

Pardus'un 64 bit Mimarisine Port Edilmesi

Necdet Yücel¹, Meltem Parmaksız², Mete Bilgin², Metin Akdere²

¹ Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği Bölümü, Çanakkale

² Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği Bölümü, 4. Sınıf Öğrencisi, Çanakkale

nyucel@comu.edu.tr, meltem@comu.edu.tr, metebilgin48@gmail.com, metinakdere@gmail.com

Özet: Bu çalışmada TÜBİTAK UEKAE tarafından geliştirilen GNU GPL lisanslı Linux tabanlı işletim sistemi projesi olan Pardus'un 64-bit işlemci mimarisinin özelliklerini kullanacak bir sürümünün hazırlanması süreci açıklanmaktadır. Ayrıca ortaya çıkan dağıtımın 32-bit işlemci mimarisini kullanan sürüm ile performans karşılaştırmaları yapılmış ve Pardus'un başka mimariler için port edilmesi için rehber olacak bilgiler sunulmuştur.

Anahtar Sözcükler: GNU/Linux, Pardus, 64-bit, x86-64.

1. Giriş

X86-64 AMD tarafından tasarlanan 64-bitlik bir işlemci mimarisidir[1]. X86 mimarisinin bir üst kümesi olduğundan bu mimariyi de destekler. Daha sonra INTEL firması tarafından da EM64T adıyla geliştirilmeye ve kullanılmaya başlanmıştır. INTEL'in geliştirdiği ve Itanium adıyla bilinen IA-64 ise bu iki mimariyle de uyumsuzdur ve onlar için yazılmış kodları çalıştırmaz.

2003 yılından bu yana TÜBİTAK/UEKAE tarafından geliştirilmekte olan Pardus işletim sistemi geliştirilmeye başlandığı dönemde en yaygın işlemci mimarisi olan x86 mimarisinde çalışacak şekilde hazırlanmıştır. Diğer GNU/Linux dağıtımlarının durumlarına bakıldığında[2] ise sadece belli mimarilerde çalışması için hazırlanmış iki dağıtım haricindeki tüm Linux dağıtımlarının x86 mimarisini destekledikleri ve yarıdan fazlasının da x86-64 mimarisine hazır oldukları görülmektedir. Sadece iki dağıtımın (debian ve gentoo) bütün işlemci mimarilerinde çalışmaya hazır oldukları görülmektedir.

X86-64 işlemci mimarisi temel olarak bir alt sınıfı olan x86 mimarisini desteklediğinden 32-bit için hazırlanmış bütün dağıtımlar prensipte 64-bit bilgisayarlar kurularak ve çalıştırılabilirler. Elbette bir dağıtımın bir işlemci mimarisine sahip bilgisayara

kurulabilir olması onun bütün özelliklerini kullanabileceği anlamına gelmemektedir.

Pardus'un 32-bitte çalışan sürümünün 64-bitte port edilmesi için yapılması gerekenler, dağıtım ilk hazırlandığı yıllarla kıyaslandığında oldukça azalmıştır. Aradan geçen yıllarda yazılımlar büyük oranda her iki platformda da çalışacak hale gelmiştir. X86-64 mimarisinin getirdiği yeniliklerin[3] yazılımlar tarafından kullanılabilmesi için gerekli yamaların hazırlanmaları diğer dağıtımların 64-bit'e geçiş sürecinde yaşananlarla büyük ölçüde halledilmiştir.

Her ne kadar yazılımlar x86-64 mimarisine hazır olsalar da bir Linux dağıtımı sadece GNU programlarından ibaret olmadığı için yapılması gereken başka işlemler de bulunmaktadır. Pardus herhangi bir dağıtımın temel alan bir Linux dağıtımı olmadığından kendine özgü bir paket yapısı, paket yöneticisi ve kendi geliştirdiği teknolojiler bulunmaktadır ve bu yazılımların da 64-bitte hazır hale getirilmesi gerekmektedir. Eğer Pardus sadece bir dağıtımın uyarlaması olsaydı onu x86-64 mimarisine taşımak için temel alındığı dağıtımda yapılan işlemleri yapmak yeterli olabilecekti kendi paketleme sistemi olan bir dağıtımda tüm paketlerin teker teker elden geçirilmesi gerekmektedir.

Bu işlemlere geçmeden önce Pardus'un mevcut sürümlerinden birinin seçilmesi ve

onun üzerinde çalışılmalıdır. Bu çalışma gerçekleştirilirken Pardus'un kurumsal kullanıcılara sunmayı hedeflediği Kurumsal 2[4] sürümü seçilmiş ve çalışmalar onun üzerinden yürütülmüştür.

2. Yöntem

Bir Linux dağıtımı çalıştığı mimariden başka bir mimariye taşınmak istendiğinde temel olarak iki farklı yol izlenebilir. Bunlar:

- i. 64-bit kararlı sürümü olan bir Linux dağıtımı kuluçka dağıtım olarak kullanılarak 64-bit Pardus paketlerinin hazırlanacağı bir kök dosya sistemi oluşturularak burada mevcut pisi paketlerinin 64-bit sürümleri hazırlanabilir. Bu yöntemin avantajı kök dosya sistemini hazırlarken kullanılacak araçların mevcut durumda 64-bit ikilik dosyalar üretebiliyor olmasıdır. Böyle olacağından çapraz derleme yapmaya gerek kalmayacaktır. Olumsuz olabilecek durum ise kullanılacak kuluçka dağıtım ile 64-bit için temel alınan Pardus Kurumsal 2'de bulunan programların sürümleri arasında çıkabilecek uyumsuzluklar olacaktır.
- ii. Kuluçka dağıtım olarak Pardus Kurumsal 2 kullanılarak onun üzerinde önce hem 32-bit hem de 64-bit kod derleyebilen programlar ve onların bağımlılıkları, ardından da bu araçları kullanarak 64-bit paketlerin hazırlanacağı kök dosya sistemi'nin hazırlanacağı geçici kök dosya sistemi oluşturulabilir.

Bu çalışmada kuluçka dağıtım ile Pardus arasında karşılaşılabilecek sorunlarla uğraşmamak için ikinci yolu tercih edilmiştir. Kök dosya sistemi bu yolla hazırlandığında onun üzerinde çalışabilmek için ya onu kullanarak sistemi açmak ya da o alana chroot yapmak gereklidir.

Bu aşamada karşılaşılan üçlü yol ayrımı şöyle oldu:

- i. Pek mümkün görünmese de 32-bit çalışan bir işletim sistemi üzerinde 64-bit çalışacak bir dosya sistemine chroot yapmak.
- ii. Kök dosya sistemini boot edebilir hale getirmek. Bu seçenek sadece 64-bit destekli bir çekirdek hazırlamakla sınırlı değil. Pardus'un kendi geliştirdiği teknolojilerin boot süreci ile ilgili olanlarının da bu aşamada 64-bit mimari için derlenmiş olmasını gerektiriyor.
- iii. 32-bit çalışan sistem üzerinde çapraz derleme araçları kullanılarak oluşturulan bu geçici kök dosya sistemini 64-bit çalışan bir başka Linux dağıtımına taşıyarak orada chroot yapmak.

Burada da çalışmayı en kısa yoldan hedefe götürebilecek yol üçüncüsü olarak görüldüğünden onu tercih ettik.

2.1 Çapraz Derleme Araçlarının Hazırlanması

Çapraz derleme; derleyicinin çalıştığı platformdan başka bir platform için kod derleyebilmesidir. Mikro denetleyiciler gibi işletim sistemini desteklemeyen platformlarda veya kod derlemenin daha az verimli olduğu platformlar için ikilik dosya üretiminde kullanılır.

Pardus'un 32-bit platformunda çalışan derleyicileri öntanımlı olarak sadece bu platform için ikilik dosyalar üretebilir. Bu programların 64-bit platformunda çalışacak kodlar derleyebilmeleri için aşağıdaki paketlerin bu parametrelerle yeniden kurulmaları gerekmektedir.

- Kernel-headers, File, Ncurses, GMP, Mpfr, Binutils, Glibc, Gcc

Bu çalışmada programların son sürümleri değil Pardus corporate2'nin kullandığı sürümleri kullanılmıştır.

2.2 Geçici Kök Dosya Sisteminin Hazırlanması

Çapraz derleme araçları kullanılarak aşağıdaki programların 64-bit olarak derlendiği geçici kök dosya sistemi hazırlanmıştır:

- Gmp, Mpfr, Zlib, Binutils, Gcc, Ncurses, Bash, Bison, Bzip2, Coreutils, Diffutils, Findutils, File, Flex, Gawk, Gettext, Grep, Gzip, M4, Make, Patch, Sed, Tar, Texinfo, Vim

2.3 Kök Dosya Sisteminin Hazırlanması

Geçici kök dosya sistemi hazırlandıktan sonra yapılacak işlem bu dosya sistemindeki araçları kullanarak bir kök dosya sistemi oluşturulmasıdır. Yukarıda da belirtildiği gibi bu işlem için seçtiğimiz yöntem Pardus Kurumsal 2 üzerinde hazırlanan bu geçici sistemi 64-bit kararlı sürümü olan bir Linux dağıtımına taşıyıp o dağıtım üzerinde bu alana chroot yapmak yönünde oldu. Bu dağıtımı debian olarak belirledik. Bu noktada seçilen dağıtım için tek önemli nokta çekirdeğinde aşağıdaki desteklerin belirtildiği gibi olmasıdır:

- CONFIG_INOTIFY_USER = y
- CONFIG_SIGNALFD = y
- CONFIG_SYSFS_DEPRECATED = n
- CONFIG_SYSFS_DEPRECATEDv2 = n
- CONFIG_UNIX98_PTYS = y
- CONFIG_DEVPTS_FS = y

Hazırlanan geçici sistemi debian üzerine taşıdıktan ve chroot işlemini gerçekleştirdikten sonra sırasıyla aşağıdaki test araçları kurulmuştur:

- Tcl, Expect, DejaGNU

Bu sistemin 64-bit'te sorunsuz çalışması için aşağıdaki programların kurulumları yapılmıştır:

- Perl, Linux-headers, Man pages, Glibc, GMP, MPFR, Zlib, Binutils, Gcc, Sed, Ncurses, Util-linux-ng, E2fsprogs, Coreutils, M4, Bison, Procps, Libtool, Flex, IPRoute2, Readline, Autoconf, Automake, Bash, Bzip2, Diffutils, File, Findutils, Gawk, Gettext, Grep, Groff, Gzip, IPutils, Kbd, Less, Make, Man, Module-Init-Tools, Patch, Psmisc, Shadow, Rsyslog, Sysvinit, Tar, Texinfo, Udev, Vim, XZ-Utils

Bu aşamaya gelindiğinde henüz Pardus'a özel bir şey (programların Pardus'ta kullanılan sürümlerinin seçilmesi sayılmazsa) yapmamış olmakla birlikte 64-bit çalışan bir sistem hazırlanmış bulunmaktadır. Buradan sonra daha önce kurulmuş olanlar dahil, programların Pisi paketlerinde kullanılan yamaları ve parametreleri kullanılarak tekrar derlenmesi gereklidir. Bu aşama birbirine bağlı paketlerin istenilen halleri elde edilene dek bir kaç kez tekrar edilmelidir.

- Kernel-headers, Glibc, GMP, Mpfr, Zlib, Binutils, Gcc, Make

Kök dosya sistemi hazırlığı tamamlandığında Pardus'un paket yöneticisi PiSi'nin ve onun bağımlılıklarının kurulumuna geçilebilir.

2.4 Paket Yöneticisinin Kurulması

Paket yöneticisinin kurulması için öncelikle paket yöneticisinin bağımlılık ağacı çıkarılması gereklidir. Ardından diğer dağıtımlar incelenerek bu paketlerin 64bit'e geçirilmesi için, 64bit'e özel bir yama ya da derleme parametresi olup olmadığı belirlenmeli ve bu paketlere Pardus'un kullandığı yamalar ile beraber, eğer varsa, 64bit'e özel yamalar uygulanmalıdır. Bundan sonra da bu paketler 64bit derleme

parametreleri ile (Pardus derleme parametrelerini de göz önüne alarak) sisteme kurulmalıdır.

Paketlerin kurulumuna kök dosya sistemindeki programlarından başka bağımlılığı olmayan paketlerden başlanarak, devamında bu paketlere bağımlı olan üst bileşenlere (COMAR, PISI, MUDUR,vb.) ulaşmak hedeflenmelidir. Bu süreç bağımlılık ağacı[5] takip edilerek sürdürülmüştür. Pardus ekibi tarafından geliştirilen teknolojilerin 64-bit'te ortaya çıkan sorunları programların geliştiricileriyle haberleşilerek giderilmiştir. Bu ya programların yeni sürümlerinin çıkması ya da gerekli yamanın hazırlanması şeklinde olmuştur.

PiSi'nin sisteme kurulmasıyla birlikte bir aşama daha geçilmiş olur. Bundan sonraki hedef yine aynı bağımlılık ağacını takip ederek uçlardan merkeze doğru gelerek PiSi'nin ve onun bağımlılıklarının paketlerinin hazırlanmasıdır.

2.5 Paket Yöneticisinin Paketlenmesi

PiSi'nin sisteme kurulmasıyla birlikte bir aşama daha geçilmiş olur. Bundan sonraki hedef yine aynı bağımlılık ağacını takip ederek uçlardan merkeze doğru gelerek PiSi'nin ve onun bağımlılıklarının paketlerinin hazırlanmasıdır.

Yeni sisteme PiSi'nin problemsiz kurulması için gereken bütün programlar zaten PiSi paketlerindeki yama ve parametrelerle kurulduğundan sadece Pardus'un paketleme süreci ile ilgili geliştirdiği teknolojilerin bu yeni mimaride gerektiği gibi çalışıp çalışmadığının kontrolünün yapılması yeterli olmuştur. Gerektiği gibi çalışmayan programların geliştiricileriyle irtibata geçilerek ihtiyaç duyulan düzenlemelerin yapılması sağlandıktan sonra PiSi ve tüm bağımlılıkları paketlenmiştir.

Bu aşamadan sonra çözülmesi gereken iki bağımlılık ağacı kalmış bulunmaktadır. Bunlar system.base ve system.devel

bileşenleridir. Bir paketin bu bileşenlerde yer alan başka bir pakete kurulum ve/veya çalışma bağımlılığı mevcut olması durumunda bu bağımlılık pspec.xml dosyasına yazılmadığından bu bileşenlerdeki paketlerin öncelikle hazırlanarak sisteme kurulması gereklidir. Bu iki bileşenin de tamamen paketlenmesinin ardından geliştiricilerin en çok kullandıkları iki araç olan vi ve subversion da paketlendiğinde artık 64bit PiSi paketi hazırlayacak geliştiricilerin en temel ihtiyaçları karşılanmış olacaktır.

2.6 Geliştiricilere Sunulacak Sürümün Hazırlanması

Bu çalışma ile PiSi paketlerini 32-bitte hazırlayan geliştiricilere 64-bitte de çalışabilecekleri bir sistem sunarak kendi sorumlu oldukları paketler için gerekenleri bundan sonra kendilerinin yapabilmesine olanak sağlamak hedeflenmiştir. Bunun için çalışmanın bu aşamasında sadece geliştiricilerin kullanabilecekleri bir sürüm hazırlanmıştır.

System.base, system.devel ve boot bileşenlerinin tüm içerikleri paketlendiğinde işletim sisteminin açılış (boot) için eksikliği kalmamış olur. Hazırlanan bu paketler PiSi paket yöneticisinin -D parametresi kullanılarak uygun bir dizine kurulur. Bu işlem tüm paketleri bir dizine toplayıp

```
pisi it *.pisi -D /rootfs
```

komutunu çalıştırarak yapılabilir. Bu dizini arşivleyip geliştiricilere vermeden önce içine eklenmesi gereken dosyalar bulunuyor:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/fstab
- /etc/mtab
- /etc/resolv.conf
- /root

Bu dosyalar normalde kurulum sırasında oluşturulan ama geliştiricilerin sistemlerini boot edebilmesi ve oturum açması için gerekli dosyalardır. Bunlardan fstab ve mtab dosyalarını her geliştirici kendi sistemine uygun hale getirmelidir. Diğerleri uygun şekilde oluşturulmalı ve root kullanıcısına atanan parola geliştiricilere bildirilmelidir. /etc/pardus-release dosyasının da hazırlanan portu belirtecek şekilde düzenlenmiştir.

Gerekli düzenlemelerin yapılmasının ardından üzerinde çalışılan dizin arşivlenip sıkıştırılarak geliştiricilere dağıtmaya hazır hale gelir [6].

2.7 Paketlerin X86_64 Mimarisine Taşınmasında Yapılması Gerekenler

Corporate 2 Sürümünün PiSi paketlerinin derlenmesinde kullanılan bayraklar sistemdeki /etc/pisi/pisi.conf dosyasından merkezi bir şekilde alındığı için depodaki programların büyük çoğunluğunun yeniden paketlenmesinde bu dosyada yapılanlar haricinde bir değişikliğe ihtiyaç duyulmamıştır. Bu paketlerin sadece uygun kök dosya sisteminde yeniden oluşturulmaları yeterli olmuştur. Aşağıda x86_64 mimarisine taşınma sırasında yapılan değişiklik çeşitleri verilmektedir. Bu değişikliklerin nasıl yapılacağı ile ilgili temel referans çok yaygın kullanılan ve geniş geliştirici potansiyeli bulunan debian ve gentoo dağıtımları olmuştur.

2.7.1 x86_64 için farklı parametrelerle derlenen paketler

Bazı programlar derleme veya yapılandırma aşamalarında mimariye özgü parametreler kullanılarak paketlendiklerinden bu programların paketlerinde uygun değişikliklerin yapılması gerekmiştir.

2.7.2 x86_64 için ilave yamalar gerektiren paketler

Bazı programların 64-bitte derlenmeleri kaynak kodlarına ilave yamalar yapılmasını

gerektirdiğinden bu yamalar ya daha başka dağıtımlardan alınarak uygulanmış ya da geliştiriciler tarafından yazılmıştır.

2.7.3 x86_64 için farklı kaynak dosyasına ihtiyaç duyan paketler

Dağıtım dahil edilmiş kapalı kaynak kodlu programların 64-bit için uygun ikilik dosyaları kullanmalarını sağlamak için kaynak dosyaların değiştirilmesi gerekmiştir.

2.7.4 x86_64 için dosyaları farklı dizinlerde bulunduran paketler

Bazı programlar kitaplık dosyalarını /lib dizini yerine /lib64 dizinine veya /usr/lib dizini yerine /usr/lib64 dizinine yerleştirmek istediğinden /lib64 ve /usr/lib64 dizinleri sırasıyla /lib ve /usr/lib dizinlerine sembolik linklenmiş ve paketleri için gerekli değişiklikler yapılmıştır.

2.7.5 x86_64 mimarisinde derlenemeyen fakat x86'da statik derlenerek kullanılabilen paketler

Bu kategoride sayılabilecek tek paket grub[7]'tur. Grub sadece 32-bitte derlenebilen bir yazılımdır. İşletim sistemi önyükleyicisi olduğundan hangi mimaride derlendiğinin de bir önemi bulunmadığından diğer tüm 64-bit portu bulunan Linux dağıtımlarının yaptığı gibi 32-bitte statik derlenerek 64-bit dağıtımına dahil edilmiştir.

2.7.6 x86_64 mimarisinde karşılığı bulunmayan paketler

Bazı paketlerin ya 64-bitte ihtiyaç duyulmadığından (kernel/pae gibi) ya da yazılımın 64-bitte derlenmesi mümkün olmadığından (psyco gibi) x86-64 dağıtımdan çıkartılması gerekmiştir. Virtualbox paketinin de açık kaynak kodlu olan sürümü sadece 32-bitte derlenebilmekte ama 64-bit için ikilik dosya bulundurduğundan yine dağıtımdan çıkartılmasına karar verilmiştir.

3. 64-bit Performans Test Sonuçları

Bu çalışmanın sonunda hedeflen x86-64 mimarisinin getirdiği yeniliklerin kullanılması ve 32-bite göre bir avantaj elde edilmesidir. Bu ya 32-bitte yapılamayan bir şeyin yapılması ya da aynı işin daha hızlı yapılabilir olmasını sağlamak şeklinde olmalıdır. Yapılan karşılaştırmada daha önce yapılmış olan 32-bit 64-bit karşılaştırmalarında[8] alınan sonuçlara benzer sonuçlar elde edilmiştir. Test için kullandığımız donanım şöyledir:

- 4 X AMD Opteron(tm) Processor (2.3GHz)
- 4GB RAM

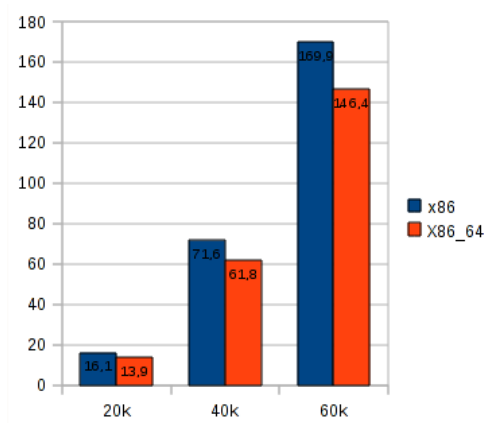
Test için aşağıdaki yazılımlar kullanılmıştır:

- bc-1.06.95-5
- gnupg-2.0.11-26
- lame-3.98.2-11
- bzip2-1.0.5-10

Dikey eksenler saniye cinsinden zamanı göstermektedir.

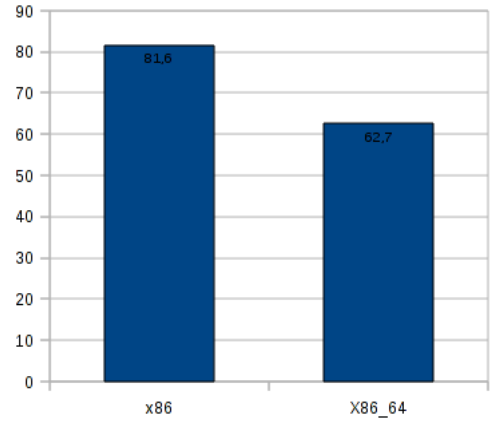
3.1 Faktöryel Hesabı Testi

Bu testte 20000, 40000 ve 60000 faktöryeller hesaplandı. Sonuçta yaklaşık %14 bir kazanç var.



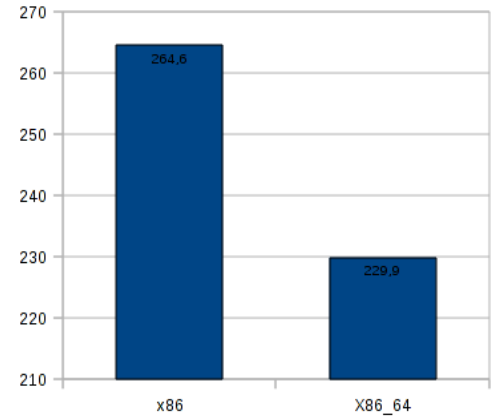
3.2 Şifreleme Testi

Bu test Pardus-2009.iso (687MB) dosyası şifrelenerek yapıldı. En büyük kazancı olduğu test bu oldu; %24 kazanç var.



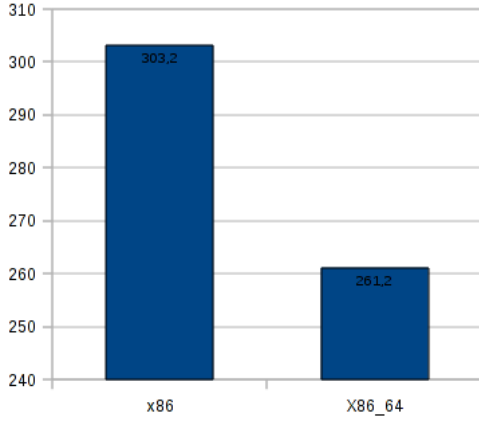
3.3 Ses Dosyası Dönüşümü Testi

Bu test wav dosyasını (647MB) mp3 dosyasına çevrilerek yapıldı, %14'lük bir kazanç elde edildi.



3.4 Sıkıştırma Testi

Sıkıştırma testinde bzip2 programı kullanılarak 687MB'lık bir dosyanın sıkıştırılması yapıldı. Yine yaklaşık %14 bir kazanç olduğu gözlemlendi.



4. Sonular ve Tartuřma

Pardus lkemizde en geniř topluluęu bir araya getirebilmiř zgr yazılımdır. TBTAK/UEKAE tarafından alıřtırılan geliřtiricilerinin yanı sıra projeye dıřarıdan katkı veren ok sayıda katkıcı/geliřtirici bulunmaktadır. Bu alıřmayı[9] gerekleřtiren grup olarak bu katkılara ek bir boyut getirerek daęıtımın x86-64 mimarisine tařınmasını gerekleřtirdik. Bunun hem geliřtiricilerde bir bilgi birikimi oluřturması hem de Pardus'un daha geniř kullanıcı kitlesine ulařması aısından faydalı olacaęı kanaatini tařıyoruz.

Yapılan testler de 32-bit ile 64-bit daęıtımlar arasında performans ynnden bir uurum olmayacaęını ama kayda deęer bir fark olacaęını gstermiřtir.

5. Kaynaklar

[1] <http://tr.wikipedia.org/wiki/X86-64>.

[2] http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions#Architecture_support

[3] http://www.amd.com/us-en/Processors/DevelopWithAMD/0,,30_2252_875_7044,00.html

[4] <http://www.pardus.org.tr/kurumsal/>

[5] <http://members.comu.edu.tr/nyucel/Pisi.png>

[6] <http://members.comu.edu.tr/nyucel/pardus-corporate2-rootfs-0.42.tar.bz2>

[7] <http://www.gnu.org/software/grub/>

[8] <http://bingouv.blogspot.com/2008/08/desktop-linux-performance-comparison32.html>

[9] http://svn.pardus.org.tr/pardus/corporate2/dev-el-x86_64/