

GNU Pardus-Linux.Org eDergi

Baharı karşılarken...

*Google ChromeOS ve Bilişim Piyasasındaki Tepkiler
C Dili ile Yazılım Geliştirme - 1 (Giriş)*

SourceForge.Net Röportajı



Firefox'i Kişiselleştirmek

Google Buzz Üzerine

Wesnoth'a Dalış - 1

Pardus'ta Django

Yo Frankie!

Ekran Klavyesi: Plasmaboard

Bilim, Bilişim, Eğitim ve Özgür Yazılım Üzerine Yazılamalar



Giriş Yazısı	3
Ekran Klavyesi: Plasmaboard	4
Oyun Tanıtımı: Yo Frankie!	6
Oyun Tanıtımı: Wesnoth'a Dalış - I	11
C Dili ile Yazılım Geliştirme - I	20
Pardus'ta Django: Veritabanı	24
Pardus'ta Django: Blog Uygulaması	28
Firefox'u Görsel Olarak Kişiselleştirmek	36
Röportaj: SourceForge.Net	40
Google Buzz Üzerine	42
Google Chrome OS ve Bilişim Piyasasındaki Tepkiler	45
Bilim, Bilişim, Eğitim ve Özgür Yazılım Üzerine Yazılamalar	47
Kitap İncelemesi: Herkes İçin Python	52
Haberler	54

Merhaba Özgür Yazılım Dostları,

Yine, beğeneceğinizi umduğumuz, siz saygıdeğer okurlarımız için yararlı olması için çalıştığımız ve iyi veya kötü eleştirilerinizden başka hiçbirşey beklemeden tamamen gönüllü bir ruhla, Özgür Yazılım'ı savunan bir düşünceye yakışır şekilde sadece Özgür Yazılımlar kullanılarak hazırladığımız bir sayımızla da karşınızdayız.

Dergimizin bu sayısı yine bir Özgür Yazılım olan OpenOffice.org ile hazırlanmış olan ve Pardus Dergi adıyla yayınlanan ilk sayımızın çıkış tarihinden tam olarak dört yıl sonra çıkıyor. Daha Pardus ülkemizde bu denli yaygın olmadığı ve Pardus'un kullanımının bugüne nazaran zor olduğu o günlerde, yani Mart 2006 sayısı için hazırlık döneminde, dergimizin ilk sayısını çıkaran Burak Bakırtaş, Mustafa Tosun, Tümel Şenel, Buğrahan Korkmaz'a sevgilerimizi iletiyoruz. Onların açtığı yolda emin adımlarla yürüyor ve siz saygıdeğer okurlarımız ve değerli katkıcılarımızla engelleri aşmayı öğreniyoruz.

İkinci ve üçüncü sayılarımız ise sayımız ise, Mart 2006'dan oldukça bir süre sonra ev, tam olarak 28 ve 29 ay sonra, Temmuz 2008 ve Ağustos 2008 aylarında sırasıyla Sevgili Metehan Özbek ve Sevgili

Ozan Aydın'ın editörlük görevini üstlenmeleriyle çıkarılmış. Pardus'un yaygın olarak kullanılmaya başlandığı o aylarda kendilerine ve o sayılarda emeği geçen tüm ekibe, özverili çalışmalarından dolayı sevgilerimizi iletiyoruz.

Dördüncü sayımız ise dergi çalışmalarını başlatmak istememize rağmen bir türlü uygun ortamı sağlayamamamız nedeniyle Ocak 2009'a dek hazırlanmayı bekledi. Sevgili Serdar Gündüz'ün girişimleri ile yeniden canlanan dergimiz, dördüncü sayımızdan beri sizlerin duymuş olduğu ilginin de vermiş olduğu gaz ile, bazı küçük aralıklar yaşansa da sürekli olarak yayınlanmaya devam etmekte. Bu süre zarfında R.M. Stallman, Mustafa Akgül, Doruk Fişek, Murat Koç, Serdar Köylü, Necdet Yücel, Sebastian Kügler, Elizabeth Krumbach, Ender Ateşman gibi önemli isimlerle röportaj yaptık ve Pardus ile GNU/Linux hakkındaki düşüncelerini öğrendik. Serdar Gündüz'e güzel hamlesinden dolayı teşekkür ederiz.

Dergimizin bu sayısında, yani 17. sayımızda, beğenerek okuyacağınızı umduğumuz oyun tanıtımları, makaleler, özelleştirmeler, yazılım geliştirme bilgileri gibi konulara değindik. Pardus-eDergi.Org, Pardus-Linux.Org ve dergimizin duyuru-

sunun yayınlandığı diğer sitelerdeki tanıtımlarımızda içeriğin neler olduğunu öğrenmişsinizdir diye düşünerek, bunların neler olduğuna yazımın içinde değinmeyeceğim.

Dergilerden söz LinuxMint'in Türkiye Topluluğu, <http://www.linuxmint.org.tr> adresinde tekrar yayın hayatına başlamış ve yeni bir dergi projesi başlatma hazırlıklarında. İsmi Root olacağını öğrendiğim dergiye başarılar diliyorum, güvenilir ve çalışmaları yarı yolda bırakmayacak insanlar çalışıyor olmalarını temenni ediyorum. Siz okurlardan ricam ise yazılarınızla, çizimlerinize, fikirlerinize Root ve tabii ki Pardus-Linux.Org eDergi'yi süslemeniz. Bizim bunlara ihtiyacımız var.

Sözlerimin sonuna gelirken, dergiler için her türlü eleştirinin bizler için değerli olduğunu bilmenizi isterim. Bu nedenle tüm fikir, öneri, soru, sorun ve eleştirilerinizi dergimizin duyurusunun yapıldığı herhangi bir sayfada veya elektronik posta olarak dergi@pardus-linux.org adresine göndermenizi rica ediyorum.

Son olarak 8 Mart Dünya Emekçi Kadınlar Günü dolayısıyla tüm emekçi kadınlarımızın bu özel günlerini kutlarız, hak ettiğini alan bir toplum dilerim.

Giriş

Plasmaboard, Pardus'ta (daha doğrusu KDE4 kullanan herhangi bir GNU/Linux dağıtımında) ekran klavyesi olarak kullanılabilen bir KDE4 masaüstü programcıdır.

Kurulum

Plasmaboard'u Pardus'a kurmak için üç farklı yöntem izlenebilir. Bunların ilki programcığın kaynak kodlarını indirip, derleyerek kurmak. İkincisi ise KDE4 masaüstü programcığı yükleme aracını kullanmak ve üçüncüsü de tarafımızca hazırlanmış olan PiSi paketini yüklemek.

KDE 4.4 ile birlikte *kdeplasma-addons* paketinin bir üyesi olarak gelecek bu programcığın, yukarıda geçen üç farklı kurulum yolu şu şekilde:

1- Kaynak Koddan Derleyerek Kurulum

Plasmaboard'un kaynak kodları [buradan](#) indirilebilir.

Plasmaboard'un kurulumuna geçmeden önce, derleme mekanizmasının ihtiyacı duyacağı bazı paketlerin önceden yüklen-

miş olduğundan emin olmak gerekiyor. Bu paketler *GCC*, *cmake*, *kdelibs-devel*, *xorg-server-devel* ve *qt*. Bu paket isimleri Pardus 2009 depolarındaki isimlere göre yazılmıştır. Paket isimlerinin GNU/Linux dağıtımları arasında değişiklik gösterebileceğini hatırlatmak isterim.

Yukarıda ismi geçen paketlerin kurulu olduğundan emin olduktan sonra, Plasmaboard'un derlenmesi ve kurulumu aşağıdaki komutların verilmesi ile gerçekleştirilebilir:

```
cmake -DCMAKE_INSTALL_PREFIX=`kde4-config --prefix`
make
sudo make install
```

2- KDE4 Programcık Yükleme Aracı Kullanarak Kurulum

KDE4, masaüstü programcıklarını kullanıcıyı uğraştırmadan kurulumunu sağlamak için, doğrudan [kde-looks.org](#) adresine bağlanan bir uygulamaya sahip. Bu uygulama masaüstünde gerçekleştirilecek şu hamlelerle çalıştırılabilir:

- 1- Masaüstünde sağ tık
- 2- Programcık Ekle
- 3- Programcık Ekle penceresinde Y

- Yeni Programcık Kur
- 4- Yeni Plasma Gereçleri İndir
- 5- Arama : Plasmaboard
- 6- Kur

3- PiSi Paketi Kullanarak Kurulum

Plasmaboard'un tarafımızca hazırlanmış olan PiSi paketini <http://paketler.pardus-linux.org/P2009-free/plasmaboard-0.91-1-1.pisi> adresinden indirip kurabilirsiniz. Veya isterseniz, Pardus-Linux.Org PiSi Paketleri Deposunu sisteminize ekleyip, doğrudan paket yöneticisini kullanarak da Plasmaboard paketinin kurulumu gerçekleştirebilirsiniz.

Küçük bir not: Pardus-Linux.Org PiSi Paketleri Deposu, resmi olmayan, resmi olarak desteklenmeyen bir depodur. Deponun amacı resmi depolarda bulunmayan uygulamaların PiSi paketlerini Pardus kullanıcılarına sunmaktır. Depomuzdaki paketlerin resmi ve katkıcı deposundaki paketlere bağımlılıkları bulunabilir. Bu nedenle resmi ve katkıcı deposunun sisteminizde ekli olması önemle önerilir. Paket depomuzu eklemek isteyenler, depo adresi olarak <http://paketler.pardus-linux.org/P2009-free/pisi-index.xml.bz2> adresini kullanabilir. PiSi çalışmalarımıza ortak olmak veya paket isteğinde bulunmak isterseniz, Pardus-Linux.Org | Forum aracılığıyla iletişime geçebilir

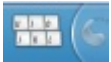
veya paketler@pardus-linux.org adresine elektronik postalarınızı gönderebilirsiniz. PiSi paketlerinin inşa dosyalarını <http://svn.pardus-linux.org/paketler> adresinde bulabilirsiniz.

Kullanım

Plasmaboard'un iki farklı ana kullanım seçeneği bulunmakta. Bunlardan birincisi doğrudan masaüstüne ikincisi panele eklemek.

Masaüstüne eklemek için *Sağ Tık > Programcık Ekle*, panele eklemek için ise ön tanımlı olarak en sağda bulunan *Plasma* işaretine tıklayıp *Programcık Ekle* yolu izlenebilir. Bu adımlardan sonra açılacak olan *Programcık Ekle* penceresinde, Plasmaboard'u "*klavye*", "*keyboard*" veya "*Plasmaboard*" adıyla bulabilirsiniz.

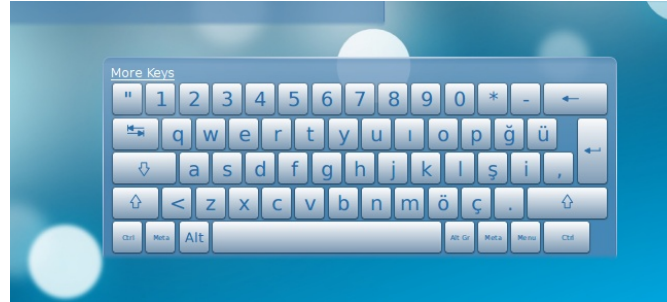
Panele eklenen Plasmaboard'un ekran görüntüsü şu şekilde:



Paneldeki simgeye tıkladığında oluşan ekran görüntüsü şu şekilde:



Masaüstüne eklenen Plasmaboard'un ekran görüntüsü şu şekilde:



Plasmaboard'un "Daha Çok Tuş" ile görünümünü şu şekilde:



Pardus'unuzda ekran klavyeniz hayırlı olsun.

Mutlu yarınlar dilerim.



Giriş

Yo Frankie!, tek cümle ile özetleyecek olursak, üç boyutlu bir platform oyun projesi. Ancak Yo Frankie!, aslında oyuncularından ziyade, Blender3D kullanıcılarının ilgisini oldukça çekeceğe benziyor. Windows kafası ile oyun oynayacak, yani "Ben oyunumu kurarım arkadaş. Bana ne onun kaynak kodundan, yok geliştirilmesinden falan." diyecek bir oyuncu için, fazla bile banal kalabilecek bir oyun.

Bak şimdi acıdım Windows oyuncularına işte. Oyunlarını sadece alıp, oynayabiliyorlar. Linux oyuncuları gibi sevdikleri oyunlara bir şeyler katmak veya onun ge-

liştirilmesinde katkıda bulunmak gibi bir şansları yok zavallıların. :-) Neyse, konuyu dağıtmayalım...

Oyunu, resmi sitesinden indirebilirsiniz. Oyun, Blender ve Big Buck Bunny projelerini de barındıran DVD versiyonundan 7 ay sonra, indirilebilir olarak yayınlandı.



Buraya hemen bir not düşeyim. Big Buck Bunny, Peach Projesi kapsamında, Blender kullanılarak yapılmış, üç boyutlu bir açık kaynak çizgi animasyon film projesi idi. Onu da sitesinden indirebilirsiniz. Hem Yo Frankie! oyununu, hem de Big Buck Bunny'yi, Blender Vakfı'nı desteklemek amacıyla, DVD olarak da satın almanız mümkün.

"Orange" ve "Peach" projelerinden sonra, 1 Şubat 2008 tarihinde, "Apricot" adıyla, Blender Institute geliştiricileri tarafından yeni bir açık kaynak projesi başlatıldı. Ancak bu sefer bir animasyon film değil, bir oyun projesi olarak başlatıldı. Projenin amacı, animasyon ve modelleme

için Blender, 3B motor olarak Crystal Space 3D, kodlama içinse Python kullanarak, platform bağımsız bir oyun oluşturmaktı. Projenin ortasında ise, Blender'ın dahili oyun motoru olan BIGE (Blender Internal Game Engine) de Apricot projesine dahil oldu. Böylece bu proje, Ağustos 2008 tarihinde tamamlandı. Apricot projesi, CCL (Creative Commons Attribution License) altında lisanslanmıştır. Yani,

serbestçe dağıtımını, kopyalamasını ve geliştirilmesini yapabilirsiniz.

Kurulum

Linux versiyonu, BIGE'yi kullanıyor. Mevcut sürüm olan 1.1 sürümünü zip formatında indirdikten sonra, unzip komutu ile zip arşivini açıyoruz (veya öntanımlı ola-

rak Ark'ı da kullanabilirsiniz). Aynı bir dizin belirttiyseniz, belirttiğiniz dizinde; belirtmediyseniz, zip dosyasının olduğu dizinde "yofrankie_1_1b_bge" isminde bir klasör oluşacaktır. Yo Frankie! oyununu oynayabilmek için, Blender3D 2.49a veya daha üst sürüm Blender'ın bilgisayarınızda kurulu olması gerekiyor. Oyunun kendisi herhangi bir kurulum işlemi gerektiriyor. Blender kurulu değilse, yönetici yetkilerini ile "pisi it blender" komutu ile kurabilirsiniz.

Daha sonra, oyunu açtığımız klasöre giderek "yo_frankie_stub.blend" dosyasına tıklıyoruz ve Blender açılıyor. Blender'da, Game > Start Game yolunu izleyerek oyunu başlatıyoruz. Yandaki ana menü önümüze geliyor.

Oynanış

Seçenekler kısmından oyunun grafik ayarlarını, shader efektlerini, her iki oyuncu için tuş kombinasyonlarını yapılandırabiliyoruz. Çok daha fazla ayarı, Blender ile de yapabiliyorsunuz aslında. Oyunu başlatmadan önce, Blender'da ayar yaparak, bilahare oyunu başlatıp, oyunu yaptığınız ayarlarla oynayabilirsiniz. Ben, -hiç olmayan Blender bilgimle- rastgele





ışkın olduğumuz -hatta pekçoğumuzun aradığı- fare ile etrafa bakınma yok.

Saldırı hareketini; karşılaştığımız düşmanlara kuyruk darbeleri vurmak ya da dibinde iken tekme atmak için kullanıyoruz. Ayrıca saldırı tuşuna basılı tuttuğumuzda karakterimiz koşuyor. Yine bir düşmana yakınlaşıp saldırı tuşuna basınca, karakterimiz ona tekme atıyor. Oyunda düşmanlardan başka koyunlar ve zıplatici yaylar var. Yanlarına gidip saldırı tuşuna bastığınızda bunları tekmeyle hava-

landırıyor; altlarına girerek bunları taşıyabiliyorsunuz. Fırlatma tuşu ile bunları istediğiniz yere fırlatıyorsunuz. Mesela koyunlar belli bir süre suyun üzerinde kalabiliyorlar. Onları, nehirleri geçmek için platform olarak kullanabiliyorsunuz. Bu arada, karakterimiz yüzemiyor, bilginiz olsun. Su, boyu geçtiği anda boğuluyor-sunuz.

Zıplama tuşu -haliyle- zıplamak için kullanılıyor, ancak iki defa basıldığında, salto atarak, daha yüksek yerlere zıplamamızı

denediğim ayarlarda, sadece "zoom" ayarını -oyunu bozmadan- becerebildim. Diğer tüm denemelerimde, oyunun bir tarafını uçurdum.

Oyunu tek kişi mi, yoksa iki kişi mi oynayacağınıza da seçeneklerden karar veriyorsunuz. Oyun, bölünmüş ekran moduyla, iki kişi olarak da oynanabiliyor.

Tek kişilik modda, Big Buck Bunny'deki şeytan ruhlu, kötü sincap Frankie'yi oynatırken; iki kişilik modda, Frankie'ye Maymun Momo eşlik ediyor.

Oyunda; saldırı, zıplama, fırlatma ve yönlendirme hareketlerimiz var. Tüm bu hareketleri, varsa Joypad'imiz, yoksa klavyemiz ile yapıyoruz. Bu tip oyunlarda a-





sağlıyor. Yine basılı tutunca, karakterimizin belli bir süre havada süzülmesini sağlıyor (Frankie bir uçan sincap, Momo ise kuyruğunu kullanıyor).

Fırlatma ise yerlerden topladığımız fındıkları ve öldürdüğümüz düşmanlardan çıkan kemikleri fırlatmaya yarıyor. Önümüze çıkıp da bize saldıran fareler, koçlar ve piranaları kuyruk ya da tekme ile öldürdükten sonra, üstlerine bir yıldırım düşerek onları parçalıyor. Bu yıldırım düştüğünde yakınında olursanız, siz de nasibinizi alıyorsunuz. Çıkan kemikleri toplayarak, fırlatmak için cephane edinmiş olu-

yorsunuz. Olan yaratıklar, bir süre sonra, eski yerlerinde tekrardan beliriyorlar.

Oyunun amacı bir noktadan, diğer bir noktaya varmak. Ancak bu o kadar kolay olmuyor. Önce yolu bulmanız ve ardında da o yoldan gidebilmeniz gerekiyor. Tahmin edeceğimiz gibi sorun karşınıza çıkacak düşmanlar değil; yolun bizzat kendisi. Çünkü bölüm sonuna varmak için, olmadık yerlerden geçmeniz, abuk subuk yerlere tırmanmanız, hoplayıp zıplamanız gerekiyor. Mesela "Nut" bölümünde, büyük bir mancınığa geliyorsunuz. Geçebilmek için, karşı ağırlığa yetecek kadar ko-

yunu toplayıp mancınığın sepetine atınca, mancınık, sizi uçuşumun öbür tarafına fırlatıyor.

Oyun başlayınca kademe seçme ekranına geliyoruz. Önümüzdeki levhada da belirttiği gibi, bölüm isimlerinin yazılı olduğu tabelalara değerek o bölüme giriyoruz. Girdiğimiz bölümün sonunda vardığımız bir yeraltı tüneli ise, bizi tekrar bölüm seçme bölümüne geri getiriyor.

Oyunda bir puan sistemi ya da bir amaç mevcut değil. Karakterimiz ölebiliyor, ancak tekrar ortaya çıkıyor. Hak sınırı diye bir şey yok. İstedığınız kadar ölebilirsiniz. Bir yeri geçmek için sınırsız sayıda deneme şansınız var. Başta absürt gibi görünse de, oynadıkça, "yahu iyi ki hakkım sınırlı değil" cümlesini kullanmaya başlayacaksınız. Çünkü bazı yerleri geçebilmek, joypadi (ya da klavyeyi) yedirtecek kadar zor.

Sonsöz

Yo Frankie!, sadece oyuncu olanlar için, pek fazla tatmin edici olmayabilir. Oyunda hatalar mevcut. Koşarken, gerek klavye gerekse joypad ile, Frankie'yi sola döndüremiyorsunuz. Sağa dönüyor ama

sola dönmeyi deneyince, duruyor. Oyunu oynadığımız üçüncü şahıs kamera perspektifi genelde başarılı da olsa, zaman zaman Frankie ile bir nesnenin arasında sıkışıyor ve yerini bulmakta zorlanıyor.

Shader efektleri açıldığı zaman, oyun motoru, o an ekranda olmayan nesnelere de efekti uygulayarak, sisteme oldukça fazla ve gereksiz yük bindiriyor. Bu da, önünüzde uzanan arazi geniş ise (yani oyun alanının sınırı uzak ise), yavaşlamalara sebep oluyor. Shader efektlerini kapadığınız zaman ise, gitmeniz gereken bazı küçük yerler, görünmesi neredeyse imkansız hale gelebiliyor. Nerede olduklarını bilmiyorsanız, bulmanız neredeyse imkansız oluyor.

Sesler de başarılı sayılmaz. Ancak bu tip bir oyun için, çok da önemli bir eksiklik değil. Yaklaşan bir düşmanın ayak seslerini duyabiliyor ve yaklaştığı istikameti kestirebiliyorsunuz. Bu fazlasıyla yeterli. Ancak müzik felaket. Sürekli tekrarlayan bir ritmden ibaret olan müzik, bir süre sonra kabak tadı veriyor. Oyunun bölümleri, çok güzel bir doğa içinde geçiyor. Bence, sürekli tekrarlayan müzik yerine; kuş sesleri, arada bir esen rüzgar, cır cır böceği sesi gibi doğal ses efektleri olabilirdi. Müzik ise, düşman saldırdığında ya

da geçilmesi zor bir bölgeye gelince başlayıp, bir süre çalıp susan bir müzik olabilirdi. Ama Frankie'nin lava düştüğü zaman, ölürlen attığı çığlıklar, çok başarılı olmuş gerçekten. Bazen sırf duymak için, bilerek Frankie'yi lavlara atıyorum. :-)

Yo Frankie!, oyun olarak tatminkarlıktan biraz uzak olsa da, oyunculardan ziyade, Blender kullanıcılarının ilgisini çekecek türden bir proje. Oyunun sitesi; kaynak kodları, Blender dosyaları, eğitim ve demo filmleri ile oldukça zengin bir içeriğe sahip. Blender'i hiç bilmeyenlerin ve baş-

langıç seviyesinde olanların,"Vay be! Blender böyle şeyler de yapabiliyormuş demek ki!" şeklinde tepki verecekleri bir proje olurken (mesela ben, Blender'da bir dahili oyun motoru olduğunu bilmiyordum); üstatların ise yaratıcılıklarına, Blender bilgilerine, hayal güçlerine, kısacası Blender ile ilgili tüm yeteneklerine, "hodri meydan" diyerek meydan okuyan bir proje olarak ortaya çıkıyor.

Hamit Giray Nart
hamit@pardus-linux.org



- 1. BÖLÜM -

İlk bilgisayarım, bir Commodore 64 idi. 1989 yılında aldırılmıştı. Her ilkokul çocuğu gibi ben de oyunların hevesindeydim ve her ilkokul çocuğu gibi aynı palavrayı atmıştım: "Derslerime yardımcı olur."

Bilgisayar oyunları, pek çok konuda tartışma konusu olsa da, bilişim dünyasındaki yerleri tartışılmaz. Oldukça büyük bir pazar payına sahiptirler. Onlara, bu kadar büyük bir yer ettiren şey nedir peki? Pek çok kişi bu konuda konuşuyor ve konuşmaya da devam edecekler. Ben, televizyonlarda boy gösteren havalı psikologlardan ya da sosyologlardan biri değilim. O yüzden oyunların, sadece benim için anlamını açıklayabilirim.

Bilgisayar oyunları, benim için, gerçek dünyanın keskinliğine karşı sığındığım bir limandır. Tek başıma yaşadığım hayatımda, evden işe gider; sabahtan akşama kadar her gün hayatımı tüketir; akşam yine evime dönerim. Hiçbir sosyal aktivitem, arkadaşım, eşim-dostum yoktur. Kapıyı anahtarla açtığım anda, karanlık evin içinde pusuya yatmış olan yalnızlık, üzerime saldırır. Her defasında, bildiğim halde, pususuna düşerim. Var mıdır ki başka seçeneğim?

The Battle for Wesnoth'a Dalış

Sonra bilgisayarımı açarım. Aslında bir FPS oyuncusu olmama rağmen, fantezi tabanlı oyunları çok severim. Ekranın öbür tarafına geçince, hiçbir anlamı olmayan hayatım, cüce yapıımı kadim bir zırha ya da büyük bir savaşçının ağır kalkanına çarparak parçalanır. Gerçek hayatta kimsesiz bir hiç olan Hamit, artık kadim bir büyücü veya ülkeleri yöneten bir kraldır. Ama heyhat! Her bilgisayarın bir kapatma komutu vardır. O komutu verdiğinizde, elinizde kalan tek şey, yine gerçek hayatınız olur.

Peki, neden böyle olsun? Bir GNU/Linux

oyuncusu olarak, bir oyunda olma şansım var benim. Neden gerçekten oyunun içinde olmayayım ki ben? Elimin altında özgür yazılımın gücü var. Kendi hayal gücümün oluşturacağı bir hikayeyi, neden oyunda yaşamayayım ki? Benim işletim sistemim ve benim oyunlarım buna imkan sağlıyor.

Çok mu fantastik-kurgu oldu dersiniz? Yaratıcı olmaya çalışıyoruz işte, n'apalım?!

GNU/Linux dünyasının özgür yazılım felsefesi altında geliştirilmiş olan Battle For Wesnoth oyununu, sanırım pek duymayan yoktur. Yazımızın bütünlüğü açısından, başlarken, bu oyunu biraz tanıtacağım. Ancak bu, sadece bir tanıtım yazısı olmayacak. "Wesnoth'a Dalış" yazı dizisinde, kendi hikayemizi yazacak; kendi haritalarımızı yapacak; kendi ırklarımızı, kahramanlarımızı, savaşçılarımızı ve hat-



ta kendimizi oluşturacak ve Wesnoth dünyasında adımlarımızı atmaya başlayacağız. Bunun için Wesnoth Harita Düzenleyicisi'ni, Wesnoth veri ve konfigürasyon dosyalarını, WML dilini inceleyeceğiz. Kendi haritalarımızı, karakterlerimizi oluşturduktan sonra oyuna entegre edip; test edeceğiz. Sonrasında ise, kodlama ve entegrasyon işlerimizi oldukça kolaylaştıran "CampGen" isimli bir Python yazılımı olan, Wesnoth Seferberlik Oluşturucu uygulamasını inceleyeceğiz. Haydi Wesnoth'a dalalım. Darısı çıkışımıza... :-)

Klasik Bir Oyuna Klasik Bir Yazı



The Battle for Wesnoth, fantezi-kurgu temalı, tur tabanlı bir strateji oyunudur. Basit kurallarıyla eğlenceli bir strateji oyunu olan Battle For Wesnoth, 2003 senesinde başlayarak, günümüze kadar gelişerek geldi.

Oyun, Pardus PiSi Paketleri deposunda mevcut. sudo pisi it wesnoth komutu ile kurabilirsiniz. Wesnoth, harita düzenleyicisi ile birlikte kurulacaktır.



Oynayış

Oyuna başladığınızda ana menü ekranı açılacak. Buradaki seçeneklerimiz:

1- **Öğretici:** Karakterlerimizi hareket ettirme ve haritayı kullanma konusunda te-

mel kuralları öğreten bir senaryo. İlk oynayacak olanların mutlaka uğraması gereken bir yer.

2- **Seferberlik:** Wesnoth tarihini anlatan birbirine bağlı 14 adet hikayeyi seçip, oynayabileceğimiz bölüm. Acemi, deneyim-

li, uzman olarak üç zorluk kategorisinde sıralanan hikayeler; kendi içlerinde de üç zorluk derecesinde oynanabiliyorlar.

3- Çok Oyunculu: Burası diğer oyunculara karşı oyun oynayabileceğimiz bölüm. Buradan, öncelikle oyuncu adınızı belirleyebilirsiniz. Resmi sunucu bağlantısından, eğer Wesnoth forumlarında bir hesabınız varsa, buradaki kullanıcı adı ve şifrenizi kullanarak, Wesnoth resmi sunucularına giriş yapabilirsiniz. “Sunucuya Bağlan” bölümünde bir IP adresi vererek, ağ üzerinde bir sunucuya bağlanabilir; “Sunucuyu Başlat” ile de kendiniz bir sunucu olabilirsiniz. “Yerel Oyun” ise LAN üzerinden oyun oynayabileceğiniz seçenek. Ayrıca bilgisayara karşı tek senaryoları da buradan oynuyoruz. Bunun için, yerel ağ oyununu seçip, düşman oyuncuyu (oyuncuları) bilgisayar kontrollü yapmak yeterli.

4- Yükle: Kayıtlı oyunlarınıza devam edersiniz. Eğer seçenekler kısmından kapatmazsanız, oyun, her el sonunda otomatik kayıt yapacaktır.

5- Eklentiler: Wesnoth sunucularına; oyuncular ve geliştiriciler tarafından yüklenmiş haritalar, seferberlikler gibi paketleri indirebileceğiniz bölüm.

6- Haritacı: Yazımızın sonraki bölümünde kullanımını anlatacağımız, Wesnoth Harita Düzenleyicisi.

7- Dil: Oyunun dilini seçebileceğiniz bölüm. Mevcut sürümün desteklediği 49 ayrı dilden, birini seçmenize yarar. Ancak PiSi deposundan indireceğiniz Wesnoth, sadece İngilizce ve Türkçe dil desteği yüklü geliyor. Diğer dilleri istiyorsanız, siteden indirmeniz gerekiyor.

8- Tercihler: Oyunun ses, görüntü, genel ve çoklu oyuncu ayarlarını yapabileceğiniz bölüm.

9- Teşekkür: “Yahu kimler yapmış bu oyunu?” şeklinde merak ederseniz, cevabını bulacağınız yer.

10- Çıkış: Var mı bunun da ne olduğunu soracak? Varsa da söylemeyeceğim.

Sağ alttaki zarf simgesine tıklamak suretiyle açılan pencerede ise, seferberlikleri dengelemeye yardımcı olacak istatistiklerin, Wesnoth'a gönderilmesini etkin kılabilir veya iptal edebilirsiniz. Varsayılan olarak, bu özellik kapalıdır.

Oyunda ırkların ayrıldığı, altı sınıf var. Bunlar:

1- Kralcılar: Askerler ve köylülerden oluşan sınıf. Genelde insan ırklarının olduğu seferberlikleri bu grupla oynarsınız.

2- Asiler: Bazı Elfler ile Haydutlardan oluşan sınıf. Saklanma, gizlenme ve pusu bu sınıfın özellikleri arasındadır.

3- Kuzeyliler: Kuzey Elfleri ve Orklardan oluşan bir sınıftır. Güç ve dayanıklılık en belirgin özellikleridir.

4- Kanalga İttifakı: Kanalga Cüceleri ve insanlardan oluşan bir sınıftır.

5- Yarı Ölüler: Kara büyü ustaları Necromancer denilen büyücüler ve onların yaklandıkları yarı ölü birliklerinden oluşan sınıf.

6- Ejderler: Tüm ırklardan ayrı yaşayan, ejder savaşçılarından oluşan bir topluluk.

Ancak bu sınıfları seçmek gibi bir şansınız yok. Oynayacağınız seferberlik hikayelerinde karşılaşacağınız sınıflar.

Oyunun ana kısmını “Seferberlik” bölümü oluşturuyor. Buradan, Wesnoth tarihi ile ilgili senaryolardan herhangi birisini seçip oynayabilirsiniz. Elbette, seviyenize göre olan senaryoları seçmeniz iyi olacaktır.



Hatta oyunu ilk defa oynayacaksanız, var. Menü altında senaryo amaçları, kaydetme ve yükleme seçenekler, tercihler ve çıkış gibi komutlarla; oyunun durumunu izleyebileceğiniz, istatistik tablosu ve

Ekranın en üstündeki çubuğumuzda, birim tablosu gibi seçenekler var. Eylem- "Menü" ve "Eylemler" seçeneklerimiz ler menüsünde, seçili birimler için yapı-

cağınız eylemler ile hamleleri geri al ve tekrarlar gibi komutlar bulunuyor.

Yanındaki renkli bayrak, sıranın hangi oyuncuda olduğunu gösterir. Sıranın olduğu oyuncunun rengini alır. Rakamlardan soldaki, o an kaçınıncı elde olduğunuzu; sağdaki ise maksimum oynanacak olan el sayısını belirtir. Eller bitene kadar dayanma senaryoları dışında, genel olarak, bu eller sona ererse oyunu kaybedersiniz. Yanında ise, sahip olduğunuz altın miktarını görebilirsiniz. Altın miktarı yeni asker almak ve deneyimli eski askerleri geri çağırmak için kullanacağınız kaynaktır.

Yandaki köy göstergesinde ise, soldaki rakam sizin sahip olduğunuz köy miktarını gösterirken; sağdaki rakam ise haritadaki toplam köy miktarını göstermektedir. Köyler sizin altın kaynaklarınızdır. Zorluk seviyesi ile değişmekle beraber, genelde köyler size, el başlarında iki altın getirirler. O yüzden elinizde mümkün olduğunca fazla köy bulundurmanız gerekir. Köyler sadece gelir kaynağı değil; yaralı askerleriniz için birer sağlık merkezi olmakla beraber, savunma için de size iyi bir konum sağlarlar.

Yanındaki hücre, sahip olduğunuz birim miktarını gösterir. Onun yanında ise sıra-

sıyla gider ve gelir hücreleri var. Almış olduğunuz birimler hem ücretlidir, hem de savaş giderine sahiptirler. Bu gider, genelde, birimin seviyesi kadardır. Eğer en sağdaki hücre eksi (-) işaretini alırsa, elinizdeki köy miktarı, ünitelerinizin giderlerini karşılayamıyor demektir. Ünite azaltma ya da köyleri arttırma yolu ile durumu düzeltmezseniz, her el başı, gelir tablosunda yazan eksi miktar kadar altın kaybedersiniz. Altınınız biterse, yeni ünite almak veya deneyimlileri geri çağırmak gibi bir şansınız olmaz.

Onun yanında ise saat bulunur. Sonraki rakamlar ise, imlecin bulunduğu altıgenin (oyun haritası altıgenlere bölünmüş durumdadır) x ve y cinsinden koordinatlarını gösterir. En sağda ise, imlecin bulunduğu altıgenin arazi durumunu gösterir.

Sağ kontrol sütunumuz da, oldukça basit yapılandırılmış. En üstte, haritanın tamamını ve üniteleri renklerine gösteren, ölçeklendirilmiş harita mevcut. Onun altında ise, günün vaktini gösteren bir gösterge mevcut. Bu gösterge boşuna değil. Çünkü bazı üniteler, özelliklerine göre, gece veya gündüz daha iyi savaşıyorlar.

Altında ise, seçili birimin özelliklerini gösteren bir tablo mevcut. Burada ilk olarak,



birimin, Yaşam Puanı (YP) ve Tecrübe Puanı (TP) göstergelerini görüyoruz. Sağdaki rakamlar mevcut durumu gösterirken; soldaki rakamlar ise en üst seviyeyi göstermektedir. YP geriye doğru gider ve darbe aldıkça azalır. Sıfırlanırsa birim ölür. TP ise ileri gider. Girdiği bir savaşta dövüştüğü birimin seviyesi kadar tecrübe kazanır. Eğer düşman birimini öldürürse, öldürdüğü düşman biriminin seviyesinin sekiz katı kadar tecrübe puanı kazanır. Yani 2. seviyedeki bir düşmanı öldüren birim, $2 \times 8 = 16$ tecrübe puanı kazanır. Mevcut puan sağdaki rakamı aştığı zaman, birimiz, seviye geliştirir. Birimler genelde, güçlenerek seviye atlarlar. Ancak bazıları size seçenek sunar. Mesela büyücüler, 2. seviyeye ulaştıklarında, büyücünün hangi birime terfi edeceğini seçenek olarak sunarlar. Bir büyücüyü Ak Büyücü'ye (ışık) ya da Kızıl Büyücü'ye (ateş) terfi ettirebilirsiniz. Ancak unutmayın, bu seçiminiz, ileri ki terfilerini değiştirecektir. Bir de "Noob" tabir edilen yeteneksiz birimler vardır. Bun-

ların terfisi diğer birimlerden farklı olur. Bunlar terfi etmezler, ancak size bazı seçenekler sunarlar. Daha iyi kılıç vuruşu, daha iyi ok vuruşu, daha fazla vuruş, yaşam puanını arttırma gibi seçeneklerden birini seçerek, ünitenin gelişmesini sağlarsınız. Normal olarak seviye atlayan ünitelerin TP yazı rengi kırmızıya dönüştüğü zaman, o birim, artık en üst seviyeye gelmiş demektir. Daha fazla terfi edemez. Noob'ların ise hep kırmızıdır.

Daha altında ise, birimin bilgileri ve özellikleri bulunur. Her birim, ırk ve sınıf olarak çeşitli özelliklere sahip olabileceği gibi; rastgele gelmiş özellikler de taşır. Bunlardan en çok karşınıza çıkacak olanlar:

Sınıfsal Özellikler: Irk veya sınıflar has özelliklerdir.

Adil: İnsan ırkına ait bir özelliktir. Bu birimler, gündüz vakitleri normal saldırılarından %25 daha etkin saldırılar yapabilirler. Ancak gece vakti ise, normalden -%25 etkinlikte saldırı yapar. Yani, 12 saldırı gücü olan adil bir birim, gündüz 15 saldırı gücüyle saldırırken; gece 9 saldırı gücüne düşecektir.

Kaypak: Troll, Ork ve Yarı-Ölü sınıfına ait

bir özelliktir. Adil özelliğinin tam tersidir. Kaypak birimler, gece daha güçlü saldırırlar.

Korkusuz: Elf ve Cüce'lerde olan bu özelliğe sahip üniteler, gece veya gündüzden etkilenmezler.

Dalış: Yarı-Ölü sınıfına ait bir özelliktir. Derin sularda görünmeden ilerleyip, düşmana saldırabilirler.

Mekanik: Kukla, korkuluk veya savaş arabası gibi birimler mekanik birimlerdir. Sömürü, zehirlenme ve veba gibi saldırılardan etkilenmezler.

Becerikli: Elf'lere ait bir özelliktir. Bu özelliğe sahip birimler, menzilli silahlarla saldırdıkları zaman +1 hasar gücüne sahiptirler.

Sağlıklı: Cüce'lere ait bir özelliktir. Sağlıklı birimler, seviye başına +1 YP'ye sahiptirler. Dövüşmedikleri zaman, iki YP iyileşirler. Ayrıca zehirlere de %25 dirençlidirler.

Karabasan: Yarı-Ölü sınıfına ait bir özelliktir. Bu özelliğe sahip birim, saldırmadığı sürece, geceleri görünmez olur.

iyileşir: Trollere ve Entlere (ağaç) ait bir özelliktir. Birim, her el başında 8 YP'ye kadar kendini iyileştirir. Zehirlenmişse, zehri temizler.

Rastgele Özellikler: Her birimin taşıyabileceği özelliklerdir. Irka bağlı değildir.

Zeki: Bu birimler, normalden %20 daha az tecrübe puanı ile terfi ederler. Üst seviyeye normal birimlerden daha çabuk ulaşırlar.

Hızlı: Bu birimlerin, normalden bir fazla hareket puanlarına karşın; %5 daha az yaşam puanları vardır.

Dayanıklı: Bu birimlerin, normalden dört fazla yaşam puanları vardır ve her seviye için bir yaşam puanı fazladan alırlar.

Güçlü: Bu özelliğe sahip birimler, yakın dövüş silahları ile, normalden bir fazla hasar verirler ve normalden bir fazla yaşam puanına sahiptirler.

Pusu: Bu özelliğe sahip birim, ormana girdiği zaman, saldırmadığı sürece görünmez olur.

Gizlenir: Bu birimler, köylere girdikleri zaman, saldırmadıkları sürece görünmez olurlar. Bu birim köydeyken, başka bir birim köye giremez veya saldıramaz.

İlaç verir: İlaç veren birimler, el başında, komşu altıgenlerde bulunan dost birimlerin zehirlerini temizlerler.

iyileştirir +4 veya +8: Bu özelliğe sahip birimler, komşu altıgenlerde bulunan ve yaralı dost birimleri belirtilen rakam kadar iyileştirirler. Ancak zehirlenmeyi temizleyemezler.

Bunlar, en sık karşılaşacağınız özellikler. Bunlardan çok daha fazlasını, oyun içerisinde sağ fare tuşu ile açacağınız menüden "Birim Tanıtımı" maddesini seçerek görebileceğiniz, Wesnoth kılavuzunda bulabilirsiniz. Ayrıca her seferberlik tasarımcısı, kendi oluşturduğu birimlere ayrı özellikler eklemesi suretiyle de, Wesnoth oyunundaki birim özellikleri -karakter yapımı aşamasında göreceğimiz gibi farklılıklar ve fazlalıklar gösterebilir.

Özelliklerin altında ise Hareket Puanı ve saldırı özellikleri bulunur. Hareket puanı, birimin bir el içerisinde gidebileceği altıgen sayıdır. Atlı birimler, yaya birimlere göre daha fazla hareket puanına

sahiptirler. Dolayısıyla daha uzak mesafeye, daha çabuk varırlar.

Saldırı özellikleri ise, birimin, saldırı tiplerini, hasar gücünü ve sayısını gösterir. Silahlarının altında belirtilen rakamlardan soldaki rakam, saldırının hasar gücünü gösterirken; sağdaki rakam saldırı sayısını gösterir. Mesela 6-3 şeklinde bir gösterge, birimin saldırı başına 6 hasar puanı verecek şekilde 3 defa saldırabileceğini gösterir. Saldırıları tip olarak "Yakın" ve "Menzilli" olarak ikiye ayrılır. Kılıç, sopa, mızrak, yumruk gibi silahlarla yapılan saldırılar yakın; ok, balta ya da bıçak fırlatma, büyü gibi saldırılar ise menzilli saldırılardır. Siz bir saldırı yaptığınız zaman, saldırıya maruz kalan düşman birimi de, yakın saldırıya yakın; menzilli saldırıya menzilli olarak karşılık verir. Ancak saldırıya uğrayan birimin, aynı tip saldırısı yoksa, saldırıya karşılık veremez. Mesela menzilli silahı olmayan süvarileri, okçu saldırılarına karşılık veremez. Saldırı özelliklerinden en sık karşılaşacaklarınız:

Destekli: Bu saldırı tipi kullandığınızda, düşman biriminin arkasında sizden bir birim varsa, saldıran üniteniz, saldırdığı düşman birimine iki misli fazla hasar verecektir.

Yükleme: Bu tip bir saldırı, saldırı sırasında, hedefe iki misli zarar verir. Ancak hedef birimin cevabından da iki misli zarar görür.

Cinnet: Saldırı ve savunma durumunda, birimlerden birisi yıkılana ya da 30 ellik dövüş karşılığı kadar, dövüş devam eder.

Sömürür: Yarı-Ölü saldırı tipidir. Bu saldırı, düşmandan götürdüğü gücün yarısı kadar da, saldırıyı yapan birime ekler.

İlk Vuruş: Bu tip saldırı özelliğine sahip birim, saldırı ve savunma durumundayken, ilk vuruşu yapar. İki ilk vuruş karşılıksa da savunan önce vurur.

Sihirli: Büyüyle desteklenen sihirli saldırılar, ne olursa olsun, her zaman %70 isabet şansına sahiptirler.

Nişancı: Saldırı durumundayken bu tip saldırının, hep %60 isabet şansı vardır.

Gizemli: Yarı-Ölülere normalden %50 daha fazla hasar verir.

Veba: Bu da bir Yarı-Ölü saldırı tipidir. Zombi ve Umacılarda olan bu saldırı tipi ile öldürülen düşman birimi, zombiye dönüşerek, kendisini öldüren birimin tara-

fına geçer.

Zehirler: Bu tip bir saldırıya maruz kalan birim zehirlenir. Zehirlenen birim zehirden ölmez ama 1 YP'si kalana kadar, her el 8 YP kaybeder. İlaç veren bir birim tarafından ya da bir köye giderek ancak zehirden kurtulabilir.

Sağ sütunumuzun en altında "Eli Bitir" butonu ile de eli bitirerek, sırayı diğer oyuncuya veririz.

Oyunun savaş ve ekonomi kuralları ile kontrolü, benim gibi "Ben anlamam kardeş, stratejiden mstratejiden. Yüklenir silahlarımı, bodoslama ön kapıdan dalayım." diyerek bu tip şeylerle uğraşmaktan hiç hoşlanmayan FPS oyuncularını için bile, oldukça basit.

ANCAAAAK...

Oynanış ise bu kadar basit değil. Oyuna strateji öğeleri yanında, fantastik öğeler de etki ediyor. Mesela fantezi-kurgu bilgisi olan birisi, Elflerin ormanlarda; cücelerin dağlarda ustaca dövüştüklerini bilir. Bu tip özelliklerin tümü Battle For Wesnoth oyununda mevcut. Mesela mızrakçı birimlerin, atlı birimlere karşı saldırı

üstünlüğü vardır. Bir düşman birimine saldırı yaptığınızda, en güçlü saldırı yerine, düşmanın karşılık veremeyeceği saldırıyı seçmek. Önemli köy ve geçiş noktalarını belirleyip, oraları tutmak. Tüm bunların yanında, bir de fantastik-kurgu kurallarına dikkat etmeniz gerekiyor. Mesela bir Cüce'yi ormana ya da bir Elf'i mağaraya sokmak demek; onu ölüme göndermek demektir.

Oyunda, ilah gibi görünmeyen bir güç olarak birimlerinizi yönetmiyorsunuz. Hikayenin başkahramanı olarak, bizzat, haritada bir karaktersiniz. Durumunuz, avantajınız, gücünüz ne olursa olsun; bir şekilde ölürseniz, oyunu kaybedersiniz. Ayrıca hikayenin devamında rolü olacak karakterler de, haritada sizinle birlikte olabilirler. Onların da ölümü, oyunu kaybetmeniz demek olacaktır.

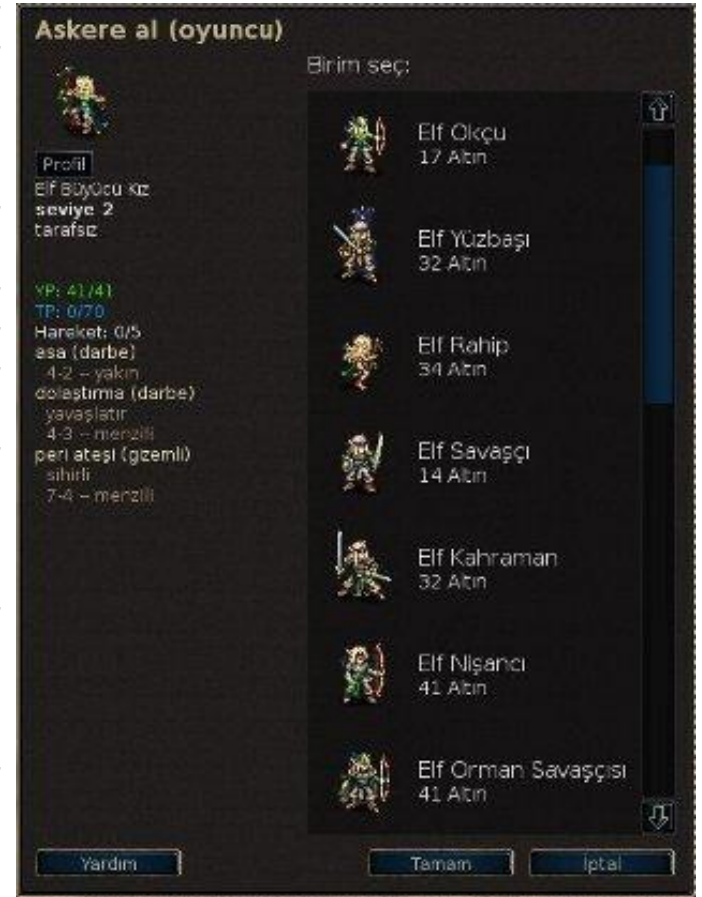
Tüm birimlerin solunda, birimin gücünü gösteren bir enerji barı vardır. Bu bar, birim darbe aldıkça azalır. Ancak, düşman birliklerde olmamakla birlikte, sizin veya dost birimlerin enerji barlarının üzerlerinde bir küre görürsünüz. Bu küreler, birimlerin hareket durumunu belirtirler. Sizin kontrolünüzde olan ve hiç hareket etmemiş birimlerin küreleri yeşil renktedir. Sizin kontrolünüzde olup da hareket etmiş,

ancak daha hareket edebilecek veya saldırabilecek kadar hareket puanı kalmış birimlerin küreleri sarıdır. Sizin kontrolünüzde olup da, hareket puanı tamamen sıfırlanmış birimlerin küreleri ise kırmızı renktedir. Oyun esnasında "n" tuşu ile, hareket puanı kalmış ünitelerinizi görebilirsiniz. Birimin küresi mavi ise, o birim, kontrolünüzde olmayan bir dost birim demektir. Bazı birimlerin kürelerinin hemen altlarının ise taçlandırılmış olduğunu görürsünüz. Bu birimler, hikayenin kahramanı olan birimlerdir. Ne pahasına olursa olsun, onları korumanız gerekir.

Oyunda, diğer strateji oyunlarından alıştığımız, bina yapmak gibi bir fonksiyon mevcut değil. Asker almanın tek yolu, sizin (hikayenin başkahramanı olan karakterin), bir iç kale ya da bir kamp merkezine gelerek, boş komşu kale altıgenleri üzerine sağ tıklayarak açacağınız menüden "Askere Al" maddesini seçmek.

Açılan menüden, alabileceğimiz askerleri seçerek onları alabiliriz. Askere alma bedeli birimin altında yazan miktar kadardır. Yine aynı şekilde "Çağır" diyerek de eski ve deneyimli askerlerinizi çağırabilirsiniz. Ancak çağırma bedeli sabit olup, 20 altındır.

Haritada, günün saatinden başka, dikkat edeceğimiz bir başka husus ise, zemindir. Bir birimi seçtiğinizde, o birimin hareket alanı içerisinde bir yere imleci getirdiğiniz zaman, imlecin üzerinde bir yüzde (%) yazdığını görürsünüz. Bu yüzde, o birimin, o noktadaki etkinlik oranıdır (vuruş isabet şansı). Bu oran ne kadar yüksekse, birim, o kadar etkili savaşır. Mesela bu oran %70 ise birimin, üç vuruşundan en az



ikisi isabetli olacak demektir. Ancak bu oran %20 ise, beş vuruştan en az dört vuruş boşa gidecek demektir. Bu oran, saldırı ya da savunma durumlarının her ikisinde de geçerlidir. Hele biriminizin bir ya da iki gibi saldırı sayısı az ise, bu birim, en az iki el etkisiz kalacak demektir. Ancak köyler ve kalelerde bu oran %60 olur.

Oyunda yine pek çok küçük ayrıntılar var. Bazıları, strateji oyuncularının yaptığı stratejilere göre değişmekle beraber, uyulması gereken genel kurallar var. Öncelikle, mümkün olduğu kadar az asker kaybetmeye çalışmalısınız. Galibiyet için, gerçekten çok dikkat etmeniz gereken bir husus. Yaralı askerlerinizi ön saflardan kaçırmalısınız. Gücü, yarının altına düşmüş bir birim, düşmanı durduramaz. Öleceğini bile bile adamlarınız saldırtmayın. Çünkü altının azaldığı ya da karakterinizin kalelerden uzak olduğu durumlarda askersiz kalmak, gerçekten felaket oluyor. Ayrıca deneyim kazanan birimler, bir sonraki haritalarda geri çağrılıyorlar. Ölen deneyimli bir birimse, beraberinde deneyimlerini de götürüyor.

Araziyi oldukça dikkatli kullanın. Dar geçitleri ve köprüleri tutmak için, o arazi tipinde etkili olabilecek birimleri seçin. Mesela bir nehir kıyısındaki bir ormanda bu-

lunan birkaç Elf okçu, arkalarında bir de iyileştirici bir birim olursa, koca bir birliği nehirde telef ederler. Aynı şekilde bir mağara girişi ya da bir dağ geçidini tutmuş birkaç cüce savaşçısı, geçilmesi çok zor bir duvar oluşturur.

Savunma hatlarını doğru oluşturmak çok önemlidir. Hem az kayıp vermenizi hem de altın biriktirmek için zaman kazanmanızı sağlar. Savunma hattındaki bir birime, düşman en fazla iki birim ile saldırabilir. Bu hattı koruyarak ilerlemek, en sağlam ilerleme yöntemidir. Hem yaralı askerlerinizi arka saflara almayı kolaylaştırır, hem de safin arkasında bulunan iyileştirici birimlerin, düşmana karşı korunmasını ve işlerini daha etkin yapmalarını sağlar.

Sonsöz

Battle For Wesnoth, oldukça güzel bir klasik. Wesnoth'a Dalış dizimizin ikinci bölümünde, Wesnoth verilerini, harita yapısını ve harita kodlarını inceleyeceğiz; Haritacıyı tanıyacak, yeni harita paletleri oluşturup, bunları haritamıza yerleştirmeyi göreceğiz. Yani ikinci bölümümüzü, Wesnoth haritalarına ayıracağız. Önümüzdeki ay görüşmek üzere...

GNU/Linux üzerinde C uygulamaları geliştirmeye başlamadan önce, bilmeyenler için dilin bazı özelliklerini ve kurallarını açıklayacağım. Bilgisayar, kullanılmaya başlandığında makine diliyle programlama yapıyordu. Makine dili denilen şey, 0 ve 1'den oluşur. Aslında buradaki 0 ve 1 rakamları, elektrik sinyallerini ifade etmektedir. Temelde, bilgisayarın anladığı tek şey bu elektrik sinyalleridir. İşte makine dilini oluşturan bu 0 ve 1'leri kullanarak program yazmak çok zor olduğundan ve yazılan programlar fazla yer kapladığından şöyle bir yol izlenmiştir: Önce, makine dilinde bir derleyici yazılır. Bu derleyici, yeni oluşturulan dilin kodlarını makine diline çevirecek bir programdır. Yeni oluşturulan dile, "Assembly" adı verilmiştir. Bu dil, İngilizce sözcüklerin baş harflerinden meydana gelen komutlar dizisinden oluşuyordu. Bir noktadan sonra, bu dilde de uygulama geliştirmek zorlaşmaya başlamış ve yeni bir dil daha oluşturulmuştur. Bu dilin adı, B dilidir. Bunun nedeni; Assembly dilinin, A dili olarak ifade edilmesidir. Bundan sonra bir çok programlama dili oluşturulmuştur. Bunlar arasında en köklü olanlardan biri de C dilidir. Neyse, artık eskiyi bir kenara bırakıp C dilinin yapısına bir göz atalım. :)

C Dilinin Yapısı

Fonksiyonlar:

Bu dilde her şey fonksiyonlar aracılığı ile gerçekleştirilir. Bir C programı, ana fonksiyon ile başlar ve bu fonksiyon ile sonlanır. Aşağıdaki hiçbir işlem yapmayan C programımızı inceleyelim:

```
int main(void)
{
    /* Bu program hiçbir işlem yapmamaktadır. */
    return 0;
}
```

C dilinde, her şeyin fonksiyonlar aracılığı ile gerçekleştiğini söylemiştik. Bir fonksiyon, 3 özelliğe sahiptir; fonksiyon ismi, geri dönüş değeri ve parametre değişken(ler)i.

Fonksiyon ismi, fonksiyonu çağırmak için gerekli olan isimdir. Örneğin, yukarıdaki "main" fonksiyonunu çağırmak için "main()" biçiminde belirtmek gerekir. "main", fonksiyonun ismidir. "main" 'den önce gelen "int" ise, fonksiyonun geri dönüş değerinin türüdür. "int", C dilinde tamsayı türünü ifade etmek için kullanılan bir anahtar sözcüktür. "main" fonksiyonunun yanında yer alan parantezlerin içine, fonksiyonun parametre değişkenleri yazılır. Burada "void" yazarak, parametre değişkeninin olmadığını belirtiyoruz. Bunları izleyen küme parantezleri ise, fonksiyon bloğudur. Fonksiyonun yapacağı tüm işlemler burada kodlanır. /* */ karakterleri arasına yazılanlar, derleyici tarafından derlenmezler. Sadece açıklama yapmak için kullanılır.

Değişkenler: Değişkenler, içlerinde bilgi barındıran, dolayısı ile bellekte yer kaplayan nesnelere dir. Değişken tanımlama işlemi, aşağıdaki gibi gerçekleştirilir:

```
"Değişken Türü" "Değişken İsmi";

int sayi;
```

"int" sözcüğünün, tamsayı türünü ifade ettiğini belirtmiştik. Yukarıda, "sayi" isminde, tamsayı türünden bir değişken oluşturulmuştur. Bu değişkene değerler atayabilir ve daha sonra bu değerleri kullanabiliriz.

Kütüphane Dosyaları: C dilinin, standart olarak bulunan fonksiyonlarını barındırır. Örneğin; ekrana bir şeyler yazdırmak için "printf()" fonksiyonu kullanılır. Bu fonksiyon, "stdio.h" adlı bir kütüphane dosyasında tanımlanmıştır.

Bir örnek verip açıklayarak, asıl konuya geçeceğim:

```
/*ornek.c*/
/*Bu program girilen 2 sayıyı toplayıp ekrana yazdırır.*/

int topla(int a, int b)
{
return a + b;
}

void main(void)
{
int sayi1,sayi2,toplam;
clrscr();
scanf("%d%d",&sayi1,&sayi2);
toplam = topla(sayi1,sayi2);
printf("Sayılar ve toplamları: \n%d+%d=%d",sayi1,sayi2,toplam);
getch();
}
```

Yukarıdaki kod derlenip çalıştırıldığında program, her C programında olduğu gibi, "main()" fonksiyonundan başlar. Bu fonksiyonun bloğuna göz atarsak;

* 1. satırda, "sayi1", "sayi2" ve "toplam" adında üç tamsayı türünden değişken tanımlanmıştır. ";" karakteri, sonlandırma karakteridir. Derleyici bu karakteri gördüğünde, deyinmin bittiğini anlar. Yani bir sonraki komutu çalıştırır.

* 2.satırda, "clrscr();" ifadesi görülüyor. Bu ifadede, "clrscr()" adlı fonksiyon çağırılmıştır. Bu fonksiyon, C'nin kütüphane dosyasında tanımlanmıştır. "clrscr()" fonksiyonu, ekratemizlemek için kullanılır. "(" işleci, fonksiyon çağırma işlecidir.

* 3. satıra gelindiğinde, "scanf()" fonksiyonunu görüyoruz. Bu fonksiyon, klavyeden bilgi girişi yapmamızı sağlar. Fonksiyonu çağırırken, fonksiyon çağırma operatörü içine yazdığımız veriler, argüman olarak nitelendirilir. "scanf()" fonksiyonu, klavyeden girilecek bilgilerin yerleştirileceği nesnelerin türlerini ve adreslerini parametre olarak alır. Biz burada, "sayi1" ve "sayi2" adlı değişkenlerin önüne "&" işlecini katarak; bu değişkenlerin bellekteki adreslerini, "scanf" fonksiyonuna göndermiş olduk. Yani "&" işleci, sağ tarafındaki nesnenin adresini üretir. Bunlardan önce, "%d%d" diye bir argüman göndermişiz. "%d" ifadesi, "sayi1" adlı değişkenin türünün "int" yani tamsayı olduğunu belirtir. Diğer "%d" ise, aynı şekilde ikinci değişkenimizin türünü gösterir. Yani, "sayi2" de "int" türündendir diyoruz. İlk sayı girilir; enter tuşuna bastığımızda program, ikinci sayıyı ister. Tekrar enter tuşuna bastığımızda, ilk sayı "sayi1" değişkenine; ikinci sayı ise, "sayi2" değişkenine yerleştirilir.

* 4.satırda, önce "topla()" adlı fonksiyon çağrılır. Bu fonksiyona, "sayi1" ve "sayi2" adlı değişkenler, argüman olarak gönderilir. Bu gönderimde şu olay gerçekleşir. Birinci argüman, fonksiyonun tanımlama işleminde belirtilen birinci parametre değişkenine aktarılır. İkinci argüman, ikinci parametre değişkenine aktarılır ve bu işlem bu şekilde sırasıyla devam eder. "topla" fonksiyonu çağrıldığında, programımız, bu fonksiyona atlar; yani dallanır. "topla()" fonksiyonun bloğuna baktarsak; "return a + b" işlemi görürüz. Fonksiyonların, geri dönüş değerlerinin olduğunu söylemiştik. C dilinde bir veriden bahsediliyorsa, bunun türü çok önemlidir. Bu dilde kullanılan tüm verilerin, türlerinin de belirtilmesi gerekir. Fonksiyonların geri dönüş değeri dediğimizde de, bunun mutlaka bir türünün olduğunu da anlamalıyız. Bu geri dönüş değerinin türünü, ilgili fonksiyonu tanımlarken belirttiğimizi anlatmıştık. "topla()" adlı fonksiyonun geri dönüş değeri, "int" yani tamsayı türüdür. Şimdi gelelim "return" anahtar sözcüğüne. Bunu, ilgili satırı açıklayarak anlatacağım. "return a + b" diyerek programa, "*a değişkeni ile b değişkeni toplayıp, "topla" adlı fonksiyona geri döndür*", diyoruz ve bu işlemden sonra "topla" adlı fonksiyon sonlanıp, "main" fonksiyonda, kalınan yerden devam ediliyor. Bu işlemlerden sonra, "main" fonksiyonun 4. satırı şu şekilde bürünür: *toplam = "topla() fonksiyonun geri döndürdüğü değer"*. Bu değer de "toplam" adlı değişkene atanır.

* 5.satırda, "printf()" fonksiyonu ile ekrana bir şeyler yazdırıyoruz. Biz burada "sayi1", "sayi2" ve "toplam" adlı değişkenlerin içeriğini yazdırdık.

* 6.satırda, "getch()" fonksiyonu ile klavyeden bir karakter girilmesini bekliyoruz. Bir C programı, "main" fonksiyonundan başlar ve bu fonksiyonun bloğunu ifade eden küme

parantezinin kapanması ile biter. O halde, "getch()" fonksiyonunu çağırmadığımızı düşünürsek, "}" karakteri ile programımız sonlanacak ve ondan önceki işlemleri göremeyeceğiz. Ama "getch()" fonksiyonunu çağırdığımızda, klavyeden bir karakter girilmesi beklenecek ve biz bu anda yaptığımız işlemleri görebileceğiz.

C dili ile için kaynak arıyorsanız; "*Kaan Aslan - A'dan Z'ye C Kılavuzu*" adlı kitabı tavsiye ederim.

GNU/Linux Üzerinde C İle Uygulama Geliştirmeye Giriş

Aslında bu konuda ben de henüz yeniyim. Araştırmaların sonucunda öğrendiğim bilgileri sizlerle paylaşmayı istedim. İlk olarak, Linux üzerinde C programları yazmak için gerekli editörümü belirtelim. Bu iş için geliştirilmiş, kaliteli programlar mevcut. Fakat şimdilik, Linux'un herhangi bir metin editörü ile bu işi gerçekleştirebiliriz. Ben Kate adlı programı kullanıyorum. Kullandığım Linux dağıtımı ise Pardus. Kate programı, Pardus'u yüklediğimde kurulu değildi. Paket yöneticisini çalıştırıp, arama kutusuna "kate" yazarak, bu programın da içinde bulunduğu paketi yükleyebilirsiniz. Linux üzerinde yazdığımız C programlarını derlemek için ise, "GCC" adlı derleyiciyi kullanacağız. GCC derleyicisini yüklemek için, komut satırını açtıktan sonra

```
opsi it gcc
```

yazıp, enter tuşuna bastığımızda yükleme başlayacaktır. Araç gereçlerimizi hazır ettikten sonra, artık GNU/Linux üzerinde C programlarımızı yazmaya başlayabiliriz.

Aşağıdaki programımızı, Kate adlı editörde yazıp, "ornek.c" olarak kaydedelim. Bir sonraki dersimizde görüşmek üzere, herkese sağlık ve mutluluk dolu günler dilerim.

```
/*ornek.c*/
#include <stdio.h>
main()
{
    printf("Linux üzerinde yazdığımız ilk C programımız.");
}
```

"ornek.c" adlı programımızı derlemek için önce konsolumuzu açıp, aşağıdaki komutu girelim:

```
gcc ornek.c
```

Not: Burada "ornek.c" adlı program masaüstündedir ve gcc komutu uygulanmadan önce masaüstü dizinine geçilmiştir.

Bu komutu girdikten sonra, kaynak kodumuzun bulunduğu dizinde, "a.out" adlı bir program oluşturulacaktır. Bu programı da, konsoldan ./a.out yazarak çalıştırdığımızda, yazdığımız C programını çalıştırıp görmüş olacağız.

GCC derleyicisi ile derleme işlemi yapmak için daha bir çok parametre mevcut. Bunlardan birisi, aşağıda görülmektedir:

```
gcc ornek.c -o yeni
```

Yukarıda ki kodu uyguladığımızda "ornek.c" adlı uygulamamız, a.out yerine, "yeni" ismi ile oluşturulacaktır.

Pardus'ta Django

Django'yla geliştirdiğimiz Web uygulamalarında veritabanı kullanmaya başlayacağız. Django'da veritabanı teorisini ve SQL bilmenize gerek yoktur. Ancak biraz veritabanıyla ilgili bir şeyler bilmeniz iyidir.

viewlarda veritabanı sorgusu döndürebiliriz. Herhangi bir Python kütüphanesini kullanarak sorgu döndürürüz ve sonuçlarla bir şeyler yaparız. Bir MySQL veritabanına bağlanmak için MySQLdb kütüphanesini kullanarak, kayıtları alabiliriz ve bir Web sayfasının HTML kodlarına koyabiliriz.

```
from django.shortcuts import render_to_response
import MySQLdb
```

```
def book_list(request):
    db = MySQLdb.connect(user='me', db='mydb',
                        passwd='secret', host='localhost')
    cursor = db.cursor()
    cursor.execute('SELECT name FROM books ORDER BY name')
    names = [row[0] for row in cursor.fetchall()]
    db.close()
    return render_to_response('book_list.html',
                            {'names': names})
```

Bu yaklaşım doğrudur ama kullanışlı değildir:

- Veritabanı bağlantı parametrelerini sabit kodluyoruz.
- Klişe kodlar yazmalıyız.
- MySQL kullanmak zorunda bırakıyor. MySQL'i Oracle satın aldı. MySQL'i paralı kullanmak zorunda kalabiliriz. PostgreSQL'e

çevirmek isteyebiliriz. Bunun gibi başka veritabanı dillerini kullanmak isteyebiliriz.

Django'nun veritabanı API'siyle bu kodları şöyle yazabiliriz:

```
from django.shortcuts import render_to_response
from mysite.books.models import Book

def book_list(request):
    books = Book.objects.order_by('name')
    return render_to_response('book_list.html',
                            {'books': books})
```

Veritabanı konfigürasyonu yapmamız gerekiyor. Django'ya hangi veritabanı sunucusunu kullanacağını ve nasıl bağlanacağını söylememiz gerekiyor. settings.py dosyasını açalım. Şu kodları görmelisiniz:

```
DATABASE_ENGINE = ''
DATABASE_NAME = ''
DATABASE_USER = ''
DATABASE_PASSWORD = ''
DATABASE_HOST = ''
DATABASE_PORT = ''
```

Kodların özeti şöyle:

DATABASE_ENGINE: Hangi veritabanı motorunun kullanılacağını söyler.

Veritabanı kullanıyorsanız **DATABASE_ENGINE**'de bir tane metin belirtmelisiniz. Veritabanı motoru için kullanacağınız metin ve

gerekli paketler şöyle:

PostgreSQL: postgresql, psycopg version 1.x

PostgreSQL: postgresql_psycopg2, psycopg version 2.x

MySQL: mysql, MySQLdb

SQLite: sqlite3, Python 2.5 ve üstü kullanıyorsanız herhangi bir paket gerektirmez.

Oracle: oracle, cx_oracle

DATABASE_NAME: Veritabanının ismini söyler.

DATABASE_USER: Veritabanına bağlanmak için hangi kullanıcı adının kullanılacağını söyler.

DATABASE_PASSWORD: Veritabanına bağlanırken hangi şifrenin kullanılacağını söyler.

DATABASE_HOST: Veritabanı bağlanırken kullanılacak yollar.

Veritabanı kullanarak bir uygulama yazabiliriz. site dizini altında books adında bir uygulama yazalım. Şu komutu verelim:

```
python manage.py startapp books
```

site dizini altında books dizini açıldı. models.py ve views.py dosyalarına bakalım. models.py veritabanı dosyamız ve views.py de HTML dosyalarını gösteren dosyamız. Basit bir veritabanıyla uygulamamızı geliştirelim. Verilerimiz book, author, publisher olacak.

- authorun name, surname ve e-maili olacak.

- publisherın, name, address, city, state_province, country ve websitei olacak.

- bookun, title ve publication_datei olacak.

models.py dosyası şu şekilde olacak:

```
from django.db import models

class Publisher(models.Model):
    name = models.CharField(max_length=30)
    address = models.CharField(max_length=50)
    city = models.CharField(max_length=60)
    state_province = models.CharField(max_length=30)
    country = models.CharField(max_length=50)
    website = models.URLField()

class Author(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=40)
    email = models.EmailField()

class Book(models.Model):
    title = models.CharField(max_length=100)
    authors = models.ManyToManyField(Author)
    publisher = models.ForeignKey(Publisher)
    publication_date = models.DateField()
```

Veritabanımızı yükleyelim. settings.py dosyamızda şöyle bir kod olmalı:

```
INSTALLED_APPS = (
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.sites',
)
```

Şu kodu ekleyerek veritabanını yükleyebiliriz:

```
INSTALLED_APPS = (
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.sites',
'site.books',
)
```

Veritabanını doğrulamalıyız. Şöyle bir komut vererek doğrulayabilirsiniz:

```
python manage.py validate
```

Tabloları üretmek için de şöyle bir komut verebilirsiniz:

```
python manage.py sqlall books
```

manage.py startapp komutunu vererek şöyle bir çıktı alıyosak sorun yok:

```
BEGIN;
CREATE TABLE "books_publisher" (
  "id" serial NOT NULL PRIMARY KEY,
```

```
  "name" varchar(30) NOT NULL,
  "address" varchar(50) NOT NULL,
  "city" varchar(60) NOT NULL,
  "state_province" varchar(30) NOT NULL,
  "country" varchar(50) NOT NULL,
  "website" varchar(200) NOT NULL
)
;
CREATE TABLE "books_author" (
  "id" serial NOT NULL PRIMARY KEY,
  "first_name" varchar(30) NOT NULL,
  "last_name" varchar(40) NOT NULL,
  "email" varchar(75) NOT NULL
)
;
CREATE TABLE "books_book" (
  "id" serial NOT NULL PRIMARY KEY,
  "title" varchar(100) NOT NULL,
  "publisher_id" integer NOT NULL REFERENCES
"books_publisher" ("id")
DEFERRABLE INITIALLY DEFERRED,
  "publication_date" date NOT NULL
)
;
CREATE TABLE "books_book_authors" (
  "id" serial NOT NULL PRIMARY KEY,
  "book_id" integer NOT NULL REFERENCES "books_book" ("id")
DEFERRABLE
INITIALLY DEFERRED,
  "author_id" integer NOT NULL REFERENCES "books_author"
("id") DEFERRABLE
INITIALLY DEFERRED,
```

```
UNIQUE ("book_id", "author_id")
)
;
CREATE INDEX "books_book_publisher_id" ON "books_book"
("publisher_id");
COMMIT;
```

SQL'i veritabanına işlemek için şöyle bir komut verelim:

```
python manage.py syncdb
```

Şöyle bir çıktı alıyorsanız sorun yok:

```
Creating table books_publisher
Creating table books_author
Creating table books_book
Installing index for books.Book model
```

Django'da veritabanı kullanımına temel bir şekilde ele aldık. Temel uygulamalar geliştirerek, diğer dillerle çok kod yazarak geliştirdiğiniz uygulamaları Django'yla az kodla geliştirebileceksiniz.

Onur TUNA
onur@pardus-linux.org

Kaynak:
<http://www.djangobook.com>

Bu sayımızda Pardus'ta Django blog uygulaması hazırlamayı ve html dosyaları üzerinde ekrana bastırmayı öğreneceğiz. Daha öncede katagorize sistemini öğrenmiştik aşağı yukarı aynı şekilde `models.py`, `views.py`, `settings.py` ve `urls.py` ayarlarını yapıp template dizini içinde html dosyalarımızı yazacağız.

Öncelikle `django-admin.py` dosyamız yok ise `/usr/lib/python-django/bin/` klasöründen bulabilirsiniz. Kullandığınız sürüme göre klasör yolu değişik olabilir arattırma yaparak bulmanız daha çok hızlı olacaktır. (Python 2.5 için `/usr/lib/python2.5/site-packages`) Daha önceki dersten hatırlarsanız bu dosyamız bize yeni bir proje oluşturmamızda yardımcı olacaktır.

`/home` klasöründe kullanıcı adımız altında `django` adlı bir klasör oluşturuyoruz. Komut satırımızı açıp `django/` klasörüne geçiyoruz. `django-admin.py` dosyasını buraya yapıştırıyoruz. Komut satırında `django-admin startproject blogproje` yazarak `blogproje` adında bir proje klasörü oluşturuyoruz.

```
muslu@muslu-laptop:~$ cd django/
muslu@muslu-laptop:~/django$ django-admin startproject blogproje
muslu@muslu-laptop:~/django$ cd blogproje/
muslu@muslu-laptop:~/django/blogproje$ ls
__init__.py  manage.py  settings.py  urls.py
muslu@muslu-laptop:~/django/blogproje$
```

Komut satırında `cd blogproje` yazarak proje klasörümüze giriyoruz. Otomatik olarak oluşturulmuş 4 tane dosya göreceksiniz. Bu dosyaların görevleri şu şekilde:

`__init__.py`: Python'da bu dizinin bir Python paketi olarak değerlendirilmesi gerektiğini söyler.

`manage.py`: Django projesi ile çeşitli yollarla iletişim kurabileceğiniz bir komut satırı aracı. Bu dosya ile veritabanında bilgileri ekleme ve serveri çalıştırmak için kullanacağız. Ayrıca SQL komutları çalıştırmayı ve hataları kontrol etmeye de yarar.

`settings.py`: Django projemiz için ayarlar dosyası. Bir çok ayarı bu dosyadan yapacağız. (Dil, tarih, saat, uygulamalar, veritabanı vs..)

`urls.py`: Django projemiz için URL tanımlamaları. Gelen her url'ye bir görev verebiliriz.

İlk olarak `settings.py` dosyasında sistem ve veritabanı ayarlarını yaparız.

En üst satıra `# -*- coding: utf-8 -*-` kodunu yerleştiriyoruz. Bu kod Türkçeyi de içinde barındıran utf-8 karakter kodlamasına izin verecek.

```
# -*- coding: utf-8 -*-
import os,sys
# os ve sys yi import ederek işletim sistemi ve sistem
#parametrelerini yüklüyoruz
klasoryolu = os.path.dirname(__file__)
# klasoryolu adında bir liste oluşturarak, dosyalarımızın
#olduğu klasör yolunu tayin ediyoruz.
```

Bu ayarlamaları veritabanı sisteminize göre yapınız.

```

DATABASE_ENGINE = 'sqlite3'
# Kullanılacak veritabanı 'postgresql_psycopg2', 'postgresql',
#'mysql', 'sqlite3' or 'oracle'
DATABASE_NAME = '/home/muslu/django/blogproje/deneme.db'
# Veritabanı adı veya SQLite kullanıyorsanız veritabanının tam
#yolu.
DATABASE_USER = ''
# Kullanıcı adı (SQLite için kullanılmıyor)
DATABASE_PASSWORD = ''
# Şifre (SQLite için kullanılmıyor)
DATABASE_HOST = ''
# Host Adınız Aynı fiziksel host için boş bırakabilirsiniz.
#(SQLite için kullanılmıyor)
DATABASE_PORT = ''
# Kullanacağımız port (SQLite için kullanılmıyor)
TIME_ZONE = 'Europe/Istanbul'
# Zaman ayarı
LANGUAGE_CODE = 'tr-TR'
# Kullanılacak dil ar-AR fr-FR en-EN
MEDIA_ROOT = os.path.join(klasoryolu, 'static/')
# media dosyalarımızın yani ses, resim, video dosyalarının
#bulunacağı klasör yolu
MEDIA_URL = '/static/'
# media dosyaları için url yolu
ADMIN_MEDIA_PREFIX = '/media/'
# admin media dosyalarının yolu
TEMPLATE_DIRS = (os.path.join(klasoryolu, 'templates'))
# proje klasörü altında templates adında bir klasör oluşturup
#html dosyalarımızın burada saklanacağını belirtiyoruz.

```

settings.py dosyasının en altında kullanılacak uygulamaların listesi var. Şimdilik sadece admini yüklüyoruz.

```

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
)

```

Artık urls.py dosyasına geçebiliriz. urls.py dosyası da adı üstünde gelecek olan url linklerini yönlendireceğimiz dosya. KWrite veya herhangi bir editörle urls.py dosyasını açıyoruz.

```

from django.conf.urls.defaults import *
from django.contrib import admin
admin.autodiscover()
urlpatterns = patterns('',
    (r'^$', 'django.views.generic.simple.direct_to_template',
     {'template': 'index.htm'}, 'index'),
    (r'^admin/', include(admin.site.urls)),
)

```

Bu düzenlemeden sonra; ilk olarak gelecek sayfayı index.htm dosyasına /admin olarak gelecek url'yi de admin paneline yönlendirecektir.

Şimdi komut sisteminde python manage.py syncdb diyerek settings.py'deki verileri veritabanına yüklüyoruz.

Tabloları oluşturduktan sonra yönetici olarak eklemek istediğimiz bir kullanıcı olup olmadığını soracak yes yazarak onay veriyoruz. Yöneticiye ait bir kullanıcı adı atıyoruz ve e-posta adresimiz ile şifremizi yazıyoruz.

```
muslu@muslu-laptop:~/django/blogproje$ python manage.py syncdb
Creating table auth_permission
Creating table auth_group
Creating table auth_user
Creating table auth_message
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table django_admin_log
```

You just installed Django's auth system, which means you don't have any superusers defined.

Would you like to create one now? (yes/no): yes

Username (Leave blank to use 'muslu'):

E-mail address: musluyuksektepe@gmail.com

Password:

Password (again):

Superuser created successfully.

Installing index for auth.Permission model

Installing index for auth.Message model

Installing index for admin.LogEntry model

muslu@muslu-laptop:~/django/blogproje\$ █

Hatırlarsanız `urls.py` dosyamızda ilk gelen url'ye `index.htm` adlı bir dosya tanımlamıştık. Direk test etmek isterseniz `TemplateDoesNotExist` at / hatası alırsınız.

Proje klasörü altında `static` ve `templates` adında klasör oluşturuyoruz. `static` klasörü `media` dosyalarımız, `templates` de `html` dosyalarımız içindi. `templates` içine istediğiniz gibi bir `index.htm` dosyası koyuyoruz. Örnek olarak:

```
<html>
<head>
<title>Yazki.com</title>
</head>
<body>
<a href="/">Anasayfa</a> |
<a href="/blog/">Blog</a> |
<a href="/admin/">Yönetim Paneli</a>
```

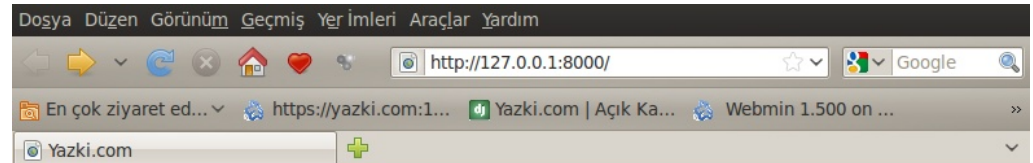
```
{%block main%}{%endblock main%} <<<----- Sayfanın altını
parselliyoruz. Birazdan göreceğiz.
</body>
```

Artık oluşturduğumuz projemizi test edebiliriz. Bunun için de yine komut satırında `python manage.py runserver` yazıyoruz.

```
muslu@muslu-laptop:~/django/blogproje$ python manage.py runserver
Validating models...
0 errors found
```

```
Django version 1.1.1, using settings 'blogproje.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Herhangi bir hata yapmadıysak bu ekranın gelmesi gerekiyor. İsterseniz tarayıcınızda `http://127.0.0.1:8000/` yazarak istersenizde komut sisteminde adres üzerinde sağ tuş yaparak adresi açabilirsiniz.



[Anasayfa](#) | [Blog](#) | [Yönetim Paneli](#)

Bitti

Artık blog uygulamamızı yazabiliriz.

Komut satırında `python manage.py startapp blog` yazarak blog adında bir uygulama oluşturuyoruz. blog adında bir klasör oluşturuyoruz. Bu klasör içinde 4 adet dosya oluşturuluyor. Biz `models.py` ve `views.py` dosyaları ile ilgileneceğiz.

`__init__.py`: Hatırlarsanız klasöründe python paketleri olarak algılanmasını belirtiyor.

`models.py`: Veritabanına ekleyeceğimiz bilgiler.

`views.py`: Veritabanından bilgi çekmeye ve `urls.py` yardımı ile html dosyalarına göndermemizi sağlayacak.

`tests.py`: İleride göreceğimiz projelerin önceden test edilip daha sonra işlem yaptırılması için

Hemen `models.py` dosyamızı düzenleyelim:

```
# -*- coding: utf-8 -*-
from django.db import models
from datetime import datetime
# tarih ve saat fonksiyonlarını ekliyoruz
class Blogum(models.Model):
    baslik=models.CharField(max_length=200,verbose_name="Başlık",
        help_text="Bu alan blog başlığını yazınız. Max:
        200 karakter yazılabilir")
    slug = models.SlugField(max_length=200,
        verbose_name="Otomatik Link",
        help_text="Bu alan otomatik olarak oluşturulur.")
    tarih = models.DateTimeField(verbose_name="Yayın Tarihi",
        default=datetime.now)
```

```
aciklama = models.TextField(max_length="3000",
    verbose_name="Açıklama",
    help_text="Bu alana blog açıklamanızı
    yazabilirsiniz.")
html = models.CharField(max_length="3000",
    blank=True,verbose_name="HTML Kod",
    help_text="Bu alana html kod yazabilirsiniz. Max:
    3000 karakter")
# tablolara hücre ekliyor ve özellik ve kısıtlamalarını
#belirtiyoruz.

def __unicode__(self):
    return self.baslik
    return self.aciklama
# unicode desteği olacak hücreleri belirtiyoruz.

class Meta:
    verbose_name_plural = "Blog Girdileri"
# Yönetim panelinde görülmesini istediğimiz ismi belirtiyoruz.
#Yazmazsak s takısı ekleyecektir.
```

views.py dosyamızı açıyoruz.

```
# -*- coding: utf-8 -*-
from django.shortcuts import render_to_response
# html dosyasına yönlendirme fonksiyonunu çağırıyoruz.
from blogproje.blog.models import Blogum
# blog uygulamasında ki Blogum tablosunu çağırıyoruz.
def blogonsayfa(request):
    gonderiler = Blogum.objects.all().order_by('-tarih')[:5]
    sayfayagidenbilgiler = {'gonderi_listesi':gonderiler,}
    return render_to_response('blog/blogliste.html',
                             sayfayagidenbilgiler)
# blogonsayfa fonksiyonu oluşturup; gonderiler listesine Blogum
#tablosunda ki tüm verileri tarihe göre sıralatıp en son 5
#kayıdı istiyoruz. Bu kayıtları sayfayagidenbilgiler adında
#demetle alıp templates/blog klasörü altında blogliste.html
#dosyasına gönderiyoruz.

def blogdetay(request,slug):
    gonderiler = Blogum.objects.all()
    sayfayagidenbilgiler = {'gonderi_listesi':gonderiler,}
    blogdetay = gonderiler.get(slug=slug,)
    sayfayagidenbilgiler.update({'blogdetay':blogdetay})
    return render_to_response('blog/blogdetay.html',
                             sayfayagidenbilgiler)
# blogdetay fonksiyonu oluşturup oluşturduğumuz kayıtların
#detaylarını slug otomatik linki ile eşitleyip templates/blog
#klasörü altında blogdetay.html dosyasına gönderiyoruz.
```

Şimdi normalde gelmeyen ama bizim admin panelinde uygulamamızı göstermemiz ve özelleştirmemiz için gereken admin.py dosyasını yazmamız gerekiyor.

```
from blogproje.blog.models import Blogum
from django.contrib import admin
class Blogum_Admin(admin.ModelAdmin):
    list_display = ('baslik','tarih','aciklama','html')
# listeleme sırası
    list_filter = ['tarih']
# filtreleme tarihe göre
    date_hierarchy = 'tarih'
# kayıtların tarihe göre sıralamasını yap. Yıl ay gün olarak
#dizeleme yapacak
    ordering = ('baslik',)
# sıralama başlığa göre
    prepopulated_fields = {'slug':('baslik',)}
#slug otomatik linkini başlıkla aynı olsun

admin.site.register(Blogum,Blogum_Admin)
# yönetim paneline Blogum ve Blogum_admin sınıflarını ekle
```

Hazırladığımız blog uygulamasını settings.py dosyamıza ekliyoruz.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
    'blogproje.blog',
)
```

Tekrar komut sisteminde python manage.py syncdb yazarak oluşturduğumuz tabloları veritabanımıza ekleyebiliriz.


```

muslu@muslu-laptop:~/django/blogproje/blog$ cd ..
muslu@muslu-laptop:~/django/blogproje$ python manage.py syncdb
muslu@muslu-laptop:~/django/blogproje$ python manage.py syncdb
Creating table blog_blogum
Installing index for blog.Blogum model
muslu@muslu-laptop:~/django/blogproje$ █

```

Yazdığımız blog uygulamasında herhangi bir hata yoksa bu şekilde bir çıktı almamız gerekiyor. Şimdi urls.py dosyamıza blog adında bir url gelirse views.py'den oku ve yönlendirmeleri yap diye açıklama yapmamız gerekiyor.

urls.py dosyasını açıyoruz.

```

from django.conf.urls.defaults import *
from blogproje.blog.views import *
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    (r'^$', 'django.views.generic.simple.direct_to_template',
     {'template': 'index.htm'}, 'index'),
    (r'^blog/$', blogonsayfa),
    (r'^blog/([\w\ -]+)/$', blogdetay),
    (r'^admin/', include(admin.site.urls)),
)

```

templates klasöründe blog adında klasör oluşturuyoruz. İçine blogliste.html ve blogdetay.html adında 2 adet html dosyası oluşturuyoruz.

blogliste.html

```

{%extends 'index.htm'%} <<<----- index sayfasını sayfanın
                               başına yükle

{%load markup%}

<html>
<head>
<title>Blog onsayfa</title>
</head>
<body>
{%block main%} <<<-----sayfayı parsellemiştik. Yani index.htm 'deki
                               block main alanına sadece burası gelecek

{%for blogoku in gonderi_listesi%}
<h1>
<a href="/blog/{{blogoku.slug}}">
{{blogoku.baslik|truncatewords_html:'4'}}
</a>
</h1>
<br/>
{{blogoku.aciklama}}
<br/>
{{blogoku.tarih|date}}
{%endfor%}
{%endblock%}
</body>

```

blogdetay.html

```

{%extends 'index.htm'%} <<<----- index sayfasını sayfanın
başına yükle
<html>
<head>
<title>{{blogdetay.baslik}}</title>
</head>

```


masının sevinci....

Html dosyalarınızı istediğiniz gibi CSS, Jquery, Javascript, Flash vs.. ile görselleştirebilirsiniz.

Dersin videolu anlatımı için burayı tıklayabilirsiniz..

<http://www.yazki.com/kategori/altkategori/detay/django-da-blog-uygulamasi-hazirlama-ders-suresi-35-dk-hazirlayan-muslu-yusksektepe>

Saygılar.

Muslu Yüksektepe
musluyuksektepe@gmail.com
<http://www.yazki.com>

Firefox adres çubuğuna **about:config** yazdığınızda açılan ekran, Firefox'un ayarlarını detaylı olarak görebileceğiniz ve düzenleyebileceğiniz bir ekrandır. Bu ayarlardan bazılarını **Düzen>Seçenekler** yolu ile açılan **Firefox Tercihleri** iletişim penceresinden de ulaşabilir ve ayarlayabilirsiniz; ancak hepsini değil. Kısacası **about:config**, daha fazla ayar ve kişiselleştirme yapmak istediğinizde, size, binlerce ayara ulaşma imkanı veren bir arayüz.

Config ekranında; **ayarın ismi, durumu, türü ve değeri** sütunları yer alıyor. Durum sütunu, ayarın kullanıcı tarafından değiştirilip değiştirilmediği bilgisini veriyor. Değişiklik yaptığınız bütün ayarlarda ortak iki değişiklik olacak; hem yazı tipi kalınlaşacak, hem de varsayılan değer, **kullanıcı tanımlı** değerine dönüştürülecek. **Türü** sütunu, ayara ne tür değerler girebileceğinizi gösteriyor. **Sayı** türüne -adı üzerinde olmasına rağmen her sayıyı giremezsiniz. Sadece tam sayı olan değerleri kabul eder. **Boolean** türü ayarlar, değer olarak **true** veya **false** değerlerini, yani doğru veya yanlış değerlerini kabul ederler. Boolean türünde olan değerleri değiştirmek için, ayarın üzerine çift tıklayıp ayarı değiştirmek yeterli olacaktır. **Katar** türü ise alfanumerik değerler olabilir.

Yapılan ayar değişikliklerinden bazılarını etkin hale getirmek için, Firefox'u kapatıp açmanız gerekiyor. Eğer yaptığınız değişikliğin etkili olmadığını görürseniz, Firefox'u kapatın ve tekrar açın. Yazıdaki başlıkları kopyalayıp **Süzgeç** kısmına yapıştırırsanız, istediğiniz ayara daha hızlı ulaşabilirsiniz. Bu yazıda, config ekranında Firefox'un görünümünü değiştirebileceğiniz ayarlardan bazılarını bulabilirsiniz.



[browser.fullscreen.autohide](#)

Eğer siz de tam ekran görünümünde, yani **F11** tuşuna bastığınızda, adres çubuğunun ve sekme çubuğunun görünmesini istiyorsanız size yardımcı olacak ayar bu. **browser.fullscreen.autohide** seçeneğini bulduktan sonra üzerine çift tıklayın. Bu çift tıklama varsayılan değer olan **true** değerini, **false** haline getirecektir. Firefox'u kapatın. Yeniden açtığınızda, artık tam ekran modunda adres çubuğunu ve sekme çubuğunu görebileceksiniz.

[browser.tabs.tabMinWidth](#) ve [browser.tabs.tabMaxWidth](#)

Bunlar, sekme başlıklarının genişliğini ayarlamak istediğinizde bakmanız gereken ayarlardır. **browser.tabs.tabMinWidth**

ayarı, sekmelerinizin başlığının bulunduğu alanda (sekme çubuğu da deniyor) birden fazla, mesela abartalım ve 20 tane diyelim, sekme açıldığında; sekmelerin başlıklarına sürüklemeye yapmadan ulaşmanızı sağlayacak şekilde, sekmelerin başlık alanlarının kaç piksel değere kadar küçültüleceğini ayarlıyor. Varsayılan değer olarak 150 piksel girilmiş, bu değeri 50 (yani 50 piksel) yapar ve 20 tane sekme açarsanız, ne demek istediğimi anlarsınız.

Buradaki değer, piksel olarak verildiği için, ekran çözünürlüğü ile doğrudan ilişkili. Eğer ekranınız 1024 piksel genişlikte ise, bu değere 40 verdiğinizde, sekme çubuğunda herhangi bir sekme başlığı kaybolmadan (aradaki boşluklar ve sağda ve solda yer alan simgelerin genişliklerini de düşüğünüzde) 21 tane sekme açabilirsiniz. Deneyin ve görün.

browser.tabs.tabMaxWidth ayarı ise, tek bir sekme açarken, sekme başlığının ne kadar geniş olacağını belirliyor. Varsayılan değer olarak 250 piksel girilmiş. Ben, bu değeri 150 piksel olarak değiştirdim; siz de zevkinize göre bir değer girebilirsiniz.

browser.tabs.closeButtons

Sekme başlıklarında yer alan kapatma düğmelerini kaldırmak veya sadece aktif sekmede kapatma düğmesinin bulunmasını isterseniz, sekme çubuğunun en sağında sadece bir tane sekme kapatma düğmesinin olmasını isterseniz veya ekranın hiçbir yerinde sekme kapatma düğmesi görmek istemiyorsanız; bunun çözümü de **about:config** sayfasındaki **browser.tabs.closeButtons** ayarı. Eğer siz de benim gibi

sekmeleri açmak/kapatmak için farenin tekerleğini kullanıyorsanız, bu ayar, özellikle hoşunuza gidebilir.

Bu ayarın üzerine çift tıkladıktan sonra, girebileceğiniz dört değer var:

- 0** > Sadece aktif sekmede kapatma düğmesi gösterir.
- 1** > Bütün sekmelerde birer kapatma düğmesi gösterir.
- 2** > Ekranda herhangi bir sekme kapatma düğmesi göstermez.
- 3** > Sekme çubuğunun en sağında bir tek kapatma düğmesi gösterir.

browser.enable_automatic_image_resizing

Bir internet sayfasının dışında bir resme bakarken, eğer resim, tarayıcı penceresine sığmayacak kadar büyükse, tarayıcı penceresine uyacak şekilde küçültülür. Bu, Firefox'un varsayılan davranışıdır. Ayarın değeri **false** yapılırsa, ekrana sığmayan resimler orijinal boyutları ile gösterilirler; ekrana sığması için küçültülmezler. Güllü seven, dikenine katlanır. Bitmap resimlerde, büyüdükçe oluşan tırtıklı ve bozulmuş görüntüden hoşlanmıyorsanız, bu ayarın bir diğer etkisi de resimlerin orijinal boyutları dışında büyütülememesi. İçerisinde resim ve yazı olan bir sayfayı, bu ayarı **false** yaptıktan sonra büyütürseniz, ayarın nasıl davrandığını hemen kavrayabilirsiniz.

browser.search.openintab

Arama çubuğuna arama için bir metin girip **enter** tuşuna basıldığında, arama sonuçlarının yeni bir sekmede açılmasını sağlamak için, varsayılan değer olan **false** değerinin üzerine çift tıklanıp, değerinin **true** olarak değiştirilmesi gerekiyor.

browser.tabs.closeWindowWithLastTab

Bu ayar, son sekme kapatmanın Firefox'u kapatıp kapatmayacağını kontrol ediyor. Varsayılan değeri **true**. Yani son sekme

kapatılırsa, Firefox kapatılıyor. Değer **false** yapılacak olursa, son sekme kapatmak Firefox'u kapatmıyor. Varsayılan ayar Firefox 3.5 sürümü öncesi **false**; yani son sekme kapatmak Firefox'u kapatmıyormuş.

mousewheel.with[alt|control|no|shift|meta]key.action

Bu ayar, klavyede ilgili tuş basılı tutulurken fare tekerleği hareket ettirildiğinde, Firefox'un nasıl davranacağını ayarlıyor.

Süzgeç: mousewheel.with

Ayar ismi	Durum
mousewheel.withaltkey.action	varsayılan
mousewheel.withaltkey.numlines	varsayılan
mousewheel.withaltkey.sysnumlines	varsayılan
mousewheel.withcontrolkey.action	varsayılan
mousewheel.withcontrolkey.numlines	varsayılan
mousewheel.withcontrolkey.sysnumlines	varsayılan
mousewheel.withmetakey.action	varsayılan
mousewheel.withmetakey.numlines	varsayılan
mousewheel.withmetakey.sysnumlines	varsayılan
mousewheel.withnokey.action	varsayılan
mousewheel.withnokey.numlines	varsayılan
mousewheel.withnokey.sysnumlines	varsayılan
mousewheel.withshiftkey.action	varsayılan
mousewheel.withshiftkey.numlines	varsayılan
mousewheel.withshiftkey.sysnumlines	varsayılan

mousewheel

Firefox'ta fare tekerleğinin davranışını kontrol eden ayarlar

mousewheel.with[alt|control|meta|no|shift]key
mousewheel.horizscroll.with[alt|control|meta|no|shift]key

Girilebilecek olası değerler:

- 0 : Belge X satır sayısı aşağı/yukarı/sağa/sola kaydırılacak
- 1 : Belge bir sayfa aşağı sürüklenecek
- 2 : Geçmişte ileri/geri hareket edilecek
- 3 : Metin büyütülecek/küçültülecek
- 4 : Belge X piksel kadar aşağı/yukarı/sağa/sola kaydırılacak

mousewheel.with[alt|control|meta|no|shift]key.numlines

Kaydırılacak satır sayısı (eğer uygunsa)

mousewheel.with[alt|control|meta|no|shift]key.sysnumlines

Kaç tane satır kaydırılacağı konusunda sistem öntanımlı değerlerini kullan

Aşağıdaki resimde kırmızı ile işaretli ayarların üzerine tıklan-
ğında, girilecek olası 0 ile 4 arasındaki değerler; **alt, control, meta, no (yani hiçbir tuş basılı olmadığında)** veya **shift** tuşlarına basıldıktan sonra fare tekerleği yukarı aşağı hareket ettirildiğinde, tekerleğin nasıl davranacağını veya ne hızda kaydırma yapacağını belirliyor. Yukarıdaki resimde, bu ayara girilebilecek olası değerlerin nasıl etki edeceği anlatılmıştır. Eğer çift tıkladıktan sonra açılan pencereye **0** veya **4** değerlerin birisini girdiyse, bir başka ayar daha yapabiliyorsunuz.

mousewheel.with[alt|control|no|shift|meta]key.numlines ayarına çift tıklayarak, eğer bir önceki ayarda **0** değeri girdiyse, fare tekerleği hareket ettirildiğinde kaç satır hareket edileceği; eğer bir önceki ayarda **4** değeri girdiyse, fare tekerleği hareket ettirildiğinde kaç piksel hareket edileceğini belirleyen değeri giriyorsunuz. Eğer bir önceki ayara **0** değeri girdiyse, 15-20 üzeri değerler, -tabi bu ekran çözünürlüğüne ve sistem ayarlarınıza da bağlı- tek bir fare tekeri hareketinde internet sayfasının ekranda görünen kısmından fazlasının kaymasına neden olabilir. Yani, çok büyük değerler girerseniz internet sayfasının bazı kısımlarını görmeden atlayıp, sayfanın sonuna gelebilirsiniz.

mousewheel.with[alt|control|no|shift|meta]key.sysnumlines ayarını ise; bu iki ayarda, yani **mousewheel.with[alt|control|no|shift|meta]key.numlines** ve **mousewheel.with[alt|control|no|shift|meta]key.action** ayarlarında yapılan değişiklikleri etkin hale getirmek için, **false** yapmanız gerekir. Eğer yukarıdaki iki ayarı değiştirdiniz, ancak bu ayarın değerini **true** olarak bıraktıysanız yaptığınız değişiklikler işe yaramaz.

Kaynaklar:

<http://getir.net/q9x>

<http://kb.mozillazine.org/Browser.fullscreen.autohide>

<http://preferential.mozdev.org/preferences.html>

<http://getir.net/y61>

http://www.timeatlas.com/5_minute_tips/general/introduction_to_firefox_preferences

Aşağıdaki sorular Geeknet'in başkan yardımcısı Jeff Bates tarafından cevaplandı.

SourceForge nedir ?

SourceForge, açık kaynaklı yazılımları geliştirmek ve indirmek için en köklü, en geniş ve benim fikrime göre en iyi sitedir.

SourceForge'un amacı nedir ?

Site, açık kaynaklı projeleri bir araya toplamak ve bu projelerin indirilmesini sağlamak için yaklaşık on yıl önce oluşturuldu. Sitenin gelişimiyle ilgili ayrıntılı bilgiyi [bu adresten](#) edinebilirsiniz.

SourceForge'ta yayınlanacak projelerin sahip olması gereken özellikler nelerdir ?

Yayınlanacak tüm projelerin [açık kaynak lisansına](#) sahip olması gerekiyor. Ancak bu lisanslar, BSD Lisansı ya da GPL gibi popüler lisanslarla sınırlı değil, çok sayıda [seçenek](#) bulunuyor.

Projelere ne tarz hizmetler sağlıyorsunuz ?

Sağladığımız hizmetlerin başında hosting geliyor. Hosting sayesinde çok sayıda geliştirme aracını bir arada sunuyoruz; hata denetimi, e-posta listeleri, forumlar, wiki'ler, web sayfaları, on ikiden fazla yardımcı uygulama gibi. Beş kıtada bulunan sunucularla daha hızlı erişim sağlamak mümkün. Ayrıca ABD'de bulunan teknik destek ekibi, kullanıcıların sorunlarını çözmesine yardımcı oluyor. Sunduğumuz hizmetlerin geri kalanını [buradan](#) görebilirsiniz.

Sourceforge'u benzer sitelerden farklı kılan şey nedir ?

Eski ve oldukça geniş çaplı olmasından dolayı herkes SourceForge'u açık kaynaklı yazılımın ilk adresi olarak görüyor. Milyonlarca ziyaretçimiz var ve arama motorlarında yüksek görünürlüğe (high visibility) sahibiz. Yani insanların bizi ve yayınladığımız projeleri bulması oldukça kolay.

SF'de yayınlanan kaç proje var ?

Yaklaşık 300 bin proje olduğunu söyleyebilirim.

SF tarafından geliştirilen projeler var mı ?

Elbette ! Son projenin adı Ming, ilişkisel olmayan (non-relational) MongoDB ile etkileşimi sağlayan bir araç.

SF'de yayınlanan en sevdiğiniz projeler hangileri ?

Projeler çocuğumuz gibi, bu yüzden birini seçmek zor. Bazen kafa yormam gerekenlerle uğraşmayı seviyorum, bazen de daha kolay ve eğlenceli olanları tercih ediyorum.

Her ay, ayın projesini seçiyorsunuz. Bunun için ölçütleriniz neler ?

Herkes tarafından beğenilen ve hareketlilik sağlayan ya da sıkça indirilen projeleri seçiyoruz. Ayrıca SourceForge'un onuncu yılı olduğu için, bu yıla özel bazı

planlarımız var. Önümüzdeki on ay boyunca, [ayın projesi](#) seçilen her proje bir yıl boyunca SourceForge'ta yayınlanacak. Her ayın ilk haftası [Community Blog](#)'a bakmayı unutmayın, tanıdık isimler göreceksiniz.

SF'te çalışma olanağı sağlıyor musunuz ?

Kullanıcılarımıza iş olanağı sağladığımız oluyor. İlanları [bu sayfada](#) yayınlıyoruz.

Yakın zamanda SF'e eklemeyi düşündüğünüz özellikler var mı ?

Yakında bazı büyük çaplı değişiklikler olacak, ancak henüz bir şey söylemek için erken. Yine de eleştirilerin bir kısmına yanıt vereceğimizi ve istenen özelliklerden bazılarını siteye ekleyeceğimizi söyleyebilirim.

SF projesinin açık kaynaklı yazılımları yayınlamasının ve geliştirilmesine olanak sağlamasının arkasında yatan fikir nedir ? Neden SF özgür yazılımı desteklemek istiyor ?

Microsoft bunun için bize para ödüyor ve bunu her şeyi tekelinde buldurmak için yapmıyor. Hayır, şaka yapıyorum ! Biz de açık kaynaklı yazılımları kullanıyoruz ve bunları geliştirenlere minnet duyuyoruz. Projeleri yayınlamak ve bunun için araçlar geliştirerek üzerimize düşeni yapıyoruz. Yayına başladığımızda SourceForge farklı şekilde kazanç sağlayan bir şirketin küçük bir parçasıydı. Bugünse SourceForge şirketin en can alıcı kısmı haline geldi ve artık reklamlar sayesinde kazanç sağlıyoruz. Site her zaman kullanıcılar ve geliştiriciler için ücretsiz olacak.

Bir kaç gün önce SF'in bazı ülkelerin erişimini engellediğini duyduk. Bunun sebebi nedir ?

Şirket olarak herhangi bir kişiyle, şirketle ya da ülkeyle sorunumuz yok. İş birliği yaptığımız yerleri seviyoruz ve onlar için gerekeni yapıyoruz. Ancak SF'nin merkezinin bulunduğu ABD'deki hükümetin diğer devletlerle bazı sorunları var. Bu yüz-

den hükümetin bize dayattığı bazı koşullar var ve bunları [şu adresten](#) okuyabilirsiniz.

Son olarak okurlarımıza söylemek istediğiniz bir şey var mı ?

İlgi gösterdiğiniz için teşekkürler. Pardus oldukça saygı duyuyoruz. Linux.com alan adını Linux Foundation'a çevirmeden önce dağıtımı [inceleyip](#) yüksek notlar vermiştik. Bence kaliteye ve özgür yazılım felsefesine aynı ölçüde değer veriyoruz. Başarılarınızın devamını dilediğiyle.

Rportajı Gerçekleştiren:
Erdem Artan
erdem@pardus-linux.org

Onur Tuna
onur@pardus-linux.org

Çeviren:
Melike İteralp
melike@pardus-linux.org

Rportajın İngilizce sürümünü dergimizin ekinde bulabilirsiniz.

Google Buzz ve Kullanıcıların Gizlilik Hakkı

Google, İnternet'teki bilgiyi organize etmek için emin adımlarla projeler üretti ve uyguladı. İlk başlarda siber kullanıcının gizlilik hakkı tartışma konusu olmadı. Ama yapılan aramaların nerdeyse %70'lik bir kısmı bu platform ile yapılıncaya, sorun her yerde tartışılmaya başlandı. E-posta, arama bilgileri, açık adres, resimler vs.. Bütün bu bilgilerin saklandığı bir veri tabanı şüphe uyandırdı ve bu fikir hala değişmiş değil. Özellikle, FBI'ın 'gerekliği gördüğü bilgilerde' bu bilgilere ulaşma imkanı varken; insanların akıllarındaki soru işaretleri daha da arttı. Google'ın büyümesinin sırrı siber kullanıcının güveniyken, kendini bitirecek bir strateji izlemesini beklemek ne kadar mantıklı tartışılır. İran ya da Çin gibi devletlerin istediği servisleri keyfi yasaklamasını da hesaba katarsak, bu konulara detaylıca değinmek lazım.

Gizlilik Hakkı ve Sosyalleşme İkilemi

İnternet, Microsoft-Google arasında yaşanan bir savaşı, her haber ve gelişmeyle gösteriyor. Kişisel bilgisayar devriminin devi Microsoft, tahtını 'yeni yetme'



Robert Sanzalone

Google'a kaptırmamak için elinden geleni yapmaya devam ediyor. Google, sunduğu servisler ya da aktivitelerle hiçbir zaman kullanıcıları bir zorlamaya tabii tutmadı. Bütün servisleri eleştiriye açık ve bu servislerin gelişimini servisleri kullanan kullanıcılar yönetti; yetkisini kullanıcı ile bölüştü. Ama Microsoft tam tersi, despot taktiklerle hakimiyetini kullanıcılarına hissettirdi. Demoklates'in kılıcı her zaman kullanıcının tepesinde durdu. Google'ın gelişmesi ve büyümesi; bu tepkilerini topluca dile getiren kullanıcılar sayesinde oldu. Microsoft, her mantıklı ya da mantıksız yolla kullanıcı gizlilik hakkını ezdi geçti ve halen yapmaya devam ediyor. Google'a verilen tepkiyi, Microsoft'a veremeyenlerin tepkisi de tam burada şüpheli duruma düşüyor.

Google, özgür yazılım camiası ve kendi kurumsal gücüyle korkulan bir rakip oldu. Veritabanı sistemi için, özel derlenmiş bir Linux çekirdeği sürekli güncelleniyor ve doğacak ihtiyaçlara göre çekirdek üzerinde düzenlemeler yapılıyor. Android ve Chrome OS sistemleri, özgür yazılım

camiasının gücüyle geliyor. Bunun karşılığında, AdSense ve bağışlar yoluyla, Google karşılık vermeye çalışıyor, ama bu yeterli değil. En son olarak, FSF'nin Google'a duyurduğu açık mektup çok önemli. Siber kullanıcıların özgürlüğü birçok şirket için önemli değil. Özgür yazılım camiası, destek verdiği Google'dan, YouTube'da özgür yazılıma uygun bir lisansa benzer bir tür video türünü yaygınlaştırmasını istiyor. Eğer yapmazsa, İnternet'in gücünü sömürdüğü anlamına gelebilir. Özgür yazılım ve Google işbirliği kadar, siber kullanıcıların, içeriğe özgür ulaşma isteği var mı yok mu.. Google'ın tepkisi, soruya açık bir cevap olacaktır.

“Google, hatasızdır.” demek mantıksızlıktır. Her bilişim kurumu bilerek ya da bilmeyerek hata yapar. Ama önemli olan; hatanın sonucu nasıl olduğudur. Google'ın nerdeyse bütün servisleri, beta versiyonu ile piyasada halen. Bu yolla, “Servislerimiz hatalı olabilir. O zaman, eleştirin. Biz de sorunu çözelim” diyorlar. Siber kullanıcı istemediği halde, hangi servisi ya da zorlamayı uyguladı Google? Daha böyle bir örnek görülmedi. Ama Microsoft ve 3. Parti yazılım şirketleri için uzun bir liste çıkarabilirim bu konuda. Google aynı hatayı inatla yaptığında, Microsoft'un aldığı eleştirileri; hatta daha fazlasını ala-

caktır. Burada önemli nokta; gizlilik hakkı ve sosyalleşme arasındaki farkı belirleyebilmektir.

Hiçkimse kafasına silah tutulduğu için, özel hayatı hakkındaki bilgileri İnternet'e yayacak kadar şaşkın değildir. Aile sırları, banka kartı şifresi veya şantaj malzemesi olabilecek bir bilgi, normal şartlarda İnternet'e yayılacak değildir. Türkiye'de çok yaygın konuşulan, "Dinleme ve özel hayat bilgilerinin medyaya sunulması" sorunu, Google yaptığı için konuşulmuyor. Bu sorunu, devlet, İnternet sağlayıcı şirket ya da benzer bir kurum çok rahatlıkla yapabilir. Öte yandan Google yüzünden olan bir şikayet daha haberlere düşmedi. Bu olmayacak anlamına da gelmez, sadece olmadan herhangi bir kurum önyargılarla yargılanmamalıdır. En yakın örnek; İran, Gmail e-posta servisini yasakladı. Neden peki? Kendi e-posta istemci servisini oluşturacaklarını açıkladı İran yönetimi. Peki, kendi e-posta servisi ile özel hayata karışması ya da bu durumun denetlenmesi daha zor değil mi?

Her sektör için sosyalleşme önemli bir araçtır. Kurumun büyümesi ve piyasa devleri karşısında ayakta kalması için kritik bir noktadır. Eğer Twitter'da küçük ölçekli

şirketler anlık güncelleme yapamamasalardı, piyasa devleri büyük reklam atağıyla liderliğini daha da sağlamlaştırabilirlerdi. Blog dünyası ve bilişim sektörü yazarları, Hürriyet, Milliyet, New York Times, Wall Street Journal gibi büyük medya devleri karşısında ayakta kalabiliyorsa, bunun sebebi sosyalleşmeleridir. O yazarlar hakkında bir çok bilgiye istediğimiz gibi ulaşabiliriz; nerede okuduğu, yaşadığı şehir, aldığı eğitim, medeni durumu vs.. Bu özel hayat mıdır veya sosyalleşme midir? Bu iki terim farklı kültür, felsefe ve kuruma göre yoruma açıktır. Şimdi kime, neye ve hangi standarda göre bu terimleri kalıplaştırabiliriz?

Facebook, çok büyük bir boşluğu doldurdu. Bilerek ya da bilmeyerek, siber kullanıcıların ihtiyaçlarını karşılayan bir platform oldu. Bu güçle hızla büyüdü. Ama Twitter'ı da bünyesine dahil edince, kendi alanında kartel konumuna geldi. Piyasada ona denk bir rakip yoktu. Ta ki, Google Buzz duyuruluncaya kadar. Birkaç özellik çok dikkat çekici: "Nearby" özelliğiyle GPS teknolojisinin avantajını kullanıyor. Bu durumda, yerel küçük işletmelerin hepsi, kendi tanıtımını yapıyor. O civarda olan herkes, anahtar kelimeyi girdiğinde o işletmelerin tanıtımları duyuruluyor. Herhangi bir özgür yazılım projesinin top-

lantısını herkes kolaylıkla takip edebiliyor. Ses tanımlaması ve İnternet'te konferans teknolojisine yatırımla Google Buzz farkını gösterecek gibi görünüyor. Önümüzdeki yıl, Facebook gibi sansanyonel haberler, Google Buzz için de yapılacaktır.

Piyasada büyük holdingler, var olan güçleriyle rakip olabilecek şirketleri, reklam ve kampanyalarla etkisiz hale getiriyorlardı. Google Buzz servisi ile bu durum apaçık değişiyor. Küçük ölçekli kurumlar, büyük sermayeye ihtiyaç duymadan, yakın şartlarla rekabet edebilecek. Artık büyük sermayenin piyasa hakimi olmaya yetmediğini herkes daha iyi anlayacak.. Microsoft, büyük bir serveti reklam ve pazarlamaya ayırabilir. Ama özgür yazılım projeler ve dağıtımlar, her bölgedeki destekçilerini daha iyi organize edebilecek. Büyük bir ofis ya da büyük bir sponsora ihtiyaç duymadan büyümenin zeminini olması büyük bir olasılık. Bu hedefe Twitter yaklaştı. Google Buzz, gelişmeyi bir adım öne çekecek. Facebook, rakibinin olduğunu bildiği için; her kafasına estiğinde kullanıcı profiline görünümüyle oynayamayacak mesela. Örnekler çoğaltılabilir.

Sayılan avantajlara rağmen, sorunlu taraflar da yok değil. Google Buzz servisi,

Gmail e-posta istemcisine bağlı aktif edildi. Diğer e-posta istemcisini kullananların şikayetlerini duyar gibiyim. Özellikle, kullanıcı gizliliği sorunu çözülmeden, kullanıcı tepkisi gün geçtikçe artmaya devam edecektir. Bu sorun, mahkemeye açılan bir dava ile meyvesini vermeye başladı. "Gmail kullanıcılarının gizlilik haklarının çiğnenmesi ve bunun sonucunda; Gmail kullanıcılarına miktarı belirsiz bir tazminat ödenmesi.." isteği ile bilişim habeleri çalkalandı. Diğer muhtemel davalar da yolduğu olduğu kesin bir şey. Esas sorun orada değil ama.

Google, servislerini kullanıcılara sunmadan evvel, belli bir deneme döneminden geçirir. Önce, Googleplex'deki çalışanlar tarafından ve belirlenen bir kısım kullanıcı tarafından sınanır, muhtemel sorunlar rapor edilir, düzenlemeler yapılır. En son aşamada, 'beta' olarak kullanıcılara duyurulur. Ama Google Wave ve Buzz servislerinde bu uygulanmadı. Servis duyurulmadan önce, ekran görüntüsü verilse, uyarılara göre alınacak tepki daha az olabilirdi. Google'ın hata yaptığı kesin. Fakat Microsoft'un başını çektiği rakipleri, bu hatayı çok iyi değerlendirdi. Bilişim haberlerinde, "Google, Microsoft'un kurumsal mantığını mı yansıtıyor?" sorusu çok sık soruldu. Hatasından ders çıkarırsa,

Google yine bu durumdan kurtulabilir. Ama hanesine bir hata olarak işlendi. Bu hatalar düzelmeden sürerse, kullanıcı desteğini kaybedebilir.

Biraz beyin fırtınası yapalım. Kötümser bir tablo çizelim; Google bütün bilişim sektörlerini etkisiz kıldı, piyasada tek kaldı. Amerika'da rekabet kurumu ve Avrupa Birliği Microsoft'a neler yaptı ve yapıyor, hepimiz görüyoruz. İster istemez şirket ufalmak zorunda kalıyor. Google-Yahoo ortak stratejisinin iptal edilmesi de çok iyi bir örnek. Herhangi bir şirket, haddinden fazla büyüdüğünde, kontrol mekanizması ilk önce İnternet kullanıcılarıdır. Sonra da, Rekabet Kurulu veya Avrupa Birliği gibi resmi otoritelerdir. Kendi kendimize paranoya yaparak, sadece kendi yorumumuzu sunabiliriz.

Sonuç Olarak...

Google, büyük holdinglerin kabusu oldu. Bilişim dünyasını, yavaş ve emin adımlarla yeniliklerle tanıştırdı. GNU/Linux dağıtımlarına ve Mozilla vakfı gibi önemli projelere destek verdi. Mozilla Firefox, şu anda İnternet Explorer'ın kabusu oldu. Daha böyle çok projeye de destek sürüyor. Şartlar değişirse, YouTube'a eleştiri dolu videolar yayımlanır. Google'ın hatası

her haberde yüzüne çarpılır ve bedeli her zaman ödetilir. Microsoft ve diğer şirketler bu konuda öncülük yapacaktır kesin.. Google'ın gücünü suistimal etmesini engelleyen; önce destek veren siber kullanıcılar ve Microsoft gibi dış bileyen rakipleridir. Piyasa devleri arasında olan rekabet her zaman iyidir. Rekabet bittiğinde sorun başlar.

Google, 1999 yılından beri piyasayı iyi takip etmeyi ve duruma en iyi biçimde uyum sağlamayı başaran bir bilişim şirketi. Yazılım şirketi olmasına rağmen, geçen perşembe açıklanan karara kadar Google, İnternet tarayıcı tabanlı işletim sistemi hakkında ipucu vermedi. Bu dönemde, Windows 7 duyuruldu, Ubuntu'nun 9.10 sürümü yayınlandı ve netbook piyasası iyice analiz edilebilir hale geldi. Böylelikle Chrome OS piyasaya duyurulabildi.

Bilişim piyasasında 70'li yıllarda kişisel bilgisayar devrimi yapıldı. Bu ciddi bir dönüm noktasıydı, ancak çıkan ürünler hep aynı standartta oldu. Bu süreçte Windows ve Macintosh kendi standartlarını piyasaya kabul ettirdi. Geniş bant İnternet'in yaygınlaşmasıyla şartların değişimi de hızlandı. Değişen kullanıcı istekleri doğrultusunda çıkan Netbook'larla bu şartlar iyice olgunlaştı, klasik bilgisayar mimarisi artık değişmişti. Şimdi, sadece boyutların küçülmesiyle değil, cloud computing (Bulut teknolojisi) ile bilgisayarlar yeni bir değişim yaşayacak.

Chrome OS'un geçtiğimiz perşembe günü duyurulan Beta versiyonu bile bir çok piyasaya devini düşündürmeye başladı. İnternet tarayıcı tabanlı işletim sisteminin

GNU/Linux dağıtımını olarak çıkarılması, farklı fikirlerin ortaya atılmasına neden oldu. Popüler GNU/Linux dağıtımlarının piyasa payına hitap eden bir ürün olabileceği düşünülüyor ama Chrome OS'nin geliştirilmesinde Canonical'ın geliştirici kadrosundan bir kısmının destek olduğu konuşuluyor.

Chrome OS'nin, GNU/Linux dağıtımlarının büyümesini etkileyip, etkilememesi olasılığını da düşünmek gerek. Bu olasılığın olması çok az görünüyor çünkü Chrome OS'nin geliştirilmesinde GNU/Linux dağıtımlarının toplulukları'nın destek ve katkıları da olacak. GNU/Linux camiası: Dağıtımları ve zengin geliştirici ve program kapasitesiyle istediği kitleye hitap edemedi şu zamana kadar. İntel Moblin Projesiyle dikkat çekebilmeyi başardı ama yine de eleştirel bakış açısında olan kullanıcılar, önyargılı fikirlerine devam ediyorlar. Chrome OS, 2010'nun ikinci yarısında Netbook'larda önyüklemeli olarak geldiğinde, bu GNU/Linux camiası'nın yararına da olacak. Microsoft'un kaynak kodlarını kapatarak, GNU Gpl lisanslı kodları çalması gibi bir durum da olmayacak bu. Tümünüyle özgür yazılım programcılar tarafından desteklenen ve kaynak kodları herkese açık bir proje bu. FSF'nin felsefesine karşıt bir durum çıkma ihtimali az.

İnternet tarayıcı gelişimini bilen kişiler, bu sisteme kuşkuyla bakabilir, ancak iyi planlanan ve istenen amaca ulaşması beklenen bir işletim sisteminden söz ediyoruz. Microsoft ve Apple, "İnternet'in büyüyen gücüyle bir platform." olabileceği ihtimalini yok saydı. Bu gerçeğe göre işletim sistemlerini ve sundukları özellikleri ayarlamadı. GNU/Linux dünyası zaten elektronik olan her cihaz için yeniden derlenebiliyor, gelişen koşullara her zaman bir yenilik çıkartabilecek esnekliğe sahip. Chrome İnternet tarayıcısı ile zaten tarayıcılarda olan sorunlara kalıcı çözümler bulmanın yolundaydı. Google ürünü piyasaya sürdü ve kullanıcıların performans sorunlarını değerlendirerek, var olan sorunları da çözdü. 2010 yılında karşımıza alışkanlıklarımızı toptan değiştirebilecek bir işletim sistemi çıkabilir.

Google'un Eleştirisi Alabilecek Hataları

Google, arama motoru sektöründe aldığı eleştiriler biliniyor. Kullanıcıların bilgilerine sahip olması endişe sebebi oluyor. Chrome OS sisteminde sabit disk olmadığına göre, sistemi kullanacak kişilerin kişisel bilgilerinin nasıl, nerede ve hangi koşullarda saklanacağını açıklanması

çok önemli. Ayrıca Netbook'lar için düzenlendiği düşüncesi de yanlış anlatılıyor. Chrome OS, kendi standartları, oluşturacağı kullanıcı kitlesi ve yepyeni fikirlerle sunuluyor. Nasıl ilk İpod ya da Netbook kendi standartlarıyla piyasa girdi, Chrome OS'de kendi standartlarıyla piyasaya girecek.

Google'ın güvenlik için, kendi istediği firmaların modelleriyle sistemini piyasaya

sürmesi de eleştiriye açık olabilecek bir nokta.. Google tarafından yapılan açıklama yeterli olmamasına rağmen, bu konu tek bir makaleyle açıklanabilecek bir konu değil. İyi ya da kötü yanları düşünürüz ama bilişim piyasası'nın makul bir rekabet ortamına gireceğinin işaretlerini görmek mümkün. Tek bir yazılım şirketinin tekelinde olmayan bir piyasa, bütün kullanıcılar için seçim özgürlüğü demektir.

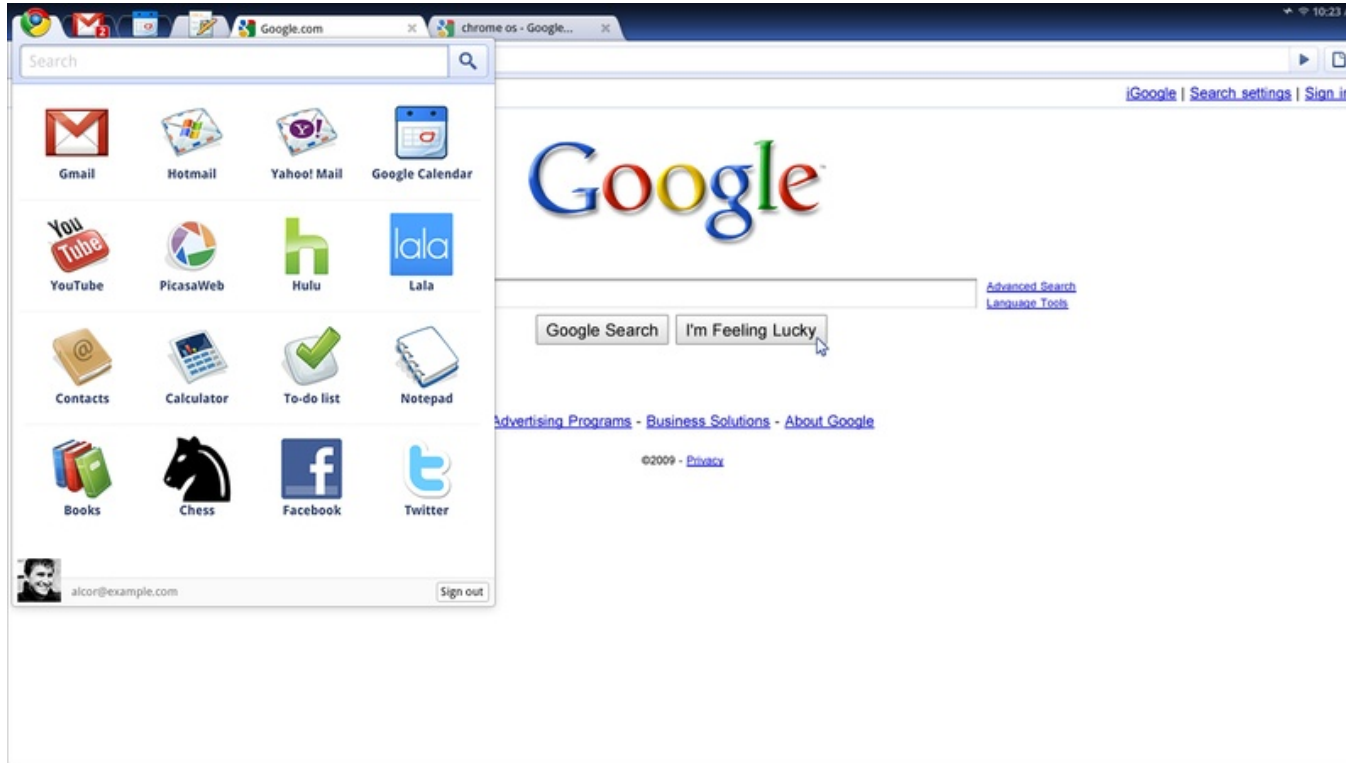
Kaynak:

<http://arstechnica.com/open-source/news/2009/11/chromeos-announcement.ars/1>

http://blogs.computerworld.com/15127/ubuntu_s_canonical_and_google_partner_to_create_chrome

http://www.pcworld.com/article/168028/google_announces_chrome_os.html

<http://www.cnn.com/2009/TECH/11/20/google.os/>



Wikipedia'dan

Kemal Karataş
kemal@pardus-linux.org

Giriş

Cebit Eurasia 2009 Bilişim Fuarı'nın bu seneki söylemi; "Sektörler Bilişimle Buluşuyor" idi. Peki sahiden de sektörler bilişim ile buluştular mı? Yoksa köşe başında kendilerini bekleyen bilişim tekelleri miydi? Üstelik Ar-Ge yatırımlarının peşinde koştuğumuz Avrupa Birliği ülkelerinin kat be kat altında olduğu düşünülürse değil bilişimle buluşmak teknolojiye dahi uzagız. Birde tüm bunların üzerine ulusal bir bilişim, Ar-Ge ve teknoloji vizyonsuzluğunu; siyasal iktidarın kalkınma planlarıyla uyguladıkları arasındaki uçurumu da eklersek ne dediğimiz ortaya çıkacaktır.

Kavramlar üzerinde biraz gezinti

Bilişim ve bilgi teknolojileri, kısa bir tanımla mesafeleri kısaltan, zamandan kazandıran günümüzün büyük teknolojisi. Sadece mesafeleri kısaltmak değil maliyetleri de düşürmekte üstüne yok; verimli, hızlı, sonuç odaklı çalışabilmenin gerek kişisel, gerek firmalar, gerekse kamu kurumları bazında kapılarını aralamıştır. Üstelik an be an gelişen bilginin de sınıflandırılması, işlenmesi, değerlendirilerek verimli hale getirilebilmesi bilişim teknolojilerinin kullanılması sayesinde. (dip-

not: Küçük bir örnek için Pardus-Linux eDergi'nin 14. sayısında yayımlanan... söyleyişi okuyunuz.)

Teknoloji ise ilk mekanik halı dokuma tezgâhı icat edildiğinden bilgisayarlar er meydanına çıktığından beri sürekli gelişen ve insanların aktif olarak bulunduğu her türlü alanı, iş kolunu düzenleyen, iyileştiren, kolaylaştırıcı öneme sahip bilimsel bir alandır.

Gerek bilişim gerekse teknolojinin hızına bugün yetişmek mümkün değildir. Basitçe artık zaman ve mekan gerçeklikleri bu iki kavram sayesinde ki kısalmıştır. Devlet kurumları, firmalar ve içinde buldukları sektörler, insanlar da erişebildikleri orada bilişim ve teknolojiye ve de sahip oldukları bilgi-birikim-beceri ekseninde işlerini görmektedirler. Ancak hepsinin ana amacı aynıdır: daha verimli, daha hızlı, sonuç odaklı, uygun maliyetli ve güvenli bir çalışma.

Bilişimin tüketilmesi

Bugün kuşku yok ki, bilişim teknolojilerinin en büyük tüketicisi devlettir; ardından firmalar gelir ve sonrasında ise her ikisinden bağımsız fakat her ikisiyle de iç-

li dışı insanlar. Devlet büyük olmasına karşın ne yazık ki bu gücünü yeterince kullanamamaktadır. Özellikle de işin içinde yandaşlarını, kendisini seçimlerde destekleyenleri zengin etme duygusallığı işin içine girince büyük alıcı baskı kurup bilişim teknolojisini üretilen kendisine pazarlayan firma ya da firmanın ülkedeki dağıtıcısına kendi işlemlerine, kullanım amacına yönelik yazılımlar ürettirip, fiyat kırdırarak yerde alakasız ihaleler yüzünden deyiz yerindeyse kazıklanmakta, kaynaklar başka yerlere aktarılmaktadır. Muhalefet tarafından, tüyü bitmemiş yetimin savunucusu yazarlardan sıklıkla söylenir, filanca il milli eğitim kurumunun yazılım alım ihalesinde usulsüzlük oldu, satın alınan donanımlar kullanışsız ve sonraki sürümü desteklememekte, ihale işin ehli olmayan yandaş partizan firmaya bırakıldı...

Ancak firmalar ne istediklerini çoğunlukla bilirler. Çünkü amaçları üretim veya hizmettir ve verimliliklerini iyi planlayıp kar etmek isterler. Bu yüzden yazılım-donanım tedarikçileri ve bilişim teknolojilerini (BT) kendilerine pazarlayanlarla iyi alışveriş ve pazarlık yapıp isteklerine uygun alımlar yaparlar. Hatta teknolojik danışmanlık firmalarından yardım ve destek alırlar.

Son kullanıcı ise maalesef araştırdığı va-
kit işini görecektir tüketim yapar; yoksa işi
teknoloji zincir mağazalarının aç gözlülü-
ğüne kalır ki, bu kazıklanmak demektir.

Bilişim Teknolojileri pazarı olarak ül- kemiz

Hiç kuşkusuz son yıllarda bilişim ürünleri-
nin tüketimi ülkemizde yükselişe geçti ve
BT üreticileri; donanım olsun, yazılım ol-
sun, her türlü bilgisayarlar olsun; için ül-
kemiz büyüyen bir pazar. Ancak sadece
tüketim açısından. Yazılım alanında ise ü-
retim çok az ve çoğunluğu da kaynak ko-
du kapalılığını esas ilke benimseyen yazı-
lımların, işletim sistemlerinin yerelleştir-
me çalışmaları ve pazarlanması ile ilgili.

BT'ye hakimiyetin tüketmekten değil de
üretmekten, Ar-Ge faaliyetlerine kaynak
ayırp ağırlık vermektense geçtiği diğer ül-
kelere bakılınca görülmekte. Bugünün
cep telefonu ve mobil iletişimin şeklini
değiştiren lider firması Nokia'nın dünün
orman ürünleri alanında iş yapan firması
olduğuna kimi inandırabilirsiniz? Örnek-
leri çoğaltmak mümkün Blackberry mucu-
zesini yaratan Kanada, Silikon Vadisi ile
Amerika, uzak doğu ülkelerindeki tekno-
loji üretim üsleri, İkinci Dünya (Paylaşım)

Savaşının mağlubu Japonya.

Unutulmaması gerekir ki, bir zamanlar ül-
kemiz ve Güney Kore aynı sınıfta gelişim
ve kalkınmaya aday ülke olarak gösterili-
yordu. Ancak bugün yarışta Güney Kore'
nin bizi refah ve teknolojik ilerleme ala-
nında geçtiğini, dünyada bilinen sayıca
teknolojik markalar yarattığı ayan beyan
ortada. Bunu sakın "biz adam olmayız,
bizi ancak yabancılar ihya eder" ezikliği
olarak algılamayın. Bu tümüyle bilişim
okur-yazarlığının, bilim ve teknoloji üreti-
minin eksikliğinden; eğitimde yaratıcı ak-
lın, bilimsel düşünce ve sorgulamanın,
imkanların kısıtlanarak, hatta dışlanarak
ezberci ve sınava dayalı bir sisteme dö-
nüştürülmesi yüzündendir. Kısaca vizyon-
suzluk!

Refahın ve zenginliğin adı zamanımızda
bilim ve teknoloji. Bu ikisine giden yol ise
araştırma, sorgulama ve sürekli merak
peşinde koşmaktan geçer. Eğer bu alan-
larda yoksanız bu alanda kendisini yet-
kinleştirmiş, zenginleştirmiş ülkelerin sö-
mürgesi olmaya mahkumsunuz demektir
ki, bugün yaşanan da bu!

Ülkemiz teknoloji ürün ve hizmet pazar-
ları arasında en büyük pazarlardan birisi;
ancak yalnızca tüketen bir pazar. Hatta

kendisine eksik ve geç gelen düşük mo-
del ürünlerden kaynaklanan üretim ya da
kullanıcı hatalarını, eksiklikleri dahi geri
bildirmeyen bir tüketici pazarı. Gençlerin
ellerinden son model cep telefonları, bil-
gisayarlar, netbooklar, mekandan bağım-
sız iletişim araçları düşmüyor ama bunca
teknolojik, bilimsel alete, yaratıcılığa ö-
zenmek ise maalesef yok. Sadece kuru
kullanıcı.

Bu durumu biraz açalım: kullanılan ürüne
ve arkasındaki AR-GE, üretim süreçlerine
ve bilimsel yöntemlere karşı yabancı ve
ilgisiz kalan kullanıcılar haliyle bu tekno-
lojik aletlerin kendileriyle iletişime geç-
melerini sağlayan işletim sistemlerine de
yabancı kalmaktalar. Öyle ki, kullanıcı-
ların çoğu Microsoft ürünlerini ve diğer ka-
palı kaynak kodlu yazılım ve işletim sis-
temlerini kullanmakta. Bu duyarsızlaşma
ise sonuçta kendi güvenliklerini, özgür-
lüklerini tehlikeye attığı gibi yaratıcılık-
larını, meraklarını köreltmekte. İzlediğim
bir video da Avustralya sokaklarında bir
dizüstü bilgisayara yeni çıkan ve çığır a-
çan KDE 4 masaüstü yöneticisi kuran iki
genç, insanlara bunun Microsoft Windows
Seven işletim sistemi olduğuna inandır-
mıştı. Oysa insan doğası gereği değişir ve
gelişir; yeniliğe açıktır. Ancak koşullu şart-
lanma ve öğrenilmiş tek bilgi ile bilinen

yoldan sapmadan işlerini olanaklar arasındaki tek seçenikle yapma alışkanlığı insanları bağınaz, tutucu ve gelişime/değişime kapalı hale getirir.

Çoğu Windows kullanıcısı GNU/Linux dağıtımlarının cıvıl cıvıl ve bambaşka genişlik, canlılık ve yetkinlik içindeki masaüstü yöneticilerine, uygulamalarına ilk bakışta hayran kalmaktadır. Ardından yıllardır baskılanmış merak bir anda patlama yapmaktadır ancak beraberinde "göç korkusunu" da getirmektedir. Aslında biz Özgür Yazılım gönüllülerin yaptığı tüm yerleştirme, proje, duyuru ve eğitsel çalışmalar da bu merakı sürekli kılmak ve "göç korkusunu" yok etmektir.

Sektörler ve tüketiciler kiminle buluştu: Bilişimle mi yoksa Vizyonsuzlukla mı?



Cebit Eurasia 2009 (eski adı ile Bilişim) fuarında binlerce cıvıl cıvıl tanıtım noktası kuruldu, alanlarda mankenler eşliğinde en son model teknolojik ürünler, bilgisayarların masaüstü, dizüstü, ufak, mi-

nik, cepte-elde taşınan yığınla modeli tanıtıldı, binlerce basılı reklam broşürleri dağıtıldı, karvizitler alınıp verildi, işler bağlandı, müşteri portföyleri genişletildi...

Peki bunca üretilen ürün ve hizmetin tek bir ürüne mahkum olduğunu inkar edebilir miyiz? Eşyaya adını vermekten çekinmem ancak sıklıkla vurgu yapmaya da gerek yok: Microsoft ürün ailesi.

İşletmeler, kamu kurum ve kuruluşları her yıl korsan kullanımlar ile başları derde girmesin diye yazılımların lisans yenileme ve donanımların bakım-onarım, hizmet alımı konularında yığınla kaynak ayırmaktalar ki, bu aynı zamanda karları azaltan önemli bir maliyet kalemi. Ancak işletmeler kendilerine sunulan bu tek ürüne kendilerini o kadar kaptırmışlardır ki, gidip gezdikleri fuarlardan bir şey kapmadan sadece sunulanın yeni makyajlısını alıp geliyorlar: Microsoft Xp, Vista, şimdi de Seven; MS Office 2000, 2003, 2007... Liste uzar gider, haliyle bunlara uygun donanımlar da satın alınır durur.

Araya Cumhuriyet Gazetesinden[1] bir haber sıkıştırmak yararlı olur kanısında-yım: **'ÖZGÜRLÜK İÇİN PARDUS' / BARIŞ YAMAN**

KONYA - TÜBİTAK bünyesinde geliştirilmekte olan özgür yazılım lisanslı, ulusal işletim sistemi PARDUS, başta Savunma Bakanlığı, Milli Savunma Bakanlığı, Çanakkale 18 Mart Üniversitesi, Adıyaman Üniversitesi gibi onlarca kurum tarafından kullanılıyor. PARDUS'un 150 binden fazla bireysel kullanıcısı bulunuyor. Selçuk Üniversitesi'nde düzenlenen BİLMÖK Kongresi'ne katılan PARDUS geliştiricilerinden Bahadır Kandemir, "Özgürlük için PARDUS" sloganı ile çalışmalarını yürüttüklerini belirtti. PARDUS'un Türkiye'nin en büyük ve en güçlü yazılım projesi olduğunu ifade eden Kandemir, "Diğer programlardan farklı olarak edinimi ücretsiz ve dağıtımı özgür olarak yapılır. İstenildiği gibi değiştirilebilir ve açık kaynak kodludur" dedi.

Kar-zarar ve fayda-maliyet kısır döngüsü içinde verimlilik, zaman ve hız kazanmak, tasarruf etmek için tek üretici ve onun ürünlerine göre hayatını yönlendiren donanım üreticilerine mahkum olan bir mecbur-seçeneksiz tüketici, doğrudan alıcı pazarı mevcut.

[1]Cumhuriyet Gazetesi'nin 4 Mart 2010 Perşembe günü yayınlanan sayısının 7. sayfasında Barış Yaman imzası ile yayımlanmıştır.

Değil yılda bir-iki her gün bilişim ve teknoloji üzerine fuarlar düzenlensin bu durum söz konusu oldukça neye yarar? Kazanan teknolojiyi üreten ve üretemeyen pazarlara pazarlayıp satanlar olacaktır elbet.

Neye mi yarar? Sadece teknolojinin tüketilmesine. Onu anlamlandırmaya değil, bilgiyi çoğaltmaya hiç değil. Zaten satın alınan bir bilgi bilgi değildir ki! Kaldı ki, kaynak kodu kapalı ve özgür olmayan yazılımlar size sadece kullanıma hakkını satıyorlar, bilginin kaynağını değil. Okumadan (nasıl okunsun ki, çoğu Türkçe dahi değil ve bu açıkça tüketici haklarının da ihlalidir; neticede satın alınan mal ayıplı çıkarsa dili Türkçe olmadığı için tüketicinin zararı olacaktır) kabul edip kurduğumuz çoğu kaynak kodu kapalı ve özgür olmayan yazılımın sözleşmesinde bunlar açıkçası belirtilmiştir: kopyalayamazsın, dağıtamazsın, kendi evinde dahi olsa ikinci bir bilgisayara kuramazsın; kurmak istersen çoklu lisans alırsın; isteğine göre düzenleme yapamaz ve teşebbüs dahi edemezsin, satamazsın, iade edemezsin (evet, bu çok saçma değil mi?) gibi yığınla kısıtlama.

Oysa Özgür Yazılım ve evriminin birer sonucu olan GNU/Linux dağıtımları kullanıcıların yaratıcılığını tabiatları gereği özgür olduklarından kısıtlamak da ne, daha da genişletirler. Şimdi söz Stallman'da[2]:

Özgür yazılım kullanıcının özgürlüğüne ve kullanıcı topluluklarının sosyal dayanışmasına saygı duyan yazılımdır. Özgür yazılım, sizin yani kullanıcının dört temel özgürlüğe sahip olmanızı sağlar:

***Özgürlük 0**, bir programı istediğiniz gibi çalıştırma özgürlüğüdür.*

***Özgürlük 1**, programın kaynak kodunu çalışabilme ve programın ne yapmasını istiyorsanız o şekilde değiştirebilme özgürlüğüdür.*

***Özgürlük 2**, istediğiniz zaman programın tam kopyalarını dağıtabilmenizi sağlar.*

***Özgürlük 3**, değiştirdiğiniz programı dağıtabilmenizi sağlar.*

Bir program bu özgürlükleri sağlıyorsa özgür yazılımdır. Özgür yazılımla kullanıcıların hem bireysel hem olarak kendi he-

saplamaları üzerinde kontrolü vardır.

Biz bu özgürlükleri sağlamayan yazılımlara özgür olmayan, telif haklarıyla korunan yazılımlar diyoruz; bunlar kullanıcıya etik olmayan bir sosyal sistem sunar ve yazılımı geliştirene kullanıcıya göre adil olmayan bir güce sahip olmasına yol açar. Bu kullanıcıları böler çünkü paylaşma hakları yoktur ve çaresiz kılar çünkü kaynak koduna sahip değildirler. Hiç kimse bu güce sahip olmamalıdır, bu yüzden özgür olmayan yazılımlar var olmamalıdır.

Eğer kullanmak ve özgürlük istiyorsanız, özgür yazılıma bağlı kalmak tek yoldur.”

Son olarak:

Araştırma alanlarından uzak tutulan, bilime sırt çevirmiş, akılcılığı ve laikliği dışlayan günü kurtarmaya yönelik vizyonsuz politikaları benimseyen bir Türkiye bilimde, teknolojiye, bilişimde, yazılım alanlarında ve daha nice bağlantılı alanlarda ve de kültür-siyaset-ekonomi bağlamında

[2]“Dijital sömürgeleştirmeye karşı Özgür Yazılım”

Richard M. Stallman'ın 11 Ekim 2009'da SoL.org.tr ile yaptığı söyleşi. Dileyen tamamına bu adresten erişebilir: <http://haber.sol.org.tr/bilim-teknoloji/dijital-somurgelestirmeye-karsi-ozgur-yazilim-haberi-19103>

gelişmiş; atı alıp kaçırmış; ülkelerin sömürgeci politikalarının esiri, pazarı olmaktan kurtulamaz.

Lisanslar:

Makalenin tüm içeriği GNU/GPL 3[3] ve Creative Commons (by-nc-sa)[4] ile lisanslanmış olup içeriği haber verilmek ve yeniden GNU/GPL ve Creative Commons (by-nc-sa) ile lisanslanmak koşuluyla kopyalanabilir, düzenlenip değiştirilebilir, a-tıfta bulunulabilir, yeniden

[3] GNU Genel Kamu Lisansı sürüm 3'ün gayri resmi Türkçe çevirisi için Pardus Viki ekibine teşekkürler:
http://tr.pardus-wiki.org/GNU_GPL_%28Genel_Kamu_Lisans%C4%B1%29_S%C3%BCr%C3%BCm_3_Gayr%C4%B1resmi%C3%A5_%C3%87evirisi

[4] Creative Commons (by-nc-sa): Bu lisansa sahip eseri kopyalayabilirsiniz, üzerinde değişiklik yapıp yenisini üretebilirsiniz. Sağlanması gereken üç şart var. İlki, eserin tüm kopyalarında eserin ilk sahibinin belirtilmesi. İkincisi, eserin hiçbir kopyası ya da eserden üretilmiş yeni eserlerin hiçbirinin ticari ortamda kullanılmaması. Üçüncüsü, eserin tüm kopyalarında ya da eserden üretilmiş yeni eserlerde de aynı lisansın kullanılmasına devam edilmesi.
http://tr.wikipedia.org/wiki/Creative_Commons



Free as in Freedom



Aydın Bez
Ocak-Şubat 2010, İstanbul
<http://ozguryazilimsendikasi.org/bilgi@ozguryazilimsendikasi.org>
aydin@pardus-linux.org



Fırat ÖZGÜL

PYTHON

KODLAB
www.kodlab.com

ISBN :9786054205103
Yayınevi :KODLAB Yayınları
Yazar :Fırat ÖZGÜL

Pardus-Linux.Org'un emekli yöneticilerinden Fırat Özgül, Şubat 2010'da, ülkemizde sıklıkla karşılaşılmayan bir projeye imza attı. 2007 yılında Pardus-Linux.Org | Wiki'de yazmaya başlayıp, istihza.com adresinde yazmaya devam ettiği Python programlama diline ait bilgi ve belgeleri derleyip düzenledi ve bir kitap haline getirip, hepimizin kullanımına sundu.

Daha dumanı tüten bu kitap, herkesin anlayabileceği şekilde, özellikle de yeni başlayacak olan kişiler için, Python programlama dilini en temelden başlayarak anlatıyor. Kitapta, Python 2.x ve onun resmi grafik arayüz takımı olan Tkinter hakkında ayrıntılı bilgiler ve anlatımlar bulacaksınız.

Kitap, Fırat Özgül'ün dört satırlık biyografisinin ardından, kendi kaleminden çıkmış bir önsöz ile başlıyor.

Fırat Özgül önsözünde, Python dilinin kullanım alanları, çalıştığı platformlar ve kitabın bölümleri hakkında kısa bir ön bilgi veriyor. Ardından bodoslama dan konuya giriyor. Python'un, Linux ve Windows platformlarına kurulup çalıştırılmasının anlatımının ardından,

"print komutu" başlığı ile kullanıma başlıyoruz.

Kitap temel olarak üç kısımdan oluşuyor:

1. Temel Konular
2. Özel Konular
3. Arayüz

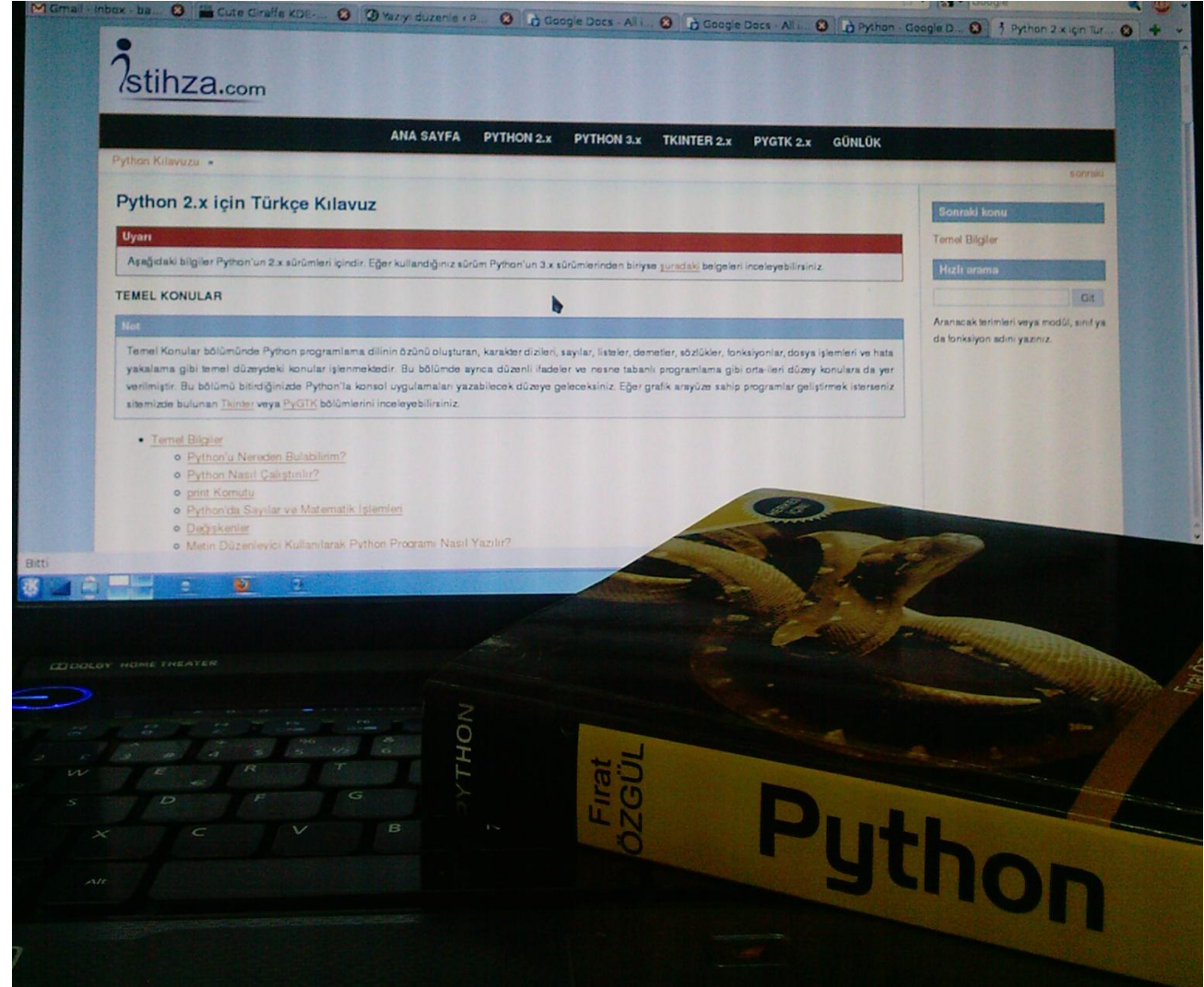
Birinci kısım olan Temel Konular bölümünde, Python dilinin özünü oluşturan temel fonksiyon ve işlevler işleniyor. Bu bölümde Python isminin nereden geldiği, Python'un nasıl edinilebileceği, nasıl çalıştırılabileceği gibi bilgilerin ardından Python ile yazılım geliştirmeye basit bir giriş yapılan "Temel Bilgiler" konusunun ardından, "Koşullar", "Döngüler", "Listeler, Demetler, Sözlükler", "Fonksiyonlar", "Modüller", "Dosya İşlemleri", "Hataları Gidermek", "Karakter Dizelerinin Metotları", "Düzenli İfadeler", "Nesne Tabanlı Programlama", "Ascii, Unicode ve Python" ile "Biçim Düzenleyici" konuları işleniyor.

İkinci kısım olan Özel Konular bölümünde ise matematik fonksiyonları içeren math modülü, PDF ile ilgili fonksiyonlar içeren pyPDF modülü ile OpenOf-

face.org ile ilgili fonksiyonlar içeren PyUno modülü hakkında bilgilere yer verilmiş; Python'da Id() fonksiyonu ve ls işleci ile Windows'ta Python'u YOL'a Ekleme konuları işlenmiş.

Son kısım olan Arayüz bölümünde Python ile arayüz tasarımının nasıl yapılabileceği konusuna değinilmiş. Bu bölümde, Tkinter arayüz takımı ile neler yapılabileceği konusu iyice işlenmiş.

Öğrenimi kolay ve sade bir yapısı olan Python programlama dili, anlaşılır bir kitap ile bizlere sunulmuş. Programlama dili temeline sahip olmayan bir kişi bile olsanız, kitabı kolaylıkla anlayabiliyorsunuz. Fırat Özgül'ün herkes için yazdığı Python programlama dilini anlatan kitap, programlama dillerini anlatan alışageldiğimiz inşaat tuğlası gibi kitaplardan değil. 534 sayfalık konu anlatımı olan kitap, Python hakkında kendinizi geliştirirken, her zaman elinizin altında bulunması gereken bir kaynak.



Erdem Artan
erdem@pardus-linux.org

Hamit Giray Nart
hamit@pardus-linux.org



LinuxMint Türkiye Topluluğu Yeniden!

LinuxMint Türkiye Topluluğu yeniden yayın hayatına başladı. Şu anda aktif olarak forum üzerine kurulu topluluk, dergi projesi üzerine de çalışıyor.

Google'a Açık Mektup!

Özgür Yazılım Vakfı, Google'a V8'i özgürleştirilmesi ve YouTube'da kullanması için bir açık mektup yazdı.



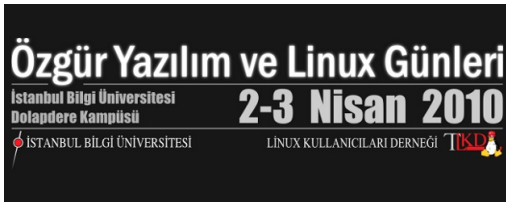
Özgür Yazılım, webin ve kullanıcıların özgürlüğüne değinilen mektuptaki şu ifade oldukça dikkat çekici:

"Eğer aksini yaparsanız, amacınızın kullanıcının web özgürlüğü değil de Google'ın hakimiyeti olduğunu biliyor olacağız."

Mektubun Türkçe çevirisinin tamamını [buradan](#) okuyabilirsiniz.

Özgür Yazılım ve Linux Günleri

Linux Kullanıcıları Derneği ve İstanbul Bilgi Üniversitesi işbirliğiyle 'Özgür Yazılım ve Linux Günleri' ismiyle düzenlenecek olan etkinlikler, 2-3 Nisan 2010 tarihleri arasında, İstanbul Bilgi Üniversitesi Dolapdere Kampüsünde gerçekleştirilecek.



Davetliler arasında Ubuntu projesinin teknik lideri Matt Zimmerman, Mozdev'in başkanı Brian King ve GNOME Vakfı'nın danışma kurulu üyesi Leslie Hawthorn bulunuyor.

Bilgisayar Bilimcisi Gibi Düşüme: Python ile Öğrenmek 2. Baskı

Tahir Emre Kalaycı tarafından çevirisi yapılan, GNU FDL (Free Document License) ile yayımlanan yani özgür olan bu kitabın yazıları Jeffrey Elkner, Allen B. Downey ve Chris Meyers tarafından, görselleri ise Udit Bhatnager ve Chris Schmeelk tarafından hazırlanmış.

Python Programlama Dilini içeren kitapta, "Programlama Yolu", "Değişkenler, Deyimler ve Cümleler", "Fonksiyonlar", "Koşul İfadeleri", "Ürün Veren Fonksiyonlar", "Yenileme", "Karakter Dizileri", "Birleşik Veri Tipi", "Listeler", "Modüller ve Dosyalar", "Özyineleme ve İstisnalar", "Sözlükler", "Sınıflar ve Nesnelere" ana başlıklarında konulara değinilirken, "Ubuntu'yu Python Geliştirme İçin Yapılandırma" konusu ek olarak sunuluyor.

Tam 206 sayfa olan eseri buradan [özgürce](#) indirebilirsiniz.

Kaynak: *KodveUs*



Bilgisayar Bilimcisi Gibi Düşünmek
"Python ile Öğrenmek"
2. Baskı

Ankara'daydık

Pardus-Linux.Org ve Pardus Kullanıcıları Derneği olarak, GNU/Linux gönüllüleriyle Ankara'da buluştuk.

İşleri nedeniyle gelemeleyenler nedeniyle katılımın biraz düşük olduğu buluşmada, Bursa ve Tekirdağ'dan gelen arkadaşlarımızın var oluşu bizlere moral kaynağı oldu.

Ankara Barosu Eğitim Merkezi'nin toplantı salonlarının birinde 11 kişiyle başlayan buluşma, daha sonra yine ABEM'in kafeteryasında devam etti. Kimi zaman havadan sudan, kimi zaman derin Pardus ve Özgür Yazılım konularında konuşulan buluşmada, neler yapılabileceği ve sorunlar tartışıldı.

Buluşmaya katılan arkadaşlara teşekkürlerimizi sunuyor, katılmayanlar ile 2-3 Nisan tarihlerindeki Özgür Yazılım ve Linux Günleri'nde görüşmeyi ümit ediyoruz.



64 Bit Pardus Çalışmaları 'Resmen' Başladı

18 Şubat 2010 tarihinde yürürlüğe giren protokol kapsamında ÇOMÜ Bilgisayar Mühendisliği Bölümü'nde yer alan bir ekip, Pardus'un 2010 yılının ikinci yarısında yayınlanacak olan ve KOBİ'lerden kurumsal işletmelere kadar geniş bir kitleye hitap etmesi amaçlanan Pardus Kurumsal 2 Geliştirici Sürümü'nü 64 bit sistem mimarisine port etme işine resmen başladı.

Protokolün resmen imzalandığı 18 Şubat tarihinde üniversitede bir atölye çalışması düzenleyen Pardus projesi, hem sürümü port eden ekip hem de bilgisayar mühendisliği bölümünün diğer öğrencileri ile yakından çalışma fırsatı buldu. İki gün süren etkinlikte Pardus geliştiricileri atölye çalışmalarında sürümün devamlılığı ve kararlılığının sağlanması için tecrübelerini ÇOMÜ ekibi ile paylaşma fırsatını buldu. Üniversitelerde özgür yazılım ruhunun ve felsefesinin yaygınlaşması açısından da son derece önemli olan projenin ilk kararlı ürününü 2010 yılı baharında vermesi planlanıyor.

Kaynak: Pardus.org.tr

Bu Sayıda Emeđi Geenler

Armađan Can
Aydın Bez
Erdem Artan
Gürhan Şükürođlu
Hamit Giray Nart
Kemal Karataş
Melike İteralp
Muslu Yüksektepe
Onur Tuna
Uđur aylık

Dergide yayımlanan tüm ierik,
yazar tarafından aksi belirtilmedike
Creative Commons 2.5
lisansı ile yayımlanmaktadır.

Altıntıların orijinal lisansları geçerlidir.



Pardus-eDergi,
Pardus Kullanıcıları Derneđi
hizmetlerinden olan Pardus-Linux.Org topluluđu
tarafından hazırlanmaktadır ve
Pardus-eDergi.Org adresi
üzerinden yayımlanmaktadır.

Pardus-eDergi'nin tasarımında,
hazır alınan resimler hari tutulursa,
özgür yazılımı destekleyen
bir topluluk olmanın verdiği sorumluluk ile
sadece özgür yazılımlar kullanılmıştır.

Pardus-eDergi'ye ulaşmak için
Pardus-eDergi.Org sitesinin iletişim formunu,
Pardus-Linux.Org forumlarını,
Freenode üzerindeki #pardus-destek ve
#parduslinuxorg kanallarını,
dergi@pardus-linux.org elektronik posta adresini
kullanabilirsiniz.

Pardus-eDergi'ye katkıda bulunmak için
Pardus-Linux.Org forumlarını, katkıda bulunarak neler
kazanabileceđinizi görmek için ise
Pardus-eDergi.Org adresini ziyaret edebilirsiniz.

