

GNU  
Pardus-Linux.Org  
eDergi



Şubat '10 - Sayı 16

## **N** Bu sayıda Ne var?

Michael Foord (Python) Röportajı

Kayıpsız Resim Büyütme

Pardus'ta Django

Pardus 2009.1

Coldwar

Giriş Yazısı	3
Kayıpsız Resim Büyütme: SmillaEnlarger	4
Pardus 2009.1 Melez CD Özelliği	7
Pardus 2009.1 Kurulum ve İlk Adımlar	10
Oyun Tanıtımı: ColdWar	12
Pardus'ta Django: Uygulama Geliştirme	19
Pardus'ta Django: HTML Dosyaları	25
Röportaj: Michael Foord (Python)	32
Google ve Microsoft Rekabeti Son Hız Devam	42
2010 GNU/Linux Dünyasına Ne Getirebilir?	44
Serbest Piyasa Düzeni Masalı ve Yazılımlar	46
OEM Davası Üzerine	52
Penguart	54
Haberler	57

Şubat '10 - Sayı 16

Merhaba **Özgür Yazılım** Dostları...

Pardus-Linux.Org Topluluğu ve değerli zamanlarını dergimiz ve siz okurlar için harcıyıp, sessiz sedasız ortalıktan kaybolmayan katkıcıları tarafından, tamamen gönüllülük esasına göre, hiçbir menfaat beklemesizin hazırlanan yeni bir elektronik dergimizle sizlerin karşısındayız. Dergimizin yine, her zaman olduğu gibi özgür yazılımlar kullanılarak hazırlanmasına, yani Nihad Ağabey'imın tabiri ile domuz eti kullanılmamasına özen gösterdiğimizi temin ederek sözlerime başlamak isterim.

Geçen sayımız, hani şu kapağında **Uğur Mumcu**'nun resminin bulunduğu sayımız, başta olmak üzere son iki sayımızda dergimizde siyaset yapıldığı yönündeki bazı eleştirileri çeşitli forum ve elektronik postalardan okuduk. Bu konuda söylemek gereği duyduğumuz tek şey, biz **Özgürlük** ve **Özgür Yazılım**'ın siyasetini yapıyoruz, evet. Bu konuda da **Özgür Yazılım** ile aynı kulvarda olduğunu düşündüğümüz çeşitli terimleri kullanmaktan çekinmediğimiz doğrudur ve olaya **Özgür Yazılım** ideolojisi ile yaklaştığımız gerçeği mevcuttur. Bu nedenle yazıları derseniz, hiçbir ön yargı barındırmadan okumanızı tavsiye ederiz.

Şubat '10 - Sayı 16

Bu sayımız, önceki sayımıza göre biraz daha dolgun bir içeriğe sahip. Yaşar Yeğın, resimleri neredeyse hiç kayba uğramadan büyütebilen bir yazılım olan SmillaEnlarger adlı yazılımı tanıtırken, Muslu Yüksektepe ve Onur Tuna tarafından biri işın kuramında, diğeri uygulamasında olan iki farklı Django yazısı paylaşıyor. Aydın Bez ve Kemal Karataş ise **Özgür Yazılım** adına kalemelerini son sürat kullanmaya devam ediyor. Pardus 2009.1'in çıkışıyla beraber, getirdiğı birtakım yeni özelliklere de değinilen sayımızda, Python geliştirici takımından Michael Foord ile yaptığımız röportaja da, Melike İteralp'in çevirisi ile yer verilirken, Hamit Giray Nart soğuk savaş yıllarını konu alan ColdWar adlı oyunun tanıtımını yapıyor. Kapak tasarımıımız Gürhan Şüküroğlu'ya ait, kendisine teşekkürlerimizi sunuyoruz. Tüm bunların yanında, dergimizin üçüncü sayısından beri unuttuğumuz Penguart adlı bölüme tekrardan yer vermiş bulunuyoruz. Sevgili Hamit Giray Nart ve Yaşar Yeğın'in masaüstü görünümelerini ve bu görünümelerin nasıl oluşturulduğunu anlattığımız Penguart bölümümüzde, derseniz sizin de masaüstü görünümünüzü yayınlayabiliriz.

Beğeneceğınızı umduğumuz dergimizle sizleri baş başa bırakırken, dergimize nasıl katkıda bulunabileceğiniz dergimizin arka kapağında bulabileceğınızı hatırlatır, **Abdi İpekçi**'yi öldürülüşününün 31. yıl dönümünde saygıyla anarız.

Erdem Artan  
baDibere@gmail.com

Resimlerinizi kayıpsız olarak büyötmek ister misiniz?

Cevabınız evetse, bu iş için hazırlanmış SmillaEnlarger adlı uygulamayı deneyebilirsiniz.

Uygulamayı çalıştırmak için, Qt4 arayüz takımının bilgisayarınızda kurulu olması gerekiyor. Qt4 arayüz takımına sahip tüm GNU/Linux dağıtımlarında, dolayısıyla Pardus 2009 serisinde de uygulamayı kolayca derleyip kullanabilirsiniz. Ancak Qt4 sürümüne sahip olmayan önceki Pardus sürümlerinde ya da diğer GNU/Linux dağıtımlarında, sorunla karşılaşmanız olasıdır.

SmillaEnlarger uygulaması, yazımız hazırlanırken Pardus, Ubuntu, Debian ve Sabayon üzerinde denenmiş ve başarılı sonuçlar alınmıştır.

Uygulama, derleme yöntemi ile sistem geneline kurulmuyor. Taşınabilir (portable) bir mantıkla çalıştığı için derleme sonrası oluşan ikili dosyayı çalıştırarak kullanıyoruz.

Uygulamanın derlenmesi için "qmake" ve "make" uygulamaları gerekiyor. "qmake" uygulaması, Qt4 arayüz takımı ile birlikte gelirken, "make" uygulamasını paket yöneticisi kullanarak aşağıdaki komutla kurmanız gerekiyor:

```
psı it make
```

Dilerseniz, "make" uygulamasının da aralarında bulunduğu, tüm geliştirici paketlerini aşağıdaki komutla kurabilirsiniz:

```
psı it -c system.devel
```

Uygulamanın kaynak kodlarını, SourceForge üzerindeki proje sayfasından[0] edinebilirsiniz. Uygulamanın kaynak kodlarının indirilmesi, arşivden çıkarılması ve derlenmesi için komut satırını kullanabilirsiniz. Böylece komut satırına olan alışkanlığın körelmemesi ve olası hataları görebilme olanağı sağlanırken; GNU/Linux felsefesine uygun olarak, günlük egzersizinizi de yapabilirsiniz.

Komut satırını açtıktan sonra, "wget" dosya indirme aracını kullanarak, uygulamanın kaynak kodlarını şu komutla indirmeniz gerekiyor:

```
wget  
http://sourceforge.net/projects/imageenlarger/files/  
/imageenlarger/SmillaEnlarger%20Release%200.8.9/SmillaEnlarger_0.8.9_src.zip/download
```

Ardından, indirilen zip arşivini, "unzip" komutu ile dizine açmalısınız. "unzip" komutunu kullanabilmek için, "unzip" uygulamasının bilgisayarınızda yüklü olması gerekir. Yüklü değilse, "psı it unzip" komutu ile yükleyebilirsiniz.

```
unzip SmillaEnlarger_0.8.9_src.zip
```

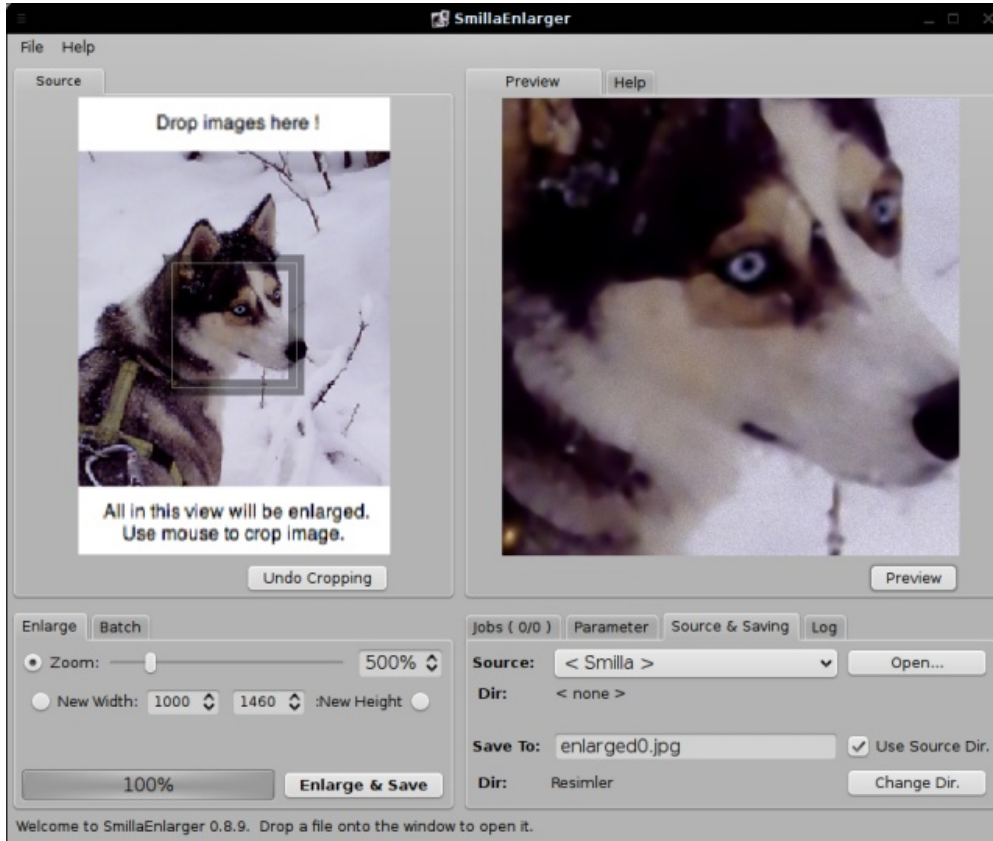
Yukarıdaki komutu girdikten sonra hatayla karşılaşmazsanız, zip arşivinin bulunduğu dizinde "SmillaEnlarger\_0.8.9\_source" adlı bir dizinin oluşması gerekir. Derleme işlemine geçmeden önce, "EnlargeParam.h" adlı dosyayı, ana dizine kopyalamanız gerekiyor:

```
cd SmillaEnlarger_0.8.9_source/  
cp SmillaEnlargerSrc/ImageEnlargeCode/EnlargeParam.h ./
```

Kopyalama işleminin ardından, derleme işlemine geçebilirsiniz:

```
make ImageEnlarger.pro  
make
```

Bu komutların ardından, hata ile karşılaşmadığınız sürece derleme yapılacaktır. Derleme sonucunda "SmillaEnlarger" adlı çalıştırılabilir bir dosya oluşacaktır. Bu dosyaya tıklayarak, SmillaEnlarger ile çalışmaya başlayabilirsiniz.



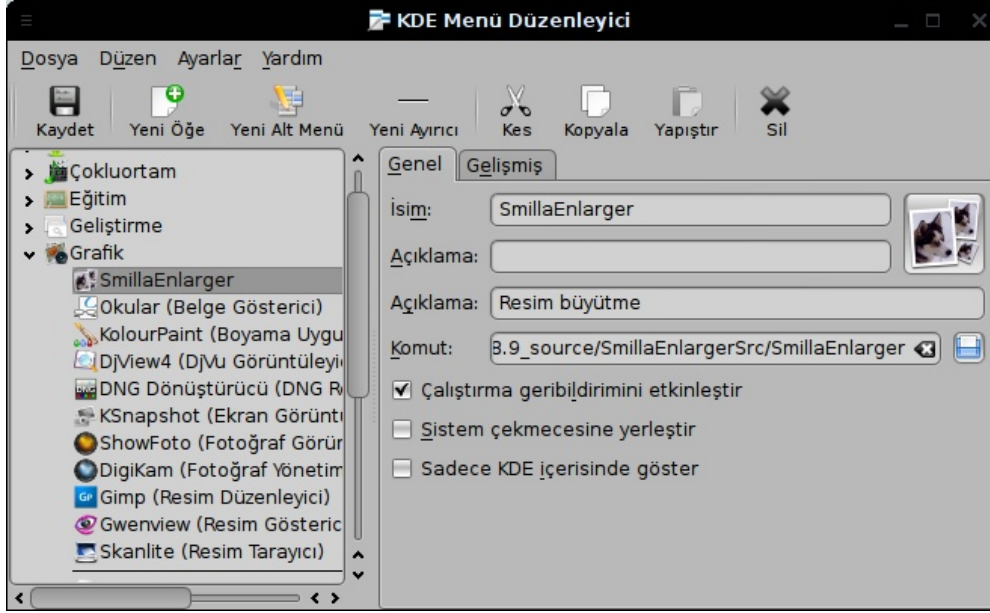
Resimde görüldüğü gibi, uygulamanın arayüzü pek karışık değil. Özel ayarlar yapmayacaksanız, File > Open yolunu izleyerek istediğiniz resmi seçebilir, ardından "New Width" ve "New Height" bölümleri ile çözünürlüğü, "Zoom" ile derinliği ayarlayabilirsiniz.

"Preview" kısmında, yeni resmin ön izlemesini inceledikten sonra, "Enlarge&Save" düğmesini kullanarak, yeni resmin oluşturulmasını ve kaydedilmesini sağlayabilirsiniz.



Önceden bahsettiğim gibi, uygulama taşınabilir olduğundan derleme sonrası istediğiniz yere dosyaları taşıyabilirsiniz. Uygulama için, her seferinde uygulamanın bulunduğu dizini açmak yerine, uygulamaya ait bir kısayol kullanmak isteyebilirsiniz. Bunun için şöyle bir yol izlenebilir:

- 1- KDE menüsüne sağ tıklayıp, Menü Düzenleyici > Grafik yolunu izleyin.
- 2- "Yeni Öğe" deyip, öğe adına SmillaEnlarger yazın.



3- Komut olarak, "Gözet" düğmesini kullanarak, SmillaEnlarger'in bulunduğu adresi yazın.

4- Simge olarak, yine SmillaEnlarger'in bulunduğu dizinde bulunan, Smilla.png dosyasını seçin.

İyi günlerde kullanmanız dileğiyle...

Yaşar YeğİN

yasaryegin@gmail.com

Pardus-Linux.Org - aware

Doctus.org - felâsife

[0] <http://sourceforge.net/projects/imageenlarger/>

Pardus 2009.1'in aday sürümünün duyurulması ile birlikte, Pardus'un USB bellek gibi ortamlardan kurulması kolaylaşıyor. Pardus için ISO kalıpları artık melez(hibrit) olarak geliyorlar. Melez ISO kalıplarının özelliği, önceki kalıplar gibi sadece CD'lere yazma mecburiyetini kaldırarak kolayca DVD, USB bellek, SD Kart gibi ortamlara yazılarak, bu ortamlardan kurulum olanağı sağlaması.

Bu yazımızda, ISO kalıplarının dd uygulamasıyla hem GNU/Linux hem de Windows'ta USB belleklere nasıl yazılabileceği incelenecektir.

## GNU/Linux Kullanıcıları

İlk adım olarak USB belleğin adresinin öğrenilmesi gerekiyor. Adresin öğrenmek için aşağıdaki iki yoldan birini kullanabilirsiniz:

### fdisk komutu ile

GNU/Linux dağıtımlarında, `fdisk -l` komutu bilgisayara bağlanmış olan depolama ortamlarının adresleri belirlenebilir. Komut verildiğinde, çıktılar arasında şöyle bir bölüm olacaktır:

```
Disk /dev/sdb: 1049 MB 1049100288 bayt
```

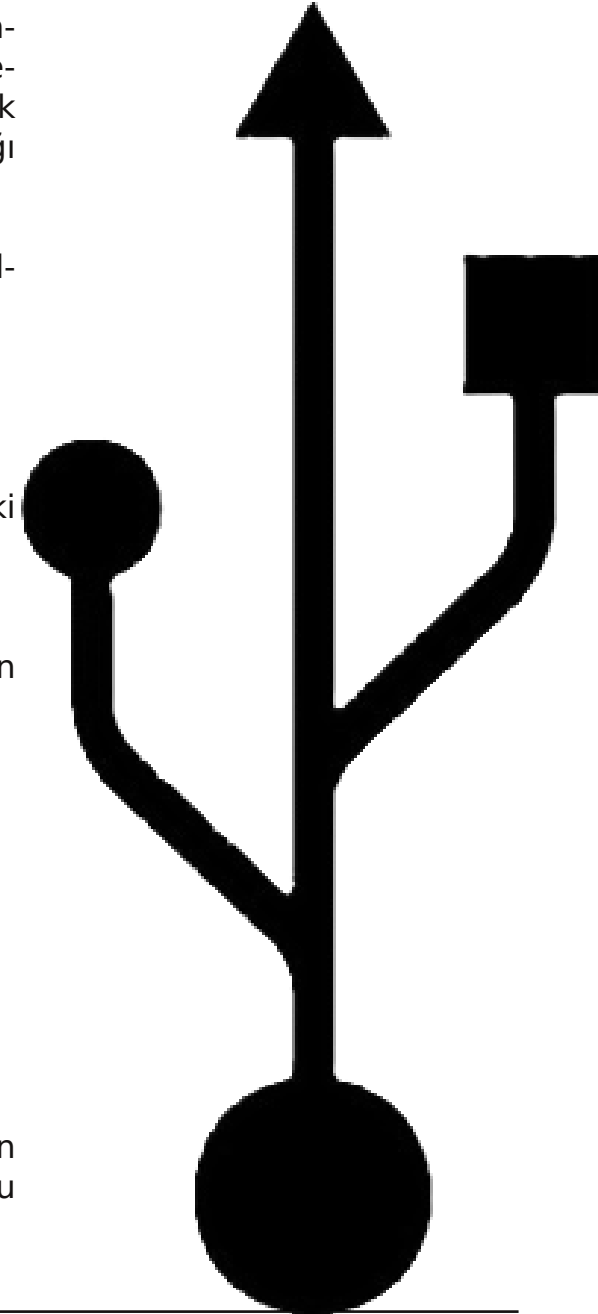
```
255 heads, 63 sectors/track, 127 cylinders
```

```
Units = silindir of 16065 * 512 = 8225280 bytes
```

```
Disk identifier: 0x1ba65e7b
```

Aygıt	Açılış	Başlangıç	Bitiş	BlokSayısı	Kml	Sistem
/dev/sdb1		1	127	1020096	b	W95 FAT32

Bu çıktının renklendirilmiş kısmı, bizim ilgilendiğimiz kısım. Görüldüğü gibi, USB belleğimizin adresi **/dev/sdb**. Bunu belleğimizin kaç MB dosya saklayabileceğinden öğreniyoruz. Bu nedenle yanlış belleğin adresini seçmediğinizden emin olmalısınız.



## ls komutu ile

GNU/Linux dađıtımlarında, USB belleklerin adresini `ls -la /dev/disk/by-id/usb-*` komutuyla da öđrenebiliriz:

```
lrwxrwxrwx 1 root root 9 0ca 23 23:26 /dev/disk/by-id/usb-USB_2.0_Flash_Disk_211452018D114A00-0:0 -> ../../sdb
lrwxrwxrwx 1 root root 10 0ca 23 23:26 /dev/disk/by-id/usb-USB_2.0_Flash_Disk_211452018D114A00-0:0-part1 -> ../../sdb1
```

Yukarıdaki çıktıda ilgilendiđimiz bölüm renklendirilmiş olan, yani **/dev/sdb**.

Bu adımdan sonra USB belleđimize ISO kalıbını yazmaya başlayabiliriz. GNU/Linux kullanıcıları, hiçbir ek yazılım yüklemeyen aşıđıdaki komutu kullanabilirler:

```
dd if=/home/guDa/Masaüstü/Pardus_2009.1_Anthropoides_virgo.iso of=/dev/sdb bs=4M
```

Burada dikkat edilmesi gereken, "if=" parametresinden sonra Pardus 2009.1 ISO kalıbının adresinin yazılması ve "of=" parametresinden sonra ilk adımda bulmuş olduđumuz USB bellek adresinin yazılması.

Bu komutun ardından aşıđıdaki gibi bir çıktı görürseniz, herşey yolunda demektir:

```
170+1 records in
170+1 records out
716177408 bytes (716 MB) copied, 359,698 s, 2,0 MB/s
```

## **Windows Kullanıcıları**

İlk adım olarak dd uygulamasını <http://www.chrysocome.net/downloads/dd-0.5.zip> adresinden indirmek gerekiyor. Ardından indirilen zip arşivinin ayıklanması gerekiyor. Ayıklama işleminin ardından MS-DOS açılarak, dd.exe dosyasının bulunduđu dizine gidilir:

```
cd Desktop/dd-0.5
```

Ardından `dd --list` komutu verilir. Komutun çıktısında aşıđıdaki gibi bir bölüm olacaktır:



```
\\.\Volume{6cea3218-2db5-11dd-884d-001e8cc17471}\  
link to \\?\Device\HarddiskVolume4  
removable media  
Mounted on \\.\f:
```

Yukarıdaki çıktının ilgilendiđimiz bölüm olduđunu "removable media" yani çıkarılabilir ortam ifadesinden anlıyoruz. Belleđin adresini ise "Mounted on \\.\f:" ifadesinden anlıyoruz. Bu bilgiler ile birlikte, ISO kalıbını belleđe yazmak için vermemiz gereken komut řu řekilde:

```
dd if="C:\Downloads\Pardus_2009.1_Anthropoides_virgo.iso" of=\\.\f: bs=4M
```

Yukarıdaki komutta, aynı GNU/Linux kullanıcıları için olan uygulamadaki gibi, "if=" parametresinden sonra Pardus ISO kalıbının adresini, "of=" parametresinden sonra ise depolama ortamının adresinin yazılması gerekmektedir.

Bu komutun karşısında ařađıdaki gibi bir çıktı görürseniz, işlem tamamlanmıř demektir:

```
rawwrite dd for windows version 0.5.  
Written by John Newbigin <jn@it.swin.edu.au>  
This program is covered by the GPL. See copying.txt for details  
170+1 records in  
170+1 records out
```

Bu işlemlerden sonra tek yapmanız gereken, boot edilebilir olarak ayarlanmıř USB belleđinizi açılıřta çalıştırmak üzere BIOS ayarlarınızı düzenlemek ve Pardus kurulumuna başlamak.



Temmuz 2009 ortalarında duyurulan Pardus 2009'un ilk ara sürümü olan Pardus 2009.1, soyu tükenme tehlikesi altında ve Türkçesi Telli Turna olan Anthropoides virgo kod adıyla, Pardus 2009 serisinin sürüm yönetici Onur Küçük tarafından 15 Ocak 2010 tarihinde duyurulmuştur.

## Kurulum

Pardus 2009.1 kurulum konusunda kullanıcının işini kolaylaştıracak bazı yenilikler içeriyor:

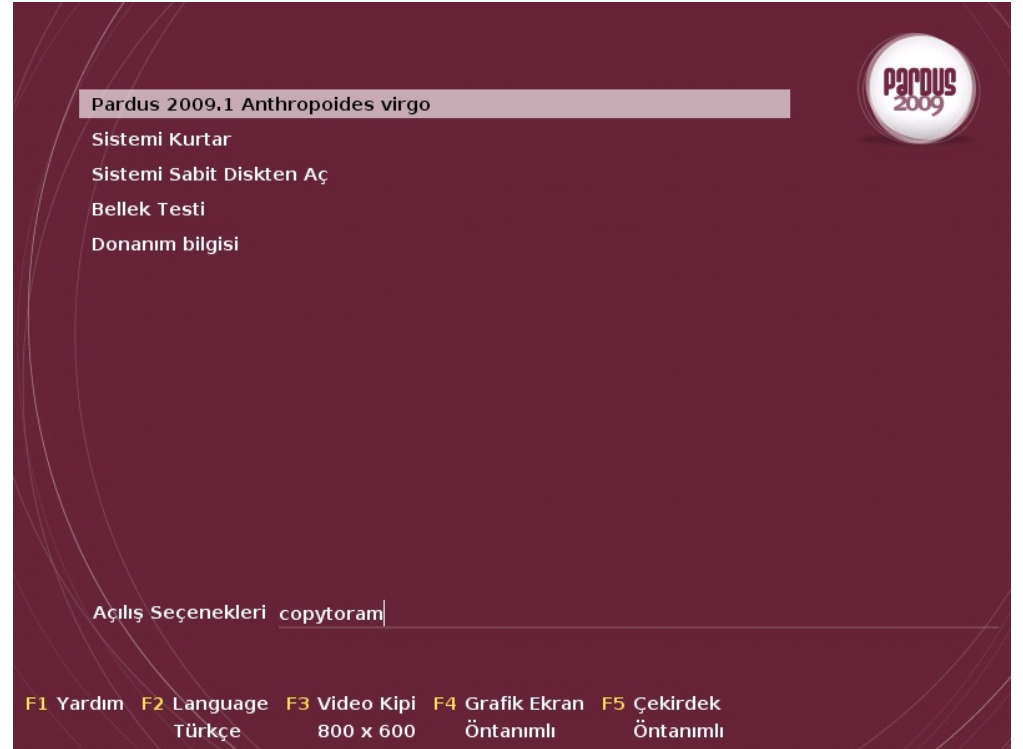
- \* USB bellek gibi ortamlardan kolayca yüklenebilme
- \* RAM bellek üzerinden yüklenme

Pardus 2009.1'in USB bellekler ile nasıl yüklenebileceğini, bir önceki makalemizde anlatmıştık. RAM bellek üzerinden yükleme olayı ise, Pardus 2009.1 Kurulan CD'nin tüm içeriğinin açılış sırasında RAM belleğe kopyalanarak, yükleme işleminin RAM bellek üzerinden devam etmesini ve böylece daha hızlı bir kurulum elde edilmesini amaçlıyor. Hatta, kurulum başladıktan sonra kurulum USB veya CD'sini bilgisayardan çıkarıp başka bilgisayarlarda başka işlerde kullanabilirsiniz.

Çalışan CD'ler ise yine copytoram özelliği sayesinde, daha hızlı bir şekilde çalışabilme özelliğine sahip oluyorlar.

1.5 GB ve üzeri RAM bellek bulunan bilgisayarlar ile kullanılması önerilen bu özelliği etkinleştirmek için, bilgisayarı Pardus 2009.1 Kurulan CD/DVD/USB Bellek/SD Kart ile başlattıktan sonra, açılan kurulum menüsünde parametre olarak

"copytoram" yazılması yeterlidir.



1.5 GB RAM bellek bulunmayan bilgisayarlar için tabii ki normal kurulum yolu izlenebilir.

Pardus 2009.1 kurulumu, Pardus 2009'a göre neredeyse hiç bir değişiklik içermiyor. Bu nedenle kurulumu anlatmayı gereksiz görüyoruz. Merak edenler, <http://www.pardus-linux.org/pardus-indir> veya <http://www.pardus.org.tr/kilavuz> adreslerini inceleyebilirler.

Pardus 2009.1'in normal kurulum süresi, kurulumun yapıldığı bilgisayarın donanımına bağlı olarak ortalama 20 dakika sür-

mekte. Bu süre USB bellek kullanarak 15 dakikaya, "copytoram" parametresini kullanarak ise 11 dakikaya kadar indirilebilmekte. (\*)

(\*) Paketlerin kurulması ve yapılandırılmasını içeren süredir.

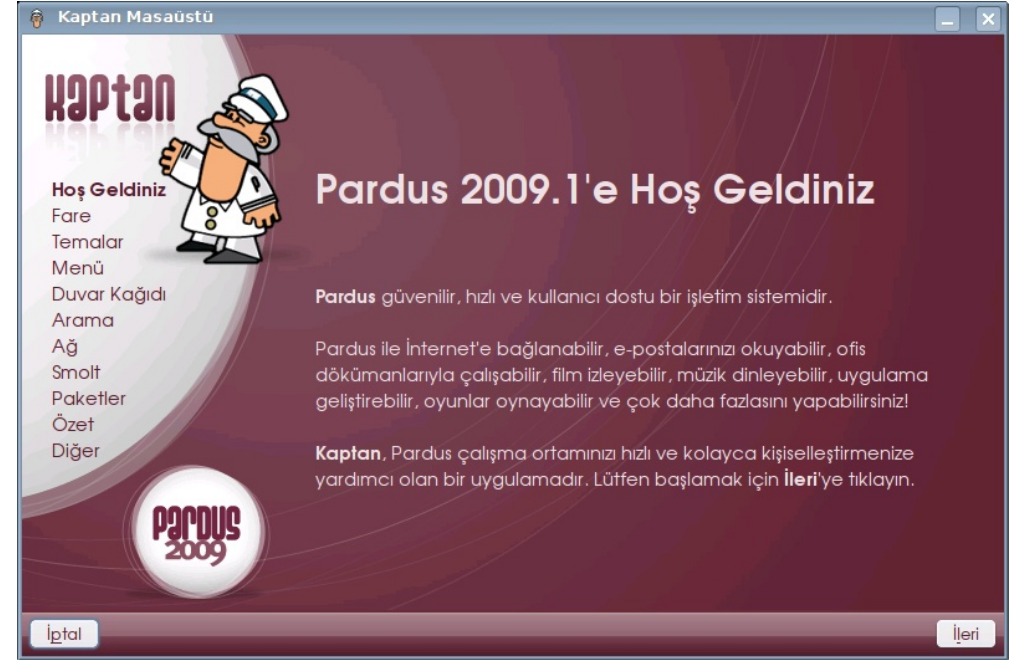
## İlk Adımlar

copytoram parametresi ile yaklaşık 11 dakikaya kurulabilen Pardus 2009.1, bir önceki sürüme göre daha kullanıcı dostu bir yapıya kavuşmuş. Kurulum sonrası yüklenen paketler şu şekilde:

- \* Linux Çekirdeği 2.6.31.11
- \* Glibc 2.9
- \* X.org X Sunucusu 2.6.5
- \* KDE 4.3.4
- \* Gtk 2.18.5
- \* Qt 4.5.3
- \* OpenOffice.org 3.1.1.5
- \* Mozilla Firefox 3.5.7
- \* Gimp 2.6.8
- \* Amarok 2.2.2

Böylece doğrudan son kullanıcıyı ilgilendirecek şekilde, daha çok donanım tanınması, İnternet'te daha güvenli gezinti, daha güzel ve daha fonksiyonel bir masaüstü sağlanmış oldu.

Kurulum sonrası, Pardus'un ilk sürümlerinden itibaren bizleri büyük bir konukseverlik ile karşılayan Kaptan, elbette ki bu sürümde de bizleri yalnız bırakmıyor. Üstelik daha çekici ve daha kullanıcı dostu:



Kaptan ile sağ ya da sol el kullanıyor olmanıza ve dosyaları çift/tek tıklama ile açmak istemenize göre fare ayarlarını, göz zevkinize göre pencere teması, masaüstü tipi (KDE3 veya KDE4 görünümü), masaüstü sayısı, masaüstü duvar kağıdını, Pardus menüsünün türünü (Basit, Kick-off, Lancelot); pratik masaüstü aramasının etkinleştirilmesini, ilk ağ ayarlarının yapılmasını, donanım bilgilerinizin Pardus ekibine gönderilmesini, Paket Yöneticisi ile ilk ayarların yapılmasını gerçekleştirebilirsiniz.

Kaptan'dan sonra göze ilk çarpan kullanıcı dostu yenilik, Kilitle/Çık Programcığının panelde ön tanımlı olarak gelmesi. Kilitle/Çık Programcığı ile, tek tık ile ekranı kilitleyebilir veya bilgisayarı kapatma ekranını açabilirsiniz.

# COLD WAR

Soğuk Savaş yıllarında rolünüzü almaya hazırlanın...

## GİRİŞ

Cold War, Dreamcatcher Games ve Mindware Studios tarafından geliştirilmiş, 3. şahıs görüş açısına sahip bir aksiyon oyunudur. Aslında oyunu tek kelime ile tanımlayacak olursak, bir "Splinter Cell" klonu olarak tabir edebiliriz. Sam Fisher'ı tanıyanların, "hımmmm...." dediklerini duyar gibiyim :-). Oynanış tarzı, kamera açıları, gizlilik, karanlıklarda sessizce hareket etme, bodoslama dalmak yerine arka kapıdan sızmak açısından SC ile benziyor ama Cold War kendi yolunu çiziyor. İlk olarak SC'den daha zorlu bir oyun. Çünkü oyunda, Sam Fisher gibi üst seviyede eğitilmiş bir gizli ajanı değil; Matthew Carter isminde, suçu sadece yanlış zamanda yanlış yerde bulunmak olan, serbest çalışan bir gazeteciyi oynuyorsunuz. Yani attığını vuran, vurduğunu da deviren siz değilsiniz bu oyunda.

Oyun, 1981 yılında Sovyetler Birliği'nde geçiyor. Dünyanın iki süper gücü olan Birleşik Devletler ve Sovyetler, II. Dünya Savaşı'ndan sonra su yüzüne çıkan düşmanlığı sona erdirecek biçimde somut adımlar atmaya başlamışlar. Ancak her iki süper güç arasında ittifak ya da iş birliği biçiminde bir ilişki söz konusu değil. Yine de, gelişmelerin normal seyri devam ederse, soğuk savaş sona erecek gibi görünüyor. Ancak herkes soğuk savaşın sona ermesinin, iyi bir fikir olduğu görüşünde değil. Sovyet hükümeti içinde bulunan ve KGB Başkanı ve başkanın danışmanlarından olan Georgiy Barinsky tarafından yönetilen köktenci bir grup, Komünist rejimi eski katı haline döndürecek ve yakınlaşmayı sona erdirecek, nükleer silah ele



geçirme ve Sovyet Başkanı'na suikast eylemlerini de içeren, bir planı uygulama hazırlığındalar.

Tüm bu gelişmeler karşısında, dünya basınının gözü kulağı da elbette Sovyetler'in üzerinde. Mühendislik eğitimi almış, ancak New York'ta serbest gazetecilik yapmakta olan Matthew Carter da gelişmelerle ilgili olarak aldığı bir istihbarat için için Sovyetler Birliği'ne geliyor. Lenin'in mozolesinde, CIA ile Sovyetler Birliği Başkanı'nın görüşmesini görüntülemek amacında. Ancak mozolede işler ters gidiyor. Carter'ın sıradan fotoğraf makinesi, ileri teknoloji ürünü bir izleme cihazı ile karışıyor (karıştırılıyor) ve kıyamet kopuyor. Carter, casusluk ve başkana suikast girişimiyle suçlanarak, KGB'nin politik nezarethanesine konuyor. Carter'ın Pulitzer Ödülü hayali, bir anda, adını temize çıkarma savaşına dönüşüyor. Olaylar geliştikçe, Carter, aslında sandığından çok daha derin bir olayın içine karıştığının farkına varıyor. Ancak adını temizlemek ve üzerine atılan suçun komplosunu ortaya çıkarmak için, önünde küçük bir engel var: Bulaştığı bu bataktan sağ kurtulmak.

## KURULUM

Oyunu çalıştırabilmek için gereken yazılım ve donanımlar şu şekilde:

Linux Çekirdeği 2.2.x ve daha üzeri çekirdek sürümü  
Glibc 2.1.x veya üzeri kütüphane sürümü  
Xfree86 4.3.x ve üzeri sürüm grafik uzantısı  
1.5Ghz veya üzeri işlemci  
512 MB Ram  
OpenGL 1.3 destekli 3B grafik kartı

ALSA veya OSS uyumlu ses kartı  
DVD-ROM  
1.1 GB disk alanı

Eğer bilgisayar donanımınız oyunun ihtiyaçlarını karşılıyorsa, oyunu Pardus altında çalıştırabilmeniz için, ayrıca bir şeyler kurmanız gerekmiyor.

Oyun DVD olarak geliyor. Ücretsiz bir oyun değil. Linux Game Publishing'den (LGP) £27 ücretle satın alabiliyorsunuz. Kurulumu gayet basit. Oyunun DVD'sini DVD-ROM'unuza takıp, konsoldan root olarak DVD'nin bağlı olduğu dizine gidip, `sh setup.sh` komutunu vermeniz yeterli. Sonrasında oyunun kendi kurulum ekranı geliyor. Kurulum ekranı standart olarak; kurulum klasörü, kısayol oluşturulacaksa oluşturulacak yer ve oyunun kurulumu aşamalarından geçerek oyunu kuruyor. Daha sonrasında ise oyunu kurduğunuz klasöre, `chmod -R 777 /oyunu_kurduğunuz_klasörün_konumu` komutuyla diğer kullanıcıların kullanımı için izinleri yapılandırın.

**Not:** Eğer ilk defa bir LGP oyunu kuruyorsanız. İlk olarak, güncellemeler için `LGP_Update`; kolay kaldırma içinse `LGP_Uninstall` isimli iki araç kuruluyor. İkisi de kullanışlı araçlar. İsimlerinden de anlaşılacağı üzere LGP\_Update, kurulmuş olan LGP oyunlarının güncelleme ve yamalarını indirip kuruyor. LGP\_Uninstall ise, kurulu LGP oyunlarını kaldırmada kullanılıyor. Ama bu araçları kurmak istemiyorsanız, Cold War kurulumuna kadar `iptal` diyerek süreci ilerletin.

Oyunu kısayol oluşturduysanız kısayoluna, oluşturmadıysanız kurduğunuz klasör altındaki `coldwar` dosyasına tıklayarak çalıştırabilirsiniz.

Oyunu kaldırmak ise kurmaktan da kolay. Ne demişler: Bozmak, yapmaktan kolaydır. :-) Klasör altındaki "uninstall" dosyasını çalıştırmanız yeterli. Kurduysanız, LGP\_Uninstall aracından da, listeden Cold War'ı seçip "Uninstall" butonuna basmanız yeterli.

## OYNANIŞ

İlk çalıştırmada önünüze; dil, çözünürlük ve ses sürücüsü seçebileceğiniz bir ayar ekranı gelecek. Dil seçeneğinde İngilizce, Fransızca, Almanca ve Rusça oyunun konuşma dili ve alt yazı dili olarak gelirken; diğer dil seçenekleri sadece alt yazı olarak geliyor.



Mozoleye sızıp da CIA ile Başkan'ın görüşmesini görüntüleme görevi, aslında bir eğitim ve alıştırma görevi. Oyunda gizlilik ve saklanmak esas. Evet, silahlarınız var. Oyunda düşmanlarınızdan almış olduğunuz silahları kullanabiliyorsunuz. Ne var ki, Carter bir komando olmadığı gibi, oyunun genelinde -karmanın zaten başlamış olduğu birkaç bölüm hariç-, çatışmaya girmek bir seçenek bile değil.

Yandaki ekran görüntüsü, oyunun genel görünümünü göstermektedir.

Carter'ı standart WASD tuşları ile hareket ettirirken, fareyle etrafınıza bakabiliyorsunuz. Farenin orta topu ile hızlanıp yavaşlayabiliyorsunuz. Shift tuşu eğilmenizi sağlarken;

Ctrl ise depar atmanızı sağlar. Ama unutmayın, ne kadar hızlı hareket ederseniz, o kadar fazla gürültü yaparsınız. Sol fare tuşu, ateş tuşu görevi yaparken; sağ fare tuşu ise kullanma tuşu olarak çalışıyor. Yani bir nesneye yaklaştığınızda, sağ üst köşede o nesneye yapabileceğiniz işlerin bir listesi çıkıyor (bir kapıya yaklaşıncaya "Open", bir düşmana arkadan sessizce yaklaşıncaya "Knock Down" gibi), sağ fare tuşuna basılı tutup, hareketi seçip, sağ fare tuşunu bırakınca, Carter, seçtiğiniz hareketi yapıyor. L tuşu ile parmak şıklatıp, çevrenizdekileri bulduğunuz yere çekebilirsiniz.



1'den 7'ye kadar olan tuşlarla ekipmanlarınızı seçebiliyorsunuz. Aynı tuşa tekrar tekrar basarak, aynı grup altındaki diğer ekipmanları seçip, “e” tuşu ile bu ekipmanı elinize alabilirsiniz. Bunlar:

1 - Ateşli silahlar: Sadece tabanca ve AK47 tüfeği var. Ancak bunları kullanırken dikkatli olun. Zira, defettikleri beladan fazlasını davet ediyorlar.

2 - Sapan: Bildiğiniz sapan. Ancak silahlardan daha çok işinize yarayacak. Sapanla, düşmanlarınıza uyuşturucu iğnelerden, izleme cihazlarına kadar gereçler fırlatırken; sağa sola da bozuk para, taş vs. atarak, dikkatlerini başka taraflara çekebiliyorsunuz.

3 - Uyuşturucular: Buradan oluşturduğunuz uyuşturucu ve eter mendillerini kullanabilirsiniz.

4 - Patlayıcılar: Aynı şekilde oluşturduğunuz patlayıcı mayınlara buradan ulaşabilirsiniz.

5 - Diğer Araçlar: Sigara, bozuk para gibi dikkat dağıtacak malzemeler ise buradadır.

6 - Sağlık : Oyunda düşmanlarınızdan veya sağdan soldan topladığınız ilkyardım çantalarına buradan ulaşabilirsiniz.

7 - X-Işını Görüntü Cihazı: Bu özel cihaz, size sadece duvarların arkasındaki kişileri değil; kimlerin silahlı olup olmadığını gösteren bir cihazdır. Odalara girerken veya köşeleri dönerken, bir sürpriz ile karşılaşmamak için kullanacağınız; ağırlığınca altın değerindeki bu cihazınız, oyun boyunca defalarca

hayatınızı kurtaracaktır. Bu cihaz ayrıca menzilli bir silahtır da. Bununla bakarken ateş tuşuna bastığınızda, baktığınız yere, yoğun bir radyasyon dalgası yollar. Bir insanın kafasına nişanladığınızda, onu kısa bir süre için sersemletir. Elektronik cihazlara (güvenlik kamerası, lazer çitler gibi) nişanlarsanız, onların iflas etmelerine yol açar.



Yandaki küçük kare ise ekranın sağ alt köşesinde olup, durumunuzu göstermektedir. Karenin en altındaki çubuk, sağlığını göstermektedir. Sağa doğru azalır ve tükendiğinde ölürsünüz. “6” tuşuna basarak, ilkyardım çantası seçip, “e” tuşu ile kullandığınızda sağlık kazanırsınız. Etkisiz hale getirdiğiniz insanların üstlerini aramayı ve sağa-sola bakınmayı ihmal etmeyin.

Karenin sağındaki dikey çubuk ise görünürlüğünüzü göstermektedir. Bu bar ne kadar boş ise, diğerlerinin sizi görmesi o kadar zor demektir. Görünmemek için, bir şeylerin arkasına çömelmek ve gölgelerde kalmak şarttır. Yine çömelerek görünürlüğünüzü azaltabilirsiniz. Bar tamamen boş ise, birinin sizi fark etmesinin tek yolu, size çarpmasıdır. Bu bar dolu ise kabalık gibi meydandasınız demektir. Ancak Spestnatz komandoları ile karşılaşırsanız, gölgelere saklanmak bir işe yaramaz. Gece görüş dürbünü olan bu adamların karşısına çıkmamak en iyisidir.

Sol dikey bar ise, X-Işını cihazınızın güç durumunu göstermektedir. Kullandıkça azalan bu bar biterse, cihaz devre dışı kalır. Kullanmadığınızda kendi kendine tekrar şarj olur.

Sağ en alttaki küçük kare ise, düşmanın alarm düzeyini gösterir. Üç renk olur:

Yeşil: Düşman alarm durumunda değil demektir.

Sarı: Düşman bir şeylerden kuşkulmuş, ancak ne olup bittiğini anlamaya çalışmaktadır.

Kırmızı: Düşman varlığınızın farkında ve sizi arıyor demektir. Bu durumda kaçıp izinizi kaybettirmeli veya bir yerlere saklanıp beklemelisiniz.

Ortadaki karede ise kullanımda olduğunuz (yani e tuşuna bastığınızda elinize gelecek olan) ekipman görüntülenir. Burada AK47 kullanımdadır. Soldaki 30 rakamı silahtaki kurşun sayısını; sağdaki 0 rakamı ise, elimde olan kurşun sayısını göstermektedir.

Oyunda dikkat etmeniz gereken bir başka unsur da, ateş ettiğiniz kişilerdir. Oyunda kişiler, düşmanlar, müttefikler ve siviller olarak üç sınıfa ayrılır. Bunları birbirine karıştırmamız pek mümkün olmasa da, nişan imleciniz her birinde ayrı renk almaktadır. Müttefiklerin üzerine gelince yeşil, sivillerin üzerine gelince sarı, düşmanların üzerine gelince ise kırmızı renk almaktadır. Sivilleri ya da müttefikleri öldürürseniz oyun biter. Ancak sivilleri bayıltabilirsiniz. Siviller silahsızdırlar ve sizi gördüklerinde genelde kaçarlar. Ancak kaçan bir sivil, kolluk kuvvetlerini alarma geçirdiği için, onlar da tehlikelidirler. Onlara da görünmememiz iyi olur.

Kaybolursanız (ki bölümlerin büyüklüğünden oldukça mümkün), "m" tuşu ile haritanızı açabilirsiniz. Burada, güvenli ka-

meralarını, lazer çitleri, üstlerine sapanla verici fırlattığınız ve normalde görünmeyen diğer karakterleri (düşman ya da sivil), kapıları(yeşil - kilitli olmayan kapılar, kırmızı - bir anahtar veya bir kontrol terminalinden açabileceğiniz kilitli kapılar, mavi - hiçbir şekilde açamayacağınız kilitli kapılar) ve kontrol terminallerini görebilirsiniz.

Bölüm içinde ne yapacağınızı unutursanız (gerçi pek olası değil) F1 tuşu ile yapmanız ve yapmamanız gerekenleri görebilirsiniz. F5 tuşu kayıt tuşu iken (oyunda en sık basacağınız tuş), F9 ile yükleme yaparsınız (bu da benim en sık bastığım tuş).

## MCCARTER

İşte oyunun en can alıcı ve kendine has bölümü. Mühendislik eğitimi almış gazeteci olan Carter'ın eğitimi burada ortaya çıkıyor. Elindeki imkanlarla zor durumlardan kurtulan ajan Mcgyver gibi Carter da sağdan soldan topladığı ıvır zıvır ile olmadık şeyler yapıyor. Mesela bir paçavra ve bir plastik şişe ile geçici bir susturucu (susturucum var diye seri saydırmayın sakın, üçüncü atıştan sonra parçalanıyor), teneke bir kutunun içine tıklımış silah mermilerinden yapılmış mayınlar gibi.

"F2" tuşu ile montaj menüsünü açıyoruz. Montaj kısmında iki kavramımız var. İlk olarak, oyun içinde bulabileceğimiz çantalar ve evraklardan kazanacağımız teknik puanlar. Bunları dolaplardan, masalardan ve baygın ya da ölü insanlardan çıkan dosyalardan kazanabiliyorsunuz. Bunlar karmaşık aletleri yapmamız için gereken seviyeleri belirleyen puanlar. İkinci olarak, mavi çizim olarak da adlandırılan teknik çizim taslakları



(yani yapılabilecek gereçler). Bunlar basitten karmaşığa doğru 7 gruptan oluşuyorlar. Geçici susturucudan, şarjör uzatmaya; mayınlardan dikkat dağıtıcı gereçlere ve X-Işını cihazı iyileştirmelerine kadar pek çok gereç var. İlk gruplar, az sayıda ıvır zıvır isterken; üst grup gereçler daha fazla materyale, alt grup gereçlere ve belli seviyede teknik puana gereksinim duyuyorlar.



Burada yapabileceğini ekipmanlar, açık renkle gösteriliyor. Daha koyu olanlar ise yapamayacağınız (ya gerekli materyaller eksiktir ya da teknik puanınız gereci yapmak için yeterli değildir) ise koyu renkli oluyorlar. Mavi renkli olanlar eli-

nizde mevcut olanlardır. Üzerinde tik işareti bulunanlar, yaptığınız ve tekrar yapılmasına gerek olmayan ekipmanlardır. Mesela ikinci grubun başındaki silah şarjörünün kapasitesini arttırmanın üzerinde tik işareti var. Onu bir kere yapıyorsunuz ve tekrar yapmanız gerekmiyor (silahınızı kaybetmedikçe tabii). Ancak pekçok gereci tekrar yapmanız gerekiyor. Mesela plastik şişeden bozma susturucumuz, tırıskadan olduğundan, üçüncü atıştan sonra dağılıyor. Tekrar yapmanız gerekiyor. Üçüncü grubun sonundaki ekipman gibi, bazı ekipmanların üzerindeki rakamlar, o ekipmanı yapmak için gereken teknik puan seviyesini gösteriyor. Yani o aleti yapmak için, 120 teknik puan seviyesinde olmam gerekiyor.

Fare imlecini bir ekipmanın üzerine getirdiğinizde, o ekipmanı yapmak için gereken materyaller aşağıda beliriyor. Yine bunların altındaki rakamlardan soldaki, ekipmanı yapmanız için gereken materyal miktarını gösterirken, sağdaki elinizde bulunan miktarı gösteriyor.

## SONSÖZ

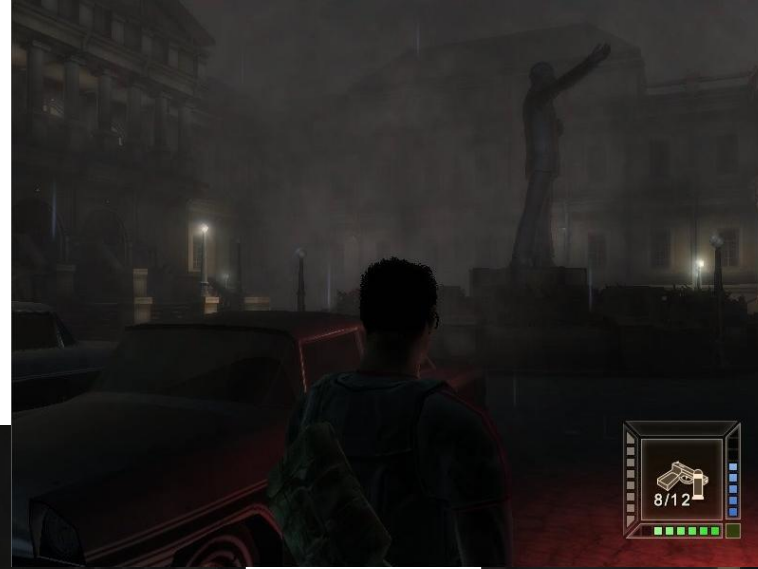
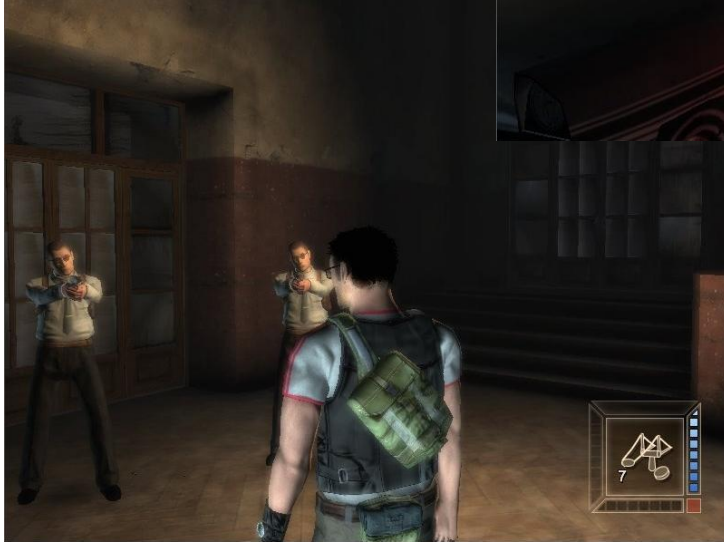
Cold War, tam anlamıyla bir casusluk oyunu. Carter'ın bir savaşçı olmaması ve yürüyen bir cephanelik gibi silah taşıyamamasının eksikliğini, oyun boyunca hissedeceksiniz. Dikkatli ve sabırlı oyuncular için çok da zor bir oyun değil. Bazı yerleri geçmek için olmadık şeyleri denemek hatta dakikalarca bir yerlerde bekleyip gözlemek zorunda kalabiliyorsunuz. Oyunun senaryosu gerçekten ilginç ve olaylar geliştikçe, pek çok sürpriz karşınıza çıkıyor. Grafikler genel olarak güzel. Bölüm tasarımları biraz şişirilmiş de olsa, karakterler -özellikle de Carter- oldukça detaylı tasarlanmış. Görüntü efektleri de iyi.

Bölümler arası demolar, çizgi roman tarzında yapılmış.

Yapay zeka ise tatmin edici düzeyde. Dolanırken pek belli etmeseler de, karakterler, oldukça gerçekçi tepkiler veriyorlar. Alarma geçen bir düşman, etrafta dolanıp sizi iyice arıyor. Hemen bakınıp gitmiyor. Çatışmalarda kendilerini ateşle koruyarak, en yakın siperin arkasına geçip oradan ateş etmeye başlıyorlar.

Sesler ise, özellikle böyle bir oyun için, oldukça yetersiz. Ayak seslerinden, kişilerin ne kadar uzakta olduklarını anlamamız pek mümkün olmuyor. Yine konuşmalardan kaç kişi olduklarını anlayamıyoruz. Halbuki bu tip bir oyunda, bunlara daha dikkat edilmeliydi diye düşünüyorum.

Hamit Giray Nart  
kuftikufti@yahoo.com



Bu sayımızda kategorize sistemini öğreneceğiz. Sistem 3 kategoride çalışacak. Merakınızı artırmamak için yazıları uzatmadan hemen örneklemeğe geçiyorum.

Önce komut satırımızda

```
python manage.py startproject yazkicom
```

yazarak yazkicom adında bir proje başlatıyoruz.

Herhangi bir metin düzenleme yazılımı ile settings.py dosyasını açıp veritabanı bilgilerini ve dil bilgilerini düzenliyoruz.

Sayfanın en başına her zaman ki gibi dil kodunu koyuyoruz:

```
# -*- coding: utf-8 -*-
```

```
DATABASE_ENGINE = 'sqlite3'  
DATABASE_NAME = 'yazkicom.db'  
DATABASE_USER = ''  
DATABASE_PASSWORD = ''  
DATABASE_HOST = ''  
DATABASE_PORT = ''
```

```
TIME_ZONE = 'Europe/Istanbul' # Ülke ve saat ayarları  
LANGUAGE_CODE = 'tr-TR' # Dil ayarları
```

Bu kadarı şimdilik yeterli.

Tekrar komut satırına dönüyoruz ve

```
cd yazkicom
```

yazarak proje klasörünün içine giriyoruz.

```
python manage.py startapp kategoriler
```

yazarak kategoriler adında bir uygulama oluşturuyoruz.

```
cd kategoriler
```

yazarak uygulamanın klasörüne geçiş yapıyoruz. Otomatik olarak oluşturulan dosyalardan models.py'yi herhangi bir metin düzenleme yazılımı açıyoruz:

```
# -*- coding: utf-8 -*-
```

```
from django.db import models #otomatik modeller  
from datetime import datetime # zaman ve tarih
```

```
class Kategori(models.Model):
```

```
# Kategori için sınıf oluşturuyoruz
```

```
    kategoriadi =  
models.CharField(blank=False,max_length=300,verbose_name="Kategori  
Adı", help_text='Max: 300 Karakter')  
# kategori adı için bir text alanı oluşturup en fazla 300 karakter  
girilecek kadar sınır koyuyoruz. blank false yazarak boş geçilemez  
yapıyoruz.
```

```
    slug = models.SlugField(max_length=300, help_text='Max: 300  
Karakter')
```

```
# slug otomatik link üretmek anlamına gelir. bu sayede kullandığımız  
temiz url'den istediğimiz gibi yararlanabiliriz.
```

```
def __unicode__(self):
    return self.kategoriadi
# kategori adının unicode desteği ile kullanılmasını sağlıyoruz

class Meta:
    verbose_name_plural = "Ana Kategori"
# Meta sınıfı ile Admin panelinde gösterilecek olan alanın adını
belirliyoruz.

class AltKategori(models.Model):
    kategoriadi = models.ForeignKey(Kategori)
    altkategoriadi = models.CharField(max_length=300, verbose_name="Alt Kategori Adı",
help_text='Max: 300 Karakter')
    slug = models.SlugField(max_length=300, help_text='Max: 300
Karakter')

def __unicode__(self):
    return self.alkategoriadi

class Meta:
    verbose_name_plural = "Alt Kategori"

class Urun(models.Model):
    altkategoriadi = models.ForeignKey(AltKategori)
    urunadi = models.CharField(max_length=300, verbose_name="Ürün
Adı", help_text='Max: 300 Karakter')
    slug = models.SlugField(max_length=300, help_text='Otomatik
olarak url için oluşturulur. Değiştirmeyiniz!.')
    tarih = models.DateTimeField(verbose_name="Oluşturma Tarihi",
default=datetime.now)
```

```
aciklama =
models.TextField(max_length=4000, blank=True, verbose_name="Açıklama",
help_text='Max: 4000 Karakter')
```

```
def __unicode__(self):
    return self.urunadi
```

```
class Meta:
    verbose_name_plural = "Ürün"
```

models.py dosyamızı bu şekilde düzenliyoruz. Veritabanında oluşturulacak alanları sınıf olarak tanımladık; içlerine oluşturulacak hücrelerinde sınırlarını, adlarını ve yetkilerini belirledik.

Settings.py dosyamızı tekrar açarak oluşturduğumuz uygulamayı projemize tanıtıyoruz.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin', # admin panelini aktif etmek için
    'yazkicom.kategoriler', # oluşturduğumuz uygulama
)
```

Uygulamayı ekledik. Şimdi de dosyaları tutacağımız klasör yollarını ekliyoruz.

```
DIRNAME = os.path.dirname(__file__)
MEDIA_ROOT = os.path.join(DIRNAME, 'static/')
MEDIA_URL = "/static/"
STATIC_DOC_ROOT = "/static/"
```

settings.py dosyasını kaydederek kapatıyoruz.

Şimdi de views.py dosyasını düzenleyerek veritabanında ki bilgileri alıp html dosyalarını göndereceğiz.

```
#-*- coding: utf-8-*-

from django.shortcuts import get_object_or_404
from django.shortcuts import render_to_response

# djangonun hazır getirdiği sayfa hatası ve yönlendirme sınıfları

from yazkicom.kategori.models import Kategori
from yazkicom.kategori.models import AltKategori, Urun

# oluşturduğumuz uygulamanın sınıfları

def kategorionsayfa(request):

    kategoriler = Kategori.objects.all().order_by('kategoriadi')

# kategoriler adında bir liste oluşturup Kategori sınıfından tüm
verileri alıp kategoriadi bilgilerini A' dan Z'ye sıralatıyoruz

    altkategoriler =
AltKategori.objects.all().order_by('altkategoriadi')
    return
```

```
render_to_response('kategori.html',{'kategoriler':kategoriler,'altka
tegoriler':altkategoriler })
# kategori.html adında bir sayfa oluşturup aldığımız liste
bilgilerini bu html dosyasına yönlendiriyoruz.
```

```
def kategoridetay(request, slug):
    gonderiler, sayfayagidecekbilgiler = init()
# gonderiler ve sayfayagidecekbilgiler adında listeleri init den al
    kategoridetay = gonderiler.get(slug=slug,)
# kategoridetay adında bir demet oluştu ve sınıftaki tüm verilerden
slugı oku ve sluga eşitle.
    sayfayagidecekbilgiler.update({'kategoridetay': kategoridetay})
# sayfayagidecekbilgilerdeki bilgileri kategoridetay daki sluga
göre oku. Yani sadece o sluga ait olan verileri al
        return render_to_response('altkategori.html',
sayfayagidecekbilgiler)
# ve aldığın verileri altkategori.html de göster.
```

```
def altkategoridetay(request, slug):
    altgonderiler, altsayfayagidecekbilgiler = init1()
    altkategoridetay = altgonderiler.get(slug=slug,)
        altsayfayagidecekbilgiler.update({'altkategoridetay':
altkategoridetay})
        return render_to_response('urun.html',
altsayfayagidecekbilgiler)
```

```
def urundetay(request, slug):
    urunposts, urunpagedata = init2()
    urundetay = urunposts.get(slug=slug,)
    urunpagedata.update({'urundetay': urundetay})
    return render_to_response('urundetay.html', urunpagedata)
```

```
def init():
    gonderiler = Kategori.objects.all()
# gonderiler adında bir liste oluştur
    sayfayagidecekbilgiler = {'post_list': gonderiler,}
# sayfayagidecekbilgiler adında bir demet oluştur
    return gonderiler, sayfayagidecekbilgiler

def init1():
    altgonderiler = AltKategori.objects.all()
    altsayfayagidecekbilgiler = {'post_list': altgonderiler,}
    return altgonderiler, altsayfayagidecekbilgiler

def init2():
    urunposts = Urun.objects.all().order_by('-tarih')
    urunpagedata = {'post_list': urunposts,}
    return urunposts, urunpagedata
```

evet views.py'yi de oluşturduk. models.py'den gelen verileri alıp html dosyalarına göndermek için listeler oluşturduk.

Şimdi de gelecek olan url sorgularına karşılık olarak urls.py dosyamızı düzenliyoruz.

```
# -*- coding: utf-8 -*-

from django.conf.urls.defaults import *
from django.contrib import admin

admin.autodiscover()
# bu satırları aktif ederek admin paneline girişi aktif etmiş olduk.
```

```
from yazkicom.kategoriler.views import *
# kategoriler uygulamasında ki tüm verileri ve yönlendirmeleri aldık

    (r'^kategori/$', kategorionsayfa),
# url /kategori/ şeklinde gelirse kategorionsayfadaki işlemleri gerçekleştir.

    (r'^kategori/([\w\ -]+)/$', kategoridetay),
# url /kategori/ altında herhangi bir url ile gelirse kategoridetayı çalıştır.

    (r'^kategori/altkategori/([\w\ -]+)/$', altkategoridetay),
    (r'^kategori/altkategori/detay/([\w\ -]+)/$', urundetay),
    (r'^admin/(.*)', admin.site.root),
# admin urlsini aktif ettik.
```

Admin panelini aktif etmişken, admin.py adında bir dosya oluşturulmamış ise (eski versiyonlarda) elle admin.py adında bir dosya oluşturuyoruz. İsterseniz buradan admin panelinde gösterilecek bilgileri özelleştirebilirsiniz.

```
# -*- coding: utf-8 -*-

from yazkicom.kategoriler.models import Kategori
from yazkicom.kategoriler.models import AltKategori, Urun
from django.contrib import admin

class Kategori_Admin(admin.ModelAdmin):
    prepopulated_fields = {"slug": ("kategoriadi",)}
    list_display = ('kategoriadi',)
    ordering = ('kategoriadi', )
```

```

search_fields = ('kategoriadi', )

class AltKategori_Admin(admin.ModelAdmin):
    prepopulated_fields = {"slug": ("alkategoriadi",)}
    list_display = ('alkategoriadi',)
    ordering = ('alkategoriadi', )
    search_fields = ('alkategoriadi', )

class Urun_Admin(admin.ModelAdmin):
    prepopulated_fields = {"slug": ("urunadi",)}
    list_display = ('urunadi',)
    ordering = ('urunadi', )
    date_hierarchy = 'tarih'
    search_fields = ('urunadi', )

admin.site.register(Kategori, Kategori_Admin)
admin.site.register(AltKategori, AltKategori_Admin)
admin.site.register(Urun, Urun_Admin)

```

Artık gelecek urlleride yönlendirdik. Adım adım incelersek:

```

#Veritabanını oluşturduk
#Veritabanına gereken alanları oluşturduk
#Oluşturduğumuz alanların adlarını, yetkilerini ve sınırlarını oluşturduk.
#Veritabanındaki bilgileri aldık ve listeleyerek html dosyalarına gönderdik.
#Admin panelinde gözükmesini istediğimiz bilgileri düzenledik ve özelleştirdik.

```

Artık bu bilgileri ekrana bastırmak kaldı.

Kategoriler için kategori.html alt kategoriler için altkategori.html ve ürünler için urun.html dosyalarını oluşturmamız gerekmektedir.

kategori.html

```
<h1>Kategoriler</h1>
```

```

{% if kategoriler %}
# if döngüsü ile kategoriler diye lsite varmı kontrol ettiriyoruz

{% for kategori in kategoriler %}
# for döngüsü ile kategori adında gelen tüm bilgileri topluyoruz ve sonuna kadar yazdırıyoruz

<ul><li><a href="/kategori/{{ kategori.slug }}"><h1>{{ kategori.kategoriadi }}</h1></a></li></ul>
{% for x in kategori.alkategori_set.all %}
# x adı ile kategorilerin altında ki alt kategorilere ait bilgileri alıyoruz

<ul><li><a href="/kategori/alkategori/{{ x.slug }}"> <h2>{{ x.alkategoriadi }} </h2> </a></li></ul>
{% endfor %}
# kategori döngüsünü bitiriyoruz

{% endfor %}
# alt kategori döngüsübü bitiriyoruz

{% else %}
# kategoriler adında liste gelmiyor ise

```

```
<p>Kategori eklenmedi.</p>
{% endif %}
# if sorgulamasını bitir.
```

### alkategori.html

```
<h2>{{ kategoridetay.kategoriadi }}</h2>
{% for x in kategoridetay.alkategori_set.all %}
<ul><li><a href="/kategori/alkategori/{{ x.slug }}">{{
x.alkategoriadi }}</a></li></ul>
{% endfor %}
```

### urun.html

```
<h2>{{ alktategoridetay.alkategoriadi }}</h2>
{% for x in alktategoridetay.urun_set.all %}
<ul><li><a href="/kategori/alkategori/detay/{{ x.slug }}">{{ x.urunadi
}}</a></li></ul>
{% endfor %}
```

### urundetay.html

```
<h1>{{urundetay.urunadi}}</h1>
<hr>
<p>{{ urundetay.aciklama|textile }}</p>
<h3>{{ urundetay.tarih|date }}</h3>
```

Artık bilgilerimiz veritabanından alındı ver urller yöntemi ile html dosyalarında gösterilmeye hazır.

Komut sisteminde projemizin klasör yoluna gidiyoruz.

```
python manage.py syncdb
```

Komutu ile veritabanını oluşturuyoruz ve ayrıca admin paneli ve yönetici bilgilerine ait soruları cevaplıyoruz.

```
python manage.py runserver
```

Komutu ile de tarayıcımızda projemizi kontrol edebiliriz.

Yalnız önce bir index sayfası oluşturmalı ve kategoriler için bir link oluşturmalısınız.

urls.py'nizi bu şekilde düzenleyebilirsiniz.

```
import os,sys
klasorYolu = os.path.dirname(__file__)
urlpatterns = patterns('',
    (r'^$', 'django.views.generic.simple.direct_to_template', {
'template': 'index.html' }, 'index'),
    (r'^kategori/$', kategorionsayfa),
    (r'^kategori/([\w-]+)/$', kategoridetay),
    (r'^kategori/alkategori/([\w-]+)/$', alktategoridetay),
    (r'^kategori/alkategori/detay/([\w-]+)/$', dersdetay),
```

Evet bir uygulama projesinide bu şekilde anlatmış olduk. Herhangi bir sorun olduğunda [www.yazki.com](http://www.yazki.com) ve/veya [www.djangoturkiye.com](http://www.djangoturkiye.com) adreslerinden bana ulaşabilirsiniz.

Muslu YÜKSEKTEPE  
musluyuksektepe@gmail.com  
[www.yazki.com](http://www.yazki.com)



HTML dosyalarının içinde Django uygulamalarını kullanmaya başlayacağız. viewların içinde HTML kodlarını fonksiyonların içine yazıyorduk:

```
def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>Now %s.</body></html>"%now
    return HttpResponse(html)
```

Bu kodları bir sitede yazmak şu nedenlerden kullanışlı değil:

#Sayfanın tasarımında bir değişiklik yapıldığında, Python kodlarında bir değişiklik yapılmalı.

#Python kodu yazmak ve HTML tasarlamak farklı işler. HTML/CSS tasarlayanlar Python kodu ekleyemeyebilir.

#Aynı anda Python kodçuları ve HTML tasarımcıları çalışırlarsa daha verimli olabilir.

Django HTML dosyası yazarak başlayalım:

```
<html>
<head><title>Ordering notice</title></head>
<body>
<h1>Ordering notice</h1>
<p>Dear {{ person_name }},</p>
<p>Thanks for placing an order from {{ company }}. It's scheduled to
ship on {{ ship_date|date:"F j, Y" }}.</p>
<p>Here are the items you've ordered:</p>
<ul>
{% for item in item_list %}
```

```
<li>{{ item }}</li>
{% endfor %}
</ul>
{% if ordered_warranty %}
    <p>Your warranty information will be included in the
packaging.</p>
{% else %}
    <p>You didn't order a warranty, so you're on your own when the
products inevitably stop working.</p>
{% endif %}
<p>Sincerely,<br />{{ company }}</p>
</body>
</html>
```

#Çift parantezle ayrılan metin bir değişkendir.

```
{{ person_name }}
```

#Parantez ve yüzde işaretiyle ayrılan metin bir etikettir.

```
{% if ordered_warranty %}
```

#Bir değişkeni biçimlendirmek için filtre kullanabiliriz.

```
{{ ship_date|date: "F j, Y" }}
```

Temel etiketlere bakalım:

### **if/else**

{% if %} etiketi bir değişkeni değerlendirir. Değişken "True" ise; sistem {% if %} ve {% endif %} arasındaki kodları gösterir.

```
{% if today_is_weekend %}
  <p>Welcome to the weekend!</p>
{% endif %}
```

Değerlendirmeyi genişletmek için `{% else %}` etiketini de kullanabiliriz.

```
{% if today_is_weekend %}
  <p>Welcome to the weekend!</p>
{% else %}
  <p>Get back to work.</p>
{% endif %}
```

Python'da ve Django HTML dosyası sisteminde şu nesnelere "False" değerini döndürür:

```
Boş liste      : []
Boş tüp        : ()
Boş sözlük     : {}
Boş dizim      : ''
Sıfır          : 0
Özel nesne     : None
Nesne          : False
```

Bir kaç değişkeni değerlendirmek ya da bir değişkeni olumsuzlamak için `and`, `or` ya da `not` kullanırız.

```
{% if athlete_list and coach_list %}
  Both athletes and coaches are available.
{% endif %}
```

```
{% if not athlete_list %}
  There are no athletes.
{% endif %}
```

```
{% if athlete_list or coach_list %}
  There are some athletes or some coaches.
{% endif %}
```

```
{% if not athlete_list or coach_list %}
  There are no athletes or there are some coaches.
{% endif %}
```

```
{% if athlete_list and not coach_list %}
  There are some athletes and absolutely no coaches.
{% endif %}
```

`{% if %}` etiketi `and` ve `or` aynı birlikte kullanmamıza izin vermez.

```
{% if athlete_list and coach_list or cheerleader_list %}
```

İşlem düzeni için parantez kullanımı desteklenmez. Sadece içiçe `{% if %}` etiketleri kullanın.

```
{% if athlete_list %}
  {% if coach_list or cheerleader_list %}
    We've athletes, and either coaches or cheerleaders!
  {% endif %}
{% endif %}
```

Bir kaç tane aynı mantıksal operatörü kullanabilirsiniz. Ancak farklı operatörleri birlikte kullanamazsınız.

```
{% if athlete_list or coach_list or parent_list or teacher_list %}
```

`{% elif %}` etiketi yoktur. Aynı işi içiçe `{% if %}` kullanarak yapabilirsiniz.

## for

`{% for %}` etiketi her maddeyi bir dizi şeklinde yapmanıza izin verir.

```
<ul>
{% for athlete in athlete_list %}
  <li>{{ athlete.name }}</li>
{% endfor %}
</ul>
```

Her `{% for %}` döngüsünde `forloop` değişkenine erişiyorsunuz. Bu değişken döngünün ilerlemesiyle ilgili bilgi veren bir kaç özelliğe sahip:

`forloop.counter`: Döngünün girilme sayısını bir tamsayıya atar.

```
{% for item in todo_list %}
  <p>{{ forloop.counter }}: {{ item }}</p>
{% endfor %}
```

`forloop.counter0`: `forloop.counter` gibi çalışır. Ancak 0'dan başlar.

`forloop.revcounter`: Döngüdeki tekrar eden maddelerin sayısını bir tamsayıya atar.

`forloop.revcounter0`: `forloop.revcounter` gibi çalışır. Ancak 0'dan başlar.

`forloop.first`: Döngünün ilk girişiyse "True" değerini döndürür.

```
{% for object in objects %}
  {% if forloop.first %}<li class="first">{% else %}<li>{% endif %}
  {{ object }}
</li>
{% endfor %}
```

`forloop.last`: Döngünün son girişiyse "True" değerini döndürür.

```
{% for link in links %}{{ link }}{% if not forloop.last %} | {%
endif %}{% endfor %}
```

Bu kodun çıktısı şöyle olacaktır:

Link1 | Link2 | Link3 | Link4

**forloop.parentloop**: İççe döngülerde ana döngüler için, **forloop** nesnesine bir referanstır.

```
{% for country in countries %}
<table>
  {% for city in country.city_list %}
    <tr>
      <td>Country #{{ forloop.parentloop.counter }}</td>
      <td>City #{{ forloop.counter }}</td>
      <td>{{ city }}</td>
    </tr>
  {% endfor %}
</table>
{% endfor %}
```

### **ifequal/ifnotequal**

Django tam olarak düşünülmüş bir programlama dili değildir ve bu yüzden size gelişigüzel Python koşullarını yürütmenize izin vermez.

`{% ifequal %}` etiketi iki değeri karşılaştırır ve değerler eşitse

`{% ifequal %}` ve `{% endifequal %}` arasındaki herşeyi gösterir.

```
{% ifequal user currentuser %}
  <h1>Welcome!</h1>
{% endifequal %}
```

Argümanlar sabit kodlu dizimler olabilir.

```
{% ifequal section 'siteneews' %}
  <h1>Site News</h1>
{% endifequal %}
```

```
{% ifequal section "community" %}
  <h1>Community</h1>
{% endifequal %}
```

`{% if %}` gibi `{% ifequal %}` etiketi de `{% else %}`i destekler.

```
{% ifequal section 'siteneews' %}
  <h1>Site News</h1>
{% else %}
  <h1>No News Here</h1>
{% endifequal %}
```

Sadece değişkenler, dizimler, tamsayılar ve ondalık sayılar `{% ifequal %}` etiketine argüman olabilir.

```
{% ifequal variable 1 %}
{% ifequal variable 1.23 %}
{% ifequal variable 'foo' %}
{% ifequal variable "foo" %}
```

## Yorumlar

HTML ya da Python'daki gibi Django da yorumlara izin verir. Yorum göstermek için `# #` kullanılır.

```
{# This is a comment #}
```

Çok satırlı yorum kullanmak isterseniz `{% comment %}` etiketini kullanmalısınız.

```
{% comment %}
This is a
multi-line comment.
{% endcomment %}
```

## Filtreler

Filtreler `|` karakterini kullanır.

```
{{ name|lower }}
```

Bu kod `{{ name }}` değişkeninin lower filtresiyle filtrelenmiş değerini gösterir. Filtreler sırayla kullanılabilir.

```
{{ my_list|first|upper }}
```

Bazı filtreler argüman alabilir.

```
{{ bio|truncatewords:"30" }}
```

Bu kod `bio` değişkeninin ilk 30 kelimesini gösterir. Bazı önemli filtreler şunlar:

**addslashes**: `\` işaretinden önce `\` işareti ekler.

**date**: `date` ya da `datetime` nesnesini parametrede verilen bir biçim dizimine göre biçimlendirir.

```
{{ pub_date|date:"F j, Y" }}
```

**length**: Değerin büyüklüğünü döndürür.

Artık HTML dosyalarını viewlarda kullanabiliriz. Şöyle bir view yazmıştık:

```
from django.http import HttpResponse
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>Now %s.</body></html>" % now
    return HttpResponse(html)
```

Bu viewı Django HTML dosyasına dönüştürcez. Şöyle yapabiliriz:

```
from django.template import Template, Context
from django.http import HttpResponse
```

```
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    t = Template("<html><body>Now {{ current_date
}}.</body></html>")
    html = t.render(Context({'current_date': now}))
    return HttpResponse(html)
```

Ancak bu şekilde yine kullanışlı olmayacak. HTML dosyasını ayrı bir dosya yapmalıyız; view bunu yüklemeli. HTML dosyasını başka bir yere yüklemeliyiz. HTML dosyasının yolunu şöyle düşünersek: /home/django/django\_test/site/time.html

```
from django.template import Template, Context
from django.http import HttpResponse
import datetime
```

```
def current_datetime(request):
    now = datetime.datetime.now()
    fp = open('/home/django/django_test/site/time.html')
    t = Template(fp.read())
    fp.close()
    html = t.render(Context({'current_date': now}))
    return HttpResponse(html)
```

Bu yöntem de iyi değil. Nedenleri şöyle:

- #Olmayan bir dosyayı ele almıyor. Time.html dosyası yoksa open() metodu IOError hatası verecek.
- #HTML dosyasının yolunu sabit kodluyor.
- #Çok fazla sıkıcı klişe kod içeriyor.

Django, HTML dosyalarını dosyasisteminize yüklemeniz için kullanışlı ve güçlü bir API sağlıyor. Bu API'yi kullanmak için framework HTML dosyalarının yolunu söylemelisiniz. settings.py dosyasındaki `ROOT_URLCONF`'u düzenlemeliyiz. `TEMPLATE_DIRS` şöyle öntanımlı şöyle:

```
TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates"
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)
```

Şöyle yapmalıyız:

```
TEMPLATE_DIRS = (
    '/home/django/django_test/site',
)
```

viewı düzenlemeliyiz. Şöyle olacak:

```
from django.template.loader import get_template
from django.template import Context
from django.http import HttpResponse
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    t = get_template('time.html')
    html = t.render(Context({'current_date': now}))
    return HttpResponse(html)
```

Site dizini altında time.html dosyasını oluşturabiliriz:

```
<html><body>It is now {{ current_date }}.</body></html>
```

Onur Tuna  
tuna.onur@yahoo.com

## Michael Foord kimdir ?

2002 yılından beri Python geliştiricisiyim. Python'la ilgili birçok makale yazdım ve bir kaç açık kaynaklı projeye katıldım. 2006 yılında, yüksek ölçüde programlanabilen bir elektronik tablola programı geliştiren Resolver Systems isimli bir şirketle çalışmaya başladım. Ardından IronPython geldi. İş arkadaşım ile beraber "IronPython in Action" isimli kitabı yazdım ve Microsoft'un ilk dinamik diller MVP'si seçildim.

2009'da Python'un ana geliştirici takımına katıldım, Python Software Foundation'ın üyesi oldum ve Python.org webmaster takımına katıldım. Ayrıca Amerika ve İngiltere'deki PyCon\* organizasyonuna yardımcı oluyorum.

Şu an Comsulting isimli bir Alman firmasında serbest geliştirici olarak çalışıyorum. Django ile web uygulaması geliştiriyoruz, büyük ölçüde Linux kullanıyoruz ancak, istemci kısmını IronPython kullanarak Silverlight üzerinden yazdık.

Python'la ve diğer teknik meselelerle ilgili yazı yazdığım bir blogum var.  
<http://voidspace.org.uk/blog>

Karımla Delia ile beraber Northampton, İngiltere' de yaşıyorum.

\* PyCon : Uluslararası Python programlama dili topluluğu, her yıl çeşitli ülkelerde programlama diliyle ilgili konferanslar düzenlenmesinden sorumludur.

## Python geliştirmenin hangi bölümünden sorumlusunuz?

Genelde standart kütüphanenin geliştirilmesine yardımcı oluyorum, Arada kaynak kodun parçalarının bağımsız olarak çalışıp çalışmadığını test ediyorum. Ayrıca standart kütüphaneyi IronPython'la uyumlu hale getiriyorum.





**Neden Python? Python'ı geliştirmenin arkasındaki fikir nedir?**

Python büyük bir codebase'e\* sahip, sık kullanılan bir dil. Kodun ve kütüphanenin bazı bölümleri oldukça eski. Elimizde genelde iyi test sonuçları var ve gün geçtikçe daha iyiye gidiyor. Ancak hala eklenmesi istenen bir çok özellik ve düzeltilmesi gereken bir çok hata var. Her zaman yapılması gerekenden fazla iş vardır ve bana katkı sağlamış bir özgür yazılım projesine katkıda bulunabilmek benim için bir ayrıcalık.

Kendimle alakalı bir not, büyük bir birim testi (UnitTest) hayranıyım. Ancak UnitTest ile sorun yaşadığımda nose ya da py.test gibi test kütüphanelerini kullandığım oluyor. Farklı test kütüphanelerinde denenmiş ve onaylanmış özellikleri kullanmayı seviyorum, standart kütüphanede test discovery' yi denemek gibi.

Son birkaç yılda Python çok gelişti. Uzun süredir var olan bazı sorunları yok etmek için Python 3 ile 2 arasındaki geriye dönük uyumluluk kalktı. Python 3 oldukça iyi bir sürüm ve bütün standart kütüphanelerle beraber geliyor, ancak Python kullanıcıları ve üçüncü parti kütüphaneler Python 3'e geçmek için biraz yavaş davranıyor.

Python'ın yeni sürümleri olan 2.7 ve 3.2 çok yakında çıkacak.

\* Codebase : Kaynak kodun tamamının kısmi bir uygulama ya da içerik yazmak için kullanıldığı yazılım geliştirme biçimi. Codebase genelde insan tarafından yazılmış kaynak kodlarını içerir; özellik ve yapılandırma dosyaları gibi, ikilik (binary) sistemdeki kütüphane dosyaları gibi değil.





Python 2.7 sürümünde; dekoratörlerden, üreteçlere (üreteçler 2.5 sürümüyle beraber çalışacak şekilde genişletildi), koşula bağlı durumlardan, “with” koşuluna ve abstract base sınıflara kadar bir çok yeni özellik ve çoklu işlemci kütüphanesi gibi yeni kütüphaneler bulunuyor

Python kullanıcılarının ve geliştiricilerinin tüm bu değişiklikleri benimsemek için zamana ihtiyaçları olacak. Python Language Moratorium'un (<http://www.python.org/dev/peps/pep-3003/>) iki yıl boyunca Python'a yeni özellik eklenmesine izin vermemesinin sebebi bu. Geliştiriciler için, dile yeni özellikler eklemektense standart kütüphaneye odaklanmaları için iyi bir fırsat.

## **Neden Python'ı seçtiniz ?**

Bir raslantı sonucu Python'la uğraşmaya başladım. Üniversitede bulunduğum süre içerisinde programlamayla uğraştım, Amiga'da assembly dili de buna dahil, ancak mezun olduktan sonra pek zaman ayırmadım. Bir gün arkadaşlarımla e-posta üzerinden Atlantis oynuyordum ve sıranın otomatik olarak ilerlemesi için bir program yazmak istedim. Kullanacağımız programlama dilinin PocketPC'de çalışması gerekiyordu, çünkü evde başka bilgisayar yoktu. Squeak kullanmaya karar vermiştik ki, son dakikada biri Python'ı önerdi ve kodu Python'la yazmaya başladık. Program yazmaktan oldukça zevk almıştım ve oyunu bırakmama rağmen Python'a ara vermedim.

## **İnsanlar neden Python'ı seçmeli ?**

Python'ın karşıma çıkması benim için bir şanstı. Programlamaya yeni başlayanlar için Python öğrenmesi kolay bir dil, ancak betik yazımından büyük çaplı uygulamaların geliştirilmesine kadar bir çok görevde

de kullanılabilir.

Python, desteklediği çok sayıda programlama paradigmasından ve stilinden dolayı oldukça esnek bir dil. Ayrıca Etkileşimli Yorumlayıcı (REPL) ile deneyim kazanmayı daha kolay hale getiriyor ve önemli uygulamaların geliştirilmesinde de kullanılabilir.

Programlamaya yeni başlayanlar betik yazımıyla başlayabilir, ihtiyaçları yoksa fonksiyon yazmalarına gerek yok. Ardından nesne tabanlı programlama için prosedürel programlamaya, daha sonralı fonksiyonel programlamaya hatta meta-programlamaya geçebilirler. Namespaceler sayesinde Python'la uğraşanlar tüm bu farklı paradigmaları kullanabilir.

Python zengin bir standart kütüphaneye sahip, bu da onu aklınıza gelebilecek bir çok görev için onu kullanıma hazır hale getiriyor. Ayrıca üçüncü parti kütüphanelerle ve genişletmelerle daha da büyüyor.

Okunabilirlik ve açıklık, Python dili ve kullanıcıları açısından büyük önem taşıyor. Satırların girintili kullanımından rahatsızlık duyanlar var, ancak kolay anlaşılabilir bir yapı sağladığı için memnun olan kişi sayısı daha fazla.

## **Diğer dillerle karşılaştırdığınızda Python ne gibi avantajlara ve dezavantajlara sahip ?**

Python anlaşılması oldukça kolay, dinamik bir programlama dili. Bu açıdan alt düzey programlama dillerinden ayrılır. Ör-

neğin; Python'da yazılmış bir uygulama olan Mercurial'ın codebase'i; C, Perl ve shell betikleri kullanılarak yazılmış aynı özellikteki Git'in codebase'nin onda biri kadardır. Java ve C# gibi statik dillerle karşılaştırıldığında Python'ın daha çabuk geliştirileceği ve sonuçta daha küçük bir codebase'e sahip olacağı görülmektedir.

Önemli bir nokta ise dinamik dillerin, statik olanlardan daha yavaş çalıştığı. Ancak kod yazmayı kısa sürede bitirirseniz, performansı geliştirmek için daha fazla zamanınız olur. Bunu gerçekten deneyip gördüm, farklı bir dil kullanmaktansa algoritmanın performansını geliştirmek daha mantıklı.

Statik programlamadan dinamik programlamaya geçenler, tip güveninin (type safety) eksikliğini hissedebilir, fakat güvenliğin sağlanmasında tip güvenli oldukça küçük bir noktadır. Bir program sadece derlenebildiği için uygulamanın mantığında hata olmadığı söylenemez, eğer bu mümkün olsaydı; statik dille yazılmış uygulamalarda hiçbir hata olmazdı. Kodun doğruluğundan emin olmanın tek yolu, onu teste tabi tutmaktır. Ben kullandığım bütün dillerde Test Driven Development'ı tercih ediyorum. Dinamik dilleri test etmek daha kolay, çünkü çalışma zamanında değişiklik yapabilirsiniz ve derleyiciyle uğraşmak zorunda değilsiniz. Python için daha güçlü test kütüphaneleri ve araçları mevcut.

## **Python'ın kullanım oranı hakkında ne düşünüyorsunuz?**

Python sık kullanılan bir dil ve son bir kaç yıldır kullanıcı sayısı artıyor. Bence Python'ın kullanıcı sayısının artmasında Ruby

ve Ruby on Rails'in tanıtımlarının da payı var. Son yıllarda dinamik diller sadece kabul görmedi; güncel ve ilgi çekici bir hal aldı. Zaten Python kullanımı çok farklı alanlarda büyüyor, sadece web geliştirme değil.

2002 yılında kullanmaya başladığımdan beri Python, Linux dağıtımlarında kullanılan temel dillerden biri haline geldi. Pardus bunun bir örneği, Ubuntu ve Gentoo'nun paket yöneticisi Portage da. Bunlar gibi daha bir çok örnek var.

Python; C, C++ ya da Fortran'la donatılmış Numpy, Scipy, BioPython, matplotlib, NLTK (Natural Language Toolkit) gibi güçlü kütüphanelere sahip olduğu için üst düzey programlama bilgisine sahip olmayan bilim adamları tarafından araştırma yapmak için de kullanılıyor.

Ayrıca Python standart olarak GIS'nin (Geographic Information Systems) ve CGI'nin (Computer Generated Imagery) içinde gömülü olarak bulunuyor. Pixar, Imageworks ve Industrial Light & Magic gibi büyük animasyon şirketleri Python kullanıyor. Blender ve Maya gibi uygulamalarda da be-tikler için Python kullanılmış.

Python donanımların denetlenmesinde, testlerde, uygulama ve web geliştirmede yani düşünebildiğiniz her şeyde kullanılıyor. Herhangi bir amaç için oldukça uygun bir dil.

**Python'da ikilik (binary) dosya yaratmak için kullanılan bir derleyici var mı? Yoksa siz böyle bir şey geliştirmeyi düşünür müsünüz?**

Python büyük ölçüde dinamik bir dil olduğu için çalışma zamanına ihtiyaç duyuyor. Aslında bytecode derlemeli bir dil olan Python, genelde yorumlanan dil (interpreted language) olarak tanımlanıyor, bunun sebebi



çalışma zamanında baytkodun yorumlanması.

Genelde Java ve C# gibi bytecode derleyen dillerin, derlenen dil (compiled language) olarak bilinmesinin sebebi; baytkodu yorumlamak yerine kodun kendisi için Just-In-Time (JIT) derleyicisine sahip olması.

Python'a JIT derleyicisini eklemek için gerçekleştirilen bazı projeler var. İçlerinden en önemlisi PyPy isimli bir aracın geliştirildiği proje. PyPy; Python'ın statik alt kümesi olan RPython'da (Restricted Python) kaynak koddan dinamik dil yorumlayıcısını derliyor. Bu proje ayrıca tüm Python kodunun; RPython'da, Prologic'te ve Smalltalk'ta yazılmasını öngörüyor. Bunun JVM (Java Virtual Machine) ve IL (Intermediate Language - .NET ve Mono uyarlamalarını çalıştırmak için Microsoft tarafından belirlenen Common Language Infrastructure'ı) back-endleri var, yani proje tek bir dile uyarlandığında bu üç platform üzerinde kullanılabilir.

Projenin oldukça ilginç ve heyecan verici yönü JIT derleyicisini içermesi. PyPy, derlediği dilin yorumlayıcısını izleyecek bir JIT oluşturuyor. Uzun bir süre boyunca bu yaklaşım oldukça deneysel ve varsayıma dayalıydı, fakat bir kaç yıllık çalışmanın ardından PyPy üzerinde CPython'da (C'de yazılmış standart Python uyarlaması) çalıştığından daha hızlı çalıştığını gösteren bir çok test sonucu var.

Python'ı sık kullanan ve ticari kaygılar açısından daha hızlı çalıştığını görmek isteyen Google'ın sponsoru olduğu Unladen-Swallow isimli bir başka proje daha var. Bu projede Python'a JIT desteğini eklemek için LLVM (Low Level Virtual Machine) kullanılıyor. Projede çalışan iki önemli Python gelişt-

tiricisi; performansı geliştirmek açısından LLVM'i Python çekirdeğine eklemek istiyor. Unladen Swallow'un son sürümü CPython'da hızın arttığını gösteriyor. LLVM, C++ ile yazılan bir proje ve Windows üzerinde tam olarak test edilmedi, bu yüzden LLVM'in Python çekirdeğine eklenmesinde sorun çıkarabilir, bekleyip göreceğiz.

Bunun haricinde Psyco adında CPython'a eklenebilecek özelleştirilmiş bir derleyici var. Nümerik kodlar için büyük bir hız artışını öngörüyor. Bu proje uzun süredir devam etmesine rağmen geliştirilmesine ara verilmişti. Kısa süre önce Christian Tismer sayesinde tekrar geliştirilmeye başlandı.

Ayrıca uygulamaların ikilik dağıtımlarını oluşturmak için bazı yaklaşımlar var. Bunlar genelde, çalışma zamanını ve kullanılan kütüphaneleri bir araya getirmeyi amaçlıyor. Windows'ta bunu yapan py2exe isimli bir araç var, Mac versiyonun adı py2app. Linux içinse cx\_freeze isimli bir araç ve adını hatırlamadığım bir kaç tane daha mevcut.

Python'da yazılmış ikilik uygulama yaratmak için diğer bir yaklaşım; Cython, Pyrex ya da Shedskin gibi melez dil kullanımından geçiyor. Python'ın statik bir alt kümesi olan Shedskin C++'ı derleyebiliyor. Python'ı, bazı dinamik özellikleri olmadan kabul edebilenlerseniz, uyumlu bir sözdizimi, performansta büyük artış ve derlenmiş ikilikler sizi bekliyor. Cython ve Pyrex, daha ziyade Python genişletmeleri için kullanılıyor. Bu ikisi, Python ve C kombinasyonundan oluşuyor. Eğer ihtiyacınız olan hızsa; Python sözdizimi ve semantiğinin C ile oluşturduğu kombinasyondan yararlanabilirsiniz.

## Python 2'yle karşılaştırıldığında Python 3 ne gibi avantajlara sahip ?

En önemli avantajı; string ve byte sorunun tamamen çözülmüş olması. Python 2'de iki tip string vardı; bytestring ve Unicode stringler. Python 2'de bu karmaşanın üstesinden gelmek mümkün olsa da, uygulamaların çözümlenmesi ve şifrelenmesinde kesin bir karar vermeniz gerekiyor, bu aşamada yanlış tercih yaparak text'e binary data gibi davranmanız oldukça kolay. Eğer Python 2'deki yaklaşımları birbiriyle karıştırırsanız - bazen Unicode, bazen text'e binary data olarak davranmak- garip yerlerde çıkan hatalarla çok uğraşmak zorunda kalırsınız. Python 3, tüm bu karışıklıkları ortadan kaldırıyor, çünkü içerdiği tüm stringler Unicode. Binary data için yeni tip bytelar ve bunu desteklemek için yeni bir IO katmanı var.

Bana göre sadece bu durum bile Python 3'e geçmek için yeterli bir neden. Bunun için yapılacak değişikliklerde geriye dönük uyumluluk korunamaz, çünkü çok esaslı bir değişiklik.

Bundan başka daha bir çok değişiklik var. Artık sadece tek tip integer var, int ve long birleştirildi. Sadece uyumlu tipteki veriler arasında karşılaştırma ve sıralama yapılabilecek, Python 2'deki gibi isteğe bağlı olmayacak. Bir çok yerleşik metod liste yerine adımlayıcı (iterator) döndürecek.

Favorilerimden biri Python 3'e eklenen nonlocal statement. Dış kapsamdaki değişkenleri değiştirmek için lexical scoping'i Python'a ekliyor. Python 2'de lexical scoping değerlere ulaşmak için uzun süre kullanılmıştı. Fonksiyon ve metod imzalarına (signature) fazladan bilgi eklenmesini sağlayan bazı de-

ğişiklikler var, bunlar contract yazımında kullanılabilir. Programcılar Python 3'ün yapabileceklerini yeni yeni keşfetmeye başladı.

Python 3'ü en büyük avantajı "yeni" olması. Bundan sonra Python 2 olmayacak. Python 2'ye eklenmesi mümkün olmayan özellikler ve kütüphaneler Python 3'e eklenecek.

## Python 3'ün bazı Python 2 kodlarını çalıştırmadığı biliniyor, bu gerçekten olması gereken bir şey miydi?

Maalesef, evet. Değişikliklerden bazıları, örneğin string/byte değişimi geriye uyumlu olamazdı. Açıkçası eğer mümkün olsaydı yapardık.

Python 2 ve 3'ün gelecek sürümlerinde ne gibi değişiklikler bizi bekliyor ?

Benim favorim, Python 2.7 ve 3.2 sürümlerinde gelen unittest kütüphanelerinin geliştirilmesi.

Python 2.7'de, 3'le birlikte yayınlanmış özellikleri azaltmayı planlıyoruz. Bunun Python 2'den 3'e geçişi kolaylaştıracağını düşünüyoruz. Python 2.7 için bunun anlamı sıralı sözlüklerin standart kütüphaneye Python 3.1 sürümünde eklenmiş olması.

Python 3.2 sürümü, yeniliklerin benimsenmesi açısından iki yıl geç çıkacağı için bir çok değişiklikle beraber gelecek. Bunun yanında çok sayıda geliştirmeyi içerecek; ondalık sayıların gösterimiyle ilgili Mark Dickinson'ın geliştirdikleri, Global

Interpreter Lock için Antoine Pitrou tarafından sağlanan performans iyileştirmeleri, "with" ifadesine bağlı çalışan çoklu içerik yöneticileri, bir çok hatanın onarımı ve daha fazlası.

Gelecekte standart kütüphaneyi geliştirmeye yoğunlaşılmasını istiyorum. Küçük bir kısmı hariç tüm standart kütüphane, Python 3'le çalışacak şekilde değiştirildi -en azından testler öyle söylüyor- , önceden stringlerle çalışan kodların baytlarla çalışıp çalışmadığını görmek gerekiyor.

## Python 2'nin süresi ne zaman dolacak ?

Muhtemelen Python 2.7 son büyük çaplı Python 2 sürümü olacak. Bunu sona erdirmek anlamında söylemiyorum ama Python geliştiriciler tarafından iki bölümlü geliştiriliyor ve geliştiricilerin çoğu sadece Python 3 üzerinde çalışmaktan yana.

Aslında bu sadece iki bölümle uğraştığımız anlamına gelmemeli çünkü Python 2.6 sürümüne devam ediliyor ve 2.7 geliştirme aşamasında. Ayrıca 3.1 sürümü devam ederken 3.2 sürümü de geliştiriliyor. Yani dört parçanın her biri için ayrı ayrı hata denetimi yapılmak gerekiyor.

Ayrıca Python 2.7 son büyük çaplı Python 2 sürümü olsa bile yıllarca devam edecek, ancak onun için bundan sonrası yok.

## Python'la programlama yaparken hangi kullanıcı arayüzünü tercih ediyorsunuz?

Geçmişte Tkinter (Python standart kütüphanesinin bir parça-

sını içeriyor) ve wxPython'ı bir arada kullanıyordum. Qt oldukça ünlü, ancak PyQt'yi Windows'ta kullanmak sadece GPL Lisansı'yla ya da ticari lisansla mümkün. Bu da bir çok işletim sisteminin bir arada bulunduğu platformlarda PyQt'nin geliştirilmesini zorlaştırıyor.

WxPython oldukça iyi. Python'da Dabo tarzında bir kütüphaneye kullanılarak geliştirilmiş ki bu da Python hissini veriyor.

Aslında geçmişte kullandığım tüm kullanıcı arayüzleri Windows Forms ile IronPython kullanarak çalışıyordu. Bu Windows'ta oldukça iyi gözüküyordu ve iyi bir yazılım programlama arayüzüne sahipti. Mono'nun Windows Forms'ta kodlanması oldukça kapsamlı bir iş, ancak varsayılan şekliyle Mac'te imde gerçekten kötü gözüküyor. Mac'te Mono'yla çalışan güzel Windows Forms uyarlamaları gördüm, ancak gereken sihirli numaraları bilmiyorum.

Eğer bir çok işletim sisteminin bir arada bulunduğu platformların lisansı ile ilgili PyQt'nin durumu değişirse, gelecekteki projeler için PyQt'yi kullanmak isterim. Kendi Qt kütüphanesine sahip Nokia'nın sponsoru olduğu PySide projesinde bu durum gerçekleşiyor gibi gözüküyor.

## Tkinter hakkında ne düşünüyorsunuz ? Geliştirilmesiyle ilgili planlarınız var mı ?

Tkinter oldukça garip, onu geliştirmek de öyle. Ama sadece basit bir şeye ihtiyacınız varsa, bunu standart kütüphaneden kullanmak oldukça iyi bir fikir.

Tkinter, Tcl dilinin bir parçası olan Tk'yi temel alıyor. Tk, Tcl'in

8.5 sürümünde sonra çok değişti ve Ttk isimli hoş bir kullanıcı arayüzüne sahip oldu. Ttk bu şekliyle oldukça iyi görünüyor ve Python 2.7 ile 3.2 sürümlerinde Tkinter'la birleşmiş durumda.

Ttk ile yazılmış bir çok uygulama gördüm, çoğu oldukça güzeldi. Ancak özgür halinden ya da doğal arayüzünden böyle-sine ilgi çekici hale getirmek için ne kadar “sihir” kullanılmış, bilmiyorum...

## Özgür yazılım hakkında ne düşünüyorsunuz?

Özgür yazılımın büyük bir hayranıyım ve açık kaynak hareketinin destekçisiyim. Ancak ticari yazılıma düşman değilim. Programcıların yazdıkları programlardan para kazanmaları bence iyi bir şey. Ticari yazılımla özgür yazılımı birbirine düşman olarak görmüyorum, bence ikisinin de birbirine faydası dokunabilir. Ticari yazılımın gelişmesi sonucu kazanılan parayla bir çok özgür yazılımın gelişmesine katkı sağlanabilir.

## Herhangi bir GNU/Linux dağıtımını kullanıyor musunuz?

Sunucular üzerinde

Linux kullanıyorum, genelde Debian tabanlı olanları tercih ediyorum.

## Pardus GNU/Linux'tan haberiniz var mı?

Python'ı kullandığı ve desteklediği için Pardus'un adını duymuştum.

## Python'ın lisansının GPL ile uyumlu olduğu bilinen bir durum. Neden buna ihtiyaç duyduunuz?

Bana göre bu pek fark yaratan bir durum değil. Kendi projelerimi lisanslamak için genelde BSD Lisansı'nı kullanıyorum, çünkü GPL Lisansı'ndan daha özgür. Diğer yandan Python'ın GPL Lisansı'yla lisanslanmaması ve GPL projelerinin Python'ı kullanamaması kötü bir durum olurdu. Bu açıdan Python'ın GPL Lisansı'yla uyumlu olması oldukça önemli.



**Sizce Python'ın .NET Framework'le bütünleşmiş kısmının açık kaynaklı alternatifinin yazılması nasıl bir anlam taşıyor?**

IronPython'ı, Microsoft'un .NET'ini kullanan geliştiricilerin ilgisini çekmek için iyi bir yol olarak görüyorum. IronPython, Microsoft tarafından geliştirilmesine rağmen tamamen özgür bir yazılım. Zaten IronPython'dan Python'a geçmek küçük bir adım.

**Kapalı kaynaklı uygulamaların yerini alabilecek açık kaynaklı yazılımlar geliştiriliyor. IronPython'ı geliştirmek yerine .NET'in yerini alabilecek bir alternatif yazılsa daha iyi olmaz mı?**

.NET'in Mono isimli oldukça iyi bir açık kaynaklı alternatifi var ve IronPython'ı çalıştırabiliyor. IronPython Microsoft tarafından geliştirilen bir yazılım, yani onu bırakıp Mono'nun geliştirilmesi gibi bir durum söz konusu değil.

**Python kullanarak yazılmış en sevdiğiniz özgür yazılım hangisi?**

Bu oldukça zor bir soru oldu. Sık kullandığım uygulamaların çoğu Python'la yazılmamış - özgür yazılım olmayan Wing IDE ve Resolver One isimli elektronik tablolaştırma programı hariç. Django, Python'da yazılmış en sevdiğim özgür yazılım, onunla web uygulaması geliştirmek çok zevkli.

**Okuyucularımıza son olarak söylemek istediğiniz bir şey var mı?**

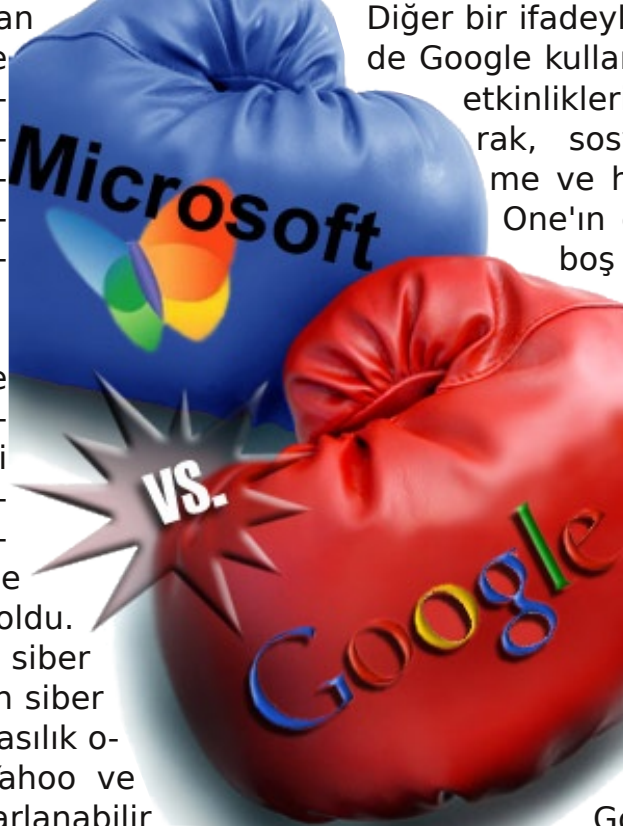
Bu röportaj benim için bir zevkti, okudukları için onlara teşekkür ederim. Son olarak, "A Little Bit of Python" isimli bir yayına katıldım, ilgilenenler bu adresten ayrıntılı bilgi edinebilir: <http://advocacy.python.org/podcasts/>

Çeviri: Melike İteralp  
mikene@gmail.com

## Giriş

Google arama motoru, 2009'da, arama motoru sektörünün hakimi olmayı başardı. Bu başarının iyi ve kötü yanlarını değerlendiren bir çok makale yazıldı ve dahası da yazılmaya devam edecek. Önemli olan Google'ın bundan sonra neler yapacağını; krizi ne şekilde yöneteceğini görmek. Google'ın Çin piyasasında yaşadığı krizi, bu açıdan değerlendirmek çok mantıklı olur. Böylece, gerçekçi ve yakın bir konuyu ele alarak, şirket büyümesini ve bunun getirdiklerini analiz edebiliriz.

Google, geçmişte Çin Halk Cumhuriyeti ile tartışmalı şartlarda altında anlaşmıştı. Ancak Çin ve Google arasındaki ilişkinin iyi olmadığı apaçık ortada. Çin'in internet özgürlüğünü kısıtladığı herkes tarafından biliniyor. Çin siber kullanıcı kitlesinin, sadece %30'luk bir bölümünün tercihi Google oldu. Buna ek olarak, Çin'den Google'a yapılan siber saldırılarla gerginlik iyice arttı. Google, Çin siber kullanıcıları piyasasından çekilmeyi bile olasılık olarak masaya koydu. Bu gerginlikten, Yahoo ve Microsoft'un arama motoru "Bing" yararlanabilir belki. Ama henüz açığa çıkan bir gelişme görünmüyor. Burada kilit nokta, mobil platformlar olarak görülüyor.



## Mobil Platformlar için yaşanan rekabet artacak..

Google; masaüstü, dizüstü ve mini dizüstü kullanıcıları tarafından büyük destek görüyor. 2009'daki eWEEK'in analizine göre; 2009 Aralık ayında 87.8 milyar aramanın kaydı tutuldu. Diğer bir ifadeyle, tüm dünyada yapılan aramaların %66.8'inde Google kullanıldı. Google, bu büyümeyi reklam ve arama etkinliklerini arttırarak sürdürmeyi planlıyor. Örnek olarak, sosyal platformlardaki eylemleri, anlık güncelleme ve her türlü alana özel arama planlarıyla Nexus One'ın dikkat çektiğini açıklamıştı. Ama rakipleri de boş duracak değil.

Microsoft ve Apple, iPhone'da Bing arama motorunu kullanma konusunu görüşüyor. iPhone piyasada çok tutulan bir ürün; bu durumda Bing, piyasa payını arttırmak için bütün makul yolları deniyor gibi gözüküyor. Makul olmayan yolların denendiğini zaten herkes biliyor. Çin ile yaşanan durumlar da hesaba katılırsa, Google için zorlu bir yıl geçecek gibi görünüyor. Yahoo ya da Bing Çin piyasasında tutulursa, Google'ın yaşadığı sorun rakipleri için önemli bir çıkış yolu olabilir.

Google, Nexus One'ı koz olarak elinde tutuyor. Chrome OS, Android gibi ürünlerle birlikte düşünüldüğünde, olabilecek gelişmelerin önceden kestirilmesi cidden zor. iPhone, piyasada büyük bir kullanıcı kitlesine hitap ediyor. iPhone'un arama motoru olarak Bing ayarlanırsa, Google'ın bu konuda ne yapacağı hala kestirilemiyor. "Chrome OS, yalnızca mini dizüstü bilgisayarlar için mi; yoksa akıllı telefonlar

“için mi kullanılacak?” sorusuna cevap bulmadan, mantıklı bir yorum yapmak zor. Chrome OS için , Ubuntu geliştiricileri de destek veriyor. GNU/Linux dağıtımları zaten mobil sektörde aktif; ama Chrome OS tüketiciye yepyeni bir ürün de sunabilir. Mini dizüstü ve mobil platformlar ortak bir noktaya gidiyor, Smartbook bunun en ilginç örneği. Mobil uygulamalar ve mini dizüstü uygulamaların bütün avantajlı yönlerini toplayan bir ürün çıktığında, çok büyük bir sürpriz olmaz herhalde.

## Sonuç olarak...

Google, kendi sektöründe lider konumunda. Ancak ona rakip çıkması sıkıntı yaratmaz. Bilişim sektörünün en son ihtiyaç duyduğu şey, yeni bir Microsoft. Ama bu rakip, Microsoft'un ürünü Bing olamaz. Piyasaya yeni oyuncuların ve makul bir rekabet ortamının gelmesi gerekiyor. GNU/Linux dünyası ve özgür yazılım geliştiricileri, bu yeni şirketlere destek verebilir ve verecektir. Küresel kriz, bir çok büyük şirketin küçülmesine neden oldu. Bu küçülmenin, makul rekabet ortamı için sebep hazırladığını umuyorum. Nasıl işletim sistemi alanında Microsoft'un tek kalmaması kesinse, bu diğer bilişim alanları için de geçerli. Ancak bu durumda siber kullanıcıların çıkarları yönünde işleyecek bir piyasa gerçekleşebilir.

GNU/Linux dolu günler dileğiyle.

Kemal Karataş  
kernelkema@gmail.com



## Giriş

Dünyamız, son yıllarda yığınla kötü olay yaşadı ve hepsi hala manşetlerde. İşgaller, küresel kriz, şirketlerin sonsuz kar açlığı, işsizlik ve kaynakların paylaşım mücadelesi vs... Sorunlarla uğraşmayan, dolaylı dolaysız etkilenmeyen kalmadı.

## Piyasa Oyuncuları

Microsoft gibi şirketlerin; kar payı artsın diye yapamayacakları taktik, kullanamayacakları ideoloji ve fikir yoktur. Aynı şey, Özgür Yazılım'ı, bilişim teknolojileri öğrencilerini, yüreklendiren projeleri destekler görünen; özgürlükçü söyleme sahip firmalar için de geçerli. Küresel kriz, en çok da bu alandaki tek şirketleri vurdu. 2007'de başlayan bu kriz dönemi, en çok özgür yazılım destekçilerine yaradı; ekonomi ve bilişim dünyası; kurumsallaşma, tüketicilerin özgürlüğü, maliyetlerin azaltılması gibi yeni terimlere aşına oldu.

Özgür Yazılım camiasının sürekli çalışması önemli bir etken, ama yine de bütün şartları oluşturmaya yetmiyor. Obama yönetimi ve AB'nin bilişim politikası da GNU/Linux dağıtımlarından yana son zamanlarda. Sol rüzgarların estiği Güney Amerika ülkelerini de unutmamak gerek. Obama yönetimi, "İnternet'in özgür ve eşit koşullarda kullanımı" konusuna olan hassasiyetini, yetkililerin konuşmalarıyla açıklıyor. Ayrıca, belli bir zümreye avantaj verecek ve tekelleşmeye yol açacak uygulamalara izin verilmiyor. Ama Microsoft'un haksız rekabeti konusunda da açıklama yapmıyorlar! Neden acaba? En son Google-Çin arasında yaşanan olayda, kaybeden önemli bir taraf da MS İnternet Explorer oldu. Almanya ve Fransa, resmi

kurumlarının, alternatif internet tarayıcılarına (Mozilla Firefox, Opera Web Browser, Google Chrome) geçeceklerini açıkladılar. Bu duruma Microsoft tepki gösterdi haliyle ve bu kararın alınmasında etkin olan güvenlik açığını çözecek güncellemeyi duyurdu. Ancak, kurumsal kullanıcılarının da bir bölümünü çoktan kaybetti. Yalnızca Özgür Yazılım destekçileri değil; yöneticiler ve bilişim piyasasının diğer tarafları tam desteklediğinde ancak böyle etkili sonuçlar olabiliyor.

## Bilişim Piyasası'ndan Özgür Yazılım'a Destek Artıyor

90'lı yıllarda yetişen bilişim uzmanları, Microsoft uygulamalarının eğitimini almadan bu sığara uygun kabul edilmezlerdi. Son birkaç yıla dek bu yine böyleydi; ama artık bu durum değişiyor. Kurumsal kullanıcılar, GNU/Linux dağıtımları ve Özgür Yazılım programlarına daha da çok destek vermeye başladılar. Artık Özgür Yazılım çözümlerini bilmek, gerek sistemler gerekse İnternet tabanlı uygulamalar üzerinde büyük bir avantaj. Microsoft sertifikalarıyla gezinen kişiler bile, harıl harıl GNU/Linux dağıtımlarını tanımaya, Özgür Yazılım'a, çalışıyor.

GNU/Linux vakfı da, bu gelişmelere göre, yeni eğitimlerini programladı. GNU/Linux dünyasına olan ilgiyi, uzmanlaştırmayla sonuçlandırmak önemli. Bu önemi kavrayan vakıf, hatırı sayılır uzmanları da bu eğitimlerde görevlendiriyor. Çekirdek derleme, gömme Linux sistemler, Özgür Yazılım programlamada kilit noktalar gibi önemli konularda eğitimler verilecek. "Windows' Seven Sins" adlı kampanya da son hızıyla devam ediyor. Hatırlanacağı gibi Microsoft, Windows 7 işletim sistemini piyasaya sürdüğü gün bu kampanya başlamıştı.

## Mobil Platformlarda GNU/Linux Büyümesi Son Hız Devam Ediyor

2010 yılı, mobil platformlarda yaşanacak kıyasıya bir rekabete gebe. Mini-dizüstü ve Mobil platformların aynı noktada buluşmasıyla, bu büyüme özellikle vurgulanmalı. Akıllı telefonlar gelişimini durdurmuş değil; mini-dizüstü ve akıllı telefon teknolojisinin en iyi ve şaşırtıcı yanı, farklı ürünler ve alanlarda karşımıza çıkmaları. Apple'ın yeni tablet ürünü iPad, buna en iyi örneklerden birisi; Amazon Kindle ürünü de. Microsoft'un bu yeni teknoloji ataklarına da karşılığı olmadığı sürece, her çıkan yeni ürün, şirketin daha çok yara almasına neden olacaktır. GNU/Linux tabanlı mobil uygulamalar, bu durumu daha da arttıracak. Nokia, Sony-Ericsson gibi önemli cep telefonu markaları, GNU/Linux tabanlı modellerini teker teker piyasaya sürüyor. Özgür Yazılıma yabancı olan kullanıcı kitlesi, bu yolla alışma ve ısınma dönemini geçiriyor. Cep telefonunda özgür yazılım uygulamalarına alışanlar, bunu neden bilgisayar için de denemesinler ki? Bu gibi sorular, Microsoft'un aklını kurcalamaya devam edecek gibi görünüyor.

Bu kadar değişime rağmen, Milliyet Gazetesi'nin Microsoft Office 2007 eğitimini övünerek vermesi ve adeta bir altın bilezik gibi sunması, Özgür Yazılım destekçilerinin daha çok çalışması gerektiğini hatırlatıyor. Dünyada olan gelişmeleri en son kabullenen Türkiye'de, bunları görmek şaşırtıcı değil; yalnızca üzücü. Türkiye'de, e-devlet ve diğer Microsoft tabanlı uygulamalardan, ne zaman vazgeçilecek diye merak ediyorum. E- Okul projesinde olan rezalet, buna en iyi örnektir. Bütün kritik işlemler, Microsoft ve 3. parti yazılımlar ile yapıldığı sürece, böyle rezaletleri daha çok göreceğiz.

## Sonuç

Dünyada olan gelişmeler umut verici. Ancak Türkiye'de olan Microsoft takıntısı'nın nedeni, cidden araştırılmalı. Yıllarca, aynı şirkete, milli servet ve gençlerin dimağlarını emanet ettiler. Sonuç ise koca bir sıfır! Amerika ve AB bu durumu anlayıp, dönüşümüne başladı. Eğitim kurumları, bilimsel projeler ve Jandarma gibi kurumları; Özgür Yazılım'a göç işlemi tamamladılar. Bu gelişmelere, Türkiye'de de başlanmasını ümit ediyorum. Dünya için gelişme dolu bir yıl, bizim için ise tüketime devam ettiğimiz bir yıl olacağı kesin gibi gözüküyor. GNU/Linux topluluklarına son hız desteğe devam etmek, umarım bazılarının kafalarında bir ışık yakar. Türkiye'de de, özgür yazılım düşüncesinden yetişmiş bilişim uzmanlarının çok uzakta olmadığını umuyorum.

Kemal Karataş  
kernelkema@gmail.com

## Giriş

Serbest piyasa ve rekabet, kapitalist sistemin olmazsa olmaz kuralıdır. Piyasaya asla müdahale edilmemelidir; çünkü riva-yete göre dengeyi kendisi bulacaktır. Bu denge ise, pazara sunulan mal ve hizmetlerin tüketimi arttıkça (alıcısı çoğaldık-ça) üretici sayısının da artması ve üreticilerin de, üretilen bunca mal ve hizmet arasında bir farkındalık yaratarak kendi müşteri yelpazesini oluşturması sonucu sağlanacaktır. Oluşan bu dengenin adı rekabet olup, üretimin tüketiciye kalite ola-rak yansımaları sağlamaktadır. Üstelik bu rekabet, müşterile-ri sosyo-ekonomik ve kültürel değerlere göre sınıflandıracak; ürünlerin maliyetinin, görünümünün, üretim biçiminin, pazarlanmasının ve fiyatlamasının da bu değerlere göre yapılmasını şart koşacaktır.

Bugün bilinen çoğu marka, kendisini, pazarda sahip olduğu ya da sahip olmak istediği müşteri yelpazesine ve rakiplerinin durumuna göre konumlandırmaktadır. Günümüzde, rekabet ederek rakibini alt etme politikası, rakibini satın alma ve te-kelleşmeyle yer değiştirmiştir. Bir çok firma, artık müşterilerini çeşitli tablo-grafik, sosyo-ekonomik-kültürel veriyle sınıf-landırmak yerine tek kalemde ikiye ayırmakta ve pazarlama işlevlerini, üretimini bu ayrıma göre biçimlendirmektedir: **Tü-keticinin ekonomik durumu.**

## Kısaca bazı pazarlama stratejileri

*Vestel-Regal, Renault-Dacia, CarrefourSa-DiaSa, Ülker-Halk, Tamek-Jucy, Milliyet-Posta, Sabah-Takvim, Arçelik-Beko-Altus, bilgisayar donanım ve sarf malzemelerde ise kutulu-OEM gi-*

bi... Peki, bu sıralamalar neden yapıldı? Cevap: Bu saydığımız ikinci markalar, geliri düşük ya da tutumlu-orta seviyede olan tüketiciler için , ilkinin güven ve kalitesiyle, üretilmekte ve pazarlanmaktadır.

Bu taktiğin dışında, firmalar ürettikleri ürünlerin özelliklerini, bir de fiyata göre değiştirirler. Örneğin; aynı otomobil üretici-sinin markasının farklı motor hacimlerine, konfor ve donanı-mına göre sınıflandırılması; aynı buzdolabının hacminin ve soğutma işlevinin farklı olması; aynı çikolatanın yağının azal-tılmışı; aynı yazılımın farklı özelliklere sahip sürümlerinin çe-şitlendirilmesi gibi. Yine sınıflamada, esas belirleyici ekono-mik durumdur ve aslında bu şekilde özgür olması gereken tü-ketici ihtiyacı, firmaların pazarlama ve fiyatlandırma faaliyet-leriyle yönlendirilmektedir. Örneklere devam edelim; motor büyüklüğü 1200 silindirin altında olan binek araçları, daha farklı vergilendirildikleri için farklı; bu hacimden yukarı olan-lar da gerek vergi oranının yükselmesi, gerekse bunu bilen ü-reticinin araca farklı donanımlar eklemesiyle fiyatı farklı ola-rak artar ve tüketici bütçesine göre hareket eder; eğer aracı illa alacaksa, önemli olan tatmin duygusudur.

Bir başka pazarlama ve satış tekniği ise, birbirine bağlı (yani birbirini tamamlayan) ürünlerden ana ürünün ambalajında, reklamında kendisini tamamlayanı tavsiye etmesidir. Örne-ğin; otomobil firmaları, araçlarında bilmem hangi petrol şirke-tinin ürünlerini; kimi beyaz eşya üreticileri, markalarının uzun süre verimli çalışması için şu deterjan firmasının sabunlarını; bilmem hangi un üreticisi şu kabartma tozunu; şu çay firması bu şekeri; gazetede tam sayfa reklamı çıkan dizüstü-masaüs-tü bilgisayar satıcısı, Microsoft Windows ürünlerini kullanma-mızı tavsiye ederek, tamamlayıcı malların satışını yaptırırlar.

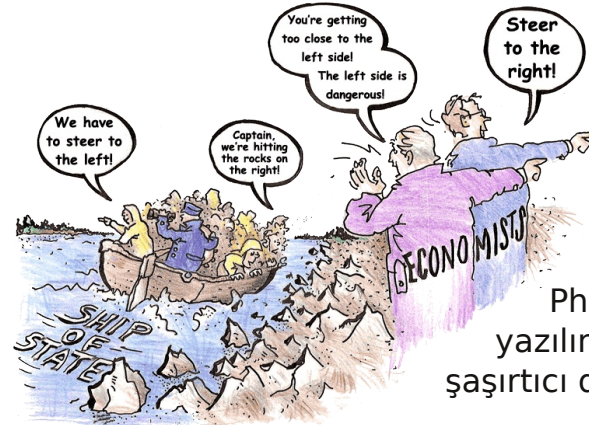
Bu, bilinçli ve tüketicinin tercihini yönlendirici bir satış tekniğidir. Bu teknik, özellikle de satın alma süreci içindeki tüketiciyi yönlendiren birinci unsur olan diğer tüketicilerin tavsiye, olumlu ve olumsuz görüşlerinden sonra gelir. Bu yönlendirme ile tüketici, tavsiye edilen tamamlayıcı malı ana ürün-hizmeti üreten firmanın güvenilirliğine, piyasadaki konumuna dayanarak satın alır.

## Bu anlattıklarımızın tüketiciler üzerinde pratiğine gelince...

Bugünün insanları için, bilgisayarların her türlü vazgeçilmez bir nesnedir. Gerek iletişimleri, gerek kişisel, gerekse iş amaçlı her alanda kullanılmaları, -fişe takılmadığı veya karşısına biri oturmadığı zaman, adeta aptal kutusuna dönen bu aletleri- değerli ve vazgeçilmez yapmakta.

Teknoloji zinciri mağazalardan hazır olarak alınan dizüstü, masaüstü ya da NetBook her türlü bilgisayarlar, eve varıldığında çevre donanımlarının kasaya takılmasıyla anında kullanılır hale gelirler; tabii içlerinde, istisnasız tamamlayıcı ürün olarak yazılımların ve kullanıcıların özgürlüğüne karşı olan Microsoft Windows işletim sistemi de çoktan kurulmuş olarak.

Yalnızca bilgisayarlar değil; birebir tüketici ile etkileşimli olan donanımlar da satın alındığında, şık kutularının içine kimi yazılım CD'leri koyarlar. Örneğin LG marka bir CD/DVD yazıcının yanında ücretli ve kaynak kodu kapalı bir yazılım olan Nero Burning Rom yazılımı; yeni anakartın kutusundan çıkan araç-gereç CD/DVD'si içinde PDF okuyucu olarak Adobe Acrobat Reader, koruma yazılımı olarak Norton Antivirüs, basit belge



Copyright 2007 Richard Stallman and Inne ten Have. Verbatim copying, distribution, and public display of this cartoon are permitted worldwide without royalty in any medium if this notice and the copyright notice are preserved.

yönetimi için Microsoft Works yazılımı; alınan yeni bir yazıcıdan, fotoğraf makinesinden basit fotoğraf işleme, arşivleme işlemleri için ACDSee, Adobe Photoshop Elements gibi yazılımların çıkması hiç de şaşırtıcı değildir.

Bu yazılımlar, kullanıcı için bir armağan değildirler; aksine deneme sürümü olan bu yazılımlardan memnun kalırlarsa, süre sonunda satın almaları istenir. Eğer satın alınmazlarsa da kullanılmaya izin vermezler ve kendilerini kullanılmadıkları için sistemden kaldırmazlar da; belki karar değişir de satın alınırsanız diye. Bu tam anlamıyla donanım üreticilerinin ve bilgisayar satıcılarının tüketiciye dayatmasıdır. Hem dayatma, hem de seçeneklerin olmadığına inandırmalarıdır; dahası korku. Oysa serbest piyasada seçenek her zaman vardır; çünkü rekabet mevcuttur. Yanılıyor muyum yoksa?

Serbest piyasanın kitaplarda yazılı olmayan rekabet anlayışına göre, iyi ve kaliteli hizmet veren değil; ürünlerini tüketici ile buluşturacak olan zincir mağazalarla yapılan, yüklü paraların döndüğü pazar payı anlaşmalarını gerçekleştirenler kazanırlar ve gerçek (!) rekabeti yaparlar. Sizlerden de, bu dayatma sonucu zorla aldığınız kutulardan çıkan ya da sizden parası bilgisayar-donanım içinde alınan yazılımların, alanında rakipsiz, seçeneksiz ve üstün olduğuna inanmanızı beklerler.

Devam edersek, çoğu bilgisayar (dizüstü, masaüstü, NetBook ve el bilgisayarları) satın alındığında bilgisayar, üreticisi - satıcı - Microsoft arasında yapılan anlaşma gereği, önyüklemeli olarak Microsoft Windows'un güncel işletim sistemi ürünü ile satılır. Bugünlerde çoğu bilgisayar Windows 7 ile ya da bu sürüme ücretsiz (tabii ki ücretli; neticede MS Windows Vista ile parası peşinen ödendi) yükseltme ürünüyle satılmaktadır. Kimsenin size sormasına gerek yok; çünkü anlaşmada sizin fikir beyan etme seçeneğiniz yok.



Bilgisayarlar, yalnızca Microsoft Windows işletim sistemiyle çalışacak biçimde tasarlanmadıkları gibi, sadece bu sistemle çalışmaları, onların gerçek gücünü, verimliliğini ve işlevlerini gösterememelerine neden olmaktadır. Haliyle de bilgisayarın yalnızca bu işletim sistemiyle çalışacağını kullanıcının aklına kazınması, serbest piyasanın altın kuralı rekabet ve ürün-hizmetlerin seçeneklerinin çokluğuna da karşı gelmektedir. Yoksa Microsoft Windows, sahip olduğu bunca hükümetler arası ilişkilere, sonsuz yazılı ve görsel basın desteğine, milyon dolarlar harcadığı pazarlama-satış-reklam-halkla ilişkiler stratejilerine, üretici ve satıcılarla yaptığı anlaşmalara rağmen hala kendisini Özgür Yazılım ve onun güzel bir ürünü olan GNU/Linux dağıtımlarıyla boy ölçüşemeyecek, rekabet edemeyecek denli güçsüz mü buluyor ki, serbest piyasa kurallarını işletmiyor?

### Bilgisayar Eşittir Windows, İnternet eşittir “em es en” denklemi

Konu bütünlüğünü bozmadan eklemek gerekir ki, yıllardır insanlar, küçük-büyük ölçekli firmalar ve kurumlar, bilgisayarların sadece Microsoft firmasının bu alandaki ürünü olan Windows serisi işletim sistemleriyle çalışabileceğini sanmaktalar; artık akıllarında **“bilgisayar eşittir Windows”** denklemi kurulmuştur.

Bu denklemin oluşabilmesi için Microsoft firması, yıllardır tekel olmanın da avantajını kullanarak, piyasaya egemen olabilmek için gerek hükümetlerle(kamu kurum ve kuruluşlarının), gerekse işletmelerle sistem yenilemeleri gibi anlaşmalar yapmış; İnternet, bilgisayar ve yazılım alanında rakip olarak gördüklerini sindirmiş; donanım üreticileriyle olsun, yazılım (oyun ve üçüncü parti, paket program) üreticileriyle olsun, satıcılarla olsun, hazır bilgisayar üreticileriyle olsun yaptığı destek anlaşmalarıyla, pazar hakimiyetini bir başka rakibe yer vermeyecek biçimde kurmuştur.

Satın alınan her yeni donanımın kutusunda mutlaka “Microsoft Windows 98, 98 SE, XP, Vista, Seven, Mac Os 9.x ile uyumludur”, “DirectX 9.c desteği vardır” gibi açıklamalar vardır ki bu açıklamalar alınan teknolojik ürünü sattırır. Oysa bugün satılan tüm bilgisayarlar ve donanımları, ülkelere ve ticaret bölgelerine göre geçerli ve uyulması gereken bazı kısıtlamalara, desteklemelere uymak zorundadırlar. Burada ana ölçüt, tüketicinin aldığı ürünü verimli ve istediği gibi kullanmasıdır.

Bir CD/DVD okuyucu, illa Microsoft Windows işletim sistemi ile



çalışacak değildir. İnternet'te yer alan birçok ev yapımı alet incelendiğinde, sayısal yayın alıcılarına yayın kaydedip çoğaltmak için, sabit disk ve CD/DVD okuyucu-yazıcı takılmış olduğu görülecektir. Hazır bir örnek ise; gelişen veri depolama aygıtları olan hafıza kartlarının da cep telefonlarından, DVD oynatıcılara dek kullanım alanının genişliğidir. Özellikle de "Microsoft Windows Vista ile uyumludur" etiketlerini, son bir buçuk yıldır, hemen her türlü bilgisayar donanım ve sarf malzemesinin üzerinde görmekteyiz. Bir LCD monitörün çalışmak için elektrik akımına ve göstereceği görüntünün kaynağına ihtiyacı vardır; Microsoft Windows Vista'ya değil. Bir örnek daha: görüntülü arama ve iletişim için vazgeçilmez olan İnternet kameraları (webcam) üzerinde ise, Microsoft Windows Live Messenger ile uyumludur etiketleri yer alır. Sonuçta kameranın da çalışma mantığı LCD monitör ile aynıdır.

Soru şu: Microsoft Windows Vista ile uyumlu olmayan donanımlar, fişe takıldığı zaman çalışmıyorlar mı? Daha önceki Microsoft Windows sürümlerinde nasıl çalışıyorlardı? Eğer çalışmazlar ise ne yapmak gerek; çöpe mi atılacak, sistemi mi güncellenecek, yoksa üzerine dantel işleme serilip evin bir köşesine mi konulacak?

Oluşturulan "tek bir sistemle çalıştırılabilir" etiketi ortaya çıkan faydayı sadece kendi ürününü sattırmak isteyen tekellerin seçenekleri dışlayıcı pazarlama faaliyetleri sonucudur. Yoksa kimse Microsoft firmasına aşık değildir.

Peki Microsoft ürünleri kullanıcıya fayda sağlıyor mu, ihtiyaçlarını karşılıyor mu? Yoksa sadece seçeneksiz ve cahil bir alışkanlık mı söz konusu? Bu soruya evet cevabını şeytanın ayrintıdaki gizliliğini önemsemeden ve tarafsızmış gibi davran-

madan cevaplamamız gerekir. Ancak bir soruyla: Microsoft başka fayda sağlayacak, ihtiyaçları giderecek yazılım firmaların ürünlerine yaşam hakkı tanıyor mu? Değil ürüne Microsoft'un evrensel standartları dahi kendi çıkarlarına yönelik olarak değiştirebileceğine şahit olmadık mı OOXML davası süresince?[1]

## Özgür Yazılım ve GNU/Linux dağıtımları pratiğine gelecek olursak...



1983 yılında, Stallman'ın, yazılımların toplumların malı olduğu ve özgürce, kısıtlama olmaksızın ticari kaygılardan sıyrılması\* (gerçektende; ticari kaygılarla bilim yapıldığını düşünebiliyor musunuz?) gerektiğini açıklayan manifestosuyla başlayan ve donanımlarla daha kolay anlaşmamızı sağlayan çekirdeğin (Kernel), Linus Torvalds tarafından bulunması ve sonuçta rekabetten hoşlanmayan, bilgisi ve bilimi kazanç amaçlı kullanan firmalardan korumak için GNU Genel Kamu Lisansı ile koruma altına alınmasından bugüne, Özgür Yazılım ve GNU/Linux dağıtımları, bir hayli yol aldılar.

Hatta artık bugün kendi aralarında yarışmaktalar ve tamamı ile insanlığa, bilginin gelişimine hizmet eden bir rekabet halini almıştır.

[1] OOXML'e Karşıyız! <http://www.ozgurlukicin.com/ooxml/>

Dünya üzerinde satılan her bilgisayarda kurulu olmasa ve satılmasa da, GNU/Linux dağıtımlarını kullananların, sayısı gün be gün artmakta. Alınan mesafe ve kullanıcı sayısının artmasında kuşkusuz itici unsur olan İnternet ile bilgisayarların ucuzlaması, her ikisine de erişimin kolaylaşmasının rolleri büyüktür.

Özgür Yazılım ve GNU/Linux dağıtımlarının devasa genel merkezleri, ülke ofisleri, satış temsilcileri, şık kutulu ürünleri yok\* (Ubuntu'nun Shiplt hizmetiyle ücretsiz gelen şık kutulu kurulum CD'leri, SuSE'nin kalın el kitapçığıyla ve kutulu gelen paralı, ama kodlarına erişim tamamen mümkün...) Hatta bu hareketin tek gelir kaynağı bağışlar, satın alınan penye, gömlek, kupa, fincan, rozet, çıkartma, fare altlığı, küçük not defteri gibi ürünlerden oluşmakta.

Üstelik sürekli birbiriyle haberleşen topluluk gönüllüleri yaptıkları harcamaları da kendileri karşılamaktalar. Kimse onlara yerleştirme ve geliştirme çalışmaları, sürüm denemeleri için bilgisayar hediye etmiyor, evlerine internet bağlamıyor!

Belki de gerçekten de bu denli güçlü olmalarının, verimli çalışmalarının, eksiklerini hızlı kapatmalarının, desteklerinin hızlı olmasının ve özgürlüklerinin, yaratıcılıklarının tek nedeni, yazılımların özgürlüğüne inanmaları ve bu uğurda mücadele etmeleri.

Mücadele kime mi karşı? Yazılımların özgürlüğüne karşı gelen ve içeriklerini, kaynağını paylaşmayı reddeden, evrensel bilgi birikimini kendi ticari amaçları için el koyan yazılım firmalarına, tekellere ve piyasada ayakta kalıp pazardan pay kapmak uğruna, bu tekellerle asalak bir ilişki kuran donanım üretici ve

satıcılarına karşıdır. Bir de Özgür Yazılım'ı ve fikri mülkiyet haklarının topluma verilmesini amaçlayan Genel Kamu Lisansı'nı, kanser gibi gören ve de bu betimlemeye destek olan kimi medya organlarına (özellikle de bilişim, bilgisayar dergileri\* -artık internetin gelişmesi ve donanımlara erişimin uygun hale gelmesiyle sadece reklamdan ibaret hale gelmişlerdir günümüzde-) karşıdır.

Üstelik mücadeleleri, kutsal olan bir görevi de kapsamaktadır. Serbest piyasanın tekelleşme olgusu karşısında seçeneksizleştiği yazılım ve işletim sistemi pazarında;

#özgür,  
#kaliteli,  
#güvenilir,  
#verimli,  
#ücretsiz,  
#erişilebilir,  
#şeffaf

yazılımlar ve işletim sistemlerinin olduğunu kullanıcılara anlatmak, tanıtmak, benimsetmek ve her koşulda verimli kullanım için yardımcı olmak.

Bilgisayar eşit değildir Windows ve eşit değildir GNU/Linux, Apple MacOS X; ne de bir başka işletim sistemi. Evrendeki her işimizi yapan, verimlilik-eglençe-öğrenme sağlayan bu araçlar, çok yönlüdürler iyi derlenmiş, verimli ve çalışabilir

## GNU/Linux



hale getirilmiş her türlü işletim sistemini çalıştırabilirler.

**Kısacası yazılımların özgürlüğü eşittir bilgisayarların özgürlüğü!** Bu ikisinin gerçekleşmesi ise insanın özgürlüğüne kapıyı açacaktır.

Aydın Bez  
İstanbul, Şubat 2010  
bilgi@ozguryazilimsendikasi.org

## Lisanslar:

Makalenin tüm içeriği GNU/GPL 3 ve Creative Commons (by-nc-sa)3 ile lisanslanmış olup içeriği haber verilmek ve yeniden GNU/GPL ve Creative Commons (by-nc-sa) ile lisanslanmak koşuluyla kopyalanabilir, düzenlenip değiştirilebilir, atıfta bulunulabilir, yeniden dağıtılabilir.



14 Aralık 2009 tarihinde, derneğimiz başkanı ve Linux Kullanıcıları Derneği üyesi Av. Nihad Karslı'nın Kasım 2008'de satın aldığı bilgisayar ile aldığı bilgisayar ile verilen Windows işletim sistemini istememesi üzerine açtığı davanın sonuçlandığını belirtmiştik.

Geçtiğimiz günlerde ise gerekçeli karar açıklanmış, böylece satın almak istediği bilgisayar ile verilen işletim sistemini almak istemeyen tüketiciler için satıcıya karşı bir dayanak sağlanmış oldu.

Gerekçeli karar hakkındaki duyurumuzun ardından, çeşitli sitelerde birtakım yorumlarla karşılaştık. Seviyesiz yorumlara da rastladığımız bu yorumlar bütününde karşılaştığımız ortak bir hata, sanki bilgisayar ile birlikte Windows işletim sisteminin satılmasını engellemişiz gibi yorumlarda bulunulması oldu. Gerekçeli kararda da belirtildiği üzere, bizim anlatmak istediğimiz işletim sistemlerinin bilgisayarların olmazsa olmaz parçaları olmadığı ve tüketiciler isterlerse bilgisayar ile verilen işletim sistemlerini satın almak zorunda olmadıklarıdır.

Dava ile ilgili gerekçeli karar ile birlikte, bilirkşi raporlarını [buradan](#) edinebileceğinizi belirttikten sonra Av. Nihad Karslı'nın Haziran 2008'deki "**Dünyaya Kimin Gözleriyle Bakıyoruz?**" başlıklı yazısıyla sizleri baş başa bırakıyoruz:

*Bu soruyu kendime sorduğumda yanıtı çok kolay verebiliyorum. Türk vatandaşı olarak bağımsızlığı ruhumuza kazımış Mustafa Kemal'in evladı olarak bakıyorum diyebilirim. Ayrıca hukukçu kimliğim gereği haklı haksız ayrımını sürekli yaparak yaşıyorum. İşte bunlar benim gözlerim. Peki biz dediğimizde, yani bu ülkenin insanları olarak nasıl bakıyoruz?*

*Asıl mesele burada başlıyor. Ayakkabısının markasından yemek yediği lokantanın adına kadar marka ve gösteriş bağımlısı Türk görünümlü gringo olarak görüyorum bizi ben. Köyünde hala taş ile büyük abdestini yapan insanlar varken Paris'te diordan alışveriş yapan bizi görüyorum. Açlık sınırının altında 12 milyon insan yaşarken, kapalı kaynak kod çetelerine milyonlarca dolar ödeyen bizi görüyorum.*

*Şimdi benle sizin çeliştiği noktada hukukçu olarak bir şeyler söylemek beyninizde fırtınalar yaratmak ve gözlerinize yeni gözler eklemek isterim. Türk Ceza Kanununda sanırım 220. maddede bir tanım var: Suç işlemek amacı ile örgüt kurmak. Bunun gerçekleşebilmesi için ise en az üç kişi gerekiyor. Bu üç kişi, bir suçu örgüt kurarak, yardım ederek, propagandasını yaparak işleyebiliyor. Nereye geleceğimi sanırım anladınız. Konumuz özgür yazılım olunca, karşıtı özgür olmayan yazılım yani kapalı kaynak kod ve onun savunucuları olduğu açıktır.*

*Bu gün tüm dünyada bu özgür yazılım karşıtı gizli kodcu yazılım firmaları ne insanların isteklerini karşılayabiliyor ne de emeklerinin karşılığını alıyorlar. Emeklerinin karşılığını değil ayakkabımızın markasına bağlı lokantamızın ismine bağlı olduğumuz gibi, marka ve patenlerini diledikleri fiyata satıyorlar. Ve bizi sömürüyorlar. Bunu da tek başına bir manda-soft olarak yapmıyorlar. Bir çete halinde tüm yazılımlarını birbirlerine uyumlu ve özgür yazılıma uyumsuz halde yapıp insanları bağımlı hale getirmeye çalışıyorlar. Tıpkı tütün satıcı firmalar gibi. Örnek: ooxml*

*Yine bu çete ve üyeleri ülke ülke gezerek kapitalist sistemi tercih etmiş yöneticileri ile birebir ilişkilere girip onlarla em-*

pati kurup kendi gizli kodlarını asıl hale getirmeye çalışıyorlar. Bunun için haksız kazandıkları dolarların bir kısmını da harcıyorlar. Kim ne kadar sebepleniyor bu işden bunu kimse bilemez tıpkı gizli kod gibi bu ilişkilerde gizleniyor.

Ülkemiz için örneği verir isek, tamamen gizli kod uygulamasını benimsemiş bir Milli Eğitim Bakanlığı bu şüpheyi uyandırıyor. Ben Milli Eğitim Bakanlığının hiç bir uygulamasını kullanmıyorum. Bana diyor ki, gideceksin şu çetenin gizli kodlarını satın alıp beni öyle kullanacaksın. Peki neden? Benim sömürülmem benim bakanlığım neden aracılık eder. Hadi bırakalım dünyadaki diğer özgür yazılımları neden Tübitak'ın geliştirdiği Türkçe dağıtım olan Pardus'u kullanmaz, kullanılmasına izin vermez. Yanıtını ben bilmiyorum, ama şunu söyleyebilirim ki ben sömürüyorum, bununla birlikte çocuklarımız da gizli kodun tiryakisi haline getirilerek bağımlı hale sokuluyorlar.

Bu çetenin reisi asıl kullanıldığı ülkede dahi haksız rekabet nedeni ile milyarlarca dolar tazminat ödemekte, hatta üyesi olduğumuz Avrupa Parlamentosu ve üye olmaya çalıştığımız Avrupa Birliği de bu çeteye milyarlarca Avro para cezası kesmekte iken benim ülkem 12 milyon açlık sınırının altında yaşayan halkım bu çeteye sömürtülmektedir.

Ceza yasası ve insan hakkı kavramlarının felsefinden bahsetmek istiyorum. Örneğin ülkemizde uygulamada hırsızlık düzenlemesinde malın değerinin az olması ya da ağır ve acil bir ihtiyaçtan dolayı işlenmesi halinde verilecek cezada indirim yapılır ya da hiç ceza verilmez. Peki bu çetenin çalışması sayesinde işletim systemsiz bilgisayar satışı neden yasak? Bir bilgisayar işletim systemsiz ne işe yarar? Bilgisayar satıcıları

bu çete ile nasıl bir işbirliğine girmişler ki gizli kod işletim sistemi olmayan bir sistemi satmazlar.

Burada temel hukuk kuralı olan yaşam hakkından bahsetmek isterim. Yaşam hakkı hukukun temelidir. İnsanın insan olabilmesi için yaşaması gerekir. Şimdi 12 milyon açlık sınırının altında yaşayan halkın ekmek çalmasına kim hırsız diyebilir. Sadece bu çete ve mensupları diyebilir sanırım. Bir benzetme yaparsak artık günümüzde iki dünya var: Biri bilgisayarlı yaşam diğeri doğal yaşam. Bir bilgisayarı insana benzetirsek onun yaşam hakkı için temel olan nedir? Tabi ki işletim sistemi. Tıpkı insanların en azından yaşamak için ekmek yiyebilmeleri gibi. Peki biz yaşamak için bu çeteye ödeme yapmak zorunda mıyız?

Ülkemin bilgisayar kullanıcılarının %90'ı bu çetenin gizli kodları ile çalışan sistemini kullanmaktaymış. Bunların %90'ı da korsanmış, hırsızmış deniyor. Bunların kimin adına konuşulduğunu göreceğiz gözlerden bakalım istiyorum dünyaya.

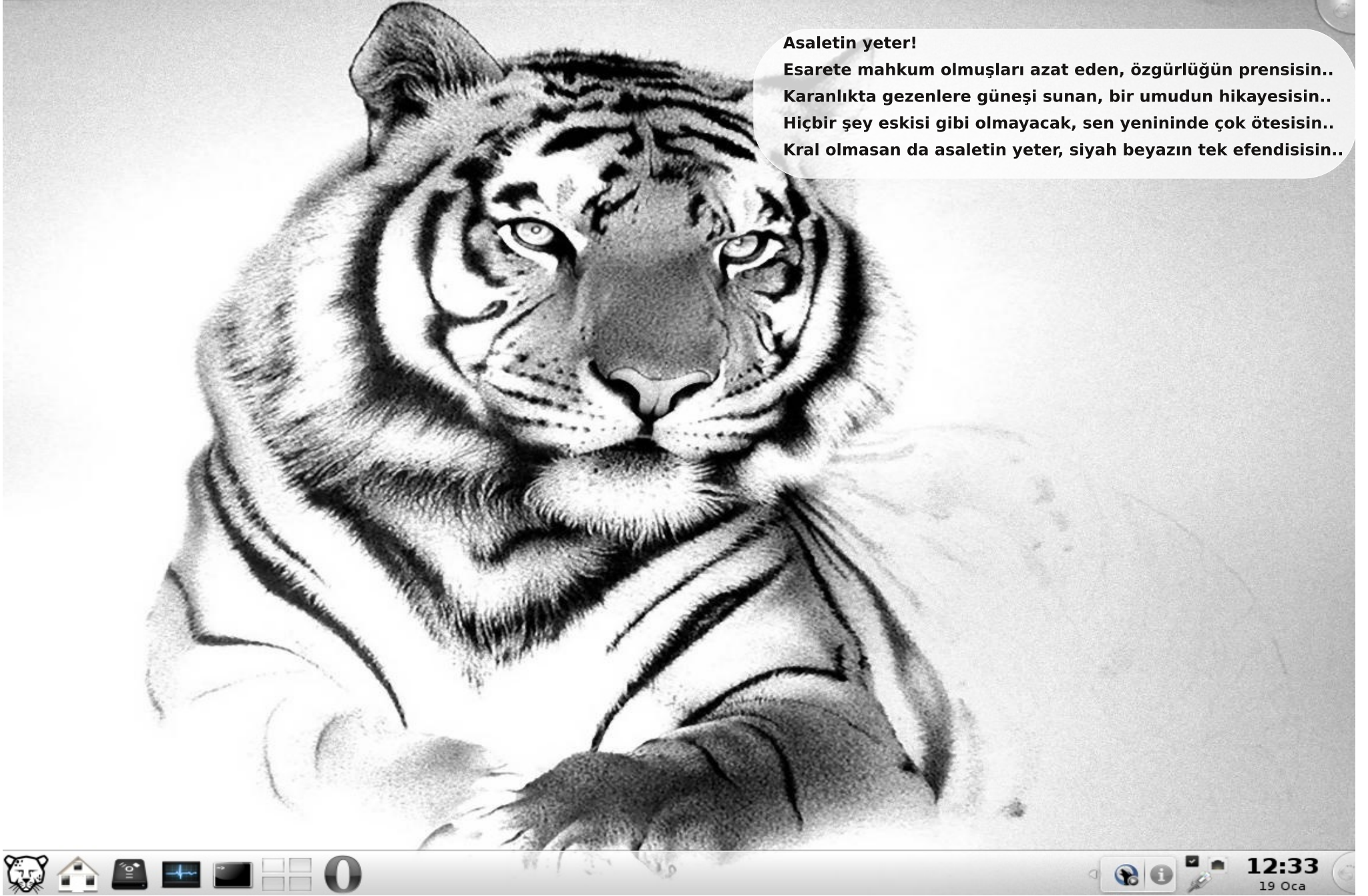
Bu yazıları okuyan birisi beni fanatik bir Stallman yanlısı olduğumu düşünebilir. İşin gerçeği, insanlığın henüz Stallman'ın düşüncelerine alışabileceğini ve uygulayabileceği bir seviyeye ulaşabileceğine ihtimal vermiyorum. Yine de iyi ki Stallman gibileri var iyi ki "robin good" lar var.

Beni dinlediğiniz için teşekkür ederim. Özgür yazılımla, özgürlüklerle olunuz.



**Firefox® 3**

- Pardus 2009.1 Telli Turna
- KDE 4.3.4 Masaüstü Yöneticisi
- Kick-off Menü Görünümü
- Glassified KDE 4 Teması
- Buuf Deuce KDE Simge Seti
- Alt panel için sayısal saat, görev yöneticisi ve sistem çekmecesi
- Gimp ile resimlerin birleştirilmesi ile oluşturulmuş masaüstü duvar kağıdı.



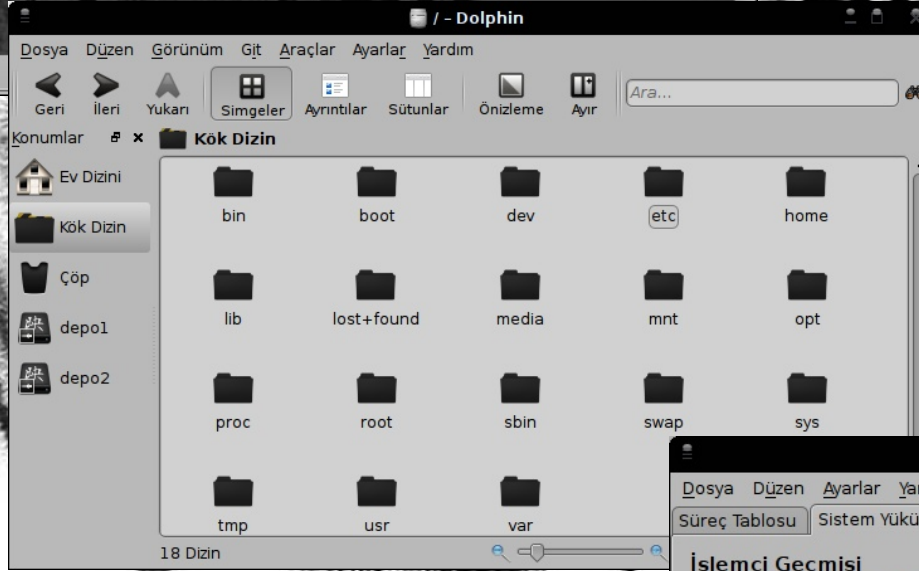
**Asaletin yeter!**

Esarete mahkum olmuşları azat eden, özgürlüğün prensisin..  
Karanlıkta gezenlere güneşi sunan, bir umudun hikayesisin..  
Hiçbir şey eskisi gibi olmayacak, sen yeninde çok ötesisin..  
Kral olmasan da asaletin yeter, siyah beyazın tek efendisisin..



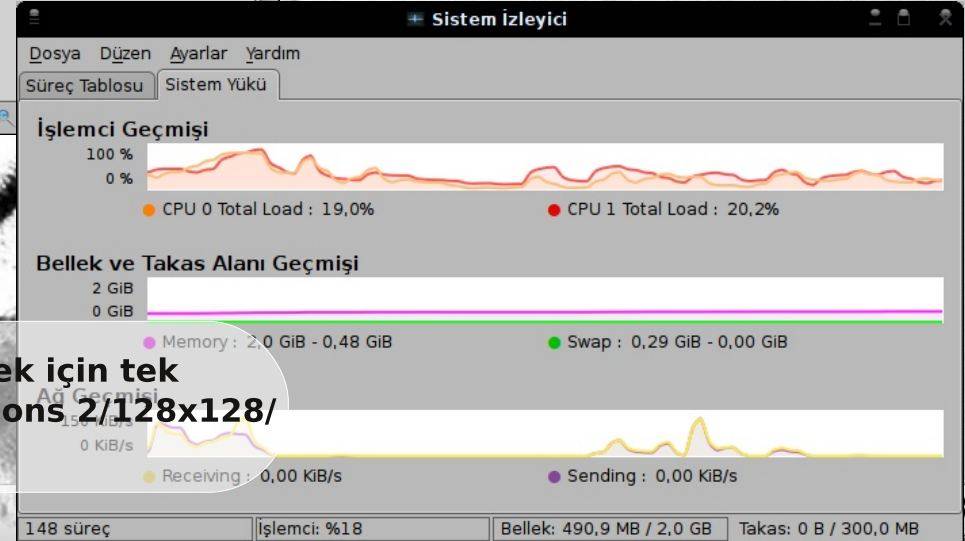
# Penguart: Yaşar Yeğin'in Masaüstü Görünümü

```
mono : bash
mono@mono-pardus ~ $ cat /etc/lsb-release
DISTRIB_ID="Pardus"
DISTRIB_RELEASE="2009.1"
DISTRIB_CODENAME="Anthropoides virgo"
mono@mono-pardus ~ $
```



- Pardus 2009.1 Telli Turna
- KDE 4.3.4 Masaüstü Yöneticisi
- Air KDE 4 Teması
- Bazı simgeleri değiştirilmiş olan Kicons2 Simge Seti
- CoolWallpapers.org adresinden indirilmiş masaüstü duvar kağıdı.

- Pardus Menü resmi değiştirilmiş. Bu resmi değiştirmek için tek yapmanız gereken `/home/kul.adı/.kde4/share/icons/Kicons 2/128x128/` altında ki `start-here-kde.png` dosyasını değiştirmek.





## 64Bit Pardus'un İlk Alfa Sürümü Hazır

Çanakkale Onsekiz Mart Üniversitesi'nden Necdet Yücel ve takım arkadaşları tarafından bitirme tezi olarak hazırlanan 64Bit Pardus'un kurulabilir ilk alfa sürümünün hazır olduğu bildirildi.

Pardus Kurumsal 2 sürümü temel alınarak geliştirilen bu sürümün depolarında 1800'ün üzerinde paket bulunuyor.

64Bit Pardus'u <http://www.pardus-linux.org/pardus-indir> adresinden indirebilirsiniz.



## Pardus Geliştiricileri DonanımHaber.Com' da Merak Edilenleri Yanıtladı

Pardus geliştiricilerinden 2009 sürüm yöneticisi Onur Küçük ve Kurumsal 2 sürüm yöneticisi Ozan Çağlayan, DonanımHaber.Com'un Teknoloji Gündemi adlı bölümünde Murat Çevik'in konduğu oldular ve Pardus hakkında merak edilenleri yanıtladılar.

Yaklaşık bir saat süren görüntüye <http://goo.gl/fb/bGjt> adresinden ulaşabilirsiniz.



## Mozilla Firefox 3.6 Nihayet Duyuruldu

Beş beta sürümü, iki aday sürüm ve bir çok ertelemeden sonra Mozilla geliştiricileri, Firefox tarayıcısının Namoroka kod adlı 3.6 sürümünü duyurdu. Mozilla'nın Gecko platformunun 1.9.2 versiyonunu esas alan bu sürüm, bir çok yeni özelliği beraberinde getiriyor.

Bu sürüm, yeni CSS, DOM ve HTML5 web teknolojilerinin - HTML5 File API gibi- desteğini, Firefox 3.5 sürümüne göre geliştirilmiş JavaScript performansını ve tam ekran video oynatma özelliğini içeriyor. Bu sürüm ayrıca Web Open Font Format özelliğini destekliyor.

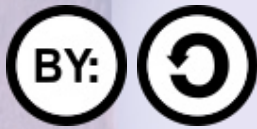
Kaynak: HaberTux.Com

## Bu Sayıda Emeđi Geenler

Aydın Bez  
Erdem Artan  
Gürhan Şükürođlu  
Hamit Giray Nart  
Kemal Karataş  
Melike İlteralp  
Muslu Yüksektepe  
Nihad Karslı  
Onur Tuna  
Uđur aylık  
Yaşar Yeđin

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

Altıntıların orijinal lisansları geçerlidir.



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

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

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

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

