

özgürlük için .com e-dergi

Pardus ve Grafik:

Fotoğrafları Onarmak

Kotalı oyuncular yaşadı:

Assault Cube

Ofis paketimiz hızlanıyor:

OpenOffice.org 3.2

Röportaj:

Semen CİRİT

Gizli Kahramanlar:

Pardus Test Takımı

03. Editörden
- 04-09. Haberler
- 10-13. Nasıl: Bugzilla'yı kullanmak: Hatamla sev beni
- 14-16. Kurulum Cd'lerinizi Kontrol Edin
- 17-20. Python'da Döngüler
- 21-24. Frescobaldi ile İşler Nasıl Gidiyor?
- 25-26. Pardus ve Grafik: Yıpranmış Fotoğrafları Onarmak
- 27-32. OpenOffice.org 3.2: Yeni özellikler
- 33-35. Plasma: Plasma'nın Gizli Yüzü-2: Plasma pratiklik kazandırıyor
- 36-38. Paket Tanıtımı: Web tasarımcıları için: Mavi Balık
- 39-41. Oyun İnceleme: Assault Cube
- 42-46. Test Süreçleri: Pardus'un Gizli Kahramanları: "Test Takımı"
- 47-48. QA ve Linux Dağıtımları
- 49-52. Pardus Test Süreçleri ve İyileştirmeler
- 53-58. Röportaj: Pardus Test Takımı Yöneticisi: Semen CİRİT
59. Son Sayfa

Bu sayının editörü:

Fahri DÖNMEZ

Bu sayıda katkıda bulunanlar:

Ahmet Hiçyılmaz, Ali Işingör,
Ali Rasim Koçal, Anıl Özbek,
Ayhan Yalçınsoy, Ceyhun Alyeşil,
Deniz Ege Tunçay, Ertan Güven,
Gaye Demirbaş, Gizem Belen,
Göktuğ Korkmaz, Hakan Hamurcu,
Hüseyin Sarıgül, Ömer Taban,
Razık Hilenoğlu, Semen Cirit,
Server Acim, Taha Doğan Güneş
ve Tuğsan Ünlü.

Tasarım:

artistanbul
(Deniz Ege Tunçay)

Özgürlükçin e-dergisi,

Creative Commons

(by-nc-sa) 3.0 ile lisanslanmıştır.

Pardus ismi ve logosu,

TÜBİTAK UEKAE'nin tescilli markasıdır

Bu yayının, Özgürlükçin topluluğu tarafından
hazırlanmaktadır.

Pardus'un bizden beklentileri...

Merhaba, özgür yazılım takipçileri!

Her sayısı heyecanla beklenen ve içeriği Özgürlükçin topluluğu tarafından hazırlanan e-dergi'nin 21'inci sayısıyla karşınızdayız.

Bu ayki kapak konumuz, Test Süreçleri. İçinde yer almaktan büyük zevk aldığım Pardus Test Takımı'nın kapak konusu olduğu bu sayımızda editörlük görevinin bana nasip olmasından ayrı bir mutluluk duydum. Röportaj yaptığımız geliştiricimiz, test süreçlerimizin sorumlusu sayın Semen CİRİT. Şimdiye kadar e-dergimizde Pardus'un beklentilerimizi nasıl karşılayacağını anlatan yazılar çıktı. Bu sayımızda "Pardus'un bizden bir beklentisi" olduğuna dair bir kapak konusu seçtik. Pardus'un bizlerden en önemli beklentisi nedir? Test edilmek ve hataların raporlanması.

Ben şahsen Linux'u öğrenmek için dağıtımları araştırıp deniyordum. Bir tanesinde karar kıldım, kullanmaya başladım. Her türlü bilgilenmeyi Türkçe sitesinden yapıyordum ancak dağıtımın Türkçe desteği pek doyurucu değildi. Daha sonra İnternet'ten TÜBİTAK'ın Türkiye'ye özel, bizim dilimize tam desteği olan bir dağıtım geliştirdiği haberlerini gördüm. İlk çıkan Pardus 1.0 Çalışan CD denemesinden sonra büyük bir heyecanla kurulan sürümü beklemeye başladım. E-posta listelerini takip ederken testçi çağrısını gördüm ve Pardus 1.0 sürümü ile Pardus testçiliği serüvenim başlamış oldu. Diğer dağıtımlarda da gördüğüm kadarıyla, dağıtımlar toplulukları ile geliyor. Bu doğrultuda Pardus topluluğunda yer almak için elimden gelen her çalışmaya katkı sağlamaya çalıştım. Ben hayatını bilgi teknolojileri vasıtasıyla kazanan biri olarak, kendi profesyonel kariyerimi de geliştireyordum.

Anladığım kadarıyla destekçiler ikiye ayrılıyor: Katkıcı ve geliştirici. Geliştirici olabilmek için özgür yazılım geliştirme araçlarını iyi bilmek ya da öğrenmek gerekiyordu. Kendimi özgür yazılım geliştiricisi seviyesine getirinceye kadar en azından katkı olabileceğimi dedim. İhtiyaç olan katkılar da test, wiki, forum ve e-posta listeleri ile başladım.

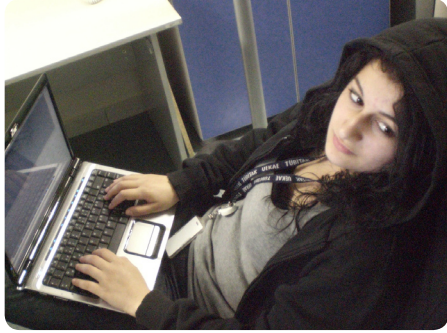
Kim istemez herkesin kullanacağı bir çalışmaya katkı sağlamak? Kültürümüzde de zaten imece diye bir kavram var. Bir işe herkes elinden geldiği kadar el atarsa, birçok eksik kapatılmış olur. Ben şahsen bu toplulukla birlikte geline noktayı gördüğümde "Keşke her gün bir test çıksa da daha da mükemmel yaklaşılsak..." diyorum.

Bu ayki sayımızda bir yazıyla mesleki, yaş ve karakter olarak ne kadar zengin bir Test Takımı'mız olduğunu size göstermeye çalıştım. Test süreçlerinde Pardus'u test eden kişilerden çok test edilen sistemlerin çeşitliliği de çok önemlidir. Ne kadar çok farklı donanım üzerinde testler gerçekleştirilirse hatalar o kadar azaltılmış olur. Siz de bu harika takıma katılmaya ne dersiniz?

Şimdiden Test Takımı'na hoş geldiniz diyorum...



Seninle Gurur Duyuyoruz Pınar!



Pardus geliştiricilerinden **Pınar YANARDAĞ**'ın adını eminiz pek çoğunuz duymuştur. Pınar, bildiğiniz gibi geçmişte TÜBİTAK UEKAE çalışanı olarak Pardus ekibinde yer almış,

birçok ilke imza atmış bir geliştiricimiz. Yıllar önce LKD'nin teknik konuda seminer veren ilk kadın üyesi olan **Pınar YANARDAĞ**, Google Summer of Code'a da kabul edilmiş ve geçen yıl Full Bright Bursu'nu kazanmıştı. Geçmişte Özgürlükçin'in e-dergisi için de [röportaj yaptığımız](#) sevgili Pınar'dan güzel bir haber, dün geldi.

Pınar YANARDAĞ, yaptığı bilimsel ve sosyal çalışmaların, liderlik özelliklerinin ve Google'a verdiği proje önerisinin değerlendirilmesi sonucunda 2010 [Google Anita Borg](#)

[AMEA](#) (Avrupa, Orta Doğu ve Kuzey Afrika) Bursu'nu kazandı.

Google 2004 yılından beri bilgisayar bilimcisi Dr. Anita Borg anısına bilgisayar bilimi alanında öğrenim gören lisans, lisansüstü ve doktora öğrencilerine Anita Borg Bursu veriyor. Burs her yıl yaklaşık 20-25 öğrenciye verilirken, öğrenciler 1 yıl boyunca akademik çalışmalarında Google tarafından finanse ediliyor.

Haziran ayında Google'ın Zürih'te düzenleyeceği ödül töreni ve etkinliklerine katılmaya hazırlanan

Pınar YANARDAĞ, bu bursun özellikle bilgisayar alanında daha çok kadının aktif olarak var olabilmesi için rol modeller ve liderler yetiştirme amaçlı olduğunu söyledi.

Çanakkale Onsekiz Mart Üniversitesi Bilgisayar Mühendisliği mezunu **Pınar YANARDAĞ**, halen Boğaziçi Bilgisayar Mühendisliği Bölümünde yüksek lisans çalışmalarını sürdürüyor.

Pardus/Özgürlükçin ailesi olarak, Pınar'ı kazandığı bu prestijli burstan ötürü kutluyor, başarılarının devamını diliyoruz.

Pardus Takım Arkadaşları Arıyor



Pardus Projesi büyümeye devam ediyor. 2004 yılında beş kişilik bir kadroyla çalışmaya başlayan proje ekibi, sürekli büyüyerek, TÜBİTAK UEKAE içinde bugün 22 kişilik bir büyüklüğe ulaştı.

Pardus'u bugüne getirenler 22 kişiyle sınırlı değil elbette. Bugüne dek dışarıdan destek veren geliştirici, paketçi, yabancı dil yerelleştirmelerini gerçekleştiren katkıcılar da dâhil olmak üzere **150'nin üzerinde** geliştiriciden "kod desteği" alan

Pardus Projesi, hedefleri doğrultusunda yeni takım arkadaşları arıyor.

Açık pozisyonlar

Tam/yarı zamanlı TÜBİTAK UEKAE çalışanı olarak çalışmak isteyenler için mevcut açık pozisyonlar şöyle:

- Pardus Dağıtım Geliştiricisi (1 kişi)
- Pardus Yazılım Geliştiricisi (2 kişi)
- Sistem Yöneticisi (1 kişi)

Özgür yazılım projelerinin geliştirme süreçleri hakkında bilgi sahibi olan, bu

projelere teknik katkıda bulunmuş ve öğrenmeye istekli, kendini geliştirmeye açık olan genç geliştiricilerin kariyeri için önemli bir fırsat. Bu fırsatı değerlendirmek ve başvuru yapmak için detaylı özgeçmişinizi kariyer@pardus.org.tr adresine göndermeniz gerekli.

TÜBİTAK UEKAE çalışanı olarak Gebze'deki Pardus ekibinde görev almak isteyenler adaylar, Pardus web sitesindeki [kariyer](#) kısmından gerekli bilgiye ulaşabilirler.

Google Summer of Code programı için uluslararası buluşma



Pardus Google
Summer of Code 2010
Öğrencileri Belirlendi!

Bu yıl üçüncü kez Google'ın, yazılım geliştirmeye uğraşan öğrencileri açık kaynak dünyasına kazandırmayı amaçladığı küresel Google Summer of Code (GSoC) programında yer almayı başaran Pardus, programa sunacağı projeler ile projelerde birlikte çalışacağı öğrencileri belirledi.

GSoC programına katılacak beş proje için Türkiye'den bir, Hindistan'dan ise dört geliştirici bir araya geliyor. Projelere **Pardus geliştiricileri, Onur Küçük, Renan Çakırek, Mete Alpaslan Katırcıoğlu, Bahadır Kandemir ve Semen Cirit** mentorluk edecekler. Projeler için çalışacak olan **Ahmet Alp Balkan, Saravanan Sundaramoorthy,**

Jain Basil Aliyas, Sukhbir Singh ve **Sarath Lakshman,** GSoC takvimine göre 24 Mayıs tarihine kadar kendilerine eşlik eden mentorları ile projelerin detayları ve çalışma yöntemleri hakkında görüş alışverişinde bulunacak, projeleri için gerekli kaynakları paylaşacaklar. Geliştiriciler daha sonra kod yazmaya, bir başka deyişle projelerini hazırlamaya başlayacaklar. 16 Ağustos tarihinde projelerin tamamlanmasını takiben, 23 Ağustos tarihinde hazırlanan projelerin başarılı olup olmadığı açıklanacak.

GSoC programı için belirlenen projelerin isimleri ve geliştiricileri ise şöyle:

- **Ahmet Alp Balkan:** Pardus için WUBI benzeri Microsoft Windows kurulum aracı.
- **Saravanan Sundaramoorthy:** Pardus sistem yedekleme ve ISO oluşturma aracı.
- **Jain Basil Aliyas:** KDE4 ayarları için yedekleme ve izleme aracı.
- **Sukhbir Singh:** Pardus için test altyapısı gerçekleştirilmesi.
- **Sarath Lakshman:** Pardus Live CD üzerinden kurulum desteği.

Pardus üçüncü kez genç geliştiricilere GSoC yolunu açıyor
GSoC'a bu yıl 365 başvuru gerçekleşti ve bunların arasından 150

danışman organizasyon seçilerek programa kabul edildi. Altıncısı düzenlenen programa Pardus üçüncü kez katılıyor.

GSoC, daha fazla açık kaynaklı kod elde edilmesi ve bu kodların herkesin yararına kullanıma açılmasının yanı sıra, genç geliştiricilerin yaz boyunca kendilerini geliştirmeleri ve açık kaynağa katkı vermeye teşvik edilmeleri açısından büyük önem taşıyor. Proje başlangıcından bu yana 300'ün üzerinde açık kaynak projesiyle binlerce öğrenciyi bir araya getirdi ve bu öğrencilerle birlikte milyonlarca satır kod da açık kaynak dünyasına kazandırıldı.

GSoC Programı, Pardus projesinde fikirleri ile yer alacak genç yazılım geliştiricilere kariyerleri açısından önemli fırsatlar ve benzersiz bir deneyim sunacak. Google, programda belirlediği birçok projeyi üç ay süresince maddi olarak destekliyor. Geçtiğimiz yıl organizasyon karşılığında öğrencilere 4500 dolar tutarında fon sağlandı.

GSoC 2010 hakkında ayrıntılı bilgiye Google Summer of Code 2010 sitesinden, Pardus'un GSoC proje fikirlerine ise [buradan](#) ulaşmak mümkün.

Özgür Yazılım Kara Listeye



Üyeleri arasında **BSA**, **Amerika Sinema Birliği (MPAA)** ve **Amerika Kayıt Endüstrisi Birliği (RIAA)** gibi lobi kurumlarını barındıran **Uluslararası Fikri Haklar Birliği (IIPA)**, Amerikan hükümetine, özgür yazılım kullanan ülkelerin telif hakkı düşmanlarının yer aldığı kara listeye eklenmesini tavsiye etti!

IIPA'nın hazırladığı ve Amerikan hükümetine sunduğu 498 sayfalık raporda, IIPA'nın tespitlerine göre fikri hakları çok az koruyan ya da korumayan ülkelerin listesi bulunuyor ve bu ülkelerin özel gözetim altına

alınması isteniyor. Raporda, özgür yazılımlar karşısında kimi devletlerin gösterdiği olumlu tutumların, fikri hakları zedelediği savunuluyor!

Amerikan hükümeti, her yıl lisans ve patent haklarını korumakta yetersiz gördüğü ülkeleri gösteren "**Özel 301**" raporunu yayınlıyor. Listeye giren ülkelere karşı ticari yaptırımlara sebep olan ve bu ülkeler üzerinde baskı kurmak amacını taşıyan liste, genellikle Çin ve Rusya'yı hedef alıyor. Çin ve Rusya, İnternet korsanlığına, korsan ve sahte ürünlere karşı önlem almamakla suçlanıyor.

Geçen sene ABD dış ticaret müsteşarı **Ron KIRK**'in girişimleriyle, Kanada bu yıl ilk defa bu kara listeye alındı. Buna sebep olarak ise Kanada'nın **Dünya Fikri Haklar Kurumu**'nun (WIPO) 1996'da çıkardığı İnternet üzerindeki telifleri korumaya yönelik kararları uygulamaması gösterildi.

Amaç, korku salmak

Eğer IIPA bu kafada devam ederse Brezilya, Hindistan, Endonezya, Filipinler, Tayland ve Vietnam gibi ülkeler de bu yılın kara listesine girecekler! Buna gerekçe olarak, açık kaynak yazılımların kullanımının yazılım sanayiini zayıflatacağı, "şirketler için yapay bir tercih" oluşturup yazılım sanayisinin "uzun dönemli rekabeti azaltacağını" iddia ediliyor.

Özgür Yazılım Vakfı'na (FSF) göre, **Yeni Binyıl Dijital Telif Hakkı Yasası (DMCA)**, özgür yazılımın gelişimini engelleyen ve ticarete olumsuz etki yaratan bir yasaya dönüştü. Özgür Yazılım Vakfı, Washington yönetimine Amerikan anayasasındaki demokratik ilkeler doğrultusunda, özgür yazılım pazarını desteklemesi ve 301 raporunu "pazarlık aracı" olarak kullanmayı bırakması çağrısında bulunuyor.

Kaynak: [The H Online](#)

Speed Dreams Pistlerde



Speed Dreams geliştiricileri geçtiğimiz günlerde, 3D motor sporları oyununun 1.4.0 sürümünü duyurdu. Speed Dreams, Özgürlükçin'de de tanıttığımız Torcs'dan türetilmiş, aynı kaynak kodu kullanan bir araba yarışı simülatörü. Torcs'dan aldığı bayrağı daha ilerilere taşıyan Speed Dreams, pek çok yeni özellik, araba, yarış pisti ve gelişmiş bir yapay zekâya sahip.

Speed Dreams'in bu yeni sürümü; görsel açıdan geliştirilmiş menüler, üç yeni yol, güncellenmiş bir TRB1 araba seti ve Super Cars, 36 GP ve LS-GT1 isimlerinde, iyi ayarlanmış ve dengeli üç yeni araba seti ile birlikte geliyor.

Speed Dreams'in bu sürümü hakkında daha fazla bilgi 1.4.0 sürüm notlarında bulunabilir. GNU GPL lisansı ile yayınlanan Speed Dreams'in Pardus için paketleme çalışmaları devam ediyor.

Kaynak: [H-Online](#)

Krita'da Önemli Değişiklikler



Sadece özgür yazılım dünyasında değil, tüm yazılım dünyasında geliştirdiğiniz bir uygulamaya eklediğiniz yeni özellikler hoş karşılanır. Ama var olan bir özelliği yok etmek isterseniz, zor bir işe kalkıştığınızı bilmeniz gerekir.

Krita geliştiricileri, kullanıcıların çoğunu sevindirirken belli bir kesimini üzen bir karara vardı. Bu karar Krita'nın fotoğrafçılıkla ilgili olan özelliklerinin gelişiminin durdurulması ve Krita'nın başarılı bir çizim yazılımı olması için gerekenlere yoğunlaşılmasıydı.

Krita geliştiricileri, uzun zamandır ortada olan bir gerçeği kabul ederek GIMP, DigiKam, Picasa, Hugin ve diğer başarılı fotoğrafçılıkla ilgili yazılımlara rakip olabilecek yeni bir fotoğraf düzenleme yazılımı daha geliştirmemeyi seçtiler. Gelişimine başladığı günden itibaren GNU/Linux üzerinde tüm çizim ve boyama ihtiyaçlarına cevap veren, yüksek seviyeli bir grafik aracı olma yolundaki Krita, fotoğrafla ilgili özelliklerini geride bırakarak en iyi ve en faydalı olduğu alanda, çizim programı olarak yoluna devam edecek.

Fotoğraf desteğinin kaldırılması, çizim ve boyamayla doğrudan ilgili olmayan araç ve özelliklerin ortadan kalmasına neden olacak. Kaldırılacak özellikler arasında, ton haritalama, HDR, mercek düzeltme aracı, gürültü giderme ve bazı sanatsal filtreler yer alıyor.

Kısacası, özgür yazılım dünyasında geliştiricilerin bazen cesur davranması ve sadece yeni özellikler eklemesi değil, bazı özelliklerden de kurtulması gerekiyor.

Kaynak: [Cyrille BERGER'in Blog'u](#)

OpenGL 4.0'ın Yenilikleri



Khronos Grubu, OpenGL 4.0'ı açık kaynaklı grafik uygulamalarının 2 boyutlu ve 3 boyutlu özelleştirmelerini duyurdu. Duyuru, San Francisco'da Oyun Geliştiricileri Konferansı sırasında yapıldı. Yeni özelliklerin başında, OpenGL Shading Language için GLSL 4.0 güncellemesi geliyor.

OpenGL.40'da yapılan en önemli yeniliklerden biri de mozaik iş desteği oldu. GPU (Grafik İşlem Birimi), yeni eklenen iki kademe sayesinde CPU'dan (Merkezi İşlem Birimi) geometri mozaikliğini devralacak. Gölgeleme için eklenen alt programlar sayesinde artık gölgeleme programlarında biraz daha esneklik sağlanacak. Nesne Örnek Alıcısı ile doku durumu ve doku verisinin birbirinden ayrılabilmesi sağlandı. Böylece, aynı durum şartları farklı doku verisine uygulanabilecek ya da tam tersi, aynı doku verisi farklı durumlara uygulanabilecek.

OpenGL 4.0'ın çapraz platformlar ve paralel programlama için olan OpenCL (Açık Programlama Dili) ile etkileşimi de geliştirildi. Khronos Grubu, aynı zamanda önceki nesil GPU donanımına, mümkün olduğunca OpenGL 4.0 özellikleri kazandırabilmek için yeni OpenGL 3.3 belirtimini de yayınladı.

"Fermi"den OpenGL 4.0 Desteği
OpenGL özellikleri, Khronos Grup'un içinde yer alan Yapı İnceleme Kurulu (OpenGL ARB) ile aralarında NVIDIA ve AMD çalışanlarının da bulunduğu, kâr amacı gütmeyen bir

konsorsiyumca geliştiriliyor. NVIDIA'dan yapılan açıklamaya göre, gelecek olan yeni nesil "Fermi" grafik işlemcisi, OpenGL 4.0'ı tamamiyle destekleyecek.

Kısaca, OpenGL 4.0, oyun geliştiricilerine GPU gücünü daha iyi kullanabilme konusunda daha çok seçenek sunacak. GPU gücünün daha iyi kullanılması ise, daha iyi görüntüyle oyunların bilgisayarlarımızda daha rahat çalışması demek.

Kaynak: [H Online](#)

2010 Django Buluşması Berlin'de!



European DjangoCon isminden de anlaşıldığı gibi, bir Django konferansı. Bu buluşmanın ilk amacı, uluslararası Django topluluğunu bir araya getirmek.

Diğer amaçları arasındaysa, "çeşitli işletmelerdeki Django kullanımı, yapılan toplantıları, panelleri ve konuşmaları geniş bir alana yayarak geliştirmek" var.

Bu sene ikincisi düzenlenen Avrupa DjangoCon etkinliği, 24-26 Mayıs tarihleri arasında, Almanya'nın başkenti Berlin'de, Palisa.de Kongre

Merkezi'nde yapılacak, 27-28 Mayıs arasında da sprints etkinlikleri (zaman kısıtlı proje kodu geliştirme atölyeleri) yer alacak.

Topluluk destekli konferans, web geliştirme ve Django Python Web çerçevesi konusunda birçok konuşma ve eğitimi içeriyor.

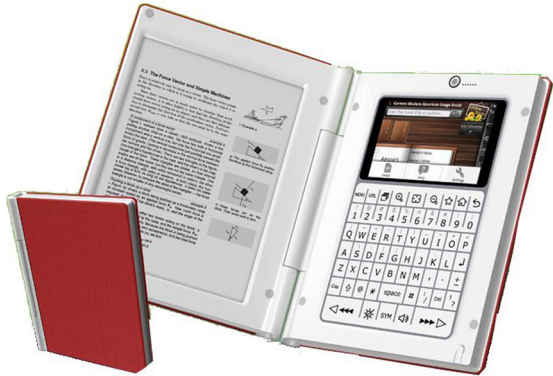
Konferans organizatörleri, sunumlar için son başvuru tarihini 1 Mayıs'a kadar uzattılar ve organizatörler hâlâ katılımcılardan yapılacak konuşmalar için sunum istiyorlar. Sunum takvimi 7 Mayıs tarihinde duyurulacak.

Topluluğumuzdaki Django kullanımı ve sevgisini göz önüne alırsak, bu konferansı ve sonrasındaki gelişmeleri takip edebiliyor olmak kullanıcıları daha çok sevindirecek.

Yıllık etkinliğin yer ve programıyla ilgili ayrıntılı bilgilere DjangoCon web sitesinden ulaşabilirsiniz. Kayıt işlemi, öğrenci indirim ve "ilk gelen kapar" kuralıyla devam ediyor. DjangoCon Avrupa 2010 etkinliği Alman Django Komitesi sponsorluğunda organize edilerek düzenleniyor.

Kaynak: H-Online

Çift Ekranlı e-Kitap Okuyucusu



1Cross Tech isimli teknoloji firması, çift ekran Android MID ve e-okuyucu aygıtını tanıttı. **MIDHybrid** isimli bu e-kitap okuyucusu, aynı zamanda bir 3G telefon, WiFi erişim cihazı, Bluetooth destekli ve bir webcam'e sahip bir cihaz!

MIDHybrid; 1Cross Tech'in sahneye ilk çıkan ürünü gibi görünüyor, ayrıca ürünün web sitesinde aygıt hakkında basit bilgiler ve bazı fotoğraflar sunuluyor. Daha fazla bilgi, CeBIT Fuarı'nda çekilen ve YouTube'a konan

uzun videodan bulunabilir. Aygıt, özellikleri şimdilik saklanan Marvell Semiconductor işlemcisiyle çalışıyor.

MIDHybrid sağda renkli Android ekranı ve solda e-mürekkep teknolojiyi Vizple gri tonlamalı ekranla birlikte, bir kitabın sayfaları gibi açılan bir kullanım sunuyor. 153 x 120 x 17 mm boyutlarındaki 1Cross Tech tasarımı, her iki yöne katlanabiliyor.

MIDHybrid 32 GB SD kart, 802.11b/g Wifi, Bluetooth ve

bir 3G TD-SCDMA kartla geliyor. İlave özellikler, seçenekli olarak GPS, hoparlörler, kulaklık girişi, bir mikrofon ve Host desteği ile birlikte bir microUSB giriş içeriyor. 2.000 MAH bataryasıyla iki-üç hafta boyunca şarja gerek duymayan e-okuyucu kullanımı sunuyor.

Ne dersiniz, Linux cephesinden iPad'e güçlü ve yaratıcı tasarıma sahip rakipler gelmeye başladı mı? :)

Kaynak: Linux Devices

Open Clip Art Library 2.0



Open Clip Art Kütüphanesi, 2004 yılının başında **Bryce HARRINGTON** ve **Fabricatorz** firmasından **Jon PHILLIPS** tarafından ilk defa ortaya atılan bir fikirdi.

Open Clip Art Kütüphanesi mütevazı bir başlangıçtan sonra bugün, dünyanın dört bir

tarafından **1.200'ün üzerinde sanatçı** tarafından oluşturulmuş, **24.000'inden fazla vektörel çizime** sahip bir kütüphane haline geldi.

Open Clip Art Kütüphanesi her geçen gün gelişen, kullanışlı ve güçlü bir buluşma noktası. Siteye yüklenen her çalışma, [Creative Commons](#) ya da "Kamu Malı" olarak lisanslanıyor. Bu sayede isteyen herkes, özgür ve ticari işler dâhil herhangi bir amaç için, burada yer alan tüm SVG biçimindeki vektörel çizimler, katmanları da açık bir şekilde, herkesin kullanımına açık.

SVG kısaca, kapalı kaynak ürünü olan Flash teknolojisinin yapabildiği birçok şeyi yapabilecek bir vektörel (çizgisel) grafik biçimi. Tabii ki en önemli farkı açık standartlara dayanması, şeffaflık gibi özellikleri içertmesi ve XML tabanlı olması.

Kütüphane'deki Yenilikler

Open Clip Art Kütüphanesi yeni bir özellik olarak, [küçük resimlerle](#) daha kolay dolaşılabilir bir koleksiyon sunuyor. Bu sayede uygun resimleri araştırıp indirmek daha kolaylaştı. Sitenin yeni düzeni ve [Andy FITZSIMON](#) tarafından yapılmış yeni teması, oluşturulan resimlerin portala

yüklenmesinin önemini daha fazla vurguluyor ve kullanıcı etkileşimini artırıyor.

Open Clip Art Kütüphanesi, 2.0 sürümüyle birlikte yeni bir altyapıya da kavuştu. [Aiki Framework](#)'e taşınan portal, eskisinden daha hızlı ve kolay yönetilebilir bir yapıya kavuşmuş.

Daha ne bekliyorsunuz? Open Clip Art Kütüphanesi'ne özgün çalışmalarınızı buraya yükleyerek yüz binlerce kullanıcıyla çalışmalarınızı paylaşabilirsiniz.!

Kaynak: [Worldlable](#)

Android Pazar Payını Artırıyor



Son aylarda, Google'ın açık kaynak kodlu mobil işletim sistemi Android, pazar payı kazanmaya devam ediyor. Piyasa araştırma şirketi [ComScore](#) tarafından yayınlanan, Şubat 2010

Mobil Abone Pazar Payı Raporu'na göre, Google'ın akıllı telefon platformu Android, Kasım 2009'dan beri tek başına %5'den fazla büyüdü. Bu olumlu gidişat, Android'in ABD pazar payının %9'unu elinde tuttuğu anlamına geliyor. ComScore'un raporu genel olarak 13 yaş ve üzeri mobil abonelerin kullandıkları cihaz üreticisi ve platformlarına yönelik.

2009'un kasım ayından bu yılın şubat ayına %1,3 artışla Blackberry üreticisi Research in Motion ([RIM](#)), hali hazırda

%42,1 pazar payını elinde tutuyor. Apple'ın iPhone akıllı telefonu %0,1'lik bir düşüşle %25,4 payla ikinci sırada yer alırken, %4'lük bir puan kaybıyla %15,1 pazar payına sahip Microsoft Windows Mobile platformu üçüncü sırada. Android %9'luk bir pazar payıyla Palm'in üzerinde yer çıkarken, Palm %1,8 puan kaybediyor ve %5,4 ile beşinci sıraya düşüyor.

ComScore'un sonuçlarına göre, 234 milyon Amerikalı mobil abone var ve bunların 45,4 milyonu kendi akıllı

telefonunu kullanıyor. Araştırmanın ortaya koyduğu bir diğer ilginç veri, bu pazarın sadece üç ayda %21,7 büyümüş olması. ComScore'un belirttiğine göre, telefon üreticileri arasında Motorola 1.,9 puan kaybederken, pazardaki toplamda %22,3'lük payıyla, en güçlü ABD mobil orijinal donanım üreticisi konumunda. LG %21,7 ile ikinci sıraya yerleşirken, onu %21,4 ile Samsung takip ediyor

Kaynak: [The H Open](#)

Bugzilla'yı kullanmak:

Hatamla Sev Beni



Pardus'ta bir hatayla (bug) karşılaştınca yapmanız gereken, işi böcek avcılarına bırakmak. Bu yazımızda "böcek avlamanın" inceliklerini anlatıyoruz.



Pardus Hata Takip Sistemi ya da diğer adıyla Bugzilla, dağıtım geliştirme sürecinin en önemli parçalarından biridir. Bu yazıda da Bugzilla üzerine konuşacağız. O halde başlayalım.

Pardus Hata Takip Sistemi Nedir?

Hata kayıt sistemi ya da Bugzilla; kısaca açıklayacak olursak, Pardus ve Pardus tarafından geliştirilen uygulamaların (PiSi, Kaptan, YALI v.s.) kullanımı sırasında ortaya çıkan hataların kullanıcılar tarafından geliştiricilere bildirilmesini sağlayan web tabanlı bir arayüzdür.

Hata Kaydı Yapmadan Önce

Bir hata kaydı yapmadan önce problemin gerçekten bir hata olduğundan emin olmamız gerekir. Örneğin, "Yazıcıım çalışmıyor?" diyerek bir hata girmeden önce yazıcınızın kartuşunun dolu olduğuna ve kablolarının doğru takıldığına emin olduktan sonra sisteminizin güncel olduğunu kontrol edin. Sisteminizde güncellemediğiniz paket varsa onları güncelledikten sonra, yaşadığınız sorunun tekrarlanıp tekrarlanmadığına bakın.

Yaşadığınız sorunun hata olduğundan emin olduktan sonra hata girmeye hazırsınız demektir. Hatanızı girmeden önce aynı hatayla ilgili daha önce hata girişi olmadığından da mutlaka emin olmalısınız.

Hataları doğru şekilde anlayabilmek ve kısa zamanda çözebilmek için ek dosyalar geliştiriciler için önemlidir. Hatanızın geliştiriciler tarafından doğru anlaşılması ve en kısa sürede çözülebilmesi için, sorun yaşadığınız uygulamanın verdiği hatayı geliştiricilere bildirmeniz gerekmektedir.

Programlarda yaşanan hatalar için, Konsol'a girerek programismi -debug (Ör: pidgin -debug) çıktısını rapora ekleyin.

Sistem açılışındaki hatalar için, Eğer sisteminiz konsol çıktıları vererek açılmıyorsa konsol ekranında yazan çıktılarından son birkaç satırını rapora ekleyin.

Pardus'a özel uygulamaların hataları için, Burada kullanacağımız komutları çalıştırabilmek için yönetici haklarına sahip olmamız gerekmektedir. su - komutunu verdikten sonra root şifresini yazarak yönetici haklarına sahip olabilirsiniz. **ÇOMAR**, Birçok durum için ÇOMAR'ın log dosyası yardımcı olabilir. cat /var/log/comar3/trace.log > /home/kullanici_adi/comar.txt komutunu verdikten sonra log dosyası ev dizininize oluşacaktır.

Ağ yöneticisi, Ağ aygıtınızı öğrenmek için, lspci -nn > /home/kullanici_adi/lspci.txt Ethernet'e özel problemler için, ifconfig -a > /home/kullanici_adi/ifconfig.txt Wireless'a özel problemler için, iwconfig > /home/kullanici_adi/iwconfig.txt çıktılarını hata kaydına ekleyebilirsiniz.



Disk yöneticisi, `fdisk -l > /home/kullanici_adi/fdisk.txt` ve `cat /etc/fstab > /home/kullanici_adi/fstab.txt` çıktılarını hata kaydına ekleyebilirsiniz. Servis yöneticisi, `service -N > /home/kullanici_adi/service.txt` çıktılarını hata kaydına ekleyebilirsiniz.

Açılış Yöneticisi, `cat /boot/grub/grub.conf > /home/kullanici_adi/grub.txt` çıktılarını hata kaydına ekleyebilirsiniz.

Güvenlik Duvarı Yöneticisi, `service -N > /home/kullanici_adi/service.txt` ve `iptables > /home/kullanici_adi/iptables.txt` çıktılarını hata kaydına ekleyebilirsiniz.

Hata Kaydı Girme

Bugzilla, 3.4.6 sürümüyle birlikte yepyeni bir görünüm kazandı. Hata raporlama süreci tamamen görselleşti. Yazı okumaktan sıkılanlar ya da okuduğundan bir şey anlayamayan kullanıcılar için güzel bir gelişme oldu.

Gelelim fasulyenin faydalarına... Pardus Hata Kayıt Sistemi'ne giriş yaptığımız sonra, simgesinden de anlaşıldığı üzere "Yeni bir hata raporla" seçeneğine tıklıyoruz. Yaşadığımız sorun ya da aldığımız hata hangi ürünle ilgiliyse o ürünü seçiyoruz. Eğer, hatanızın hangi ürünle ilgili olduğunu bilmiyorsanız, Genel bölümüne ekleyebilirsiniz. Bir sonraki ekran her üründe farklı seçeneklerle karşımıza çıkıyor.

Bugzilla - Hata Gir: Paketler / Packages

[Ana Sayfa](#) | [Yeni](#) | [Ara](#) | [Bul](#) | [Raporlar](#) | [İstediklerim](#) | [Tercihler](#) | [Yardım](#) | [Oturumu Kapat](#) eguven95@gmail.com

Bir hata raporlamadan önce, lütfen [hata yazım kılavuzunu](#) okuyun, [en sık raporlanan hatalar](#) listesini kontrol edin ve lütfen eski hatalara [göz gezdirin](#).

Gelişmiş Alanları Göster

Ürün	Paketler / Packages
Bileşen	<div style="border: 1px solid black; padding: 5px;"> <p>00 Yeni paket / New package</p> <p>01 Listede yok / Not in List</p> <p>3cp4218</p> <p>3ddesktop</p> <p>3gp-converter</p> <p>4suite</p> <p>855resolution</p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>Bileşen Açıklaması</p> <p>Paket geliştiricileri tarafından yeni hazırlanan paketlerin depoya eklenmek üzere bildirildiği bileşendir</p> <p>This component is used to report newly-packaged applications (and the like) for inclusion in the repositories</p> </div> <p>Hangi bileşeni seçeceğinizden emin değilseniz bileşen açıklamaları bağlantısını takip edin.</p>
Mimari:	<input type="text" value="x86"/>
Sürüm	<div style="border: 1px solid black; padding: 5px;"> <p>---</p> <p>Pardus 2008</p> <p>Pardus 2009</p> <p>Pardus Kurumsal 2</p> </div>
Özet	<div style="border: 1px solid black; padding: 5px; min-height: 40px;"> <p>Olabildiğince ipucu kelime kullanmaya özen göstererek yaşadığınız problemi özetleyen bir cümle girin.</p> <p>Kötü Örnek: KDE çöktü.</p> <p>İyi Örnek: Dolphin uygulamasını açtığımda eğer dizin içerisinde çok fazla dosya varsa KDE çöküyor.</p> </div>

Mimari seçeneği, eğer kullandığınız Pardus 64 bit ise x86_64 seçeneğini seçiyorsunuz. Eğer 64 bit kullanmıyorsanız Mimari seçeneğine elleşmeyin :).

Sürüm: Kullandığınız sürüme göre değişecektir. Güncel bir sistem kullanıyor olsanız bile sürümünüzü belirtin derim. Hata kaydı eskiyebilir ve giderilmesi için gereken zaman zarfı içinde yeni bir sürüm çıkabilir.

Bileşen: Bir önceki sayfadan karşılaştığımız hatayı içeren ürünün alt kategorileri diyebiliriz. Örnek vermek gerekirse, paket geliştiricileri tarafından yeni hazırlanan paketlerin depoya eklenmek üzere bildirmek için bir önceki sayfadan yani ürünler sayfasından Paketler seçeneğini seçtikten sonra, Bileşen bölümünden Yeni paket seçeneğini işaretlemeliyiz.

Özet: Yaşadığınız problemi düzgün bir şekilde buraya özetleyebilirsiniz.

Kötü Örnek: KDE çöktü.

İyi Örnek: Dolphin uygulamasını açtığımda eğer dizin içerisinde çok fazla dosya varsa KDE çöküyor.

Ayrıntılar

Özeti daha ayrıntılı açıklayın. Bu açıklamanın hataya özgü olmasına lütfen özen gösterin.

Kötü Örnek: KDE çöktü. Hep böyle oluyor :S
İyi Örnek: Dolphin uygulamasını açtığımda eğer dizin içerisinde çok fazla dosya varsa, Dolphin uygulaması dosyaları görüntüleyemiyor ve donuyor. Bir süre sonra KDE çöküyor.

Tekrarlanabilirlik

Tekrarlamak için gerekli adımlar

- 1.
- 2.
- 3.

Hatayı tekrarlamamız için gerekli olan adımları detaylandırın.

Güncel Sonuç

Yukardaki adımları gerçekleştirdikten sonra nasıl bir sonuç ile karşılaşıyorsunuz?

Beklenen Sonuç

Yukardaki adımları gerçekleştirdikten sonra nasıl bir sonuç ile karşılaşmayı bekliyordunuz?

Ek Açıklamalar

Hata için gerekli olduğunu düşündüğünüz her türlü bilgiyi buraya ekleyebilirsiniz. Bu bilgi bilgisayarınız ile ilgili bir donanım bilgisi veya herhangi bir ayar olabilir.
 Eğer bu açıklama birkaç satırdan fazla ise, lütfen **dosya ekle** bölümünden ilgili dosyayı ekleyiniz.

Önem

Karşılaştığınız problemin sizin için değil **dağıtım için** ne kadar önemli olduğunu belirtin.

Eklentiler

Dosya Ekle

Hata raporunu gönder

Ayrıntılar: Özette yazdığınız açıklamanın daha ayrıntılı ve hatada yaşadığınız sorunları detaylı bir şekilde anlatabilirsiniz.

Kötü Örnek: KDE çöktü. Hep böyle oluyor :{.

İyi Örnek: Dolphin uygulamasını açtığımda eğer dizin içerisinde çok fazla dosya varsa, Dolphin uygulaması dosyaları görüntüleyemiyor ve donuyor. Bir süre sonra KDE çöküyor.

Tekrarlanabilirlik: Hatanızın hangi sıklık ve aralıklarla tekrarlandığını belirteceğiniz bölümdür.

Tekrarlamak için gerekli adımlar: Hatanın tekrarlanması için gerekli olan adımların yazılacağı bölüm.

Güncel Sonuç: Tekrarlamak için gerekli adımları gerçekleştirirken nasıl bir sonuçla karşılaştığımızı yazacağız.

Beklenen Sonuç: Tekrarlamak için gerekli adımları gerçekleştirdikten sonra nasıl bir sonuç beklediğimizi yazacağız.

Ek Açıklamalar: Hata için gerekli olduğunu düşündüğünüz bilgileri buraya yazacağız.

Önem: Karşılaştığınız hatanın dağıtım için ne kadar önemli olduğu belirteceğiniz kısımdır.

Eklentiler: Hatayla ilgili olarak log kayıtlarını ya da uzun olduğunu düşündüğünüz metinleri dosya olarak bu bölümden ekleyebilirsiniz.

Bütün bu alanları doldurduktan sonra, Hata raporunu gönder seçeneğine tıklayarak hata kaydımızı gerçekleştirebiliriz.

Hatanıza atanan geliştiricinin size göstereceği yolu izleyerek istediği bilgileri Ek Yorumlar bölümünden gönderebilirsiniz.



nasil

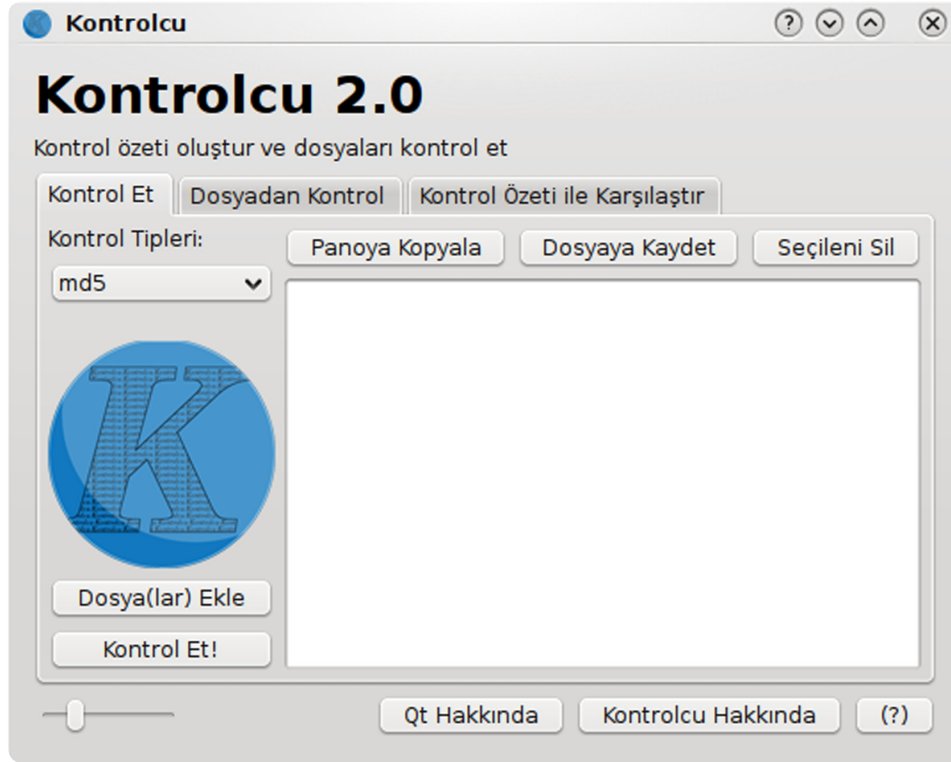
Taha Dođan GÜNEŞ
tdgunes@gmail.com



Darren Hester (CC BY-NC)

Kurulum CD'lerinizi Kontrol Edin

Kontrolcu ile kontrol toplamları oluşturabilir, Pardus Kurulum CD'lerinizi kolayca denetleyebilirsiniz.



Eskiden Pardus CD kalıplarını indirirken hep karşıma md5 ve sha1 denen numaralar çıkıyordu. Ne olduğu hakkında açıkçası hiçbir fikrim yoktu. Sonra öğrendiğime göre, eğer ben Pardus CD kalıbı indirirsem sonra onu kontrol edersem dosyanın bozulup, bozulmadığını anlayabilecektim.

Bu yöntemi uygulamak için konsol kullanmak gerekiyor. Siz konsoldan çıkarmaya çalıştığınızda sizi kontrolün ne zaman biteceği hakkında hiçbir bilgilendirme yok ve konsoldan korkan bir sürü kullanıcı olduğunu da biliyoruz. Kontrolcu işte bu sebeplerden dolayı ortaya çıktı diyebilirim.

Ne zaman, Nasıl ?

Kontrolcu, 2009'un yazında Pardus IRC kanalında PyQt4 ile nasıl md5 kontrol toplama çıkartırım diye bazı sorular sormamla başladı diyebilirim. Osman KARAGÖZ bana yardımcı oldu ve sonra onunla beraber Kontrolcu'ye başladık.

Gelişim aşaması çok uzun bir süreç oldu. Çünkü bilgimiz çok azdı. SVN'nin nasıl kullanıldığı hakkında benim hiçbir fikrim yoktu. O günden bugüne bakınca inanılmaz işler yaptığımızı düşünüyorum. Bu sadece Kontrolcu olarak değil, kendimize kattığı deneyim ölçülemez bile.

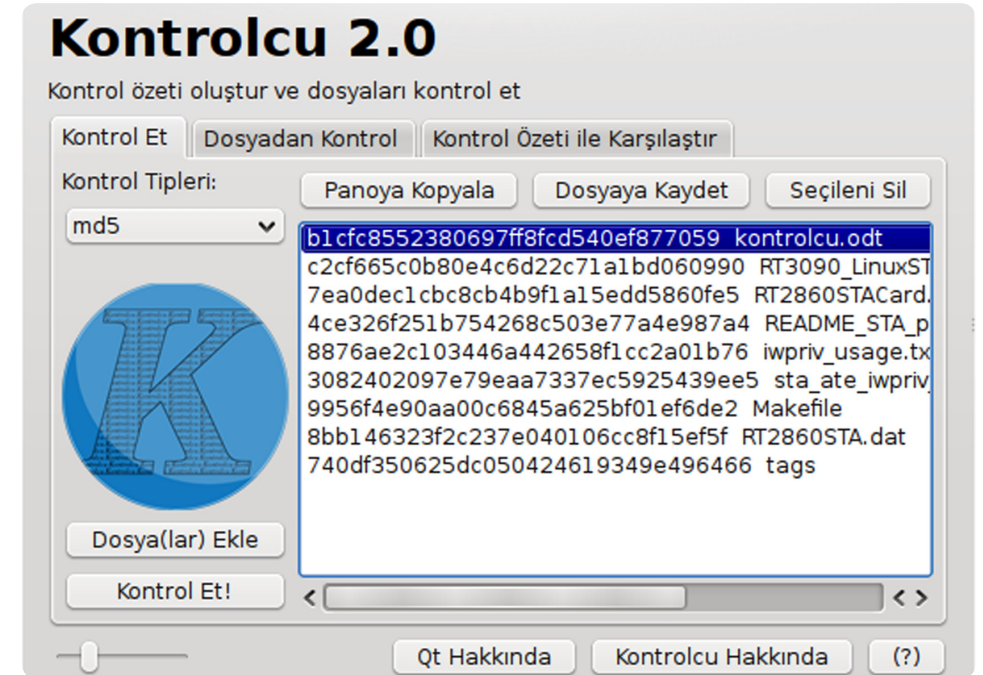
Nasıl Yüklerim?

Kontrolcu'nün her platformda çalışması için epey uğraşıldı. Şu an Linux, Windows ve Mac OS X'de güzelce çalışıyor. Pardus deposu için çalışmalar sürüyor. Hemen denemek isterseniz sırasıyla;

```
sudo pisi build http://kontrolcu.googlecode.com/svn/packages/linux/pisi/pspec.xml
```

```
sudo pisi it kontrolcu*.pisi
```

komutlarını uygulayarak Pardus'a kurulum yapabilirsiniz. Diğer platformlar için [sitemize](#) bakabilirsiniz.



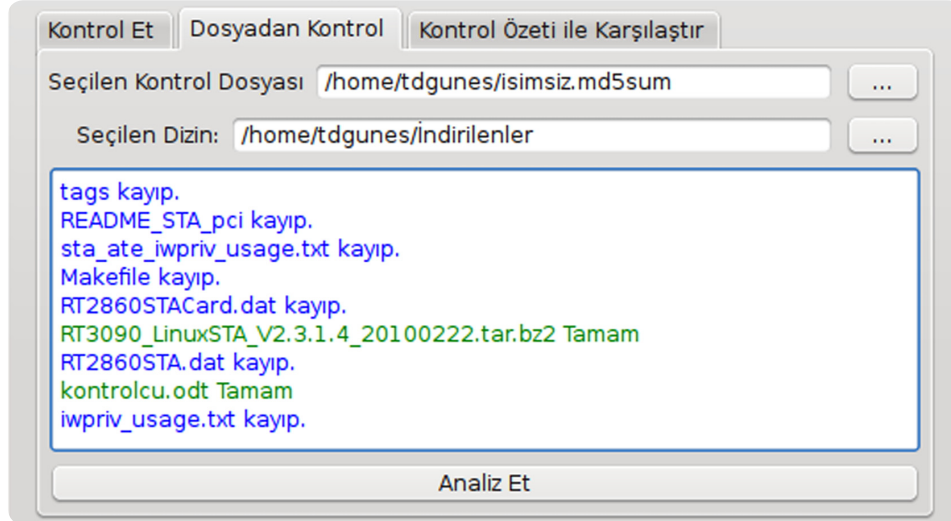
Tam olarak nedir ?

Kontrolcu, kontrol toplamları oluşturan bunları dosyaya yazabilen gelişmiş bir araçtır. Kendini KDE4'ün servis menüsüne yerleştirir, oradan kolayca erişebilmenizi sağlar. Hem konsol hem de arayüzde yükleme çubuğuyla ne zaman kontrolün biteceğini öğrenebilir, çoklu dosya kontrolü, sürükle-bırak desteği ile rutin yaptığınız işi kolaylaştırmayı hedeflemektedir.

Kontrolcu'nun arayüzü, üç bölümden oluşuyor. 'Kontrol Et', 'Dosyadan Kontrol' ve 'Kontrol Özeti ile Karşılaştır.'

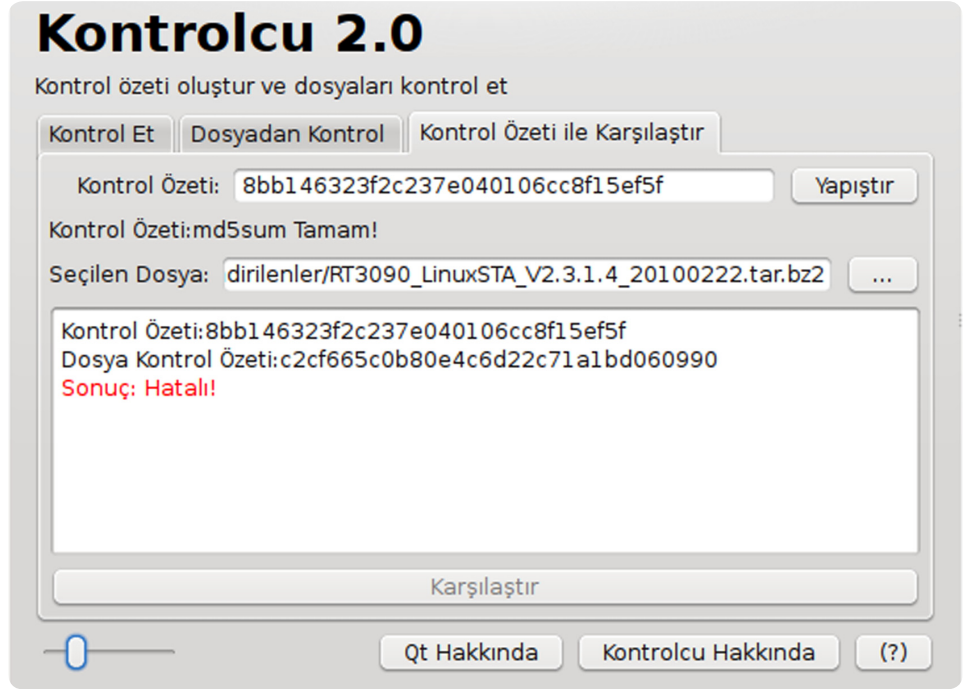
Kontrol et

Kontrol Et ile basitçe dosyalarınızı ekleyip (sürükle-bırak desteği ile dizin de ekleyebilirsiniz.) kontrol edebilirsiniz. Kontrol ettiğiniz dosyanın kontrol numarasını, 'Panoya Kopyala' dediğinizde panodan erişebilirsiniz. 'Dosyaya Kaydet' dediğinizde standartlara uygun bir kontrol dosyası oluşturacaktır. İsterseniz eklediğiniz dosyayı listeden de 'Seçilen Sil' diyerek atabilirsiniz.



Dosyadan Kontrol, çok pratik bir karşılaştırma bölümüdür. İlk önceden oluşturduğunuz bir kontrol dosyasını seçersiniz ve bu kontrol dosyasındaki toplamları oluşturduğunuz dosyaların bulunduğu dizini gösterirsiniz. Sizin için otomatik olarak karşılaştırır ve hangi dosyanın değişip, değişmediğini size gösterir.

Diyelim ki, Pardus CD kalıbı indirdiniz. Sitede verilen kontrol toplamıyla karşılaştırmak istiyorsanız bu bölüm sizin için. Sitedeki kontrol toplamını girin (Kontrol toplamını kopyaladıktan sonra yapıştır düğmesine basarsanız sizin için yapıştıracaktır.) ve indirdiğiniz kalıbı seçin, bırakın Kontrolcu sizin için karşılaştırsın.



Kontrolcu ile konsoldan neler yapabiliriz?

Çoğu kullanıcı konsol severdir. İşlerin pratik yapılması ve daha hızlı olması onlar için önemlidir. Böyle kullanıcılar için kontrolcu arayüzden yapılan her şeyi yapabiliyor. İsterseniz sonucu bir pencerede bile gösteriyor. Ayrıntılı konsol kullanım yardımı için 'kontrolcu -help' demeniz yeterli.

Son olarak:

Eğer bir hata bulursanız lütfen bizlere bildirmeyi unutmayın. Kontrolcu hakkındaki görüşlerinizi dört gözle bekliyoruz :).



Python'da Döngüler

Bir süredir ara verdiğimiz Atölye yazılarına güzel bir Python yazısıyla dönüyoruz. Haydi, kolları sıvayın!

Merhaba arkadaşlar. Bu derste bize döngü kurma olanağı veren range, for ve while deyimlerini göreceğiz.

İlk olarak döngü kavramını açıklayalım. Bir işlemin birden fazla, belli sayıda ya da sonsuza değin yapılmasını sağlayacağız. Mesela programımıza bir şifre koyduk. Kullanıcı programı çalıştırdı ve şifreyi yanlış girdi. Programımız burada kapanıyor ya da kullanıcının programı kullanabilmesi için kapatıp tekrar çalıştırması gerekiyor. Döngülerle kullanıcıya programı kapatıp yeniden başlatmadan ikinci bir veri girme şansı verebileceğiz.

Eğer aramızda matematikçiler varsa bilirler küçük Gauss'un hikayesini. Ben yine de anlatayım. Bir gün öğretmeni bir sebepten dolayı sınıfı bırakmak zorunda kalır. O sırada öğrencilerin oyalanması için onlara kendince uzun sürecek bir problem verir. 1'den 100'e kadar olan sayıları toplamalarını ister. Ancak işler beklediği gibi yürümez ve Gauss problemi gene aramızdaki matematikçilerin bileceği gibi $n(n+1)/2$ formülünü bularak çözer.

Varsayalım ki Gauss kadar dâhiyane bir çözüm bulamadık ama elimizde bir bilgisayarımız ve Python'umuz ve döngülerimiz var. Fazla uzatmadan başlayalım. Öğrendikçe anlayacaksınız.

Range Döngüsü

"Range" kelimesi Türkçe'de "aralık" anlamına gelir. Adından da anlaşılacağı gibi bu fonksiyon ile belli bir aralıktaki rakamlar ile işlem yapabileceğiz. Örneğin kabuğa:

```
>>>range(10)
```

komutunu verdiğimizde Python bize 10'a kadar olan (10 dâhil değil) sayıları;

```
cabanbilisim@tabanbilisim-pardus ~ $ python
Python 2.6.4 (r264:75706, Dec 5 2009, 21:10:59)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

şeklinde yazacaktır. Bu kodu kabukta rahatlıkla kullanabilirsiniz. Fakat bunu bir yere kaydedip çalıştırmak istediğinizde çalışmayacaktır. Bunun için kodumuzu biraz daha geliştirip;

```
print(list(range(10)))
```

şeklinde yazıyoruz. Burada "list" komutunu daha sonraki derslerimizde anlatacağız. Şimdilik verilen aralığı listelemeye yaradığını bilmeniz yeterli. Bir örnek daha verelim

```
print(list(range(2,11)))
```

Burada Python 2'den başlayıp 11'e kadar sayacaktır. Elbette 11 hariç. Şimdi biraz daha ilerletip yeni bir kod türetelim.

```
print (list(range(3,15,2)))
```

Burada 3 ile 15 aralığındaki rakamları ikişer ikişer sayacaktır. Temel kod yapısı şu şekildedir. Örnekler şu şekildedir.

```
>>>
>>>
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> print(list(range(2,11)))
[2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> print(list(range(3,15,2)))
[3, 5, 7, 9, 11, 13]
>>>
>>>
>>>
```

```
range(başlangıç_değeri,bitiş_değeri,artış_miktarı)
```

Range döngüsü bu kadar arkadaşlar. İnşallah anlamışsınızdır range döngüsünün yapısını :). Şimdi For döngüsüne geçelim.

For Döngüsü

Python'da belki en fazla kullanacağımız fonksiyondur. Şimdi "For" döngüsüyle ne

yapabileceğimize bakalım. Basit bir kodla başlayalım:

```
for i in range(1,10):
    print(i)
```

kodu çalıştırdığımızda bize

```
>>>
>>> for i in range(1,10):
...     print(i)
...
1
2
3
4
5
6
7
8
9
>>>
```

şekildeki gibi çıktı verecektir. 10'u yazmadı çünkü biz Python'a 10'a kadar saymasını söyledik.

Şimdi daha iyi anlaşılması için bir örnek daha yazalım.

```
kelime = "Tabanbilisim"

for in isim:
    print (harfler)
```

Bu kodları çalıştırdığımızda sayfanın sağ üstünde yer alan "Ekran Görüntüsü 4"teki gibi bir çıktı elde ederiz.

Peki, ama biz bu kodda Python'a ne dedik de böyle bir şey yaptı? "tabanbilisim" adlı değişkenin içindeki öğelerin her birini tek tek "harfler" olarak isimlendirdi. Sonra bu "harfler" değişkenini ekrana yaz komutunu verdik.

```
>>> kelime="TabanBilişim"
>>> for harfler in kelime:
...     print(harfler)
...
T
a
b
a
n
B
i
l
i
ş
i
m
```

(Ekran Görüntüsü 4)

Şimdi son bir tane daha for ile ilgili bir örnek yapalım.

```
toplam=0
for i in range (1,100):
    toplam = toplam + i
    print(toplam)
```

burada da Python bize 1'den 100'e kadar olan rakamların toplamını verecektir. Şimdi while döngüsüne geçelim.

While Döngüsü

"While", -iken anlamına gelmektedir. While döngüsü bir koşul sağlandığı sürece sonsuza dek devam eder. İlk önce basit bir örnekle başlayalım:

```
#!/usr/bin/env python3.0
a = 9
while a > 7:
    print ("dokuz yediden büyüktür")
```

kodunu inceleyim. Burada Python'a dedik ki "a = 9 olsun ve a değeri 7'den büyük olduğu sürece "dokuz yediden büyüktür" ifadesini ekrana bas."

Şimdi burada programı sınırlayıcı hiçbir şey yok "a" değeri hep 7'den büyük olacaktır. Dolayısıyla ekrana hep belirttiğimiz ifadeyi basacaktır. Şimdi while ile öğrendiklerimizi biraz daha oturtalım:

```
#!/usr/bin/env python3.0

while x=y:
    pass
daha genel şekilde belirtirsek:
while (koşulumuz):
    yapılacaklar
...
```

Örneğin devamlı tekrar edecek bir olay ele alalım. Mesela bir kronometre yapalım. Döngünün sonsuza dek devam etmesi için devamlı doğru olacak bir koşula ihtiyacımız var. Örneğin:

```
while 1==1:
...
...
```

Ancak hem kodun okunabilirliğini artırmak hem bazı geleneklere sadık kalmak hem de bilgisayarımızı devamlı kontrol etmekten kurtarmak için:

```
while True:
...
```

kalıbını kullanabiliriz. Kronometre örneğine geri dönersek:

```
#!/usr/bin/env python3.0

import time
sayac = 0
while True:
    time.sleep(1) //Bu satırlara şimdilik dikkat etmemize gerek yok, sadece
```

```
    bir saniye beklettiğini bilelim yeter.
    sayac += 1
    print(sayac)
```

Peki, bir döngüyü sona erdirmek istersek ne yapacağız? Örneğin while döngüsü ile 10'a kadar sayalım. Yapacağımız mantık çerçevesinde şudur:

sayı 10'dan küçük ise:
sayıyı bir arttır.
Bu basit işlemi:

```
#!/usr/bin/env python3.0

sayi=0

while sayi<10:
    sayi += 1
```

sayfanın sağındaki ekran görüntüsünde görüldüğü gibi ifade edebiliriz.

```
#!/usr/bin/env python3.0

sayi=0
```

```
while True:
    if sayi<10:
        sayi+=1
        print(sayi)
```

Yazımızı okura ufak bir alıştırmayı ekleyerek sona erdirelim:

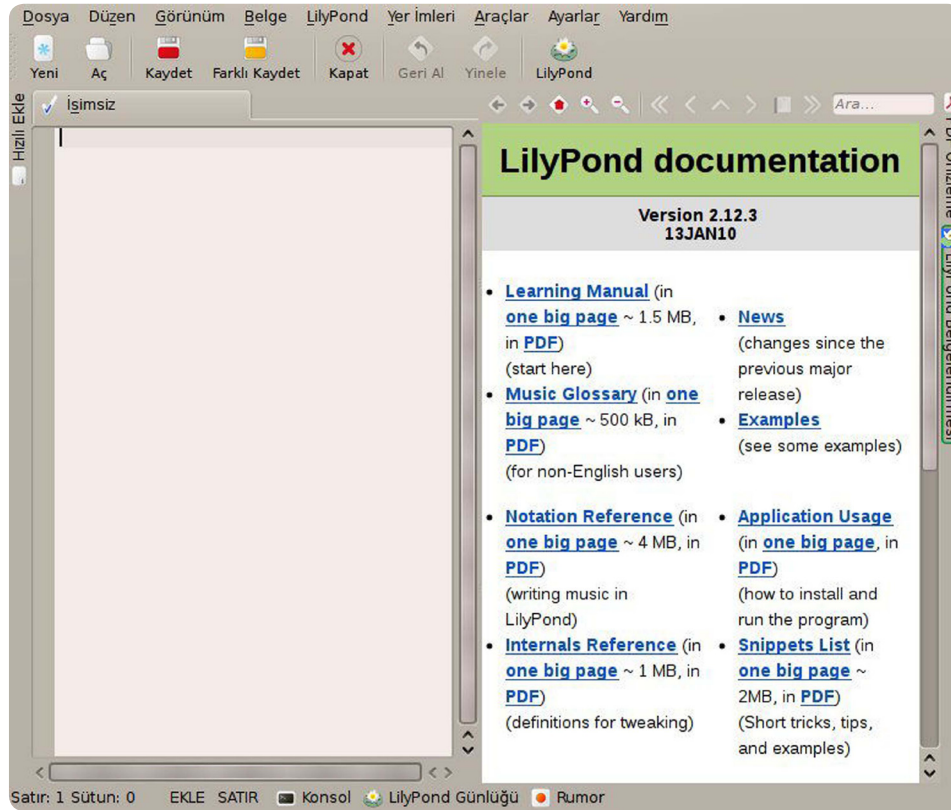
Kullanıcıdan alınan herhangi bir sayıya kadar olan asal sayıları yazdıracak bir kod yazalım. İpucu: input, while, if, else, % (kalan) deyimlerini kullanacaksınız.

Bir sonraki yazıya kadar kendinize iyi bakın.

```
>>> sayi=0
>>> while sayi<10:
...     sayi = sayi+1
...     print (sayi)
...
1
2
3
4
5
6
7
8
9
10
```

Frescobaldi ile İşler Nasıl Gidiyor?

Mü zisyenlerimizin çok sevdiği nota yazma programımız, yeni özellikleriyle göz doldurmaya devam ediyor.



Bir nota yazma yazılımı olan GNU LilyPond ve bu yazılımın editörü Frescobaldi ile ilgili yazılar gerek [Pardus Wiki](#)'de, gerek [Özgürlükçin e-Dergi](#)'nin eski sayılarında, gerekse [Özgürlükçin](#) web sayfalarında bir süredir yer alıyor.

Bunun dışında bu yazılımlar ile ilgili daha teknik detayları içeren, bu yazılımı kullanacak olan müzik insanlarına yönelik olarak hazırladığım web sayfaları da ayrıca mevcut.

Bu yazıda okuyuculara, Frescobaldi yazılımına eklenen yeni özellikleri tanıtmak istiyorum. Pardus 2009.1 sürümünü bilgisayarına yeni kuran ve/veya "Frescobaldi" yazılımının varlığından ilk kez haberdar olan okuyuculara önerim, [buraya](#) tıklayarak bu yazılım hakkında fikir sahibi olmalarıdır.

LilyPond dosyalarının Frescobaldi ile hazırlanmasında kullanıcıya sağlanan kolaylıklar, zaman içerisinde giderek artmaya başladı. Bu özellikleri çok detaya girmeden, özet başlıklar ve kısa açıklamalar halinde tanıtmaya çalışacağım.

Burada kısa bir not düşmek isterim. Bu yazıda adı geçen GNU LilyPond yazılımının sürüm numarası "2.12.2" olup, bu yazıya konu olan Frescobaldi yazılımının sürüm numarası da "1.0.2"dir. Bu yazının yazıldığı tarihte, bu iki yazılım da Pardus 2009.1 sürümünde yer almaktaydı.

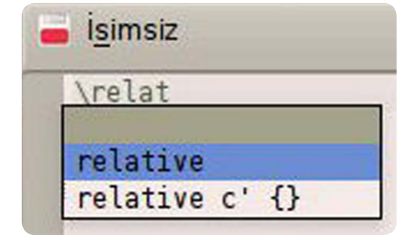
Belgelendirme Özelliği

LilyPond dosyalarının Frescobaldi ile hazırlanması aşamasında kullanıcının, orijinal dilindeki kılavuzlara göz atmasında çok büyük fayda var. Bunun için kullanıcının, GNU LilyPond yazılımının web sayfasındaki Belgeler (Documentation) sayfasına gitmesi ve bu sayfadaki kılavuzlara göz atması gerekiyor.

Artık yazılımdan çıkmadan, yazılım penceresinin sağ köşesinde yatay olarak yer alan "LilyPond Belgelendirmesi"ne tıklayarak bu web sayfasına ulaşılabilir. Belge sayfası açıkken, sağ köşede görülen "Ara..." kutusunu kullanarak belgede arama yapabilirsiniz. Bu özellik kullanıcıya zaman kazandırıyor.

Kodlama Sırasında İpuçları Görünümü

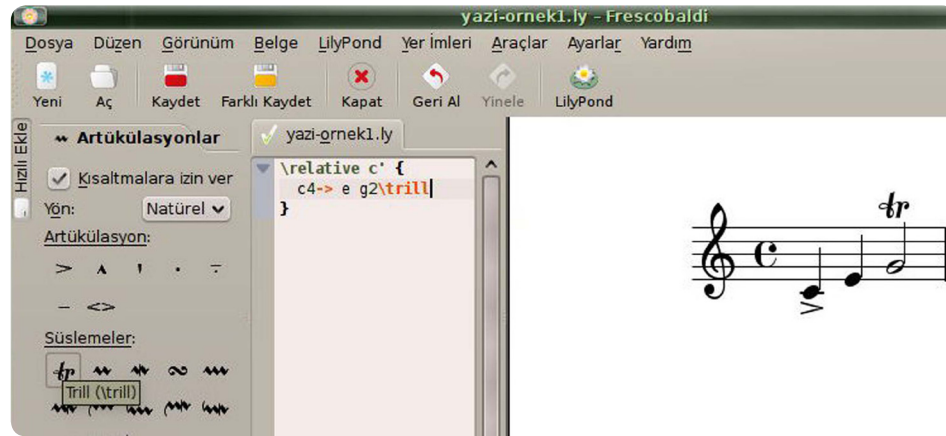
Yazılımın metin düzenleme kısmında kodlar yazılırken, zamandan kazanmak amacıyla kullanıcıya kodun devamını tamamlayabilmesi için seçenekler sunuluyor ve bu sayede işlemler hız kazanıyor.



Artükülasyon İşaretlerini Hızlı Ekleme

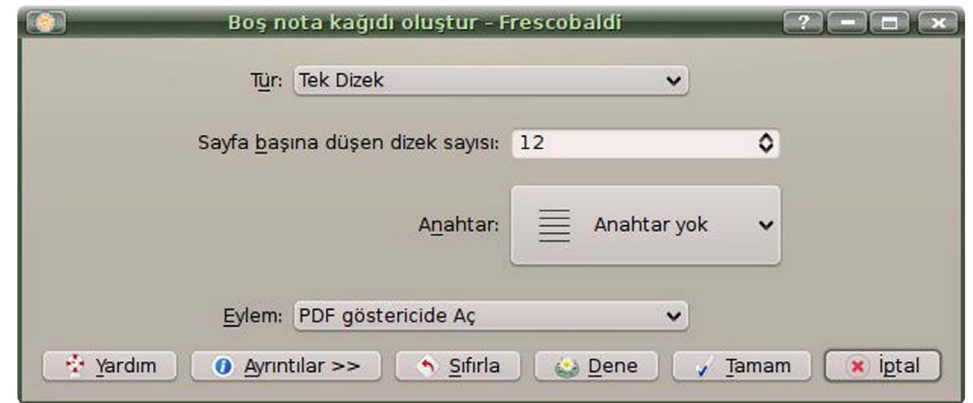
Notalara eklenen bazı simgeler, o notanın çalınması sırasında uygulanacak bazı detayları içeriyor. Türkçe çevirisinde "artükülasyon" olarak yer alan bu kategorideki simgeleri eklemek için, çalışma penceresinin solunda yatay olarak gözükken "Hızlı Ekle"ye tıkladığınızda karşınıza yeni bir küçük pencere açılır ve simgeler karşınıza çıkar.

Bu simgelerden hangisini kullanmak isterseniz önce metinde o notanın yer aldığı kısmı seçersiniz ve ardında o simgeye tıklarsınız. Böylece bu simgelerin uygulamasında hız kazanmış olursunuz. Metindeki simgelerin kod görünümündeki renk farkı da dikkatinizi çekmiş olmalı.



Boş Nota Kağıdı Oluşturma

Boş nota kağıdı bulmak, müzisyenler için önemli bir ihtiyaçtır. Notayı önce elle yazmak, bir edebiyatçının boş kağıda eline kalemi alıp bir şiir yazması gibi bir ihtiyaçtır. Kendisi de bir müzisyen olan Frescobaldi programının yazarı Wilbert BERENDSEN, bu ihtiyaca cevap veren bir özellik eklemiş yazılıma.



Menüden "LilyPond > Ekstra Araçlar > Boş nota kağıdı oluştur" seçeneğiyle açtığımız pencerede, oluşturacağımız nota kağıdı ile ilgili seçenekleri kendimize göre özelleştirebiliriz.

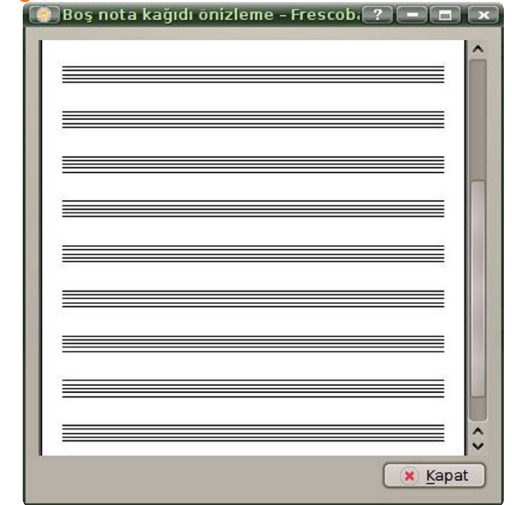
Bu özelleştirmeleri yaptıktan sonra, nasıl gözükeceğini görmek için "Dene" düğmesini tıklarız ve boş nota kağıdının önizlemesini yapabiliriz.

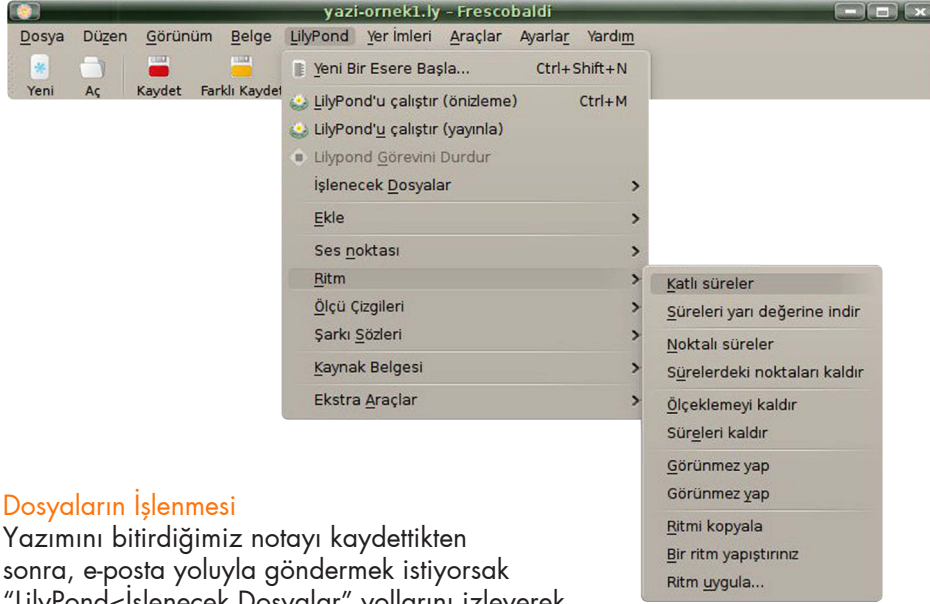
Yazılmış Notanın Ritmel Özelliklerini Değiştirme

Yazdığımız notanın sürelerini (notanın kaç vuruş boyunca uzayacağını) değiştirme şansımız var.

Önce yazdığımız notanın sürelerini de içeren kod kısmını seçtikten sonra menüden "LilyPond > Ritm > Katlı Süreler" yollarını izleyerek sürelerin tamamını bir misli büyütebiliriz.

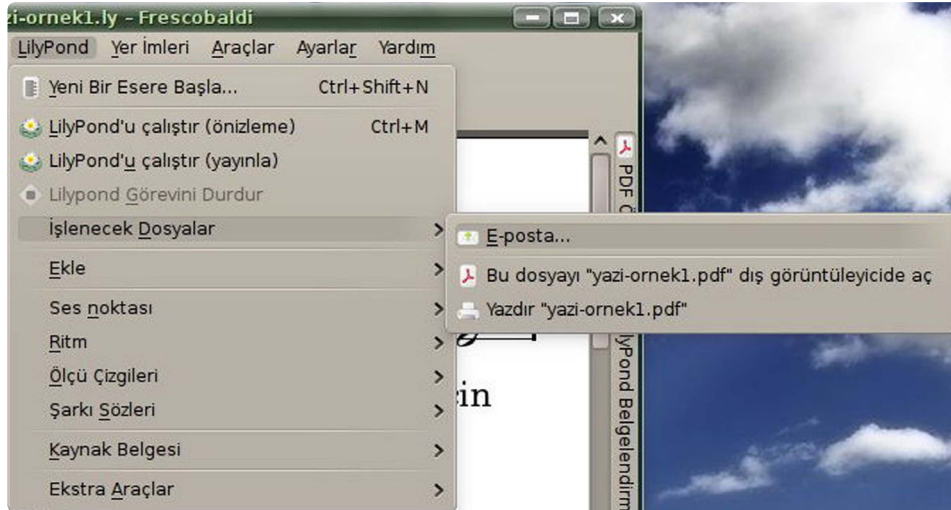
Bu seçenek, notaların sürelerini tek tek değiştirme, tümünü ya da istediğimiz bir kısmını hızlı bir şekilde değiştirme olanağı tanıması nedeniyle çalışma hızımızı artırma olanağı veriyor.





Dosyaların İşlenmesi

Yazımını bitirdiğimiz notayı kaydettikten sonra, e-posta yoluyla göndermek istiyorsak "LilyPond<İşlenecek Dosyalar" yollarını izleyerek "E-posta"yı seçiyoruz ve bilgisayarımızın varsayılan e-posta yazılımının gönderme penceresi açılıyor ve bize dosyayı hangi şekliyle



(LilyPond dosyası mı, yoksa PDF dosyası mı) göndermek istediğimizi soruyor ve biz de dosyamızı seçerek gönderiyoruz. Bu işlemi yazılımı kapatmadan yapmak bize yine hız kazandırıyor.

Sonuç

Frescobaldi yazılımına kazandırılan tüm bu özelliklerde dikkati çeken ortak nokta, kullanıcıya hız kazandırmak.

Programın KDE 3.5 masaüstü ortamında Kate için LilyKDE eklentisi olarak kullanılmaya başlandığı ilk günlerden, KDE 4 masaüstü için Frescobaldi adıyla bağımsız bir editör haline geldiği günümüze kadar Türkçe çevirilerini yaptım.

Bu yazılımın gelişimini, bu çeviriler sayesinde daha yakından izleme şansı buldum. Yazarın kendisi de bir "müzikçi" olduğu için, "bir müzisyenin yazılımından beklentilerini" tahmin etmekte hiç zorlanmıyor ve bu yazılımı kullanan müzikçilerin hızını artırmak amacıyla bazı kolaylaştırıcı özellikleri yazılıma ekliyor.

Son zamanlarda aldığım bazı e-postalar sayesinde, İnternet ortamında hazırlamış olduğum sayfaların, Özgürlükçin sayfalarında ve E-Dergi'de yazdığım yazıların bazı müzisyenlerin ilgisini çektiğini görüyorum. Umarım zamanla daha fazla insanın ilgisini çeker.

Sonuç olarak Frescobaldi ile işler gayet güzel gidiyor ve yazılım her geçen gün kullanıcının yararına kolaylıklar sağlayan bir yazılım haline geliyor.

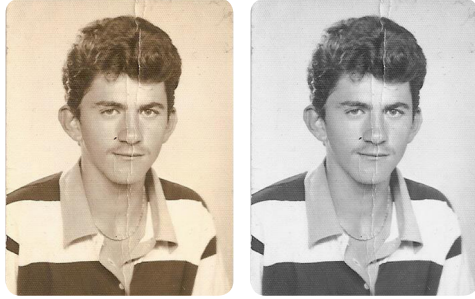


Yıpranmış Fotoğrafları Onarmak

Pardus ile eski aile ya da okul fotoğraflarını düzeltmek kesinlikle zor değil.



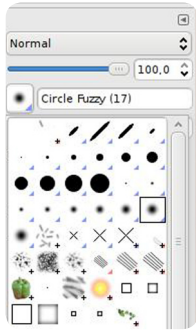
Merhaba. Bu makalemizde eski yıpranmış yırtılmış resimlerin onarılması hakkında bilgiler vermeye çalışacağım. İlk önce Gimp uygulamamızı başlatalım. Fotoğrafımızı Gimp programında açalım.



Renkli olarak taranmış siyah beyaz fotoğraf ile karşı karşıyayız. Öncelikle bu fotoğrafı gri ölçekli hale getirelim. Resim > kip > gri ölçekli yolunu izleyelim. Böylelikle fotoğrafımız siyah beyaz yani gri ölçekli oldu. Fotoğrafta görüldüğü üzere hem lekeler hemde kırıklar mevcut. Lekeleri ve kırıkları yok etmenin en kolay yolu kopyalama aracını kullanmaktır.

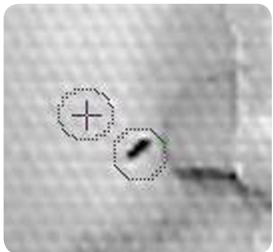


Clone Tool (Klonlama aracı ya da Kopyalama aracı da denilebilir). Araç Kutusundan Clone Tool' a tıklayalım. Fırça tipi olarak Circle Fuzzy seçelim ve fırçanın büyüklüğünü scale değerini değiştirerek sağlayabiliriz.



Büyüklik onaracağımız yerin büyüklüğüne uygun olmalı. Mesela önce yüzdeki lekeyi yok edelim. + ve - tuşları ile fotoğrafımızı ekranda büyütüp küçülebileceğimizi hatırlatalım.

Sırasıyla önce CTRL tuşuna basılı olarak fotoğrafta yıpranmamış bir yere tıklayalım. Yıpranmamış yer lekeye yakın ve aynı ten renginde olmalı. Daha sonra parmağımızı CTRL tuşundan çekelim ve leke üzerinde fare ile uzunca tıklayıp fırçayı leke üzerinde gezdirelim. Leke yok olunca parmağımızı farenin tuşundan çekebiliriz.



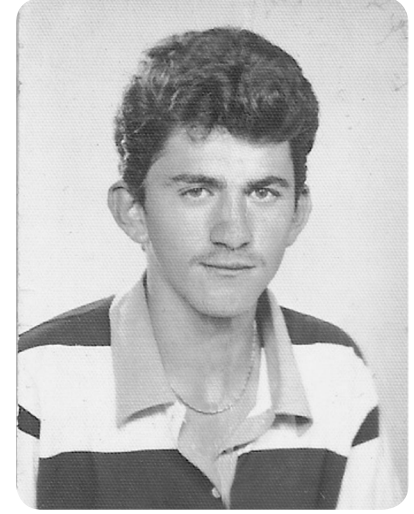
Fotoğraftaki lekeleri yok ettikten sonra fotoğrafımızı klavyeden + tuşuna basarak büyütelim ve kırıkları yok etmeye başlayalım. Bunun için gene aynı şekilde önce fotoğrafın yıpranmamış olan yerine yıpranmış yerle aynı renkteki yere Ctrl tuşu basılıyken tıklayalım.

Daha sonra kırık üzerine tıklayarak fırçayı kırık yönünde hareket ettirelim. Aynı işlemi tekrarlayarak kırığın

devamını yok edelim. Burada bir elimiz klavyede bir elimiz faredede olmalı. Gimp programını etkin bir biçimde kullanmak istiyorsanız buna alışmalısınız.

Gimp programını pratik bir biçimde kullanmak istiyorsak şunlara dikkat etmeliyiz:

- Farenin kaydırma tuşu resmi yukarı - aşağı hareket ettirmemizi sağlar.
- Ctrl + Scroll fotoğrafı ekranda büyütüp küçültmemizi sağlar.
- Shift + scroll ise fotoğrafı ekranda sağa sola hareket ettirmemizi sağlar.



Fotoğraftaki tüm leke ve kırıkları yok ettik. Şimdi lekeleri düzeltmenin ikinci bir yolunu öğrenelim. Geri alma geçmişinden geriye gidelim.

Geri alma geçmişi penceresini göremiyorsanız, Windows > dockable dialogs > Geri Alma Geçmişi yolunu izleyiniz.

Şimdi Araç kutusundan Healing tool'a tıklayalım.



Fırça tipi olarak Circle Fuzzy seçelim. Scale değerini artırarak fırça büyüklüğünü değiştirelim. Kullanımı kopyalama aracındaki gibidir.

Sağlam bölgeler CTRL tuşu basılıyken tıklanır daha sonra parmağımızı CTRL tuşundan çekerek daha sonra yıpranmış olan bölgeye tıklayabilir ya da tıklayarak fırçayı sürükleyebiliriz. Bu şekilde lekeleri ve kırıkları yok etmiş oluruz.

Bu iki yöntem hem siyah beyaz hem de renkli fotoğraflar için geçerlidir.

Fotoğraf için Ertan GÜVEN'e teşekkür ederim.



openoffice.org türkiye

Hakan HAMURCU
hakan.hamurcu@openoffice.org.tr

Yeni Özellikler

OpenOffice.org 3.2

Bugüne kadarki en hızlı OpenOffice.org ile
bir tura hazır mısınız?



OpenOffice.org'un son sürümü olan 3.2 ile çok daha hızlı, hatasız ve fonksiyonel bir ofis paketine kavuştuk. Yeni sürümün yayınlandığı günlerde sizlere yeni eklenen özellikleri bir liste halinde [duyurmuştuk](#). E-Dergimizin bu sayısı ile birlikte, artık depodaki yerini kısa bir süre içerisinde alacak olan OpenOffice.org 3.2'yi daha yakından keşfetmenin zamanı geldi.

Biz [OpenOffice.org Türkiye](#) ekibi olarak yeni sürümde kullanıcı arayüz çevirilerini %83 oranında tamamladığımızda sürüm yayınlandı. An itibarı ile arayüz çevirilerimiz %97 oranına geldi ve hızla tamamlanmaya devam ediyor. Bu sebeple mevcut sürümde Türkçe olmayan menü ve/veya mesajlarla karşılaşabilirsiniz.

Ancak ilk ara sürümde %100 Türkçe bir OpenOffice.org kullanabilir durumda olacağız. Yeni sürümlerde Türkçe çevirilerin geç kalmasını istemiyorsanız lütfen daha önce yaptığımız [çağrı mesajına](#) bir göz atın ve sizlerde aramıza katılın.

Şimdi bu muhteşem özellikleri daha detaylı olarak inceleyelim.

Genel Özellikler

Yeni özelliklerden, belki de bizi en çok mutlu edeceklerin başında "Yeni Açılış Süreleri" geliyor. OpenOffice.org 2.4 sürümü ile gayet makul ve hızlı açılış sürelerine sahip olan ofis paketimiz, yeni özellikler eklendikçe giderek hantallaşmaya başlamıştı. Doğal olarak kullanıcılardan gelen talepleri dikkate alan geliştirme ekibi 3.0 sürümünden bugüne, son sürümde ortalama %46'lık bir açılış süresi hızlandırmasını başarmış durumda. Elbetteki bu değer sizin bilgisayarınızın performansına göre farklılıklar gösterecektir. Ancak genel olarak

gayet hızlı olan 2.4 sürümünden sonra 3.2 sürümü bu süreleri daha da kısaltmayı başardı.

Bu sayfada yer alan tabloda OpenOffice.org'un 2.4 sürümünden bu güne açılış sürelerini görebilirsiniz. Umarız bu gelişim devam eder ve bir sonraki sürümde daha hızlı açılış süreleri elde edebiliriz.

Not: Değerler saniye cinsinden verilmiştir. Küçük olan rakamlar daha hızlı açılış sürelerini temsil eder.

Yeni özelliklerin ikincisiyse geliştirilen ODF desteği. Bilindiği üzere OASIS tarafından uluslararası elektronik/sayısal belgelerin depolanma biçimi olarak [Açık Belge Biçimi](#) (ODF) standardı kabul edilmiştir. Bu standartla birlikte çalışabilirlik, akıllı bilgi değişimi ve aramalarının önemi artıyor.

Örneğin, bugün herhangi bir bilgisayar sisteminde oluşturulan ofis belgelerinin, geleceğin teknolojisi de dâhil olmak üzere herhangi bir yerdeki diğer bilgisayar sistemleri tarafından da okunabilmesini garantiliyor.

OpenOffice.org 3.2, OASIS ODF/Açık Formül de dâhil olmak üzere ODF 1.2 ile uyumlu hale gelmek için büyük gelişme gösterdi. OpenOffice.org artık kullanıcıları, genişletilmiş ODF 1.2 özelliklerinin kullanılmaya başlaması konusunda uyaracak. Belge bütünlüğünü kontrol sistemi artık bir ODF belgesinin, özellikle ODF 1.2 belgelerinin, ODF özelliklerinin uygun olup olmadığını kontrol edecek. Eğer belgelerde bir tutarsızlık bulunursa, belge bozuk belge olarak etiketlenecek ve OpenOffice.org belgeyi onarım teklifinde bulunacak.

OpenOffice.org Sürüm Numarası	2.4			3.0			3.1.1			3.2M6		
Ofis Bileşeni	Writer	Calc	Draw	Writer	Calc	Draw	Writer	Calc	Draw	Writer	Calc	Draw
HızlıBaşlat Olmadan İlk Açılış	8.80	9.11	10.11	10.58	10.57	10.53	8.59	8.64	8.38	5.68	5.72	6.19
HızlıBaşlat Olmadan Yeniden Açılış	1.43	1.48	1.45	2.39	2.38	2.36	2.34	2.34	2.45	2.21	2.15	2.69
HızlıBaşlat İle İlk Açılış	5.72	6.21	6.21	6.04	6.11	5.95	5.22	5.20	4.93	3.38	3.46	3.82
HızlıBaşlat İle Yeniden Açılış	0.75	0.70	0.76	0.80	0.78	0.84	0.78	0.76	0.85	0.73	0.74	0.99

Yeni sürümde OpenOffice.org, diğer ofis belge türlerine olan desteğini de geliştirdi.

- MS Word 2007 (*.docx, *.docm)
- MS Word 2007 temaları (*.dotx, *.dotm)
- MS Excel 2007 belgeleri (*.xlsx, *.xlsm)
- MS Excel 2007 ikili belgeleri (*.xlsb)
- MS Excel 2007 temaları (*.xltx, *.xltm)
- MS Powerpoint 2007 sunumları (*.pptx, *.pptm)
- MS Powerpoint 2007 temaları (*.potx, *.potm), OLE nesnelere, form kontrolleri ve tablo özetleri
- Ayrıca yeni sürümde parola korumalı Microsoft Word 97/2000/XP dosyalarının şifre desteğiyle kaydedilmesini sağlamakta (Microsoft Office standardı olan RC4 algoritmasını kullanarak).
- Doctype temelli AportisDoc ve PocketWord filtreleri artık destekleniyor. Böylece bu türdeki dosyalar, dosya türünü seçmeden görüntülenebilir.

Dipnot için OpenType Yazıtıpi desteği: Dipnotlarda kullanmak için yüksek kaliteli ve özgür birçok OpenType yazıtıpi mevcut. Pardus topluluk portalı Özgürlükçün'den yükleyebileceğiniz çok sayıda ücretsiz ve özgür yazıtıpi biçimlendirme, çıktı alma, PDF olarak kaydetme ve görüntüleme için desteklenmektedir.

Writer / Kelime İşlemci

Otomatik kelime tamamlama: Writer bir belge üzerinde çalışırken otomatik düzeltme listesinde yapılan her değişikliği daimi olarak hatırlayabilir. Belgeyi kapatırken çıkan yeni onay kutusu, eklenen kelimelerin listeden kaldırılmasına olanak sağlayarak bu işlevselliği daha da kullanışlı hale getirmiştir.

MediaWiki desteği artık bir eklenti: Karışıklıkları önlemek için, MediaWiki filtresi artık OpenOffice.org ile beraber otomatik olarak yüklenen bir parçası olmaktan çıktı. Bunun yerine Wiki Publisher'in bir eklentisi olarak sunuluyor, böylece filtre güncellemelerinin bir sonraki OpenOffice.org dağıtımını beklemeye gerek duymadan yapılmasına olanak tanınıyor.

Calc / Hesap Tablosu

Hücre Kenarlıkları Artık Birden Çok Seçimi Destekliyor: Daha önce hücre

kenarlıkları sadece tek bir hücre aralığına uygulanabilmekteydi. Şimdiyse kenarlıklar seçilen bütün hücelere uygulanabiliyor. Seçim içinde yer alan dikdörtgen aralıklar listeden seçilebilmekte ve böylece iç/dış kenarlıklar, listedeki seçilmiş bütün aralıklara uygulanabilmekte.

Otomatik tamamlamada ek özellikler: Eğer otomatik tamamlama özelliğinin başladığı hücre bir numara ile başlıyor ve bir numara ile bitiyorsa ve ilk hücreden sonra hiç boşluk yoksa, son hücreye kadar numaralar artarak devam ediyor. Ay isimleri, gün isimleri, sayılar ve IP adresleri bu özellik sayesinde çok daha akıllı bir hal almış. Bu özellik IP adresleri gibi özel numaraların yazılmasında kolaylık sağlıyor (192.168.1.1 sayısını böylece 192.168.1.2 takip ediyor). Yalnız özellik IP numaralarında 255'den sonra 256, 257 şeklinde devam ettiğini hatırlatalım. Beklediği gibi son bölüm 255 olduktan sonra, bir önceki bölüm 1 artıyor.

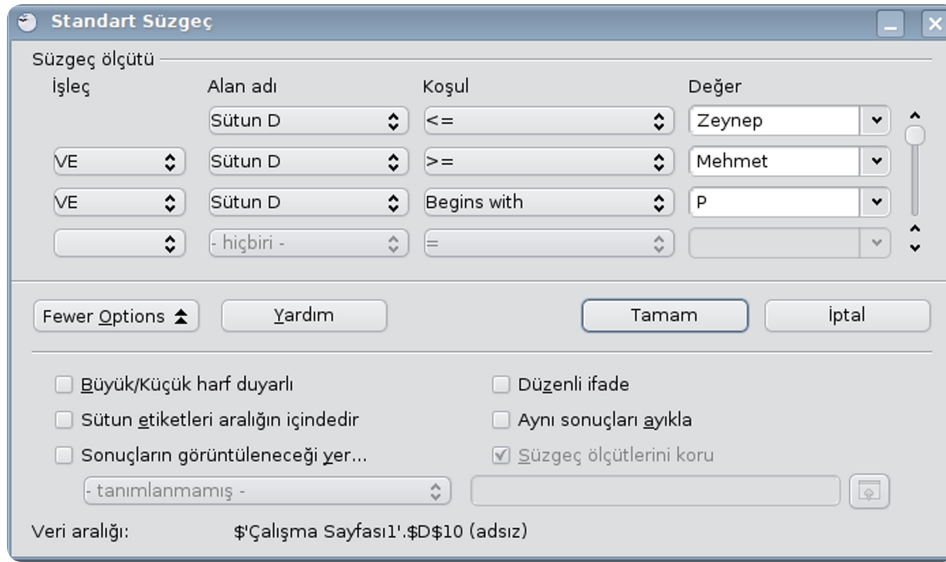
	A	B
1	Üniversite Adı	Şehir
2	Bozok Üniversitesi	Yozgat
3	Bilgi Üniversitesi	İstanbul
4	Erciyes Üniversitesi	Kayseri
5	İstanbul Teknik Üniversitesi	İstanbul
6	Ortaoğu Teknik Üniversitesi	Ankara
7	Hacettepe Üniversitesi	Ankara
8	Yıldız Teknik Üniversitesi	İstanbul

Geliştirilmiş Sıralama Özelliği: Eğer "Sırala" seçeneği kullanıcının istemediğini düşündüğü bir aralıkla karşılaşır, seçimin genişletilip genişletilmemesini soran bir pencere çıkıyor. Hücre aralıkları "Artan" ya da "Azalan" yönde sıralandıkları zaman artık hücre biçimleri de hücrelerle birlikte sıralanıyor.

	A
1	192.168.1.1
2	192.168.1.2
3	192.168.1.3
4	192.168.1.4
5	192.168.1.5
6	192.168.1.6

Geliştirilmiş Filtreler: Yeni sürümde Calc çok daha yetenekli ve esnek filtreler sunuyor. Standart Filtre diyalog kutusunda dört ayrı filtre sunulmaktayken, yeni sürümde bu sayı sekiz filtreye çıkartıldı. Ayrıca büyük eksikliği hissedilen "... ile başlayan", "... ile başlamayan", "... ile biten", "... ile bitmeyen", "içeren", "içermeyen" seçenekleri de dâhil edildi.

Daha Kullanışlı Birleştirilmiş Hücreler: Birleştirilmiş hücreler artık panodan diğer birleştirilmiş hücelere yapılandırılabilir. Hata mesajı yerine, eski hücreler ayrılıyor. Ayrıca artık birleştirilmiş hücrelerin içerisine yeni sütun ya da satır



eklenip, çıkartılabiliyor. Hatırlayacağınız gibi bir önceki sürümde birleştirilmiş hücrelerin içinden herhangi bir sütunu silmeye çalıştığınızda hata mesajı ile karşılaşılırdınız. Artık çok daha kullanışlı bir birleştirilmiş hücre fonksiyonu mevcut.

	A	B	C	D
1			Cari Hesap	
2	Adı	Soyadı	Borç	Alacak
3	Hakan	Hamurcu	15TL	5TL
4	Görkem	Çetin	7TL	26TL
5	Talat	Özpozan	0TL	50TL
6	Tahsin	Çetin	20TL	20TL

Hesap Dönemi			
Kış		Yaz	
Borç	Alacak	Borç	Alacak

Cari Hesap			
Borç	Alacak	Yaz	Alacak
15TL	5TL		
7TL	26TL		
0TL	50TL		
20TL	20TL		

Karmaşık Kopyala/Yapıştır: Calc'ın geliştirilen "Kopyala/Yapıştır" özelliği artık ardışık olmayan birden çok hücrenin kopyalanıp yapıştırılmasına olanak sağlıyor. Kopyalanan veri yatay veya dikey olarak gruplanıp düzenlenerek yeni yerine yapıştırılıyor. Kopyalarken oluşturulacak kısıtlamalarda, yapıştırırken istenilen şeklin oluşturulmasını sağlıyor.

Artırılmış İstatistik Fonksiyonları: NORMDAĞ'ın 4. parametresi artık seçime bağlı. "0" ya da "Yanlış" ibaresi olasılık yoğunluk fonksiyonunun hesaplamasını yapıyor. Bir değer belirtilmemiş, "Doğru" ya da herhangi başka bir değer girilmişse toplamsal dağılım fonksiyonunu hesaplıyor.

Eğer belirtilmezse, dosyalarda geriye uyumluluk olması için "1" yazılıyor.

Formüllerdeki Satırbaşı Karakter Koruması: Control+Enter tuş ikilisiyle oluşturulan çok satırlı metin hücreleri Calc "spreadsheet" formüllerine kaydediliyor.

Daha önce satırbaşı karakterleri boşluk karakteriyle değiştirilmekteydi. Satırbaşı karakteri "Arama" yapmak ya da "YATAY/DÜŞEY ARA" ya da "EŞLEŞTİR" fonksiyonlarının \n düzenli açıklamasıyla kullanabilmenizi sağlamaktadır.

Dizilerde Numerik Dönüşüm: Formül açıklaması yapılırken, dönüşüm belirsizse veya herhangi bir değişim yapılamamaktan dolayı hata alıyorsa diziler numaralara dönüştürülüyor.

Geliştirilmiş Referans Geçişleri: Formüllerdeki bağlantılı ve ayrı referanslar arasında geçiş yapan Shift+F4 kısayol tuşu artık hücre seçimi içinde kullanılabilir, aynı zamanda seçilen hücredeki formüle müdahale edilmesini sağlıyor.

B9 hücresine gelip SHIFT+F4 tuşuna her bastığınızda formülünüzdeki değişiklikleri görebilirsiniz.

	A	B	C	D
1	Şehir	Plaka		Bursa
2	Ankara	6		İzmir
3	Bursa	16		Kayseri
4	İstanbul	34		Kocaeli
5	İzmir	35		Yozgat
6	Kastamonu	37		
7	Kayseri	38		
8	Kocaeli	41		
9	Niğde	51		
10	Adapazarı	54		
11	Yozgat	66		

	A
1	Bu satırda birden fazla Satır kullanılarak metin girişi yapılmıştır. Eski sürümlerde bu dosyayı kaydettiğimizde Her satırın başına CTRL+ENTER yerine BOŞLUK kaydeden OpenOffice.org Artık CTR+ENTER karakterini kaydediyor.

	A	B
1	Şehir	Nüfus
2	Ankara	2.000.000
3	İstanbul	12.500.000
4	İzmir	2.000.000
5	Kastamonu	500.000
6	Kayseri	1.000.000
7	Kocaeli	750.000
8	Niğde	50.000
9	TOPLAM	18.800.000

CONVERT_ADD'e birçok yeni değişken: Hesap tablosundaki ÇEVİR_ADD fonksiyonuna (diğer uygulamalarda ÇEVİR olarak bilinmekte) OASIS ODF/ OpenFormula'da belirtilen birçok dönüştürme değişkeni eklendi.

Yeni UNICODE ve UNICHAR Fonksiyonları: Hesap tablosuna OASIS ODF/ OpenFormula özelliklerinde de belirtilen yeni UNICODE ve UNICHAR fonksiyonları eklendi.

Math / Formül

Math Menülerinde Değişiklik: Math'teki "Görünüm/Seçim menüsü", "Görünüm/Formüller" olarak değiştirildi.

Draw / Çizim

Yeni tip yorum: Draw, diğer programlarla işbirliğini desteklemek amacıyla genelde yapışkan notlar olarak bilinen yorumları artık destekliyor. Çizimlerinize üzerine açıklayıcı notlar ekleyebilirsiniz.

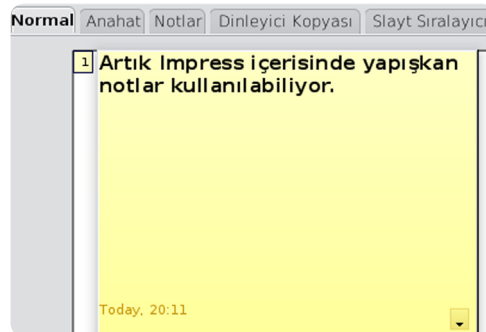
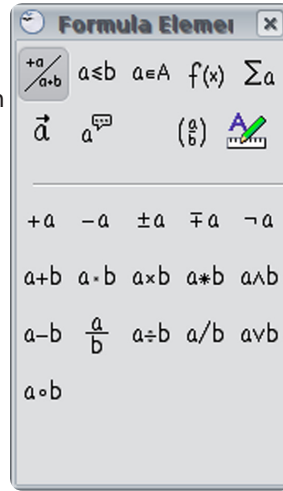
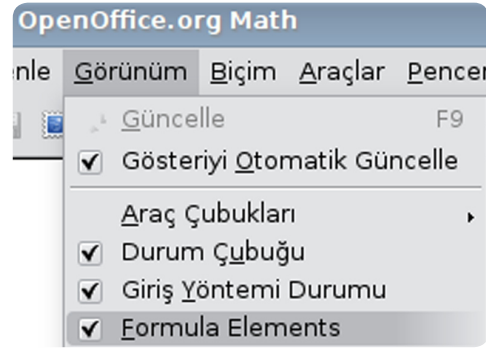
Draw'a sayfa sayma özelliği: Draw artık metin alanında "Sayfa Sayımı"nı destekliyor.

Impress / Sunu

Yeni tip yorum: Impress diğer programlarla işbirliğini desteklemek amacıyla genelde yapışkan notlar olarak bilinen yorumları destekliyor. Impress'e sayfa sayma özelliği: Impress artık metin alanında "Sayfa Sayımı"nı destekliyor.

Base / Veritabanı

Kişiselleştirilmiş "Veritabanı Oluşturma Sihirbazı": OpenOffice.



org 3.2 "Veritabanı oluştur" sihirbazında "Yeni veritabanı oluştur" seçeneğinin uygulanmasını kontrol eden yeni seçeneklerle geliyor.

Veritabanı belgeleri arasında daha kolay sorgu kopyalama: Base artık veritabanı belgeleri arasında sorgu kopyalarken, kopyalanan veri önceden olan bir tablo üzerine yazılmak istenmediği sürece size bir hedef isim sormayacak.

Veritabanı formları yakınlaştırılabilir: Veritabanı formları artık diğer belgeler gibi yakınlaştırılabilir. Genel olarak kullanılan Ctrl+Scroll Wheel fonksiyonu (Farenin tekeri) Görünüm/Yakınlaştır menüsünde yerini aldı ve durum çubuğundaki yakınlaştırma sürgüsüyle de yapılabilir.

Base için daha akıllı aktarım: Kopyalama Panosu sihirbazı artık RTF ya da HTML formatlı bilgilerinizi veritabanına kopyalamak istediğinizde etkinleşen bir seçim kutusu içeriyor. "İlk satırı sütun adı olarak belirle" olarak etiketlenen seçim kutusu seçildiğinde ilk satır sütun adları olarak aktarılabilir gibi, seçim yapılmamışken ilk satır normal bilgi satırı olarak aktarılıyor.

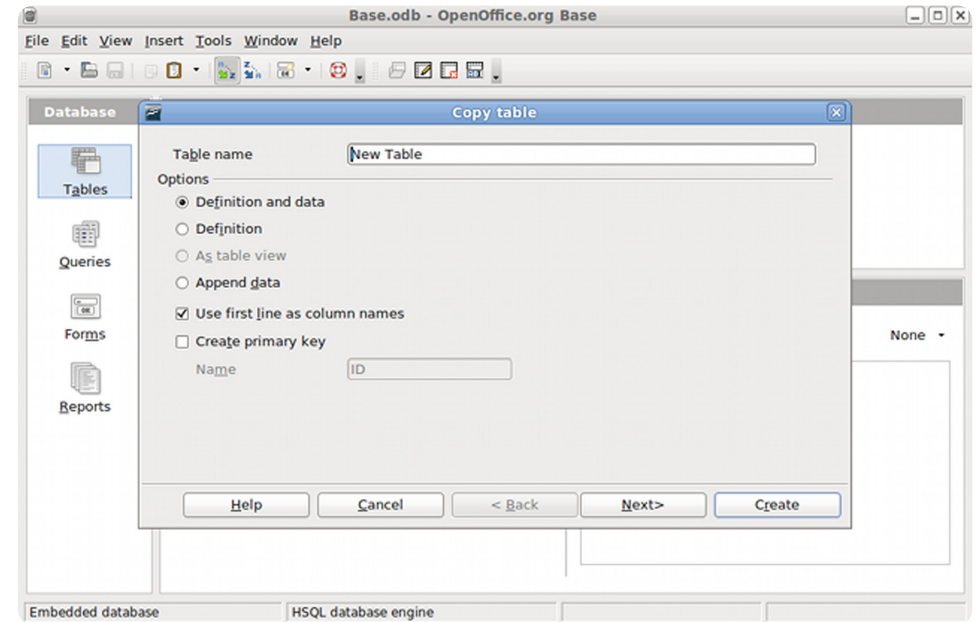
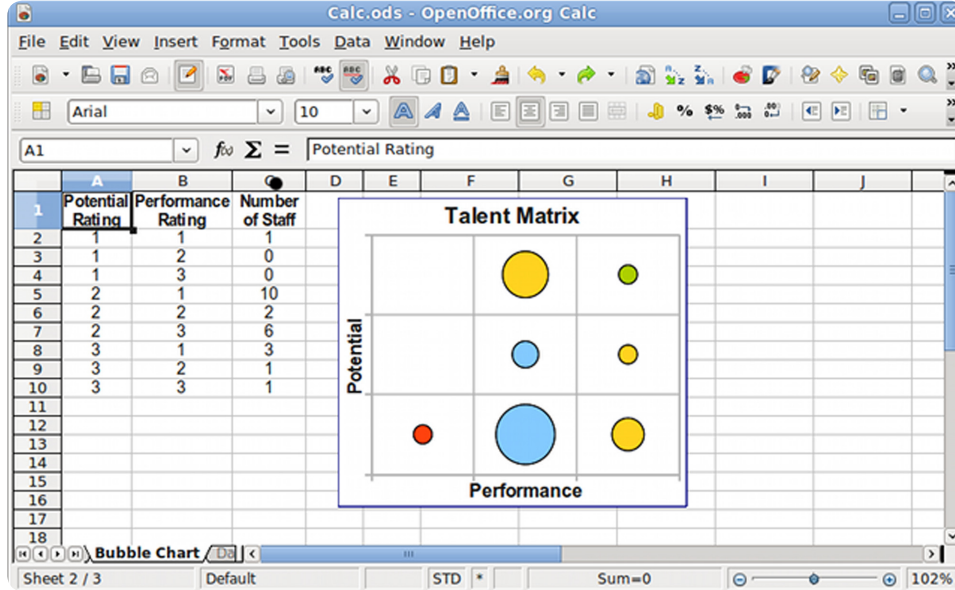


Chart / Çizelge

Yeni Chart Tipleri: Çizerge oluşturmanıza yardımcı olan Grafik Sihirbazı artık "Bubble" ve yeni bir değişken olan "Ağ" grafiklerini içeriyor.

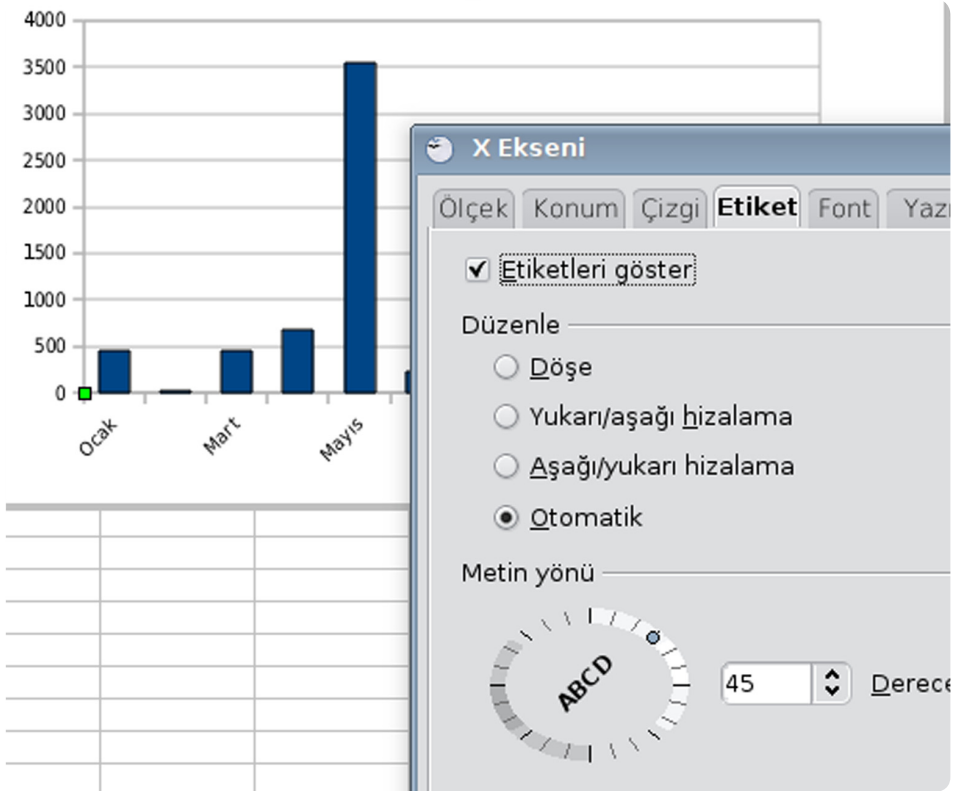


Geliştirilen Chart Kullanımı: Chart'ın grafik arayüzü kullanılabilirliği artırmak amacıyla yeniden makyajlandı. "Nesne Özellikleri" etiketi daha özel olan "Format Göstergesi" olarak adlandırıldı. Menüler artık durumlara daha hâkim, ihtiyaç duyulan araçlar sadece bir fare tıklaması uzağınızda. Geliştirilmiş Veri Etiketleri: Veri etiketleri özellikleri geliştirildi. Artık veri etiketlerini döndürebilmek mümkün. Bu kullanışlı özellik sayesinde grafiklerdeki sütun etiketleri birbirine karışmayacak.

Chart Nesneleri için Asya Tipografi Sayfası: Başlık, Gösterge, Eksen ve Bilgi serileri için artık Asya Tipografi sekmesi mevcut. Değişik yazı tipleri için aralık eklemeyi aktif/pasif yapma özelliğine sahip.

Uluslararasılaştırma ve yerelleştirme

Yeni Yerelleştirme Dilleri: OpenOffice.org 3.2; Etiyopya Oromo dili [om-ET], Uygurca [ug-CN], Antik Grekçe [grc-GR], Somalice [so-SO], Aşağı [dsb-DE] ve Yukarı Sorbça [hsb-DE], Asturya İspanyolcası [ast-ES], Yiddiş İbranicesi [yi-IL],



Umman Arapçası [ar-OM], Sardinyaca [sc-IT] ve bir Ekvador dili olan Quechua [qu-EC] dilleri için yerelleştirme bilgilerini içeriyor. Tipik olarak, yerel ayarlar karakter özelliğinden seçilebilmekte ve yazım denetimi yapılabilir. (Varsayılan yerel ayarlar, belge dili, rakam formatları ve anahat numaralandırması için geçerli. Ayrıntılar için sürüm notlarına bakabilirsiniz.)

Japon İsim Düzeni: Japon yerel ayarlarında kullanıcı bilgisi sekmesindeki Ad/Soyad sıralaması değiştirildi.

Grafit Font Teknolojisi Desteği: OpenOffice.org, dünyadaki çeşitli mikro topluluklar ve azınlıklar tarafından kullanılan yaygın olmayan yazı tiplerini daha iyi desteklemek için Grafit font teknolojisini destekliyor.

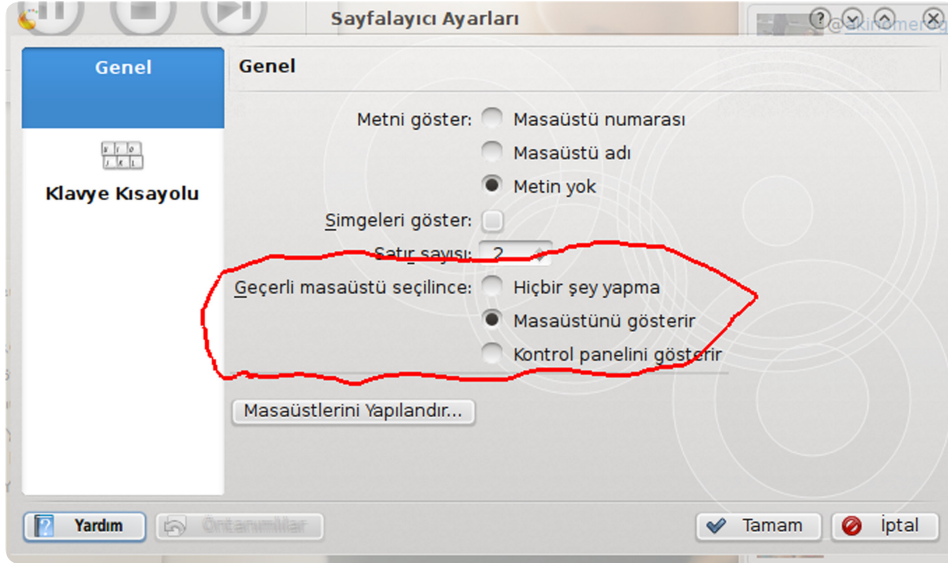
Plasma'nın Gizli Yüzü – II: Plasma pratiklik kazandırır

Hayatınızı kolaylaştıracak
Plasma programcıları burada!

Geçen yazımda Plasma'nın birkaç özelliğine değinmiştim. Bu yazıda biraz daha ufak detaylara ineceğim. Bu ufak detaylar pek çoğunuzun hoşuna gidecek. Özellikle benim gibi çoğunlukla masaüstünden çok pencerelerle yüz yüze olan kişilerin panellerini daha da işlevsel hâle getirmesi için plasmoidlerin biçilmiş kaftan olduğunu hesaba katarsak...

Masaüstünü Göster düğmesi gereksiz!

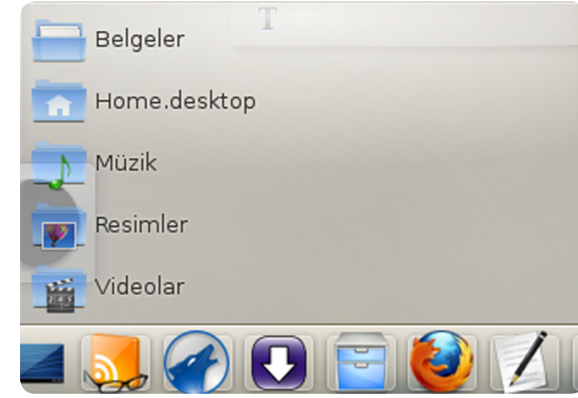
Bu düğmeye ihtiyacınız yok. Şimdi kesin "Onca pencereyi aşağı almak yoruyor be kardeşim!" diyeceksinizdir. Ben zaten o işlevden mahrum kalın demiyorum. Sadece bu özelliğin bir düğmeye ihtiyacı olmadığını söylüyorum. Bu işi, aslında başka bir işlevi olan, ama gene de bunu yapabilen ve çoğu kişinin panelinde zaten yer alan bir plasmoid yapıyor. Bu can kurtaranın adı "Sayfalayıcı"!



Evet, yanlış okumadınız. Sanal masaüstleri arasında gezmek için yaratılan sayfalayıcı, masaüstünü göster olarak da çalışabiliyor. Bu özelliği kazandırmak 2-3 tıklık bir iş. Önce sayfalayıcıya sağ tıklayın ve "Sayfalayıcı Ayarları"na girin. "Geçerli masaüstü seçilince" diye bir bölüm göreceksiniz. Oradaki eylemi "Masaüstünü gösterir" olarak seçin ve tamama basın. O da ne? Sayfalayıcıda aktif masaüstüne tıklayınca tüm pencereler aşağı mı alınıyor? İşte size bir düğmelik de olsa yer kazandıracak ilk hilemiz :).

Dizinlerle uğraşanlara ilaç!

Dizinlerle sürekli uğraş hâlinde olanlar, sürekli bir dizine girmek için masaüstüne mi geçiyorsunuz? Buna ne gerek var? Pannelle dizinlerde dolaşmak mümkünken hem de... Bunun için yapmanız gerekenler çok kolay. Panele "Dizin Görünümü" programcığını yükleyin, olsun bitsin. Geri kalan ayarlar normal bir dizin görünümü ayarları. Tek farkı dizinlerin üzerine gelince önizlemelerini yapmıyor o kadar. Ama pek çok kişi için ilaç olacaktır.



Dosya aktarımı göstergesini sevmeyenler

KDE4 ile birlikte dosya aktarımı, "Sistem Çekmecesi"nde bir bildiri şeklinde yapılıyor. Bundan rahatsız olan pek çok kişi var. KDE size bunun için ayar da sunuyor. "Sistem Çekmecesi"ne sağ tıklayıp bildiri düğmesine tıklarsanız daha rahat olur. "Sistem Çekmecesi Ayarları" diyerek ayarlarına girin. "Bilgiler" sekmesine bir bakın derim. Oradaki bir ayar (Dosya aktarımları ve diğer görevler) tam aradığınız ayar. O ayarı kaldırın ve olacakları görün.

Xbar ile pencerelerden alan kazanın

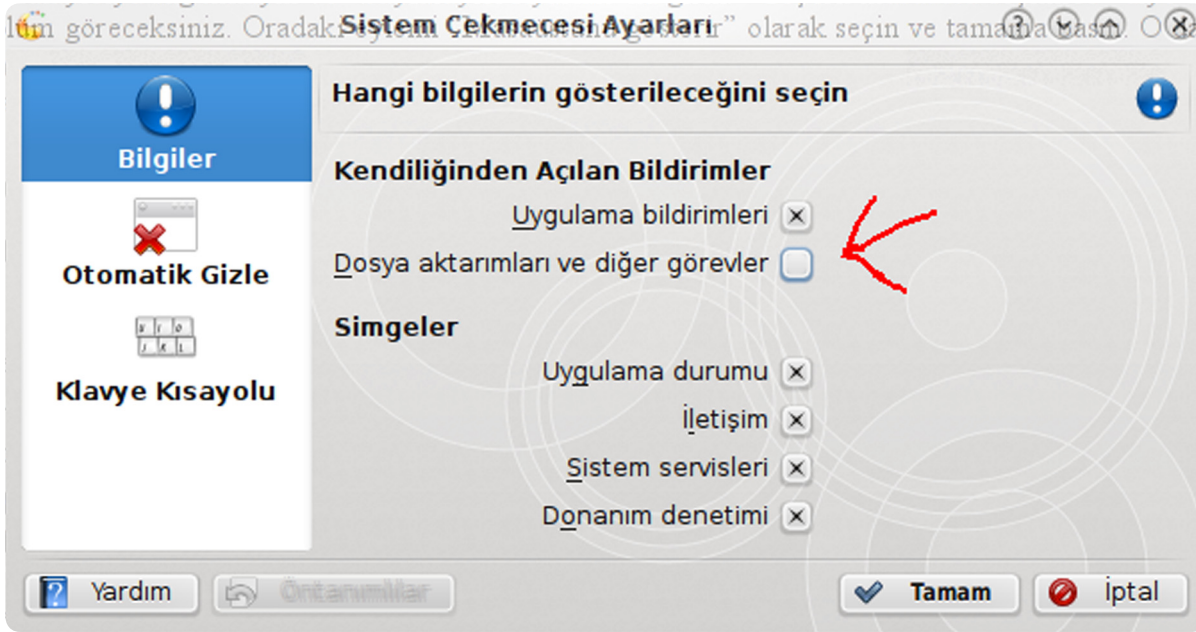
Pek çok kişi Mac OS X'in menü çubuğu (dock) özelliğinin ne kadar ilginç bir özellik olduğunu biliyordur. Peki, panelinize bunu ekleyen bir plasmoid olduğunu söylesem? Hem de Pardus'un içinde kurulu geldiğini söylesem?

Xbar adlı bu uygulama son derece hoş bir menü çubuğu. Pencerelerdeki menüleri (Dosya, Düzen vb.) panelinize aktararak sizi yer sıkıntısından kurtarıyor. Bu plasmoid Mac çakması bir arayüz yapmak isteyenler için de epey yararlı. Ayrıca bir çeşit pencere listesi gibi de çalışabiliyor.

UYARI: Bu plasmoid, sadece biçim "bespin" temasında ise çalışır. Yalnızca QT uygulamalarında işe yarar.

Pastebin ve kolay yükleme

Bir resmi ya da metni kolay yoldan İnternet ortamına aktarmak mı istiyorsunuz? Pardus ile beraber gelen başka bir Plasma programcığı da bu işi yapıyor. Pastebin ile bu işlevi kolayca halledebilirsiniz. Panelinize bu programcığı ekleyin



masaüstünde boşu boşuna yer kaplamaz. Bunun dışında xbar ile bahsettiğim özellik, yine panelin bu konudaki nimetleri arasında.

Panelde boşluk yaratmak isteyen kişiler bazen boşu boşuna hiç kullanmayacakları programcılar kuruyor. Buna gerek yok. Paneli yapılandırma gelip oradan ayırıcı ekleseniz yeter kardeşler...

Panel boyutunu otomatik ayarlar. Bunun anlamını söylemek gerekirse; eğer daha hoş görünmesi için ufacık bir panel yaptığınızı varsayalım. Sistem çekmecesi ve görev yöneticisi gibi yerlerdeki değişikliklerle panel boyutunu yeniden ayarlayabilir.

Plasma dışı ufak bir hile

Alt+F2 ile açılan başlatıcının sandığınızdan daha yetenekli olduğunu biliyor muydunuz? Bunlardan birkaç tanesini söyleyelim...

İnternet adresi yazdığınızda varsayılan tarayıcı ile o sayfayı

ve yükleyeceğinizi şeyi üstüne sürükleyin. Ayarlardan sunucu bilgilerini de yapılandırabilirsiniz.

Pencereler panelde yer kaplamasın

Birden fazla pencere açılınca panelin ne kadar sıkış tıktık olduğunu fark etmişsinizdir. Bunun için hoş bir çözüm var. Görev Yöneticisi'ne (pencere listesi) sağ tıklayarak 'Yapılandır'ı seçin ve orada sadece simgeleri göster ayarını seçin. Pencere adları boşu boşuna yer kaplamasın.

Bundan daha profesyonel çözümler de mevcut. Örneğin "Stasks" ve "Smooth Tasks" adlı iki hoş Plasma programcığı. Benim şahsi tercihim "Smooth Tasks" yönünde.

Sen neymişsin be panel?

Özellikle netbook kullanıcılarının en çok yaşadığı sorundur pencerelere yerleri alan ayıramamak. Bunun için paneli bile inceleyebilirsiniz. Ama hiçbir çözüm gizlenen bir panel kadar iyi olamaz. Sadece üstüne geldiğinizde açılarak

açar.

Başlamadan önce "=" yazıp sonra da bir işlem yazarsanız hesap makinesi olur. İşlemin sonucu, normalde uygulamaların gösterildiği alanda gösterilir.

Örneğin "2 + 2" gibi basit matematik işlemlerini yapmakla beraber uzun ve karmaşık işlemlerini de gerçekleştirebilirsiniz.

Son kullanılanlar arasında gezer.

Aynı anda birden fazla kullanıcı açıksa aralarında geçiş çok kolay. Konqueror, Konsole ve Kate oturumlarından birini seçebilirsiniz için yardımcı. Adres defterinizde kişiler arasında gezmek için yardımcı olur.

Bunlar dışında pek çok özellik daha mevcut. KDE-Apps.org ve KDE-Look.org'a biraz göz gezdirirseniz, daha yeni özellikler de bulabilirsiniz.

Bol Pardus'lu günler.



Web Tasarımcıları için: Mavi Balık

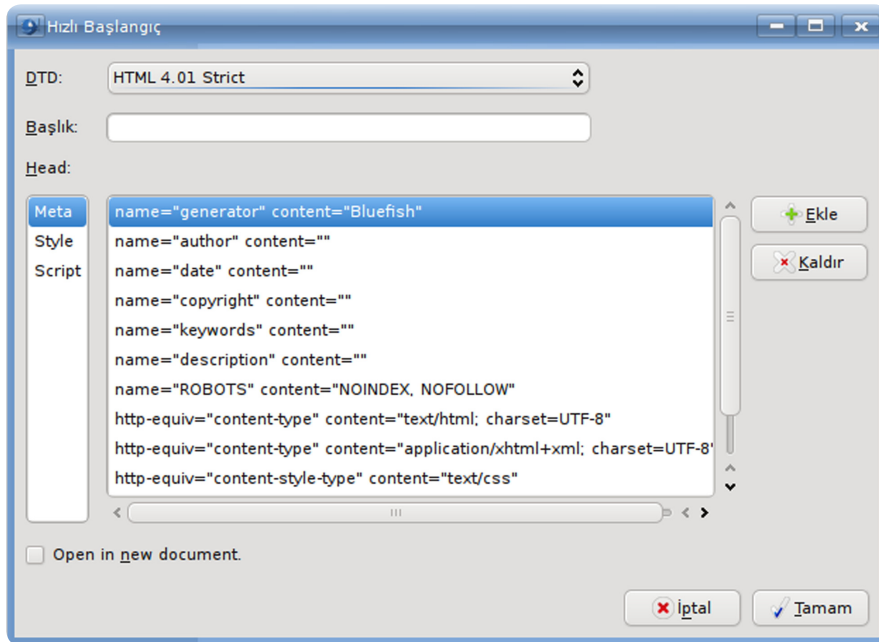
Yetenekli web editörü Bluefish ile İnternet sayfalarınız için şablon hazırlamak, son derece kolay.

paket tanıtımı

Bir web sitesi hazırlamaya kalktığınızda sayfalarınıza standart olarak kullanacağınız şablon HTML kodlarını eklemek zorundasınız. Bu kodlar ile web sayfanızın başlığını, anahtar kelimelerini, dil kodlaması gibi niteliklerini belirtebilirsiniz. Bluefish'in hızlı başlangıç özelliğiyle bu niteliklere kolayca değer atayabilir, istediğiniz etiketi ekleyip, kaldırabilirsiniz.

İlk olarak Dosya > Yeni menüsünü kullanarak yeni bir çalışma sayfası açıyoruz. Sayfamızı açtıktan sonra Dialogs > Genel > Hızlı Başlangıç Yolu ile meta taglarını tanımlayacağımız penceremizi açıyoruz. Aynı işlemi Shift+Alt+Q tuş kombinasyonu ile de gerçekleştirebilirsiniz.

Pencereyi açtıktan sonra istediğiniz meta taglarına parametre verebilirsiniz. Parametre vermek istediğiniz seçeneğe çift tıklayarak içeriğini değiştirebilirsiniz. İstemediğiniz bir meta tagını ise seçip Kaldır seçeneğiyle sayfanızdan kaldırabilirsiniz. Başlık kısmına sayfanızın üstünde gözükken (title) içeriğini girebilirsiniz. Sayfanızın DocType ayarını da pencerenin en üst kısmındaki (DTD) açılır listeden yapabilirsiniz.



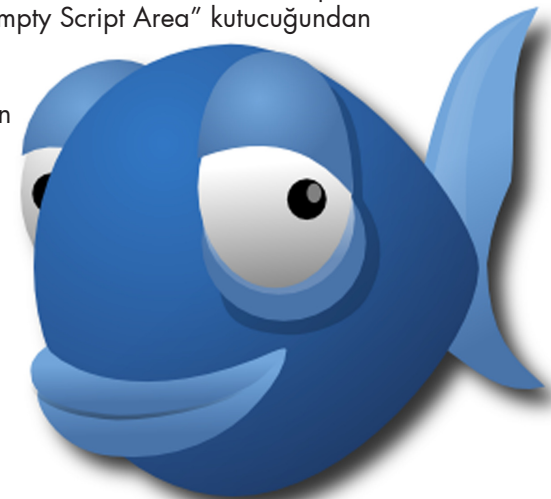
```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3 <head>
4 <title>Deneme</title>
5 <meta name="generator" content="Bluefish 2.0.0" >
6 <meta name="author" content="Tuğsan UNLÜ" >
7 <meta name="date" content="2010-03-28T15:59:54+0300" >
8 <meta name="copyright" content="" >
9 <meta name="keywords" content="" >
10 <meta name="description" content="" >
11 <meta name="ROBOTS" content="NOINDEX, NOFOLLOW" >
12 <meta http-equiv="content-type" content="text/html; charset=UTF-8" >
13 <meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8" >
14 <meta http-equiv="content-style-type" content="text/css" >
15 <meta http-equiv="expires" content="0" >
16 <meta http-equiv="refresh" content="5; URL=http://" >
17 <link href="site.css" rel="stylesheet" type="text/css" >
18 <script type="text/javascript" >
19 <!--
20
21 // -->
22 </script>
23 </head>
24 <body>
25
26 </body>
27 </html>
```

Sol kısımdaki Style sekmesinden harici veya dahili stil şablonlarınızı tanımlayabilirsiniz. Eğer dâhili stil şablonu kullanacaksanız Style sekmesinden "Create Empty Style Area" kutucuğunu işaretlemeniz gerekiyor. Harici stil şablonu kullanacaksanız "External Style Sheet" seçeneğinden "href" kısmına stil şablonunuzun adresini yazmalısınız.

Script sekmesinden ise sayfanızda kullanacağınızı harici ya da dahili JavaScript kodları için şablon oluşturabilirsiniz. Yine kaynak kısmında harici JavaScript dosyanızın adresini yazabilir ya da "Create Empty Script Area" kutucuğundan boş bir script alanı oluşturabilirsiniz.

Son olarak ise pencerenin alt kısmında bulunan "Open in New Document" kutucuğundaki seçimi, yaptığımız ayarları bulduğumuz sayfaya yerleştirmesi için kaldırıyoruz. Eğer kaldırmazsak kodları yeni bir çalışma sayfası açarak uygular.

Tamam'a basıp pencereyi kapadığınızda tek satır kod yazmadan sayfa şablonunuzun



paket tanıtımı

oluşturduğunu göreceksiniz. Kodlarda yazım denetimi olduğundan daha önce kapatmadıysanız muhtemelen altı çizili gözükecektir.

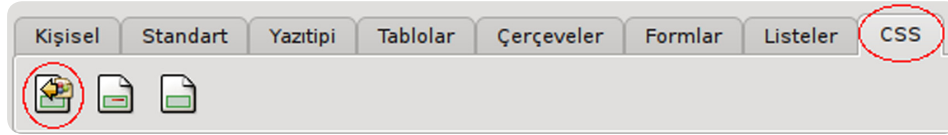
Yazım denetimini Döküman > Spell Check yoluyla kapatabilirsiniz.

Bluefish ile Görsel Arayüz Kullanarak Stil Sayfası Hazırlamak

Bir web sitesi hazırlamaya kalktığınızda web sayfanızın görselliği için bir stil sayfası yazmalısınız. Bu stil sayfasında sayfanızda yazı tiplerinin (font) özellikleri, sayfa renkleri gibi değerleri tanımlamalısınız.

Kullanılan CSS kodları uzun olduğundan, çoğu kişi bu kodları ya da parametreleri unutabiliyor. Bluefish'in en güzel özelliklerinden birisi de tek satır kod bile bilmeden kullanıcıların bu işlemlerin hepsini yapabilmesini sağlamak. Şimdi kullanımına bir bakalım.

Bluefish'i çalıştırdıktan sonra üstteki sekmelerden CSS sekmesine tıklıyoruz. Açılan menüden en soldaki Stil Sayfası Yarat seçeneğine tıklıyoruz.

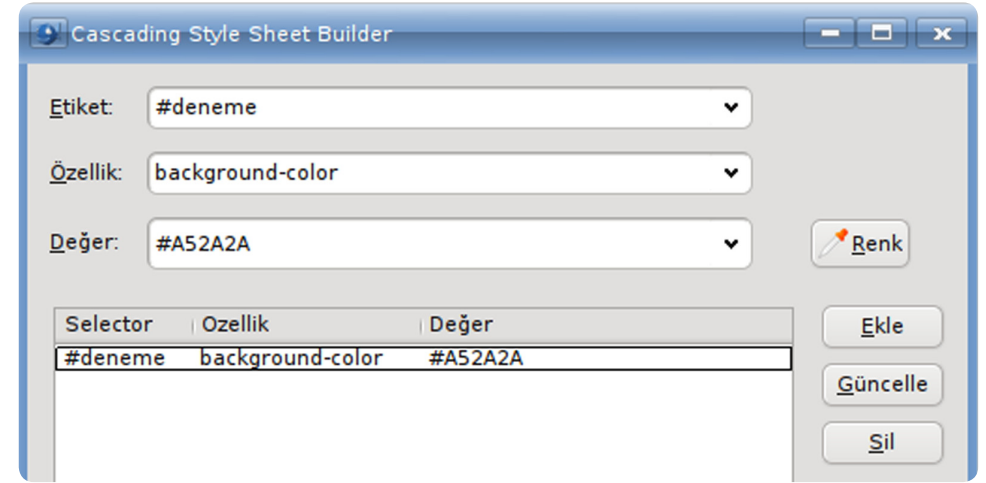


Stil sayfası yarat seçeneğine tıkladığımızda karşımıza katmanın adını ve özelliklerini girebileceğimiz bir pencere açılıyor.

Bu pencerede etiket kısmına oluşturacağımız katmanın adını, özellik kısmına tanımlayacağımız özellikleri açılır liste sayesinde kolayca seçiyoruz ve değer kısmında da bu özelliğin ne olacağını tanımlıyoruz.

Ekle diyerek yeni bir stil oluşturmuş oluyoruz. Tanımladığımız bütün stiller seçim yaptığımız alanın altında listelenir. Buradan silebilir ya da seçenekleri değiştirebilirsiniz.

Görsel arayüz kullanarak üstünde değişiklik yapmak istediğiniz katmanın başlangıç (#) ve bitiş (}) arasındaki kodları seçip, tekrar stil yarat düğmesine tıklayarak açılan pencereden "Güncelle" seçeneği özellikleri düzenleyebilirsiniz.

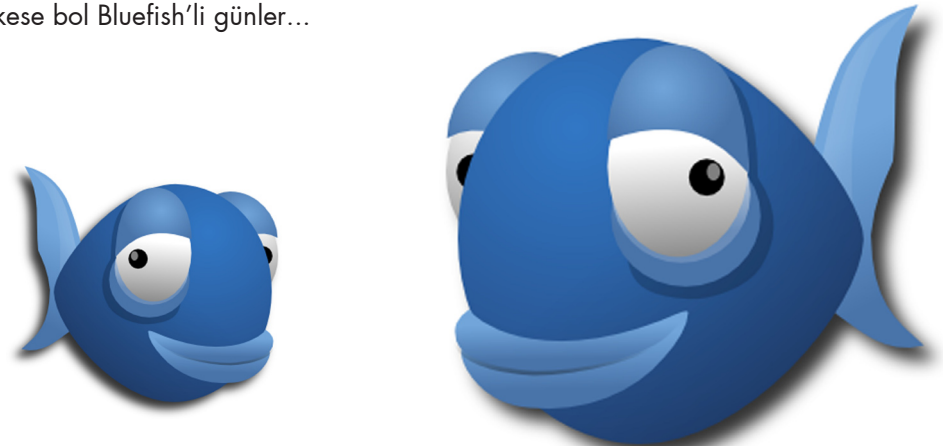


Tamam'a tıklayıp pencereyi kapattığımızda görsel olarak tanımladığımız özelliklerin arkaplane yerleşen CSS kodlarını görebilirsiniz.

```
1 #deneme {  
2   background-color: #A52A2A;  
3 }
```

Ayrıca katman ekleme, stil tanımlama gibi işlemleri yine CSS sekmesindeki üçüncü düğme olan "Div" düğmesinden de yapabilirsiniz.

Herkese bol Bluefish'li günler...



Kotalı FPS müptelalarının
dikkatine!

ASSAULT CUBE



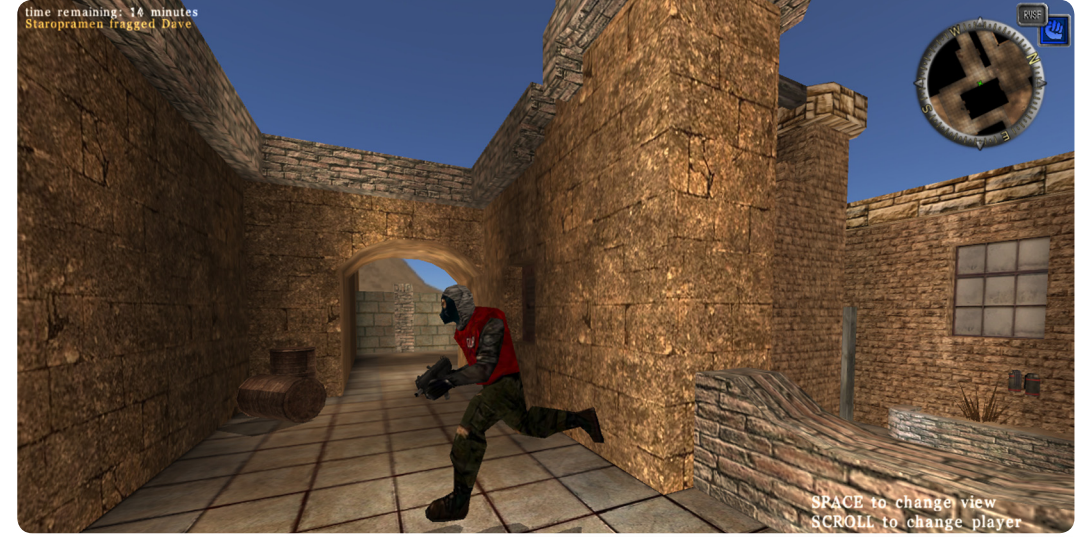
Karşınızdaki 56 K dial-up modemlerle bile rahatlıkla oynayabileceğiniz,
son derece hızlı bir takım oyunu...

oyun inceleme

AssaultCube oyunu için Counter-Strike oyunun özgür sürümü olarak benzetmeler yapılıyor. Oyunun birkaç ufak kusuru olmasına rağmen hız ve gerçekçiliğiyle etrafında büyük bir oyuncu topluluğu toplamış.

Cube geliştirici topluluğuna üye birkaç kişi tarafından daha gerçekçi ve hızlı bir oyun yaratmak amacıyla Cube oyun motoru kullanılarak üretilmiş, FPS türü bir oyundur. Çoklu oyuncu modunda çevrimiçi oynanabildiği gibi, tek oyuncu modunda da bilgisayarın yönettiği botlara karşı da oynanabilir.

Oyunun paket boyutunu diğer kendi türündeki FPS oyunları ile kıyasladığınızda epey küçük. Ülkemizde İnternet kullanıcılarının çoğu kotalı aboneler. Bu oyun onlar için İnternet'te kota dostu FPS oyunu diye tanımlanıyor. Paket boyuna bakıp yeteneklerini küçümsemeyin her ne kadar Counter-Strike oyununa benzese de oynayınca birçok noktada ondan daha iyi olduğunu göreceksiniz.



AssaultCube oyunu ücretsiz olarak Windows, Linux ve Mac OS X platformları için indirilip oynanabilir. Elbette Pardus depolarından Paket Yöneticisi ile kolayca kurma rahatlığı siz Pardus kullanıcılarının :).

Gücünü Cube oyun motorundan alıyor

AssaultCube 2004'te ActionCube adıyla geliştirilmeye başlanmış olup, ilk resmi sürümü 2006'da yayınlanmış. Planlanan, Cube motoru ile daha sade ve gerçekçi ortamlar içeren bir oyun yaratmak.

2007 yılında geliştiricilerden gelen istek doğrultusunda oyunun adı AssaultCube olarak değiştirilmiş.

AssaultCube sade ve gerçekçi grafiklerine ve hızlı oynanışı her daim

Oyun Kontrolü

İleri: W
Sol: A
Geri: S
Sağ: D
Ateş: Fare 1
Özel Hareket: (Tanımlamanız gerekiyor)
Zıplama: Boşluk tuşu
Eğilme: Sol Shift
Silaha Mühimmat Doldurma: R
Silah Menü: B
Sonraki Silah: Fare 4
Önceki Silah: Fare 5
Chat: T
Takım Chat: Y
Komut Satırı: Slash (/ veya ")
Konsol Çıktısı, Ekran Görüntüsü, Harita Krokisi, (Bayrağı atma, Tam ekran ya da daha fazlası için ayarlardan tuş tanımlamanız gerekmektedir.



çevrimiçi sunucular üzerinde yüzlerce oyuncuyu toplayabilen heyecan dolu bir oyundur.

Oyun hızlı oynanışı ve takıma dayalı bir oyun türü. Kendi türündeki diğer oyunlar ile kıyaslandığında AssaultCube Quake'ten daha yavaş ancak Counter-Strike'tan daha hızlı bir oyun. Alien Arena ve Quake III oyunları tadında oyun arayan FPS severlerin büyük bir beğeniyle oynayacakları bir oyun. Oyundaki gerçekçilik kameranın açısı ile bazen can sıkıcı olsa da bu da oyuna farklı bir eğlence katıyor.

Oyunu haritalar konusunda biraz eksik bulanlar olabilir. Diğer FPS oyunları ile karşılaştırıldığında az sayıda harita geliyor. Oyun kurulumu ile 20 harita geliyor. Diğer özgür FPS oyunlar gibi AssaultCube oyununda da harita düzenleyicisi var. Harita düzenleyicilerinin ellerinden öper. Bir gün biri çıkıp, Pardus temalı bir harita yapsa ne güzel olurdu diye aklından geçiriyor insan.

Oyunda sesler de grafikler gibi sade ve gerçekçi. Oyunda sadece silah sesi, ayak sesi gibi hareketlerin seslerinden başka bir ses pek yok. Arka fonda bir müzik çalmıyor mesela. Oyunun paket boyutunu ufak tutmak için birebir bir tercih.

Silah modelleri farklı

Oyunda üyesi olacağınız iki takım var. Cubers Liberations Army (CLA) ve Rabid Viper Special Forces (RVSF). Oyunda kullanılan silahlar da oyunun gerçekçi olma hedefinden nasibini almış ve ateş ettiğinizde silahın tepme hareketini hesaba katmayı unutmamalısınız.

Cube'tan gelen en önemli özelliklerden biri de oyuncuların hareketlerinin son derece keskin oluşudur. AssaultCube'de silahlar tüm modern FPS oyunlarında olan bilindik silahlardır: Saldırı tüfeği, makineli tüfek, keskin nişancı tüfeği, tabanca, bıçak ve av tüfeği. Oynadığınız harita üzerinde yine diğer oyunlardaki gibi toplamanız gereken mermi, el bombası, ilk yardım çantası ve çelik yelek var. Bir diğer seçenek de çift tabanca var olabilir. Çift tabanca ile saldırı son derece eğlenceli, tavsıye ederim.

AssaultCube tek kişilik bilgisayara karşı bot ve çevirim içinde çok oyunculu birçok oyun modu içeriyor.

- **Deathmatch / Team Deathmatch:** Herkesi öldürmek ya da karşı takımdan herkesi öldürmek.
- **One Shot One Kill and Team One Shot One Kill:** Keskin Nişancı Tüfeği ile.
- **Last Swiss Standing** (Bıçak ve El bombası Sadece)
- **Survivor and Team Survivor** (Karşı Takımı Yokedene Kadar)
- **Pistol Frenzy** (Tabanca, bıçak ve el bombası sadece)
- **Capture The Flag** (Bayrağı saklamak)
- **Keep The Flag and Team Keep The Flag** (Bayrağı bulmak ve en uzun elde tutmak)
- **Hunt The Flag**

Bilgiler

Lisans: GPL

Paket Boyutu: 36.1 MB

Kurulu Boyut: 47.5 MB

Web Sitesi: <http://assault.cubers.net>

İlgili Wiki Sayfası: [AssaultCube](#)

Değerlendirme

Öğrenme süresi: 1 saat

Oynanabilirlik 9/10

Grafik 8/10

Ses 7/10

Senaryo 7/10

Atmosfer 7/10

7.6

Menüdeki yeri

Uygulamalar > Oyunlar > Oyun Makinesi > AssaultCube

Pardus'un Gizli Kahramanları:

Test Takımı

Siz son kullanıcılar Pardus ile oyun oynarken ya da sıcak yataklarınızda uyurken; sizin yerinize sorunları yaşayan, sorunları size gelmeden yakalayan birileri var...



test süreçleri

Test takımı üyelerini belirli bir kalıpla sınırlamak yanlış olur. Bu yazının devamında bazı testçilerimize sorduğumuz soruların cevaplarından da göreceğiniz gibi, her yaştan, her meslekten takım arkadaşımız var. Bilgisayar ve İnternet'in yaygınlaşmasıyla insanların bu teknolojilere ilgisi ve bilgisi arttı. Test süreçlerimiz de gerek kılavuzlar gerekse mail listemizin yardımıyla öğrenilebilir aşamalardan oluşmakta.

Sanılanın aksine, testçi olmak için bilgisayar mühendisi olmanız şart değil. Özgür yazılımları kullanmaya istekli ve aşına olan ya da öğrenmeye meraklı olan sabırlı tüm kullanıcılar testçi olabilir. Elbette bazı test süreçlerinde belirli bir seviye teknik bilgi gerekiyor. Tüm bu bilgiler [e-posta listesi](#), [PardusWiki](#), [forum](#) veya IRC'den ulaşılacak uzaklıkta.

Test takımına kimler katılmalı?

- *"Ben kullandığım sistemin inceliklerini derinlemesine öğrenmek istiyorum."* diyen Pardus kullanıcıları.
- *"Ben Pardus geliştiricisi olmak istiyorum."* diyorsanız, yine iyi bir başlangıç noktalarının başında Test Takımı gelmektedir.
- *"Ben Pardus fanatığıyım, her türlü desteği vermek için can atıyorum bana nerede ihtiyaç var?"* diyorsanız, Test Takımı'na takım arkadaşı arıyoruz, buyurun.
- *"Ben bir bilişim öğrencisiyim ve okullarda açık kod yazılımlarla ilgili yeterli müfredat yok bu konuda da eksik kalmak istemiyorum"* diyorsanız, gelin hep birlikte öğrenelim.

Gelin, siz de bu yeni nesil imeceye ortak olun ve Pardus'a test yaparak katkı verin. Sistemimi biçimlendirip yeniden sistem kurmaktan bıkmayan, bunu yaparken sisteminde oluşabilecek veri kayıplarını göze alabilecek, gözü kara kullanıcıları **Test Takımı'**na bekleriz.

Testçi olmaya karar verdiğinizde sizden istenen seviyede olmanız için ufak bir sınav uygulanıyor. Bu sorular İnternet'ten araştırılabilir tarzda olduğu için kopya çekebilirsiniz :).

Burada amaç, zaten kopya çekmeniz. Kopya çekmek için istenen soruların cevaplarını PardusWiki'den araştırırken zaten istenen bilgi seviyesine gelmiş oluyorsunuz.



Ad Soyad: Aydın DEMİREL

Meslek: Harita Mühendisi / Sistem Yöneticisi

Yaş: 30

Test Takımı'na katılma sebebiniz: Pardus'un gelişimine katkıda bulunmak

Test süreçleri sizi zorluyor mu: Hayır

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Kullandığınız sistemi daha

yakından tanımak ve yapılan güzel işlerde sizin de

bir katkınız olmasını istiyorsanız, hiç tereddüt etmeyin derim.

Test Takımı'na katılmak size ne kazandırdı: Test Takımı'na katılarak kullandığınız sistemi daha yakında tanıma fırsatı buluyorsunuz.

Yapmış olduğunuz testlerin sistemi kullanan her seviyedeki kullanıcıyı düşündüğünüzde, sorumluluğunuzu artıran bir yanı var.



Ad Soyad: Süleyman BALIKÇI

Meslek: İnşaat Mühendisi

Yaş: 30

Test Takımı'na katılma sebebiniz: Hem

yenilikleri ilk kez deneme şansı elde etmek

hem de yeni sürümlerin ve programların farklı

bilgisayarlarda denenmesine yardımcı olmak.

Test süreçleri sizi zorluyor mu: Bazen.

Özellikle kişisel işlerim olduğunda, yoksa diğer türlü

test ederken zorlanmak söz konusu bile olamaz :).

Test Takımı'na katılmak size ne kazandırdı: Birçok şeyi nasıl yapacağımı öğrendim. Sorun çıktığı zaman nasıl yaklaşmam gerektiğini de öğrenmeme yardımcı oldu

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz:

Kesinlikle katılın. Daha çok donanımda test edilmek, daha kararlı bir Pardus demek :).



Ad Soyad: Uğur KAYKUSUZ
Meslek: Bilgisayar Programcılığı (Öğrenci)
Yaş: 20

Test Takımı'na katılma sebebiniz: Pardus sistemini iyi bir şekilde tanımak sistemi tam anlamıyla öğrenip geliştirici olmak.

Test süreçleri sizi zorluyor mu: Hayır.

Test Takımı'na katılmak size ne

kazandırdı: Pardus sistemini daha yakından

tanımamın yanında. Pardus depolarının işleyişini, paketlerin büyük bir kısmının ne işe yaradığını gibi konularda epey bilgilenmemi sağladı.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Eğer siz de geliştirici olmak istiyorsanız, Test Takımı'na katılın. Elbette geliştirici olmak şart değil :). Bunun dışında Pardus'a yardımcı olmak istiyorsanız, Test Takımı'na katılın.



Ad Soyad: Fahri DÖNMEZ
Meslek: Bilgisayar Programcısı
Yaş: 30

Test Takımı'na katılma sebebiniz: İşim gereği tüm platformlar hakkında kendini geliştirmek.

Test süreçleri sizi zorluyor mu: Test edilecek paket listesi uzun olduğunda belki:). Hepsini test etme zorunluluğu yok tabii. E-posta listesi üzerinden diğer test takımı üyeleriyle paylaşabiliyoruz.

Test Takımı'na katılmak size ne kazandırdı: Kendimi Linux üzerinde geliştirmek için başlamıştım. Güzel ülkemde doğru bir işe başladın mı devam ettirmeyi, takım çalışmasını ve güzel insanlar ile güzel insanlara güzel bir şeyler yapma zevki en büyük kazancım oldu.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Bilgisayar ve Linux sizin için bir tutku ve işin mutfağını yakından görmek istiyorsanız en önemli başlangıç noktalarından biri Test Takımı'dır.



Ad Soyad: Server ACİM
Meslek: Besteci ve Öğretim Üyesi
Yaş: 49

Test Takımı'na katılma sebebiniz: Test tecrübesi kazanmak ve Pardus'a bir de bu parametreyle katkıda bulunmak.

Test süreçleri sizi zorluyor mu: Çok zorladığını söyleyemem. Ancak dizüstü bilgisayarımın donanımsal sınırları bazen zorluyor.

Test Takımı'na katılmak size ne kazandırdı: "Testçi olmak" kavramını yaşayarak öğrendim. "Hata bildirme terbiyesi" konusunda tecrübe kazandım.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Sorunlarla "sabırla" mücadele etmeyi öğrenmelerini öneririm. Sorunların çözümünde katkınızın olmasını ne kadar istediğinizi kendinize sormalısınız ve eğer buna hazırsanız Test Takımı'na katılmanızı öneririm.



Ad Soyad: Mustafa ÖZBAKIR

Meslek: Memur

Yaş: 32

Test Takımı'na katılma sebebiniz: Pardus'un gelişmesine katkıda bulunmak.

Test süreçleri sizi zorluyor mu: Hayır.

Test Takımı'na katılmak size ne

kazandırdı: Pardus'u iyice tanımamı ve hiç bilmediğim birçok uygulamayı öğrenmemi sağladı.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Biraz meraklı, araştırmacı, elindekiyle yetinebilen fakat her seferinde çitayı bir basamak yukarı çıkartma çabası olan bir kişiliğiniz varsa, hiç durmayın buyurun eğlenceye :-).



Ad Soyad: Bilal ÖZÇAKIR

Meslek: Öğrenci

Yaş: 23

Test Takımı'na katılma sebebiniz: Ulusal ve özgür bir sistem olan Pardus'un gelişimine katkıda bulunabilmek ve çıkan Pardus sürümlerini kullanan ilk insanlardan olabilmek.

Test süreçleri sizi zorluyor mu: Paketler listeleri uzun olduğu zaman biraz sıkıcı olabiliyor.

Test Takımı'na katılmak size ne kazandırdı: Linux bilgilerimi geliştirdi ve bir şeyin gelişmesinde katkıda bulunabilmiş olma bilinci kendime olan güvenimi artırdı.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz:

Gerçekten Linux ve Özgür yazılımı destekliyorsanız ve bilgisayar başında boş boş oturmak bile kendinizi iyi hissetmenizi sağlıyorsa, bu süreçte yer almak tam size göre :-).



Ad Soyad: Cemalettin KARA

Meslek: Öğrenci

Yaş: 21

Test Takımı'na katılma sebebiniz: Pardus'a katkıda bulunmak, linux bilgimi genişletmek

Test süreçleri sizi zorluyor mu: Pek değil.

Test Takımı'na katılmak size ne

kazandırdı: Linux hakkında teknik bilgim arttı. Sorunları çözüm yolları üretebilmemi sağladı.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz:

Pardus'a teknik anlamada destek vermek, Linux'a dair bir şeyler öğrenmek isteyen 7'den 77'e herkesin kesinlikle katılmasını öneririm.





Ad Soyad: Ali Haluk SEVEN

Meslek: Öğrenci

Yaş: 21

Test Takımı'na katılma sebebiniz: Pardus'a katkı, pisi teknolojisini yakından görmek.

Test süreçleri sizi zorluyor mu: Son zamanlarda vakit darlığından pek katılamasam da pek zorladığını hatırlamıyorum.

Test Takımı'na katılmak ne kazandırdı:

Bir hatayla karşılaştığımda ne yapmam gerektiğini öğrendim. Bir süre sonra, hatayı gördüğümde neyle ilgili olduğunu doğru tahmin etmeye ve bununla birlikte kendi işimi kendim görmeye başladım. Bugzilla' yı keşfettik, geliştiricilerle daha yakınlaşma fırsatımız oldu.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz: Sabır.



Ad Soyad: Özgür ERDOĞRU

Meslek: Makine Mühendisi

Yaş: 35

Test Takımı'na katılma sebebiniz: Bilgi edinmek ve Pardus'un gelişimine katkı.

Test süreçleri sizi zorluyor mu: Hayır.

Test Takımı'na katılmak ne kazandırdı:

Aslında eski bir Pardus kullanıcısıyım ama test ekibine yeni katılan biri olarak, detayları incelemek durumunda kaldım ve Linux hakkındaki bilgilerim oldukça arttı.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz:

Öncelikle biraz inatçı olmak ve karşılaşılabileceğiniz zorluklar, sıkıntılar karşısında yılmamak gerektiğini düşünüyorum. Acele edilmemeli, iyi bir Pardus kullanıcısı olunmalı, işleriniz için diğer işletim sistemine ihtiyaç duymamanız önemli. Edineceğiniz bilgi ve beceriye hazır olun ve en önemlisi PC'nizdeki önemli bilgileri her zaman yedekleyin :).



Ad Soyad: Hüseyin BİLEN

Meslek: Emekli

Yaş: 67

Test Takımı'na katılma sebebiniz: Sistem tepkileri, uyumsuzluk ve yakalayabildiğim hataları bildirerek Pardus'u geliştiricilerine yardımcı olmk.

Test süreçleri sizi zorluyor mu: Hayır.

Test Takımı'na katılmak ne kazandırdı:

İşletim sistemini daha iyi tanımamın yanı sıra

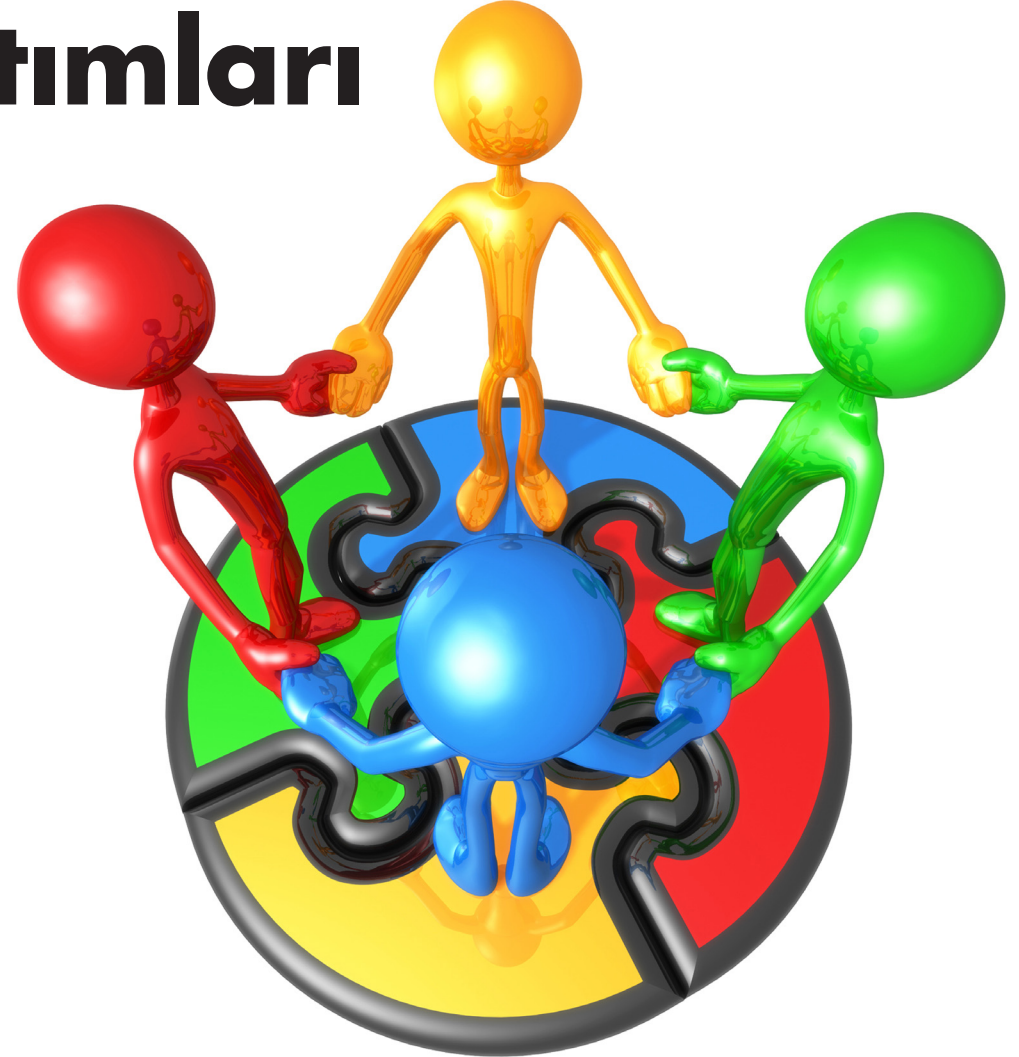
bilgisayar bileşenlerinin davranışlarını, verimli kullanımını ve onun yanı sıra birkaç upgrade ile anakart, işlemci, RAM bellek, ses/grafik kart uyumları ve işlemci besleme voltajını % 5 ila % 10 artırarak performans artırımını.

Test Takımı'na katılmak isteyenlere ne tavsiye edersiniz:

Muhtemel uyumsuzlukları yakalayıp veya test yaptığınızda sorunla karşılaşmadığınızda "OK" vererek gelişime katkı vermenin hazzını yaşarsınız. Bu arada konsol hakimiyetiniz de artacak.

QA ve Linux Dağıtımları

Ufak bir hata, milyonlarca dolarlık projelerin bir anda yok olmasını sağlayabilir.



test süreçleri

Günümüzde QA, Yazılım Sektörü için olmazsa olmazlardan. Özellikle benim de çalıştığım oyun sektöründe bu işin çok önemli olduğuna tanıklık edebiliyorum. Her yıl birçok oyun firması yazılımlarını (oyunlar) test etmek için büyük bütçeler ayırıyor ve sürekli yeni elemanlarla ekiplerini büyütüyorlar.

Ürününüz sizin markanız, prestijinizdir. Ürününüz kullanıcı gereksinimlerini ne kadar karşılayabilirse o kadar başarılıdır. Ortaya çıkan ya da çıkabilecek sorunları kullanıcılar fark etmeden hatta üründe ortaya çıkmadan önlemek yazılım sektöründe nasıl bir yer edineceğinizi belirleyicisi olabilir. Ufak bir hata milyonlarca dolarlık projelerin bir anda yok olmasını sağlayabilir. Karmaşık sistemler bir noktada birbirlerini tetikleyen hataların kurbanı olabilir ve bunu önlemek için kalite kontrolü şarttır. QA aslında tam olarak kalite kontrolü (QC) yapmaz, daha çok kaliteye ulaşmayı amaçlar. Türkçe’de kalite güvencesi, kalite teminatı gibi bir anlama gelen Quality Assurance kelimelerinin kısaltılmışıdır.

Ürün gelişimine dâhil olan QA, ürünün gelişimi, izlenmesi, standartlara ve prosedürlere uygunluğunun takibini yapar. Linux projelerinin bazıları bu işi sadece test ekibiyle yürütürken Ubuntu ve Debian gibi dağıtımlar QA ekibi aracılığıyla projelerini geliştiriyor. Kapalı kodlu firmalarda ticari sır niteliğindeki bilgilerin sızmamaları için QA ekibi geliştiricilerle birlikte yan yana çalışırken, özgür yazılımlarda yine gönüllülük esasıyla hareket edilebiliyor. Özgür yazılım projelerinde de geliştiricilerle QA ekibi sürekli bağlantı halinde oluyor.

Bir projenin aşamaları

Projelerin kontrolünde ilk aşama, amaçların ne olduğunun kararlaştırılması oluyor. Bir ürün hazırlanacak ve bu ürün şu işi yapacak, şu şekilde çalışacak ve şuna uygun olacak şekilde bir başlangıçla ürün geliştirilmeye başlanıyor. Nasıl geliştirileceği konusunda bir plan yapılıyor ve prosedürler belirleniyor. Ürün geliştirilirken bu plana uyulup uyulmadığı kontrol ediliyor.

Ürün gelişip ortaya ilk sürümü çıktıktan sonra QA ekibinin test kolu ürünü incelemeye alıyor. Çalışıp çalışmadığı kullanılabilirliğini, nelerin eksik olduğunu kontrol edip rapor yazıyor ve geliştirici ekibe iletiyor. Geliştirici ekip hataları düzeltip, iyileştirmeleri yaptıktan sonra ürünü tekrardan teste açıyor ve gereksinimler karşılanmışsa ürün müşterilere sunuluyor.

Çok kısaca özetlediğim bu süreçler sürekli olarak tekrarlanıyor çünkü farklı özellikler eklendikçe her bir güncelleme için yazılım tekrardan kontrolden

geçiyor. Linux dağıtımlarında paketlerin güncellenmesi buna bir örnek. Bir paket güncellenir ama yayınlanmadan önce test edilmesi gerekir. Mesela bir ekran kartı sürücüsü güncellenecektir, geliştirici sadece kaynak kodu paketleyip kullanıma sokar ama QA ekibi bu paketin doğru şekilde paketlenmesinin ve sunulmasının yanında kullanıcıların bu paketi kullanmaya başladıktan sonra grafiklerinde, ekran kartlarında bir sorun olup olmadığını da farklı ekran kartlı bilgisayarlarda test eder. Eğer sürücüde sorun varsa ve kontrol edilmeden kullanıma sunulursa tüm müşterilerin ekran kartı yanabilir, sistemleri arızalanabilir. Böyle bir durumda dağıtımın alacağı darbeyi söylememe gerek yok...

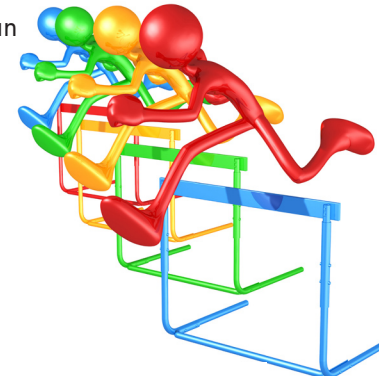
QA ekibinin önemi

QA ekibinin test süreçlerini belli standartlara bağlaması ve olabildiğince farklı sistemde ürünü denemesi çok önemlidir. Test örnekleri hazırlayıp bunu testçilere dağıtımından sorumludur. Sadece süreçleri, arayüzü, çevirileri değil aynı zamanda yazılımların bazı fonksiyonların testlerini de yapacaklarından bunlar için uygun betikler yazarlar. Bazı durumlarda kontrol ya da test için gerekli uygulamaları bizzat kendileri geliştirirler. Masaüstü uygulamalarının testi için yazılmış betikleri, ISO’ların testi için uygulamalar, arayüz test programları, stres test betikleri, paket kontrol sistemleri, hata takip sistemleri gibi örnekler bu uygulamalar için verilebilir. Çevirilerin kontrolü bile bir anlamda QA ekibinin sorumluluğundadır. Bu yüzden çeviri ekipleri de QA ile birlikte çalışırlar. Farklı dillerde yazılımda bir hata olup olmadığını da test ederler.

QA ekibi ürünün gelişimini inceler, güncellemeleri kontrol eder. Ama iş bununla bitmez. Ürünün geliştirilmesi aşamasında kullanıcılardan gelen geri bildirimler değerlendirmeye alınmalıdır. Kullanıcıların beklentilerini ölçmek adına çeşitli anketler düzenlenebilir ya da Pardus için hazırlanmış olan [Özgürlükçün Beyin](#) bölümü gibi arayüzler kullanılarak kullanıcıların fikirleri alınabilir. Bu şekilde kullanıcı beklentileri belirlenerek ürünlerin bu beklentileri karşılayıp karşılamadığı kontrol edebilir ve eğer karşılamıyorsa bu yönde çalışmalar yapılabilirler.

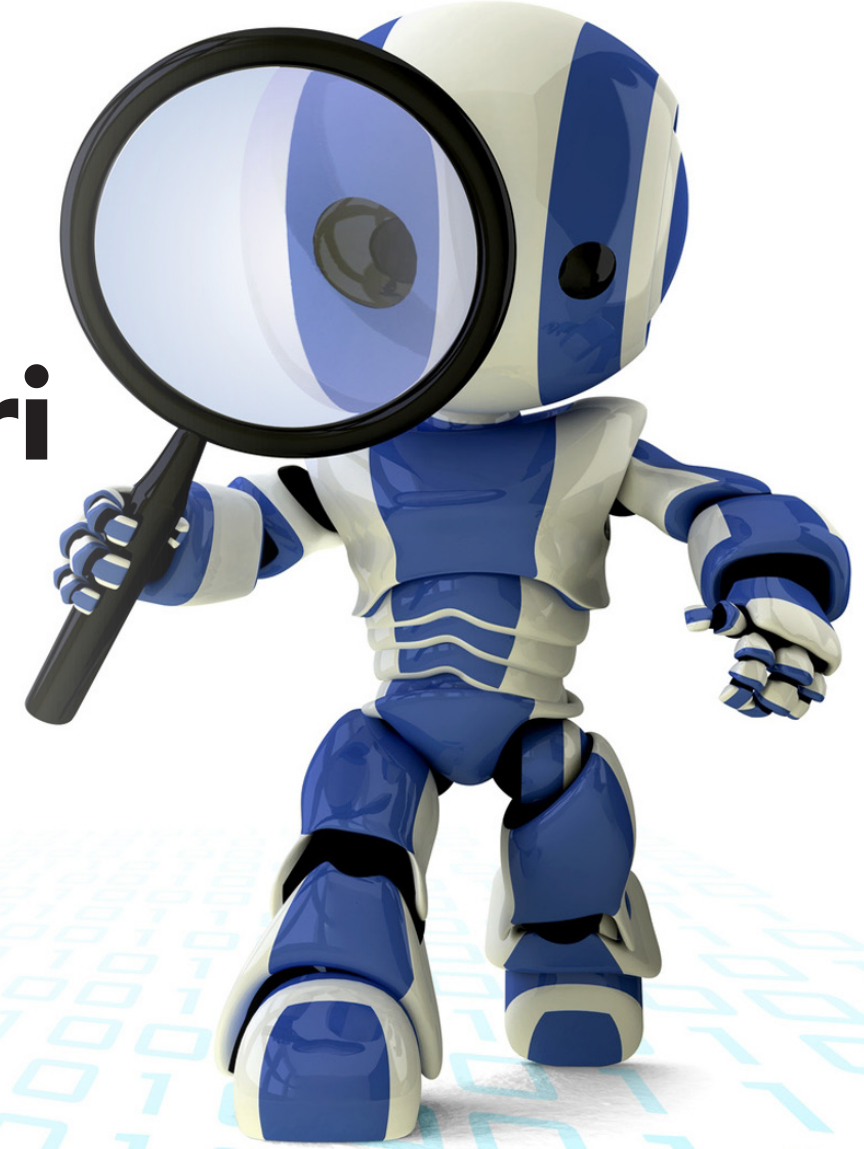
Linux dağıtımları tüm insanlığın ürünü olduklarından kullanıcıların istekleri doğrultusunda yön almaları çok önemlidir. Kullanıcılar aynı zamanda dağıtımların geliştiricileri de olabilir.

Onlarla olabildiğince yakın bir ilişki kurulursa, kapalı kaynak kodlu ürünlere göre daha “kullanışlı” ve “kullanıcı dostu” ürünlere kavuşabiliriz.



Pardus Test Süreçleri ve İyileştirmeler

Geliştiricilerimizden Semen CİRİT,
gelecekte Pardus Test Süreçlerinde
yapılması düşünülen iyileştirmeleri
anlatıyor... Kaçırmayın!



test süreçleri

Test süreçlerini anlatmaya başlamadan önce Pardus Linux dağıtımının derlenmiş depolarından ve paket kavramından bahsetmenin iyi olacağını düşünüyorum. İlk olarak Linux dağıtımlarının en küçük yapıtaşı olan paketlerden bahsedelim. Kaynak kodu, belirli bir Linux dağıtımına göre derlenmiş ve bu dağıtıma kurulabilir hale getirilmiş ikili (binary) biçimindeki dosyalara paket denir. Bu paketlerin saklandığı alanaysa derlenmiş depo (binary repository) denilir. Pardus'ta iki ana derlenmiş depo bulunuyor: Test deposu ve kararlı depo.

Geliştiricisi tarafından pakete yansıtılan herhangi bir değişiklik Pardus'un kullanmış olduğu derleme çiftliğinde otomatik olarak derlenip test deposuna aktarılıyor. Diğer bir deyişle test deposu, geliştiricilerin yaptığı değişikliklerin test sürecinden geçmeden aktarıldığı depodur. Değişiklik yapılan ya da yeni test deposuna giren paketler kararlı sürüm test sürecinden geçip kararlı depoya aktarılır.

Alfa-Beta test süreci

Kararlı sürüm öncesi dönemde gerçekleştirilen testler ve sonuçları, Pardus'un kararlı sürümünü büyük oranda etkileyecek bir güce sahip. Alınan test sonuçları, uygulamaların hatalarının kapatılmasının yanı sıra, bir uygulamaya yeni bir özellik eklenmesi, bir uygulamanın kullanılabilirliğinin artırılması gibi durumlara olanak sağlamakta.

Alfa ve Beta sürümleri, Pardus'un kullanıcı karşısına çıkan test sürümleridir. Bu sürümler boyunca sadece test deposu kullanılmakta ve bir uygulama üzerinde gerçekleşen herhangi bir yenilik ve özellik doğrudan kullanıcılara sunulmaktadır. Böylece tüm Pardus kullanıcıları, son kullanıcı test sürecine (Son Kullanıcı Testi: Kullanıcının uygulamadan beklentilerinin sağlanıp sağlanmadığının kullanıcı tarafından test edilmesidir.) dahil olup karşılaştıkları hataları hata takip sistemi üzerinden bildirebiliyorlar.

Alfa ve Beta sürümlerinin testleri öncelikle Pardus teknolojilerinin ayrıntılı bir şekilde ve belirli bir yöntem kullanarak test edilebilmesi açısından çok önemlidir. Test ekibi üyeleriyle birlikte, Pardus teknolojileri, test yönergeleri kullanılarak formal bir şekilde test edilmekte ve hatalı sonuçlar Pardus hata takip sistemine raporlanmakta. Test Yönergeleri, bir uygulamanın belirli yöntemlere bağlı kalınarak test edilmesini sağlayan bir yapıdır. Test aşamaları, belirli bir uygulama için verilebilecek girdileri ve bu girdiler verildiğinde uygulamadan beklenen sonuçları içeriyor. Eğer verilen girdi için beklenen sonuç alınmıyorsa uygulamanın bu bölümünde bir hata var anlamına gelir.

Kararlı sürüm test süreci

Kararlı sürümün çıkışı ve güncellenme süreci birbirinden farklı test süreçlerini kapsıyor.

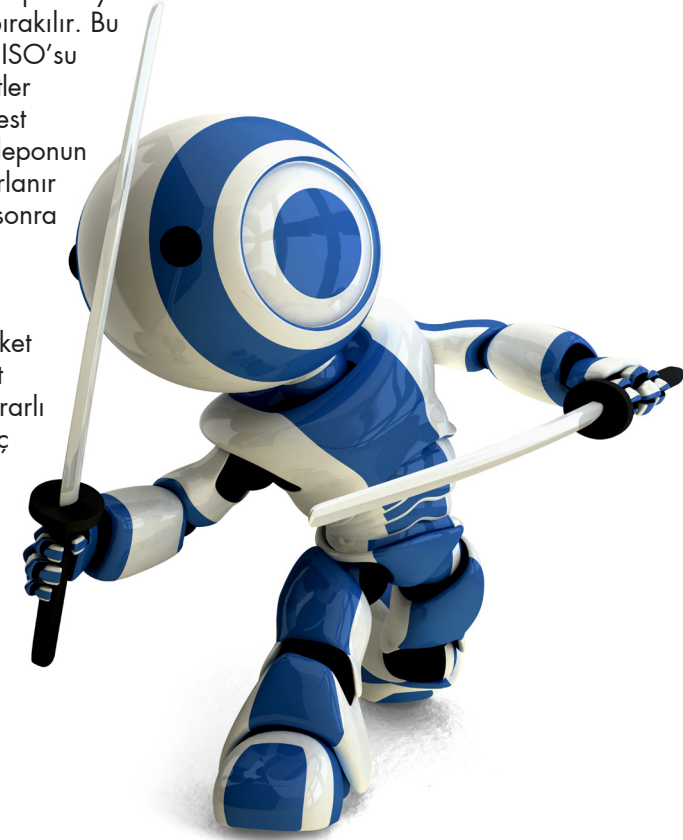
Kararlı Sürüm Çıkış Süreci: Kararlı sürümün çıkmasından yaklaşık üç hafta önce kodda değişiklik yapmayı durdurma (code freeze) evresine ve sürümden bir hafta öncesinde hataları kapatmayı durdurma (bug freeze) evresine girilir. Bu durdurma evrelerinde test ISO'ları (sürüm CD kalıbı) hazırlanıp test ekibine gönderilir. Bu ISO'lar test yönergelerini kullanılarak Pardus teknolojilerinde bulunan hataları, eksik özellikleri bulmayı ve kullanılabilirliği arttırmayı amaçlar.

Kodda değişikliği durdurma evresi sonrasında yapılan ISO'da bulunan ve hata takip sistemine raporlanan hatalar, hatanın önemine göre geliştiricileri tarafından hata kapatmayı durdurma evresine kadar kapatılır. Kararlı sürümün çıkmasına bir hafta kala kapatılmamış durumda olan hatalarsa, kritik olma durumlarına göre, geliştiricileri tarafından acilen kapatılır ya da kritik değilse bir sonraki sürüme bırakılır. Bu son değişikliklerden sonra tekrar test ISO'su yapılmakta ve değişikliklerle ilgili testler gerçekleştirilir. Bu aşamadan sonra test deposu kararlı depoya aktarılır, bu deponun bulunduğu kararlı sürüm ISO'su hazırlanır ve son kez genel testleri yapıldıktan sonra yayınlanır.

Kararlı Sürüm Güncelleme Süreci:

Kararlı sürümün çıkışından sonra, paket güncelleme, yeni paket girişi gibi test deposunda yapılan değişikliklerin kararlı depoya aktarılması için farklı bir süreç izleniyor.

Test deposunda yapılan değişiklikler sonrasında (güncelleme ve yeni paket girişi vb.) kararlı depo ve test deposu arasında paket farkları oluşur. Bu paket farkları sürüm yöneticisi tarafından listelenir ve



test süreçleri

geliştiricilerine yollarır. Geliştiricileri paketlerinin kararlı depoya geçmesi için onay verirler (ACK-NACK). Onay alan paketler test sorumlusu tarafından listelenir ve test ekibi üyelerine gönderilir. Onay alan paketler ve kararlı deponun birleşiminden bir test ekibi deposu oluşturulur ve testçilere bildirilir.

Kararlı sürüm test ortamının hazırlanması

Test ekibi üyeleri paket test yönergelerini kullanarak testlere başlamadan önce test edebilecekleri sistemin uygun olduğuna emin olmalıdırlar. Kararlı sürüm güncellemelerini test edebilmek için kararlı depo bulunan temiz bir sistem gerekmektedir. Bu sisteme test sorumlusunun hazırlanmış olduğu test deposu eklenerek testlere başlanabilir. Ayrıntılı bilgi için Kararlı Sürüm Test Süreci belgesine bakabilirsiniz.

Kararlı sürüm testleri

Test sorumlusu tarafından listelenen kararlı depoya geçecek olan paketler, test ekibi üyeleri tarafından, paket test yönergelerine bağlı kalınarak test edilirler. Alınan hatalar test ekibi üyeleri tarafından raporlanır ve hatalı olan paketler kararlı depoya aktarılacak paketler listesinden çıkarılır. Sorun yaşanmamış paketlerle ilgili sürümün yöneticisine iletilir ve kararlı depoya aktarılır.

Kararlı sürüm test sürecini iyileştirmeleri

Kararlı sürüm test sürecini hızlandırmak ve kaynak kullanımını azaltmak amacıyla iki farklı iyileştirme üzerine düşünülüyor.

Test Yönergeleri Otomasyonu - GSoC Projesi

Paket test yönergelerini çeşitli gruplara ayırmayı ve otomatik olarak test edilecek paketler sayesinde test ekibi üyelerinin karşısına daha az test edilecek paket çıkarmayı amaçlamaktadır.

Otomatik olarak test edilebilecek paketlerin testleri, makine ya da makineler üzerinde çalıştırılıp, hata alınan paketler için Bugzilla'ya otomatik olarak hata girilecek.

Diğer testler içinse test ekibi üyelerinin testleri gerçekleştirmeleri için bir arayüz sunulup ve test yönergeleri için çalıştırılacak olan komutlar bu araç aracılığıyla çalıştırmaları sağlanacak. Test yönergelerinden istenen sonucun alınıp alınmadığı testçiler tarafından arayüz aracılığıyla onaylanacak. Eğer istenen sonuç alınamamışsa testçilerin yorum yapmaları için bir alan olacak ve bu yapılan yorumlar hata takip sistemine ilgili paket için aktarılacak. Paket test yönergelerinin şu anda LaTeX biçiminde yazılmış halleri XML biçimine dönüştürülecek. XML biçimi için belirli bir şablon oluşturulmuş durumda.

Örnek biçim:

```
<testcase test="shell">
<package>vorbis-tools</package>
<case>
<release>2009</release>
<require>libvorbis</require>
<download>http://cekirdek.pardus.org.tr/~semen/dist/test/
multimedia/sound/sound/game.ogg</download>
<command> oggdec game.ogg</command>
<command> mplayer game.wav </command>
<command> oggenc game.wav </command>
<command> mplayer game.ogg </command>
</case>
</testcase>
```

- **<testcase test="">** tag'i üç farklı test kategorisini kapsayacak: gui, shell ve automatic,
- **<package>** test edilecek paket için gerekli bir bölüm,
- **<case>** testle ilgili bilgilerin tümünü içeren bir üst bölüm,
- **<release>** testin gerçekleştirileceği sürüm,
- **<text>** Test yönergelerinden istenen sonucun açıklamasını içeren soru cümlesi,
- **<require>** bu paketin testi için hangi paketin test edilmesi gerektiğini içeren bir bölüm,
- **<download>** testin gerçekleştirilmesi için gerekli dosyalarının indirilebileceği uzantıyı içeren bir bölüm,



test süreçleri

- **<command>** testi gerçekleştirmek için gerekli komutlar için bir bölüm.
- **<expected>** otomatik testlerde beklenen sonucun yazılabileceği bir bölüm.
Gerçek sonuç beklenen sonuçla aynı mı?

Bu proje seçilip sonuçlanabilirse test süreçlerini hızlandırması ve test ekibi üyelerinin yükünü azaltması düşünülüyor.

Test sürecinin proje yönetim aracı üzerinden takibi

Pardus Projesi olarak yeni bir proje yönetim aracına geçiyoruz, JIRA + Greenhopper. Kararlı sürüme geçecek olan paketlerin geliştirici onayından geçme sürecini (ACK-NACK) ve test raporlarının, onay alınan paket için takip edilmesini bu araç aracılığıyla sağlamayı düşünüyoruz.

Paketler üzerinde yapılan değişiklikler JIRA SVN eklentisi sayesinde bu araç üzerinden takip edilebilir duruma gelecek ve ACK-NACK sürecine girecek olan paketler bu araç üzerinden böylece takip edilebilecek. Geliştiricisi tarafından bu araç üzerinden onaylanıp, listelenebilecek ve böylece test ekibi üyelerine gönderilecek olan paket listesi bu araç üzerinden oluşturulmuş olacak.

Daha ileride yapılması planlanan düşüncelerden bahsetmek gerekirse; GSoC projesinde yapılan araç aracılığıyla hata takip sistemine gönderilen hata raporları, JIRA için de gönderilmeye başlayacak ve testler sırasında sorun yaşanmış paketler bu araç üzerinden takip edilebilir duruma gelecek ve kararlı depoya geçecek olan paketler listesinin elle çıkarılmasına gerek kalmayacak.

Pardus Test Takımı Yöneticisi Semen CİRİT:

**“Benim işim eksik
ve yanlışları bulmak”**

Kendinizden biraz bahseder misiniz? Semen CİRİT kimdir? Pardus geliştirici olma hikâyeniz nedir?

Manisa'da doğdum, çocukluğum ağaçlara tırmanmak, üzüm bağlarının arasında koşturmakla geçti. Bu şekilde bir çocukluk yaşamış olmaktan gayet memnunum :) zira İstanbul ya da diğer büyük şehirlerde doğup büyümüş ve ağaçları toprağı tanımamış bir kişinin çocukluğunu yaşamadığını düşünüyorum.

Liseyi Turgutlu Halil Kale Fen Lisesi'nde yatılı olarak okudum, hayatımın dönüm noktalarından biridir diyebilirim. Yatılı okumak da sanırım insanın hayatının bir döneminde yaşaması gereken bir süreç. Yatılı okulda birçok kişi arasında olmanın insan karakterlerini tanımak, insanlara karşı toleranslı olmak, insanların yerine kendini koyabilmek, iletişimi sağlayabilmek gibi birçok meziyet kattığını düşünüyorum.

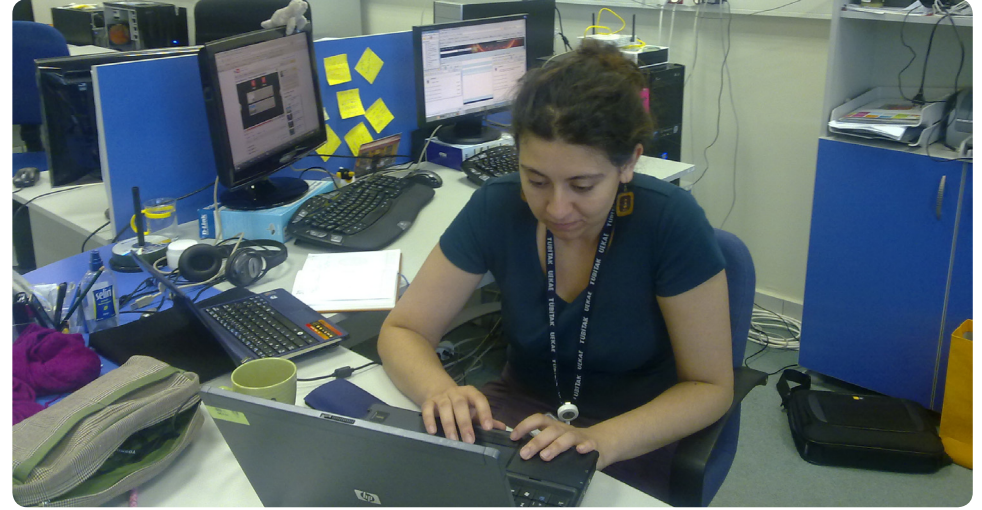
Galatasaray Üniversitesi Linux Topluluğu'na ikinci sınıfta katılmamla birlikte, Linux ile ilgilenmeye başladım. Bu topluluk haftada bir yapılan gönüllü dersler ve sonrasında yapılan gönüllü projelerle öğrencileri geliştirmeyi amaçlıyordu. Bu toplulukta aktif olarak üç yılımı geçirdim. Bu sırada öncelikle Gentoo olmak üzere, birçok dağıtımla ilgilenme fırsatım oldu.

Özellikle Gentoo, dağıtımları öğrenmem açısından çok yardımcı oldu. Bildiğiniz gibi Gentoo'nun herhangi bir kurulum arayüzü bulunmuyor. Gentoo çalışan CD'si ile sisteminizi mount edip, dosya sistem tablonuzu kendiniz ayarlayıp, çekirdekte başlayarak minimum sistem için gerekli tüm paketleri elle derleyip, sorun varsa çeşitli listelerden, forumlardan takip ederek sorunlarını giderip sistemi ayağa kaldırılıyorsunuz. İkinci sınıfın sonunda hem masaüstü hem de dizüstü bilgisayarına Gentoo kurmuştum ve Windows kullanmıyordum. O zamandan beri de arkadaşlarımla bilgisayarları ve test dışında Windows ile hiç ilgilenmedim :)

Bu gibi uğraşlarım sonrasında Pardus'ta çalışmak isteyip başvurunca, Pardus'a kabul edildim ve çalışmaya başladım.

Test etmeyi nerede öğrendiniz? Okulda mı yoksa daha önce katıldığınız özgür yazılım projelerinde mi?

Lisans'ta kendi isteğimle başlamış ve tamamlamış olduğum tez ile testle ilgilenmeye başladım diyebilirim. Bu tez sayesinde yazılım testleri nasıl oluyor, yazılım



süreçlerinde hangi tip testler uygulanıyor, hangi kısımları otomatikleştirilebiliyor üzerine çok fazla araştırma yapma imkanı bulabilmişim. Daha sonra bu araştırmaları daha ileriye götürüp bu konu üzerinde bir yüksek lisans yapmaya başladım.

Pardus'ta testlerle ilgilenmeye başlamam ise yüksek lisansa başlama sürecimden sonra gerçekleşti. Elimde olan teorik bilgileri pratiğe dönüştürme fırsatını Pardus ile yakaladım diyebilirim.

Test ekibine katılmamız için aradığınız kriterler nelerdir? Ekibe katılmak zor mudur?

Test ekibine katılmak için Linux'u sevmek ve Linux için zaman harcıyabilmek dışında hiçbir kriter gerekmemekte. Sadece yapılması gereken test ekibine katılmak için gereken soruları takıldığınız noktalarda pardus-test@pardus.org.tr adresinden danışarak cevaplamamız.

Şu anda aktif olan testçi sayısı nedir? Sizce en az kaç kişiden oluşmalı bir testçi takımı? Sayıyı artırmayı planlıyor musunuz?

Yaklaşık olarak aktif olan 10 test ekibi üyesi bulunmakta, bu sayıyı arttırma düşüncemiz her zaman var. Fakat bu sayıyı bir katkıcıda bulunması gereken

manevi sorumlulukların bilincinde olan kişilerle artırmayı planlıyoruz. Bunlar nelerdir diyecek olursak kısaca; süreklilik, doğruluk, kararlılık, iletişim başlıkları altında sıralanabilir. Bildiğiniz gibi test yazılım süreçlerinin önemli bir parçası ve kritik bir noktası. Test sonucuna güvenemediğimiz bir test ekibi üyesinin bildirimleri maalesef işe yaramayacaktır.

Hiç tanımadığınız insanlardan oluşan test ekibi üyeleriyle test süreçlerini işletmek zor olmuyor mu?

İyi bir belgeleme ve sıkı bir iletişim ile bu sorunun yaşanmayacağını düşünüyorum.

Test ekibi üyeleriyle tanışmak, onları motive etmek, test süreçlerinin daha verimli geçmesini sağlamak için ekiple yüzyüze bir toplantı yapmayı düşünüyor musunuz?

Test ekibi üyeleri ile tanışmak tabii ki isteğim, örneğin Fahri ve Uğur ile Bilgi Üniversitesi Şenlikleri'nde tanışma fırsatı bulmuştum ve bu duruma gerçekten çok seviniştim. Aynı zamanda ana sürümlerin çıkmadan önce neler yapılacağına dair genel toplantılar yapmayı düşünüyorum.

Test süreçlerine katılmanın bir programcı için faydaları nedir? Bu deneyim onlara neler kazandırır? Geliştirici olmalarını sağlar mı?

Özellikle dağıtım üzerine geliştirme yapacak olan kişilerin bir dağıtımın tüm yazılım sürecini takip edebilmek çok işine yarayacak diye düşünüyorum. Ayrıca test yönergelerinde uygulanan birçok komutla kullanıcıdan daha ileri düzeyde Linux'u bilme yetkinliği sağlayacaktır. Herhangi bir hatayla karşılaştığında çözüm yollarını aramaya başlaması zaten geliştiriciliğe ilk adımı teşkil ediyor :).

Test takımınızdaki hedefleriniz nelerdir?

Test Takımı ile yapmış olduğumuz testler tamamıyla yazılım süreçleri ile birlikte yoğunlaşıyor. Yazılım süreçlerinin düzelmesi, bir standarda oturtulması ve belgelenmesi demek aslında dolaylı olarak test süreçlerinin de düzelmesi demek. Yaklaşık son dört aydır yazılım süreçlerinin iyileştirilmesi ve standartlaşması için seri toplantılar yapıyor, tartışıyor ve raporluyoruz.

Bu standartların oturmasıyla daha tempolu ve uyumlu bir test süreci bizi bekliyor





olacak. Bunun dışında test yönergelerinin bir kısmının otomatikleştirilmesi ve gruplandırılmasıyla ilgili bir uygulama projesinden e-derginin bu sayısındaki yazımda bahsediyorum.

Birçok kişi test yapmanın gereksiz olduğunu düşünüyor. Durum böyleyken bize test süreçlerinin ne kadar önemli olduğunu ve bu süreçlerin, geliştirme sırasında ne kadar ciddi ele alındığını anlatır mısınız? Başında olduğunuz ekibin göstereceği başarı, son kullanıcıya nasıl yansır?

Aslında bu konu çok derin ve geniş bir konu ama kısaca anlatmaya çalışayım :). Geliştiriciler bir yazılımı yazmaya başladıklarında süper yaptıkları, dünyadaki en iyi yazılımı onların gerçekleştirdikleri gibi düşünceler içerisindedir. İş kullanıcı kısmına geçtiğindeyse farklı bir dünya ortaya çıkıyor. Kullanıcı tarafından uygulamanın ne derecede algılandığı, kullanma hızlığı, hata almadan

uygulamayı kullanabilmesi kısacası uygulamadan memnun olabilmesi tamamıyla yazılım testlerinin, yazılım döngüsü içerisinde doğru bir şekilde uygulanması ile ortaya çıkıyor.

Test süreçlerinde kullanılan yöntem ve süreçler açısından bakıldığında, Pardus'un diğer Linux dağıtımlarıyla karşılaştırmasını yapar mısınız?

Çeşitli Linux dağıtımları Pardus'un test prosedürlerine benzer sistemleri uygulamakta. Örneğin Ubuntu'nun benzer bir test süreci bulunmakta, bizim latex biçiminde yazdığımız test prosedürleri, onlarda text biçiminde yazılıyor ve daha sonra testçiler tarafından bir araçla test edilmekteler. Fedora da neredeyse aynı yöntemle çeşitli teknolojilerinin test yönergelerini hazırlıyor.

Çok sayıdaki paketin her biri için ayrı ayrı test süreçleri mi tanımlıyorsunuz, yoksa genel bir test süreci var mı?

Bazı paketlerin test süreçleri birbirine benzer bu yüzden ortak yazılan paket test prosedürleri olabiliyor.

Şu anda testçi e-posta listesiyle test edilecek paketler bildiriliyor. Bu araçların dışında kullanılacak başka araçlar (Forum, IRC kanalı gibi) planlıyor musunuz?

Forum ve IRC gibi araçların iletişim ve bilgi paylaşımı açısından iyi olduğunu düşünmekteyim fakat testlerin daha düzenli ilerleyebilmesi, kolaylaşabilmesi ve sonuçlarının kolay toplanabilmesi adına Bugzilla ile entegre çalışabilen ya da kendi başına bir hata takip sistemi bulunan test ya da proje yönetim aracı üzerinden testleri takip etmenin daha iyi olacağını düşünmekteyim. Bu araçlar için gerekli fizibilite çalışmalarını yapmaktayız.

Sizin onayınızdan geçmeyen paketler depoya girebiliyor mu? Test ekibindeki kişilerin gönderdikleri mesajlarla mı Geçti/Kaldı onayı veriyorsunuz, yoksa belirli sayıda test sonucunu gösteren bir takım verilere mi bakıyorsunuz?

Kararlı depoya benim son onayımdan geçmeden hiçbir paket giremiyor. Test listesine gönderilmiş olan paketlerin tamamı benim tarafımdan da test ediliyor.



Benim karşılaşmadığım, ama test ekibi üyelerinden birinin karşılaştığı hatayı tekrar etmeye çalışıyor, eğer tekrarlanabiliyorsa ve hata kritikse paketi kararlı depoya aktarmıyoruz. Bir paketin depoya girip girmeyeceği testçilerin sonuçlarıyla yön değiştirebiliyor ve bir paketin tek bir kişi tarafından hata alması bile depoya girmemesine neden olabiliyor.

Türkiye’de X donanımının üreticisi ürününün üzerine “Pardus’ta çalışır” logosu koymak isterse, bunu test edebilecek bir ekip var mı? Ya da kurulması düşünüyor mu?

Şu anda bulunan test ekibini biraz daha genişlettikten sonra farklı donanımların özel olarak test edilmesi düşünülebilir. Şimdilik kararlı depo testlerinde yeri ve sırası geldiğinde paketlerin donanım testleri testçilerin donanımına göre test edilmekteler.

Geliştiriciler, eski zamanlarda olduğu gibi neden #pardus ya da #pardus-devel kanalına sık gelmiyorlar?

Pardus Projesi bildiğiniz gibi 5 yıllık bir proje ve Pardus geliştiricilerinden beklenen istekler gün geçtikçe artmakta fakat hâlâ sayımız yeterli düzeyde değil. Bu yüzden

belirli kısımlara önem verip diğer kısımları maalesef göz ardı etmek zorunda kalıyoruz.

İnsanları Pardus’a alıştırmak için ne gibi sözler söylüyorsunuz ve Pardus’u ne tür tanımlamalarla anlatıyorsunuz? Pardus’u yaymak için ne tür faaliyetlerde bulunuyorsunuz?

Kurulumu ve kullanımı kolay, virüs ile sorunu olmayan, ücretsiz bir dağıtım. Aynı zamanda güvenlik açıklarını en hızlı kapatan dağıtımlardan biri. Kendinizi geliştirebileceğiniz ve katkıda bulunabileceğiniz toplulukları var. İstersiniz geliştirme, test, çeviri, belgeleme gibi alanlarda katkı verebilir ve bir bütünün parçası olmaktan mutlu olabilir ve mutlu edebilirsiniz.

Hem anlatıcı hem de dinleyici olarak gittiğimiz seminer ve konferanslarda Pardus’u tanıtıyor ve anlatıyoruz.

Bu kadar erkekle dolu bir ekipte, kadın olmak nasıl bir duygu?

Bilmem, hiç bu şekilde düşünmemiştim. Hepimiz bir hedef için çalışıyor ve çabalıyoruz. Bu noktada kadın ve erkek olmamız hiçbir şey değiştirmiyor.



Semen CİRİT, Ozan ÇAĞLAYAN, Işıl POYRAZ ve Fatih AŞICI



Hata Takip Sistemi'ne girilen bir hatayı çözemediğinizde kendinizi kötü hissediyor musunuz?

Hatanın durumuna bağlı, ne kadar özenle yazıldığı, tekrar edilebilmesi ve hatanın geçerli bir hata olması ile biraz ilgili.

Hata raporu iyi yazılmış ise o kadar çabuk tekrar edilebiliyor ve çözüme kavuşması için akılda soru işareti bırakıyor. Fakat hata tekrarlanamıyor ve anlaşılmıyor ise yetersiz bilgi açıklaması yapılarak hatayı raporlayan kişiye gönderiliyor. Bu şekilde süreç uzuyor ve hata raporlayıcı cevap verene kadar hata için bir şey yapılamıyor.

Sabahları ofisinize geldiğinizde, diğer geliştiricilerle ilk muhabbetiniz ne üzerine oluyor?

Hadi toplantı yapalım :) Son dönemlerde ofiste bu durumdan herkes baymış durumda ama ne yapalım hayat...

Duruma göre değişebiliyor, dün akşam gittiğim bir filmi tiyatroyu da anlatabiliyorum veya bir sürüm günü ise hangi geliştiricinin hangi hatayı çözmesi

gerektiğini söylerken kendimi bulabiliyorum.

Ofisinizde en mutlu geçen gününüzü ve en mutsuz geçen gününüzü anlatır mısınız?

Aslında ofiste her gün ne mutlu ne mutsuzum, toplantı yapmayı insanlarla paylaşıp sonralarında çok yorgun oluyorum ve bazen üzülüyorum, ama yemekler sonrasında hareketli ve neşeli olurum.

İnternet tarayıcınızda en çok girdiğiniz 5 yer iminizi söyler misiniz?

- <http://www.ntvmsnbc.com>
- <http://www.isbank.com.tr>
- <http://mail.google.com>
- <http://slashdot.org>
- <http://packages.pardus.org.tr>

Semen CİRİT boş zamanlarında neler yapar?

Düzenli olarak yüzüyorum. 3 yıl boyunca tiyatro ile ilgilendim bu dönem içerisinde oyunculuktan, kostüm ve dekora, tiyatro dramaturjisi ve rejisine kadar her kısmıyla uğraşım. Son iki yıldır iş ve master derken pek fırsat bulamıyorum, tiyatroya katkım şu sıralar ancak eski üyesi olduğum topluluğun oyunlarını izlemek ve eleştirilerde bulunmakla sınırlı. Yüksek lisansım biter bitmez tekrar yakından ilgilenmeyi düşünüyorum. Tabi fırsat buldukça tiyatro ve sinemaya giderim.

Bunların dışında kitapçılara gidip saatlerce durup, bir kucak dolusu kitapla çıkabilirim.

Keşke şu olsaydım dediğiniz ve mesleğinizden sıkıldığınız oldu mu?

Keşke şu olsaydım demeyelim ama bu mesleği yapmasaydım ne yapabilirdim diye değiştirelim. Çünkü eksik ve yanlış kısımları bulmak, gözlemek ve çıkarmak çocukluğumdan beri süre gelen bir şey ve seviyorum. Sanırım bu gözlemi ömür boyu yapabilirim. (Sadece test olarak düşünmeyebiliriz tabi :) her sistemin eksiklikleri, yanlışları ve sorunları vardır.)

Sanırım birkaç şey sayabilirim; tiyatro, iç mimari, moda tasarımcılığı.

Pardus Kullanıcıları İçin Yararlı Adresler



[Pardus'u bilgisayarınıza nasıl kurabilirsiniz?](#)

[Pardus topluluğu neler yapıyor?](#)

[Dergimizin tüm sayılarını nereden bulabilirim?](#)

[Ajans Pardus'u dinleyelim!](#)

[Nerede bu Pardus kullanıcıları?](#)

[Pardus geliştiricileri neleri tartışıyor?](#)

[Pardus kullanıcıları neleri tartışıyor?](#)

[OpenOffice.org Türkiye](#)

[Bir hata buldum, bunu nereye raporlamalıyım?](#)

[Pardus ile ilgili teknik bilgiler için nereye bakmalısınız?](#)

[Şehirimde Pardus CD Gönüllüsü var mı?](#)

[Bana Pardus 2009 Kurulum CD'si gönderir misiniz?](#)

[Pardus kullanıcıları www.linux.com'da ne yapıyor?](#)

[Windows'tan Pardus'a nasıl geçebilirim?](#)

[Pardus kullanıcılarının gezegeninde neler oluyor?](#)

[Pardus ile ilgili Sıkça Sorulan Sorular](#)

