerlerde C kütüphaneleri kullanarak daha hızlı çalışan programlar elde edebilirsiniz. O da olmadı, C içerisinde Ruby kütüphanelerini kullanarak Ruby biçiminde kod yazabilirsiniz.

Ruby'de özel durumlar ve istisnalar dışında işler umduğunuz gibi gider. Dilin yapısını bir defa anladığınız zaman neyin nasıl çalışması gerektiği hakkında fikriniz olur ve bilmediğiniz konularda çok rahat fikir yürütebilirsiniz.

Böylece her şeyi detaylı olarak öğrenmek için zaman harcamanız gerekmez.

Eğer çok hızlı geliştirebileceğiniz, hedefe yönelik, anlaşılır ve eğlenceli kodlar yazmak istiyorsanız ve bunu yaparken derin konularda boğulmak yerine programcı olduğunuza şükretmek istiyorsanız, Ruby sizin için doğru bir dildir diyebilirim.

Ruby'nin Tarihi

Ruby, çok eski bir dil olmasına rağmen, Japonya dışındaki ülkelerde yeni yeni tanınmaya başladı. İlk defa 1994 yılında Japonya'da Yukihiro Matsumoto (Kısaca Matz) tarafından yayınlandı. 2000 yılına gelindiğinde Japonya'da 20'den fazla ruby kitabı yayınlanmıştı. Ancak ruby Japonya'da bu kadar popüler olmasına rağmen diğer ülkelerde pek bilinmiyordu. Bunun tek bir sebebi vardı, o da dünyadaki bir çok programcının ne yazık ki Japonca bilmiyor olmasıydı. Japonya dışındaki programcıların Ruby ile tanışmaları ancak 2000 yılında oldu. Dave Thomas ve Andy Hunt 2000 yılının sonlarında “Programming Ruby” isimli bir kitap yayınladılar ve böylece Ruby dalgası tüm dünyaya yayılmaya başladı.

2004 yılına gelindiğinde David Heinemeier Hansson “Ruby on Rails” isimli bir web uygulama altyapısı (framework) yazdı. Rails tamamen Ruby ile yazılan, çok güçlü ve çok kolay bir uygulamaydı. Belki de ilk defa bir dil, üzerinde yazılan bir uygulama sayesinde tanınır oldu. Rails, şu anda PHP ve J2EE'ye büyük bir alternatif duruma çoktan geldi diyebiliriz.

Ruby Nasıl Kurulur?

Öncelikle Ruby'ı denemek için bilgisayarınıza kurmanız gerekmez. <http://tryruby.hobix.com/> adresinden canlı canlı Ruby'ı deneyebilirsiniz. Hatta site üzerinde yardımlar sayesinde komutlar üzerinden hızlı bir şekilde geçebilirsiniz.

Ruby'nin birçok dağıtım için hazır paketleri bulunuyor. <http://www.ruby-lang.org> adresinden size uygun paketi indirip kurabilirsiniz. Yine aynı



site üzerinde “Programming Ruby” kitabına da erişebilirsiniz. Bu kitap, başlangıçtan ileri seviyeye kadar adım adım ilerlemek için güzel bir kaynak. Ayrıca Ruby hakkında diğer Türkçe kaynaklar için <http://www.belgeler.org> adresine bakabilirsiniz.

Ruby'nin kullanımı

Yazdığınız bir Ruby kodunu çalıştırmak için konsolda

```
$ ruby merhaba_dunya.rb
```

yazmanız yeterli. Ruby ayrıca yazım denetimi de yapabilir bunun için

```
$ ruby -cw merhaba_dunya.rb
```

şeklinde bir komut verebilirsiniz. Burada c parametresi yazım denetimi yapmak için, w parametresi bulunan tüm uyarı ve hata mesajlarının verilmesini sağlar.

Ruby ile bir komut çalıştırmak için mutlaka bir Ruby program dosyasına ihtiyacınız yok,

```
$ ruby -e 'print "Adınızı girin...: "; print gets.reverse'
```

şeklinde bir komut kullanıcıdan aldığı yazıyı ters olarak ekrana yazacaktır.

Bunların dışında ruby komutlarını anında çalıştırmak için irb diye bir uygulama da ruby ile birlikte sisteminize kurulacaktır.

Kütüphaneler

Ruby kurulumu ile birlikte bir çok ruby kütüphanesi de sisteminize kurulacaktır. Bir çok kütüphane Ruby'e çevrilmiş olsa da, C kütüphanelerini, .so ve .dll kütüphanelerini de Ruby programlarınızda kullanabilirsiniz.

Debug

Ruby kendi debugger'ın sahiptir. Oldukça gelişmiş olan bu debug kütüphanesi ile yazdığınız kodlardaki hataları çok rahat bir şekilde yakalayıp, temizleyebilirsiniz. Ruby debug işlemi için kendi debug kütüphanesini kullanır. (kütüphaneler içinde debug.rb dosyasına bakabilirsiniz.)

Bir programı debug etmek için

```
$ ruby -rdebug merhaba_dunya.rb
```

Şeklinde bir komut vermeniz yeterli.

ERb

ERb (Embedded Ruby – Gömülü Ruby) HTML dosyaları gibi Ruby dosyaları dışındaki dosyalar içerisinde ruby komutlarını kullanmanızı sağlayan bir programdır. Seki Masatoshi tarafından geliştirilmiştir. ERb sayede html veya rhtml dosyaları içerisinde ruby komutları kullanabiliyorsunuz. Bunun için normal metin içerisinde <% ve %>

işaretlerini kullanarak ruby komutlarını kullanabilirsiniz. <%= kullanırsanız işlemin sonucu metin eklenir. ERb sayesinde Ruby'i PHP, ASP veya JSP gibi kullanabilirsiniz.

Ruby'nin Güzellikleri

Buraya kadar Ruby hakkında çok genel bilgiler verdik. Ruby'ı daha iyi anlamak açısından biraz daha ayrıntıya girmekte yarar var sanırım.

Değişkenler

Sayılar için Ruby'de iki tip vardır. Tam sayılar(Integer) ve ondalıklı sayılar(float). Eğer bir sayı 1.1 şeklinde yazılıyorsa ondalıklı, 100 gibi yazılıyorsa tam sayıdır. Tam sayıların sınırları yoktur. Bir sayı hafıza yettiğince uzayabilir. Ruby sizin için gerekli işlemleri yapar. Ayrıca

```
a= 123456789123456789123456789
```

şeklinde bir sayıyı

```
a=123_456_789_123_456_789_123_456_789
```

şeklinde yazabilirsiniz. Daha sonra bu sayının karesini küpünü başka hiç bir işlem yapmadan alabilirsiniz. Aritmetik işlemler için + - * / sembollerini kullanabilirsiniz. Üst alma için **, mod almak için % sembolünü kullanabilirsiniz.

Ruby yazıları da çarpmanıza izin verir.

```
>> "ho " * 3
"ho ho ho "
```

Ruby diziler konusunda da çok esnektir. Bir dizi farklı tiplerde elemanlardan veya başka dizilerden oluşabilir.

```
>> a = [1, "merhaba", [ 1, 2 ] ]
```

şeklinde bir dizi tanımlayabilirsiniz.

Nesneler ve Methodlar

Ruby'de her şey bir nesnedir. Her şeyin nesne olmasından yola çıkarak

```
>> 3.times { puts "ho " }
```

```
"ho ho ho "
```

işlemini yadırgamamak gerekir. Bir nesne üzerinde çağrılacak methodları görmek için

```
>> 3.methods
```

komutunu verebilirsiniz. Gelen sonuçlar “3” nesnesine yani bir “Tamsayı(Integer)” nesnesi üzerinde çağrılacak methodları gösterecektir. Methodlar arasında “?” işareti ile bitenler kontrol amaçlı kullanılırlar mesela .empty? methodu o değişkenin değerinin olup olmadığını kontrol ederek doğru (true) veya yanlış (false) değerleri döner. Bir de ! ile biten methodlar vardır. Bunlar da işlemi yaptıktan sonra değişkenin değerini değiştiren methodlardır. Mesela ;


```
>>a = "Merhaba"
>> a.reverse
dediğinizde a'nın değeri ters çevrilip geri
döndürülür ama a değişkenine bir atama
yapılmaz.
>>a.reverse!
dendiğinde yine aynı işlem yapılır ama işlem
sonunda a değişkeninin de değeri değiştirilir.
```

İşlem Blokları

Ruby Class'larına kendi methodlarınızı da yazabilirsiniz. Örneğin Integer Class'ına kere() diye bir mothod tanımlayarak

```
>>3.kere { puts "ho " }
```

komutunu yazabilirsiniz. Bu komutu verdiğinizde ne oluyor? Şimdi işlemi bir adım daha ileriye götürelim ve şu şekilde yazalım.

```
kere (3) { puts "ho " }
```

işlemin başı "kere (3)" bildiğimiz fonksiyon işlemi ama daha sonra gelen süslü parantezler ne olacak? İşte o da işlem blokları denilen yapı. İşlem blokları, fonksiyon içinden çağrılabilen dışsal fonksiyon kümeleridir desem sanırım hiç bir şey anlaşılmaz. Onun yerine şöyle bir örnek vermek daha yerinde olacak.

```
>> 3.times { |counter| puts
counter.to_s + ". işlem" }
0.işlem
1.işlem
2.işlem
```

times methodu kendi içinde 0'dan başlayarak 1'er 1'er artarak yeald diye bir method çağırmaktadır. Yeald methoduna geçilen parametre işlem bloğu içinde | | işaretleri içinde yazılan değişkene atanır, daha sonra işlem bloğu içindeki işlem çalıştırılır. Bu sayede çok genel fonksiyonlar yazmak mümkün olur.

Bloklar Ruby'de bir çok yerde kullanılır ve kod tekrarını önleyerek hayatı büyük ölçüde kolaylaştırır.

Ruby on Rails

Daha önce de dediğimiz gibi Rails, Ruby ile yazılmış bir altyapı (framework)'dir. Rails ile diğer dillere göre çok hızlı şekilde web uygulamaları yazabilirsiniz. Rails Model-Controller-View şeklinde 3 katmanlı yapıyı desteklemektedir. Bu üç katmanlı yapıyı bir asansöre benzetebiliriz. Hareket mekanizması "Model", asansör çağırma düğmesi ve kat düğmeleri "Controller" ve asansörün kaçınıcı katta olduğunu gösteren ekran "View" olarak düşünülebilir.

Model-Controller-View (MVC)'nin

Railsteki karşılıkları şöyledir;

Model; Tüm veritabanı işlerini yapar, arama, ekleme, değiştirme, silme, sıralama, kontrol etme

Controller; site üzerinde yapılabilecek tüm işlemleri tutar.

View; görsel içeriği tutar, ERb kullanılarak yapılır.

Rails, DRY (Don't Repeat Yourself – Kendini tekrar etme) prensibine sıkı sıkı bağlıdır. Rails içerisinde yapacağınız işlemi için nereye kod yazmanız gerektiği bellidir ve bir defa yaptığınız bir işlem tüm sistemde geçerli olur. Örneğin "Kullanıcı telefon numarası 7 haneden kısa olamaz" diye bir kural tanımladıysanız, sistem üzerinden hiçbir koşulda 7 haneden kısa bir telefon numarası veritabanına yazılamaz.

Rails'in web sitesi www.rubyonrails.org dir. Site üzerinde videolu anlatımlar da bulunmaktadır.

Sonuç

Bu yazıda Ruby kodlarından çok Ruby'nin nasıl birşey olduğunu anlatmaya çalıştım. Benim fikrime göre 1-2 sene sonra Ruby çok farklı bir yerde olacak. Bu nedenle, Ruby ile hiç ilgilenmeyenlerin bile Ruby'e bir göz atmalarını tavsiye ederim, çünkü bazı şeylerin diğer dillerde neden farklı olduğunu düşünmenizi sağlıyor. Bir defa Ruby'i kullanmaya başladıktan sonra diğer dillerde aynı özelliklere aramaya başlıyorsunuz.

Sonuç olarak, Ruby hem eğlenceli, hızlı geliştirilen uygulamalar yazmak isteyenler için biçilmiş kaftan. Özellikle 3 satır kod ile sonucu/istemci uygulamaları yazmaya başlayınca bana hak vereceksiniz. Ayrıca programcılığa yeni başlayacaklar için de Ruby'nin doğru bir seçim olacağını söyleyebilirim. Web uygulaması geliştirenlerin de RubyOnRails'e mutlaka bakmalarını tavsiye ederim.





Pardus g  n  l  n  r  

Pardus projesine katkıda bulunmak istiyor ve Linux ya da bilgisayar kullanımı konusunda farklı deneyimlere sahipseniz; ancak hen  z Pardus toplulu  unda kimseyi tanımıyorsanız, <http://kalite.uludag.org.tr> adresini ziyaret ederek   zg  r d  nyaya ilk adımınızı atabilirsiniz! Tamamen yeni başlayan bir kullanıcıysanız e-posta listeleri, forumlar gibi deneyiminize daha uygun se  eneklerden başlayabilirsiniz...

OpenVPN ile VPN Uygulamaları

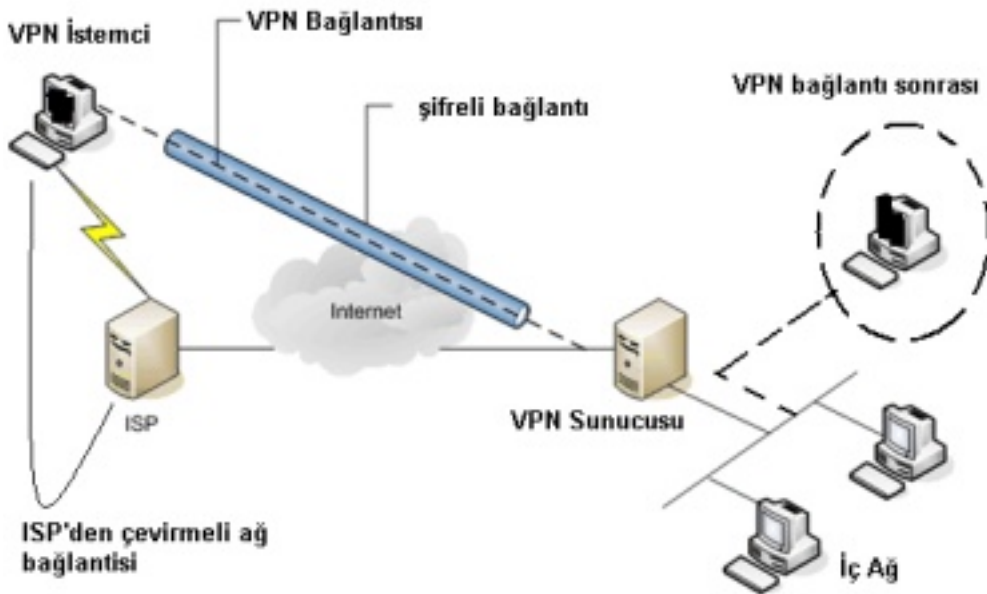
Huzeyfe Önal

VPN(Virtual Private Network/Sanal Özel Ağ) internet üzerinden şifreli ve güvenli veri iletişimi sağlamak için düşünülmüş bir teknolojidir. Lease-line gibi daha güvenli, sağlam çözümlerin yerine VPN kullanilmasının temel nedeni, maliyet ve kolay yapılandırmasıdır.

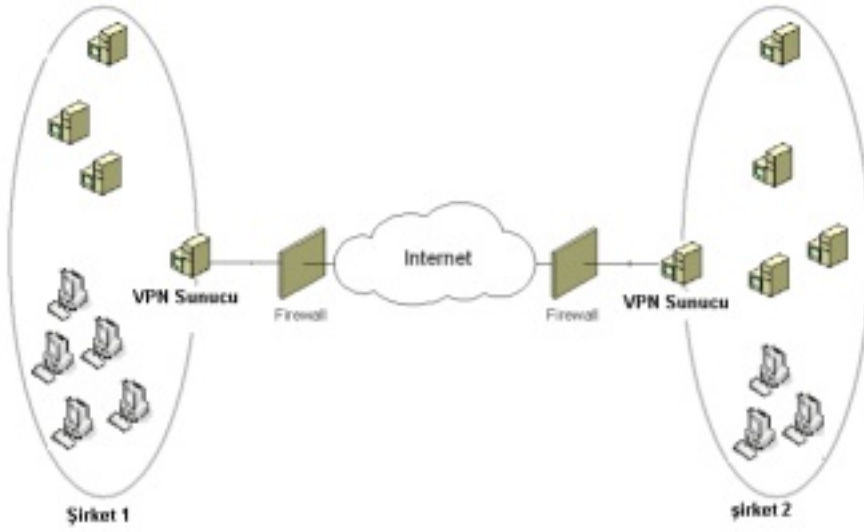
Temelde iki tip VPN teknolojisi vardır. Amacımıza göre bu iki VPN teknolojisinden birini seçebiliriz. Bu teknolojiler "Remote Access VPN " ve "Site-to-site VPN" olarak geçer.

Remote Access olarak tanımladığımız VPN türü, firmaların gezgin çalışanlarının firma ağına her yerden güvenli iletişimlerini sağlamak için kullanılır. Ya da büyük bir firmanın farklı lokasyonlardaki şubelerini merkeze bağlamak için kullanılır.

Basitçe resimleyecek olursak: Firmamızın satış elemanı Sivas'da bir görüşme sonrası bazı belgeleri print etmesi ya da ofisteki bir kaynağı kullanması gerekti, bunu normal internet üzerinden yapmak hem riskli hem de bir o kadar zordur. Bunun yerine biz elemanımıza VPN istemcisi kurarak istediği yerden şirket ağına bağlanarak "belirli" işlemleri gerçekleştirebilmesini ve "belirli" kaynaklara erişimi sağlayabiliriz -ki tercih edilmesi gereken yöntem de budur-.



Bu tip VPN genellikle farklı firmaların birbirleri ile VPN aracılığı ile güvenli iletişim kurmaları için düşünülmüştür. Aynı zamanda firmanın farklı lokasyonlardaki şubelerinin merkeze bağlanmasını da sağlar. Remote Acces VPN'den temel farkı VPN işlemini gören iki uçta VPN sunucu olması.



VPN teknolojileri hakkında daha detaylı bilgi edinmek için Penguençe Sayı 2'deki "Sanal Özel Ağ, Kriptoloji ve PKI Teknolojileri" yazısına gözatmanızı tavsiye ederim. Kısaca VPN teknolojisine değindikten sonra bu teknolojiyi en etkin ve ucuz bir şekilde uygulamanın yollarına bakalım.

Açık Kod VPN Çözümleri

Açık kod dünyasında her amaca yönelik çeşitli çözümler vardır. Bu çözümlerin bazıları diğerlerine göre daha fazla tutulur, kullanılırlar ve o teknolojinin adı anıldığında insanların aklına gelir. Mesela bugün port tarama konusunun geçtiği her kitap/yazıda açık kodlu port tarama programı Nmap'den bahsedilir. Nmap ya da benzer diğer popüler açık kodlu yazılımlar, bunu basitlik, projeyi sahiplenme ve bol dökümantasyon özellikleri ile başarmışlardır denilebilir.

VPN çözümlerinde de öne çıkmış bir iki Açık kod yazılım vardır. Bunlardan biri OpenVPN, diğeri de Linux OpenSWAN(Ya da daha kararlı bir çözüm olarak OpenBSD Ipsec).

AçıkKod VPN çözümleri ile ilgili olarak detay bilgiyi kaynaklar kısmından edinebilirsiniz[ref 4]

Bunun yanında açıkkod dünyasında farklı VPN teknolojilerini kullanarak benzer amaçları gerçekleştiren çeşitli VPN yazılımları da vardır. Bunlardan en sık kullanılanları;

- PPTP Çözümü : Poptop
- Ipsec Çözümü : Linux OpenSWAN, OpenBSD Ipsec
- SSL VPN Çözümü : SSLExplore, OpenVPN
- L2TP Çözümü : OpenL2tp

Gerçek bir VPN Çözümü Olarak OpenVPN

OpenVPN multi platform SSL VPN çözümüdür. Endüstri standardı SSL/TLS protokollerini kullanarak OSI 2. ve 3. katman seviyesinde şifreli ağ erişimi sağlar.

NOT:SSL VPN denilince aklı gelen bir browser aracılığı ile ek bir program gerektirmeksizin VPN yapmaktır. Fakat buradaki SSL VPN tanımı farklıdır.

OpenVPN ile yapılabilecekler;

- *Linux, Windows 2000/XP ve üzeri, OpenBSD, FreeBSD, NetBSD, Mac OS X ve Solaris işletim sistemlerinde çalıştırılabilir.
- *OpenSSL kütüphanesinin sunduğu encryption, authentication, ve certification özelliklerini kullanabilir.
- *Nat üzerinden sorunsuz tünelleme imkanı
- *İsteğe bağlı olarak GUI ile yönetim.
- *Kablosuz ağlar için güvenli erişim imkanı

OpenVPN'in kısa sürede bu kadar popüler olmasının nedeni hem güvenli bir altyapı sunması hem de kurulum ve yönetiminin basit olması denilebilir. OpenVPN, Ipsec gibi işletim sisteminin çekirdeğinde temel değişiklikler gerektirmez.

OpenVPN Kurulumu

Yaygın kullanılan üç işletim sistemi için kurulum adımları;
OpenVPN dosyalarının taşınacağı ortak alanı oluşturalım

```
#mkdir /usr/local/etc/openvpn
```

OpenBSD için Kurulum Adımları

OpenBSD üzerinde OpenVPN kurulumu için ister OpenBSD paket sistemi(ports) ister kaynak koddan kurulum yöntemi izlenebilir. Biz burada kaynak koddan kurulum ile ilerleyeceğiz

OpenBSD Kurulumu: OpenBSD 3.8 kurulumu için <http://www.enderunix.org/docs/openbsd.avi> adresindeki kurulum videosu takip edilebilir.

Kaynak Koddan kurulum

Kaynak koddan kurulum için sistemde wget programı kurulmalıdır.

OpenBSD için wget kurulumu;

```
#pkg_add -v ftp://ftp.enderunix.org/pub/OpenBSD/3.7/packages/i386/wget-1.8.2.tgz
```

Lzo Kurulumu

```
#cd /usr/ports/archivers/lzo  
#make && make install  
# mkdir /usr/src/openvpn  
# cd /usr/src/openvpn/
```

Son sürüm openvpn paketi indirilerek açılır

```
# wget http://openvpn.net/release/openvpn-2.0.5.tar.gz  
# md5 openvpn-2.0.5.tar.gz  
MD5 (openvpn-2.0.tar.gz) = ***
```

Not: <http://openvpn.net/sig.html> adresinden MD5 SHA1 imzaları kontrol edilebilir.


```
#tar zxvf openvpn-2.0.5.tar.gz
#cd openvpn-2.0.5
# ./configure --with-lzo-lib=/usr/local/lib --with-lzo-headers=/usr/local/include/
#make
#make install
#mv easy-rsa sample-scripts sample-config-files plugin contrib/
/usr/local/etc/openvpn/
```

OpenBSD paket yönetim sistemi kullanarak kurulum

```
#cd /usr/ports/net/openvpn
#make install
```

Kurulum sonrasında örnek yapılandırma ve gerekli scriptler /usr/local/share/examples/openvpn/ dizini altına atılmaktadır. Bu dizini /usr/local/etc/openvpn dizini olarak kopyalayalım.

FreeBSD için Kurulum

NOT: FreeBSD için port ağacından Kurulum için

```
#cd /usr/ports/security/openvpn
#make
#make install
```

Komutları verilmelidir.

NOT:Sistemin açılışında otomatik başlaması için /etc/rc.conf dosyasına openvpn_enable="YES"satırı eklenir.

Srandart kurulum için OpenBSD kurulum adımları takip edilebilir.

Red Hat Linux Enterprise için Kurulum

```
# mkdir /usr/src/openvpn
# cd /usr/src/openvpn/
#wget http://openvpn.net/release/openvpn-2.0.5.tar.gz
#tar zxvf openvpn-2.0.5.tar.gz
#cd openvpn-2.0.5
#./configure --with-lzo-lib=/usr/local/lib --with-lzo-headers=/usr/local/include/
#make
#make install
#mv /usr/local/etc/openvpn/
```

Kurulum sonrası genel yapılandırma

Her üç işletim sistemi için kurulum sonrasında yapılandırma dosyalarını ana bir dizine taşıyarak bunda sonraki işlemlerin üç işletim sistemi için de aynı olmasını sağladık(usr/local/etc/openvpn).

CA(Certificate Authority) Kurulumu ve sunucu/istemciler için gerekli sertifikaları oluşturma

CA bir yetki merkezidir. Sertifika ile güvenliği sağlanmaya çalışılan taraflar için güven onayı veren bir merkezdir. Güvenilir bir CA tarafından imzalanmış sertifika ile yapılan şifreleme işlemlerinin güvenliği sağlanmış olur.

Eğer kullanılan CA herkes tarafından kabul görmemiş ise CA tarafından imzalanan sertifikalar için güvenlikden söz edilemez.

OpenVPN ile birlikte kullanılacak sertifikalar için bir adet CA ihtiyacı vardır. Bu CA'ı kendiniz oluşturabileceğiniz gibi, internet üzerinden güvenilirliği kanıtlanmış CA'leri de kullanılabilirsiniz.

Her sunucu istemcisi ikilisi için birer adet public ve private key oluşturulur. OpenVPN'nin güzel bir yanı da çift taraflı onaylama desteklemesidir, yani hem kullanıcı sunucuyu hem de sunucu kullanıcıyı denetleyebilir.

Aşağıdaki komutlar kendi CA'nizi oluşturmanıza yardımcı olacaktır.

```
#cd /usr/local/etc/openvpn/easy-rsa/
# . ./vars
NOTE: when you run ./clean-all, I will be doing a rm -rf on
/usr/src/openvpn/openvpn-2.0/easy-rsa/keys
# ./clean-all
#./build-ca
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:TR
State or Province Name (full name) [NA]:
Locality Name (eg, city) [BISHKEK]:KOCAELI
Organization Name (eg, company) [OpenVPN-TEST]:ENDERUNIX
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:vpn-gateway
Email Address [me@myhost.mydomain]:huzeyfe@enderunix.org
```

Sunucu için sertifika ve anahtar oluşturma

sunucu için sertifika ve gizli anahtar oluşturma

```
#./build-key-server server
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'server.key'
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:TR
State or Province Name (full name) [NA]:
Locality Name (eg, city) [BISHKEK]:KOCAELI
Organization Name (eg, company) [OpenVPN-TEST]:ENDERUNIX
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:server
```

Email Address [me@myhost.mydomain]:server@enderunix.org

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:

An optional company name []:

Using configuration from /usr/local/etc/openvpn/easy-rsa/openssl.cnf

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'TR'

stateOrProvinceName :PRINTABLE:'NA'

localityName :PRINTABLE:'KOCAELI'

organizationName :PRINTABLE:'ENDERUNIX'

commonName :PRINTABLE:'server'

emailAddress :IA5STRING:'server@enderunix.org'

Certificate is to be certified until Dec 12 19:43:11 2015 GMT (3650 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

Istemciler için anahtar olusturma

```
# ./build-key istemci
```

```
./build-key istemci
```

```
Generating a 1024 bit RSA private key
```

```
.....+++++
```

```
.....+++++
```

```
writing new private key to 'istemci.key'
```

```
-----
```

You are about to be asked to enter information that will be incorporated
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

```
-----
```

```
Country Name (2 letter code) [KG]:TR
```

```
State or Province Name (full name) [NA]:
```

```
Locality Name (eg, city) [BISHKEK]:KOCAELI
```

```
Organization Name (eg, company) [OpenVPN-TEST]:ENDERUNIX
```

```
Organizational Unit Name (eg, section) []:
```

```
Common Name (eg, your name or your server's hostname) []:istemci
```

```
Email Address [me@myhost.mydomain]:istemci@enderunix.org
```

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:

An optional company name []:

Using configuration from /usr/local/etc/openvpn/easy-rsa/openssl.cnf

```
DEBUG[load_index]: unique_subject = "yes"
```

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'TR'

stateOrProvinceName :PRINTABLE:'NA'


```
# cp * /usr/local/etc/openvpn/
#cd /usr/local/etc/openvpn# cp -rp /usr/src/openvpn/openvpn-2.0/easy-rsa/keys/
/usr/local/etc/openvpn/
```

Buraya kadarki adımlarla kurulum sürecini tamamladık. Şimdi de OpenVPN'in çalışma yapısına gözatarak nasıl yapılandırılacağına gözatalım.

OpenVPN Çalışma Modları

OpenVPN iki farklı modda çalışabilir. Bridge mod ve route mod. Gereksiniminize göre bu iki çalışma yönteminden birini kullanabilirsiniz.

Bridge Mode Çalışma Yapısı

Bridge mod, WAN üzerinde bir ethernet LAN'ı oluşturmak için kullanılır. Yani birbirinden farklı lokasyonlardaki makineleri, ağları tek bir ethernet ağındaymış gibi haberleştirebilirsiniz. Bridge mode daha çok özel gereksinimler için tercih edilmektedir. Mesela broadcast paketler aracılığı ile haberleşen bir uygulamanız varsa bridge mod kullanmanız kaçınılmazdır. Bridge modda tap sahte arabirimleri kullanılır.

Route Mod Çalışma Yapısı

Routing mod biraz daha rahattır ve özel bir gereksinim olmadığı müddetçe(IPX gibi IP tabanlı olmayan protokollerin kullanımı gerektiğinde) routing mod kullanımı tavsiye edilmektedir. Route modda tun sahte arabirimleri kullanılır.

Sunucu Tarafı Yapılandırma Dosyası – server.conf

OpenVPN çalışma parametrelerini komut satırından alabileceği gibi bir dosyaya düzenli bir şekilde yazarak bu dosyadan da okuma yapabilir. Tercih edilen yöntem, tüm yapılandırma parametrelerini bir dosyaya(server.conf) yazarak bu dosyadan okutmaktır.

Öntanımlı olarak bu dosya server.conf'tur. Server.conf dosyasında sık kullanılan bazı parametreler ve anlamları;

NOT: server.conf dosyasında # ya da ; ile başlayan satırlarıyorum satırı olarak algılanır ve herhangi bir etkisi yoktur. Bir parametrenin önündeki ;, # işaretlerini kaldırarak o parametreyi aktif hale getirmiş oluruz.

VPN Sunucu hangi IP üzerinden çalışsın?

local a.b.c.d

a.b.c.d ile belirtilen IP adresi üzerinden çalışacağını belirtir. Sunucumuzda birden fazla IP adresi varsa bu adresler arasında seçim şansı verir.

VPN sunucu Portu

port 1194

VPN sunucunun hangi port üzerinden çalışacağını belirtir. Aynı makine üzerinde birden fazla OpenVPN çalıştırılacaksa bu parametre her yapılandırma dosyası için farklı olmalıdır.

proto udp

Hangi protokolün kullanılacağını belirtir. Varsayılan ve tavsiye edilen değeri UDP'dir.

Route mod mu Bridge mod mu?

```
;dev tap0  
dev tun0
```

Layer 2 VPN kullanmayı düşünüyorsanız bu değer tap olmalıdır. Eğer OpenVPN'i route modda kullanmak isterseniz tun arabirimi kullanılmalıdır.

NOT: TAP, Tun Arabirimleri;

Tun Arabirimi: Sanal bir ağ bağdaştırıcısıdır. Üzerinde çalıştığı makine için bir PPTP arabirimden farksızdır. Programcı tun arabirimini herhangi bir dosya gibi kullanarak istediği bilgileri okur ve yazar. Tap arabirimi de Tun'e benzer fakat sadece ethernet arabirimleri simüle edebilir.

```
ca /usr/local/etc/openvpn/certs/ca.crt
```

CA Sunucunun sertifikası. Burada tam yol belirtilmelidir. Bu sertifika tüm sunucu ve istemcilerde bulunmak zorundadır.

cert /usr/local/etc/openvpn/certs/server.crt

VPN sunucunun sertifikası. Sadece sunucu tarafında bulunmalıdır.

```
key /usr/local/etc/openvpn/certs/server.key
```

bu dosya çok önemlidir. Diğer tüm sertifikaları imzalamada kullanılır.

```
dh /usr/local/etc/openvpn/certs/dh1024.pem
```

Diffie hellman parametrelerinin bulunduğu dosya

VPN İstemcileri Ağ Yapılandırması

```
server 10.8.0.0 255.255.255.0
```

VPN sunucuya bağlanarak IP alacak istemcilerin IP havuzunu belirler. Havuz içinde ilk IP adresi VPN sunucunun IP adresi olacaktır.

```
ifconfig-pool-persist ippler.txt
```

VPN sunucuya bağlanarak IP adresi alan istemcilerin kayıtlarını tutar. VPN sunucuda yaşanacak bir bağlantı kopması sonrasında istemcilerin eski IP adreslerini almalarını sağlar.

```
;push "route 192.168.20.0 255.255.255.0"
```

VPN ile bağlanan istemcileri VPN sunucu arkasındaki başka ağlara da erişim izni için yönlendirme tanımı.

İstemciye Özel IP atama

Bazı istemcilerinize özel ip ataması yapmak isterseniz istemcilerin sertifikalarında kullandıkları CN tanımına göre özel ip atamaası yapabilirsiniz.

Örnek;

Sertifikasında CN'si enderunix olan istemciye 10.9.0.1 ip'sinin atanmasını istiyoruz.

```
client-config-dir özel
```

```
route 10.9.0.0 255.255.255.252
```

/usr/local/etc/openvpn/ozel dizinini oluşturarak içine enderunix adlı bir dosya açılır ve bu dosyaya aşağıdaki satır eklenir.

```
ifconfig-push 10.9.0.1 10.9.0.2
```

VPN Kullanıcının Tüm Trafiğini Yönlendirmek

```
push "redirect-gateway"
```

VPN sunucuya bağlanan istemcilerin varsayılan geçit yolunu(default gateway) VPN sunucu olarak ayarla manasızan gelir. Böylece istemcinin özel olarak yönlendirilmemiş tüm trafiği VPN gateway aracılığı ile çıkacaktır. Burada istemcileri internete çıkarmak için VPN sunucu makinesinde NAT yapılması da gerekir. Linux, FreeBSD ve OpenBSD işletim sistemlerinde NAT işleminin nasıl yapıldığı öğrenilmelidir.

OpenBSD PF için nat tanımı:

```
ext_if="fxp0"  
VPN_AGI="100.100.100.0/24"  
nat on $ext_if from VPN_AGI -> ($ext_if)
```

VPN istemcilerinin birbirini görmesi

OpenVPN varsayılan yapılandırımı ile VPN istemcileri sadece VPN sunucuyu görecektir. Birbirlerini görebilmeleri için

```
;client-to-client
```

Tanımı girilmelidir. İstemcilerin sadece VPN sunucuyu görmelerini kesin olarak sağlamak için VPN sunucu üzerindeki Güvenlik duvarı uygun şekilde yapılandırılmalıdır.

Aynı sertifika ile birden fazla İstemci

Aynı sertifika ile birden fazla istemcinin VPN yapabilmeleri için

```
duplicate-cn
```

tanımı kullanılmalıdır. Aksi takdirde VPN ağına bağlanan her istemci aynı IP adresini alacaktır. Biraz karışık bir özellik ve sadece test amaçlı kullanılması öneriliyor.

```
keepalive 10 120
```

Sunucu ve istemcilerin birbirinin durumundan haberdar olmalarını sağlayan bir yapı. Anlamı her 10 saniyede bir kontrol et, 120sn içerisinde cevap gelmezse bağlantıyı kopar.

VPN Hattında Sıkıştırma

```
comp-lzo
```

kullanılır. Bu tanım hem sunucuda hem de istemcide kullanılmalıdır.

Eşzamanlı VPN Kullanıcısı

```
max-clients 100
```

eşzamanlı 100 kullanıcıya izin ver.

OpenVPN durum Kontrolü

```
status openvpn-status.log
```

tanımı ile yapılır. VPN sunucunun durumu hakkında özet bilgi için.

Loglama

```
log /var/log/openvpn.log
```

```
log-append /var/log/openvpn.log
```

```
verb 3
```

```
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
```

VPN sunucu başlatma ve çalışma zamanı için loglarını atacağı dosya. VPN sunucuda problem yaşandığında ilk bakılması gereken dosyadır.

NOT: Tüm geçerli parametreler için örnek server.conf dosyasının incelenmesi faydalı olabilir.

Örnek Yapılandırma Dosyaları

Aşağıdaki istemci ve sunucu yapılandırma dosyaları temel bir VPN ağı oluşturmak için gerekli yapılandırmaları içermektedir. Kendi ihtiyacınıza göre bu değerlerle oynayabilirsiniz.

Örnek istemci dosyası

```
-----
client
dev tun0
proto udp
remote 194.27.72.88 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert istemci.crt
key istemci.key
ns-cert-type server
comp-lzo
verb 3
-----
```

Örnek sunucu konfigürasyonu

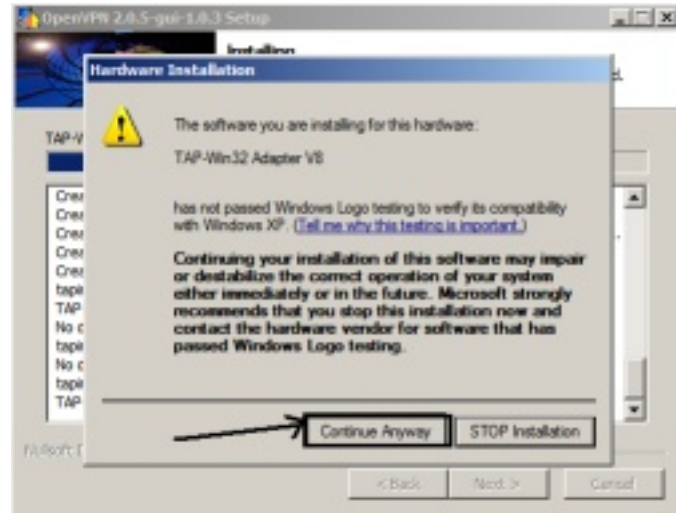
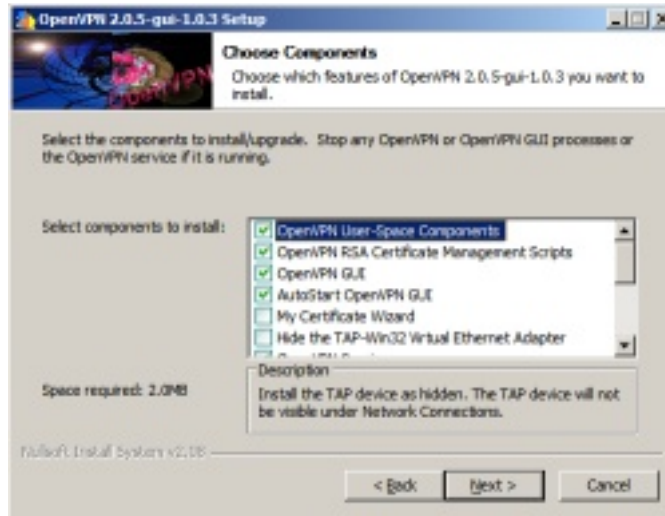
```
-----
local 14.2.2.8
port 1194
proto udp
dev tun0
ca easy-rsa/keys/ca.crt
cert easy-rsa/keys/sunucu.crt
dh easy-rsa/keys/dh1024.pem
server 100.100.100.0 255.255.255.0
ifconfig-pool-persist ipp.txt
```

```
push "redirect-gateway"
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
log /var/log/openvpn.log
verb 6
-----
```

Windows XP OpenVPN istemci Kurulumu

<http://openvpn.se/download.html> adresinden son sürüm “stable” OpenVPN-gui paketini indirerek işleme başlayalım.

Yazı hazırlarken son sürüm openvpn-gui: openvpn-2.0.5-gui-1.0.3-install.exe

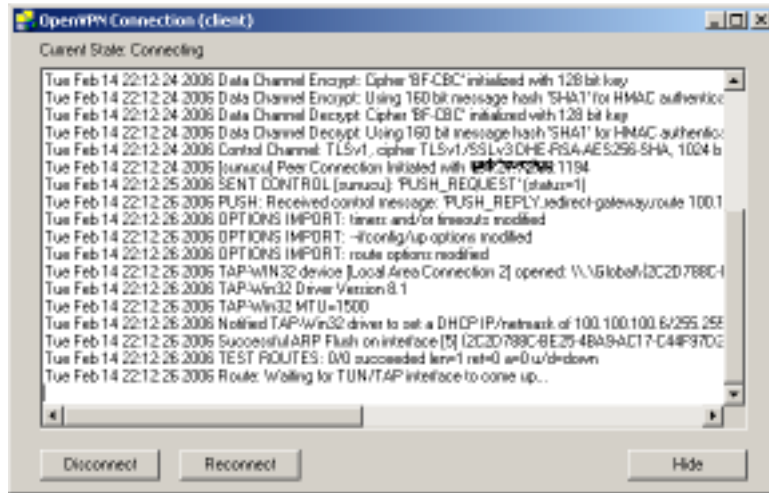


Kurulum adımları tamamlandıktan sunucu tarafında oluşturduğumuz istemci sertifikalarını C:\Program Files\OpenVPN\config dizini altına kopyalayarak istemci tarafı VPN yapılandırma dosyasını uygun şekilde düzenleyelim.

Laptop.ovpn adlı bir dosya oluşturarak içine örnek istemci dosyasındaki gibi değerleri yazarak VPN bağlantısını başlatabilirsiniz.



Bağlantı kurulumu esnasında aşağıdaki ekrana benzer bir pencere açılarak bağlantı durumunu gösterecektir.



Kaynaklar

[ref 1] "Implementing OpenVPN". Florin Andrei. Mar.26.2004

http://fedoranews.org/contributors/florin_andrei/openvpn/

[ref 2] OpenVPN Articles.

<http://openvpn.net/articles.html>

[ref 3] " Sanal Özel Ağ, Kriptoloji ve PKI Teknolojileri". Serkan YILMAZ.

<http://penguence.linux.org.tr/?~p=dergi&action=show&which=77>

[ref 4] "AçıkKod VPN Çözümleri". Huzeyfe ÖNAL.

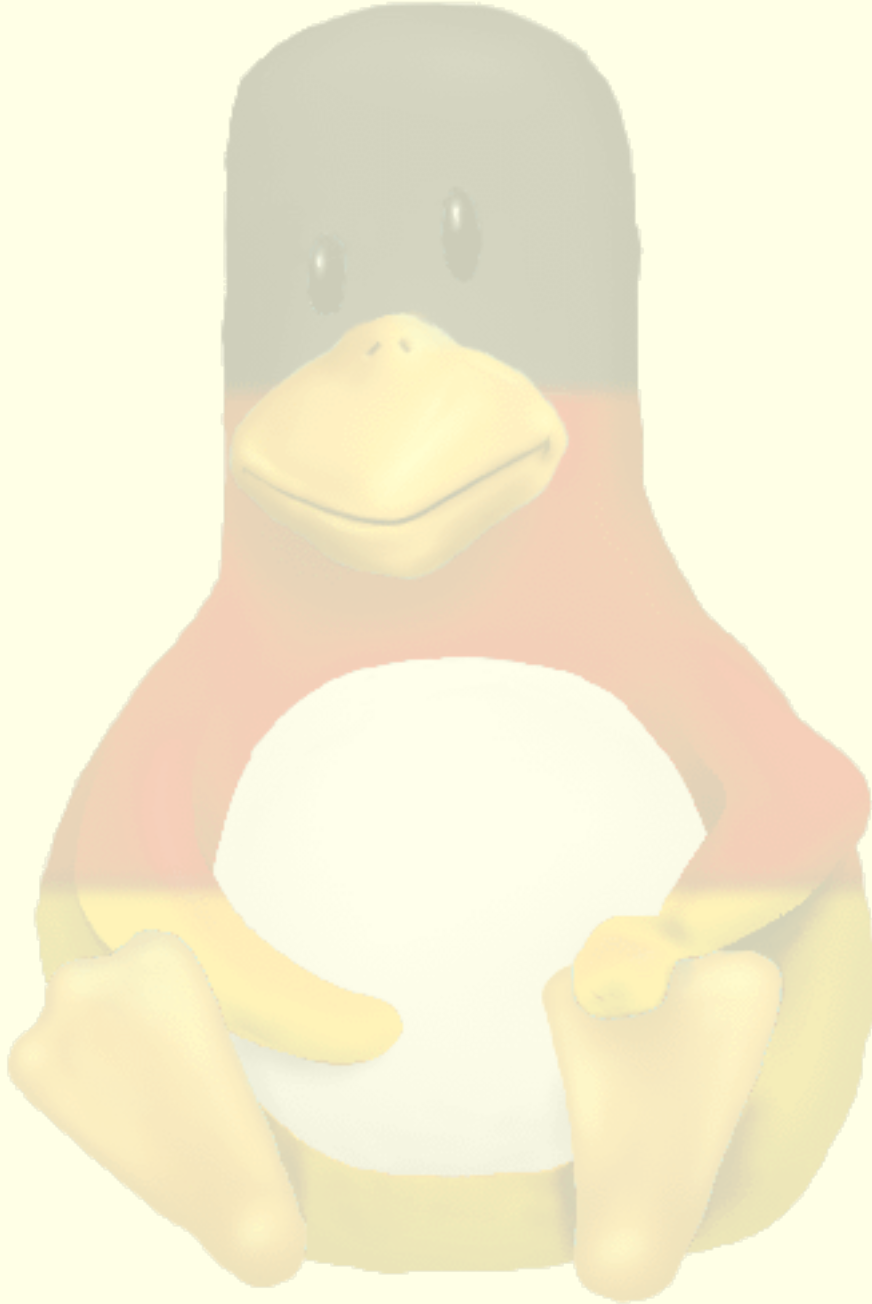
<http://www.enderunix.org/slides/Internet%20Konferanslari/acikkodvpn.pdf>

[ref 5] "OpenVPN Howto".

<http://openvpn.net/howto.html>

5. Linux Ve Özgür Yazılım Şenliği

Korsan değil , özgür yazılım !



11-14 Mayıs 2006

ODTÜ Kültür Ve Kongre Merkezi

Gezgin Penguen'in 4. turuna hoş geldiniz!

Arda Çetin

Yine oltamıza takılan bazı haber ve bağlantıları sizlerle paylaşmaya devam ediyoruz...

Pardus Cephesi

Ulusal Dağıtımımız Pardus'un 27 Aralıkta çıkan 1.0 sürümüyle birlikte internette Pardus ile ilgili bir çok adres açılmaya başlandı. Bunlardan belki de en önemlisi **www.pardus-wiki.org** adresinde yayın yapan PardusWiki. PardusWiki, "PardusMan" tarafından kurulan ve Pardus gönüllüleri tarafından hazırlanan, Uludağcıların da yazdıkları belgelerle destekledikleri bir Pardus belge deposu. PiSi paket yapımından, çeşitli sürücülerin kuruluma kadar bir çok belgeyi burada bulabilirsiniz. Ayrıca PiSi'lenmesini istediğiniz paketi PardusWiki'deki Paket İstekleri başlığı altına yazarak geliştirmeye isteğinizi ulaştırabilirsiniz.

Pardus cephesinden diğer bir adres **www.pardus-linux.org** adresi. Yine gönüllüler tarafından kurulan bu sitedeki forum sayesinde özellikle e-posta listeleri ile arası iyi olmayan Pardus kullanıcıları yardım alabilir, bilgi alış verişinde bulunabilirler. Söz e-posta listelerinden açılmışken, Uludağ listelerine **liste.uludag.org.tr** adresinden ulaşabilirsiniz.

Pardus cephesinden son adresimiz

sanat.uludag.org.tr.

Sanat.Pardus, Pardus'un daha zengin bir arayüze kavuşmasına olanak sağlamak, Pardus kullanıcılarını bir araya getirerek sosyal bir ortam yaratmak, Pardus'a gönül verenlerin masa üstünü daha da keyifli bir hale getirmek için hazırlanan bir web sitesi. Buradan çeşitli KDE stilleri,

pencere dekorasyonları, simgeler, temalar ve duvar kağıtları temin edebilir, masa üstünüzü daha da güzelleştirebilirsiniz. Hatta bu masaüstlerinizin ekran görüntülerini diğer Sanat.Pardus ziyaretçileriyle paylaşabilirsiniz. Dilerseniz sitedeki diğer uygulamalara yorum gönderebilirsiniz.



Canlı OpenSolaris geldi

SUN Microsystems, Solaris'in kaynak kodlarını açıp açık kaynak kod dünyasına OpenSolaris'i tanıttıktan sonra ilk OpenSolaris tabanlı işletim sistemi live(canlı/çalışan) CD olarak Belenix duyuruldu: **http://belenix.sarovar.org** Direk CD'den çalışan Belenix, içerisinde GTK paketleri ve XFCE masa üstü ortamı ile birlikte geliyor.

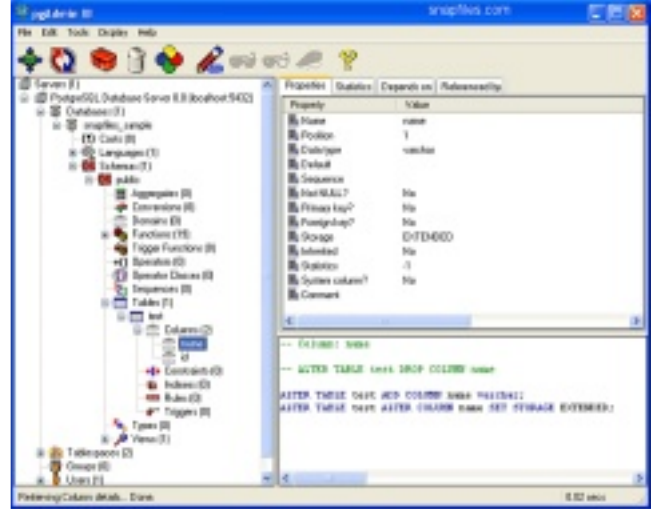
Söz OpenSolaris'den açılmışken; ücretsiz Solaris paketlerini **www.sunfreeware.com** adresinden temin edebilirsiniz

PostgreSQL ile Programlama Kitabı Projesi

PostgreSQL ile Programlama Volkan Yazıcı tarafından başlatılmış, PostgreSQL veritabanına C, PHP ve Python programlama dilleri kullanılarak nasıl bağlanılacağı hakkında incelemede bulunan oldukça kapsamlı bir kitap projesi. Proje GNU Genel Kamu Lisansı altında korunmakta:

<http://www.students.itu.edu.tr/~yazicivo/doc/postgresql-ile-programlama.html>

Projeye her türlü katkı sağlamak mümkün; direk para göndermekten, arkadaşlarınızı bu projeden haberdar etmeye kadar yazar her türlü teklife olumlu bakıyor.



RMS ile Ropörtaj!

LinuxP2P.org sitesi Richard Stallman'la P2P paylaşımları ve Creative Commons lisansı üzerine okunası bir ropörtaj gerçekleştirmiş.

Meraklılarına duyurulur;

<http://www.linuxp2p.com/forums/viewtopic.php?t=374>

ID Software'dan Hediye!

Çoğumuzun çocukluğundan bildiği ID Software firması açık kaynak kod severleri sevindirecek bir haber duyurdu: Wolfenstein: EnemyTerritory tamamen açık kaynak kodlu ve ücretsiz olarak GNU/Linux kullanıcılarının hizmetinde bir oyun. Quake ve Doom benzeri oyunlara meraklı penguenele mutlaka tavsiye edilir:

<http://www.3dgamers.com/games/wolfensteinet/>



Şenlik Geliyoor!

5. Linux ve Özgür Yazılım Şenliği'ne gün geçtikçe az kaldı ve şenlik web sayfası sizlerin ziyaretini bekliyor: <http://senlik.linux.org.tr>

Siteden edinebileceğiniz şenlik bannerlarını kendi web sayfanıza/blogunuza koyarak siz de şenliğin daha çok penguene duyurulmasını sağlayabilirsiniz. Ayrıca şenlikte aktif görev almak isteyenler için bir de Gönüllü Penguin başvuru formumuz var.

Şenliğe kayıt ve seminer programını da etkinliğe yakın tarihlerde sitede bulabileceksiniz.

Bu sayımızdaki turumuz bu kadar. Bir sonraki sayıda görüşmek üzere...