

Windows Ağında Linux Terminalleri

Yazan:
Nilgün Belma Bugüner

<nilgun (at) belgeler.gen.tr>

7 Temmuz 2004

Özet

Bu belge asıl olarak, güncelliğini yitirmiş bilgisayarlarla kurulmuş ve ancak W98'lerin çalışabildiği bir ağdaki bilgisayarların, ağa sunucu olarak hızlı bir makina eklenmesi ve buna Linux kurulması ile nasıl hızlı bilgisayarlar dönüşürülebileceğinden bahsedilmiştir.

Geçen aylarda, Boğaziçi Üniversitesi Mütercim–Tercümanlık Bölümünden Linux uygulamalarına ve belgelerine türkçe çeviri desteği verebileceklerini belirten bir eposta geldi. GNU–TR projesi kapsamında bazı çalışmalar yapıyorlar. Bu arada, bilgisayar laboratuvarlarında Linux kullanabilmek istediklerini de belirttiler, ama Windows'larından da vazgeçemiyorlardı. Bir çözüm, bir öneri istendi. Deniz hanım, İstanbul'da olduğundan onlarla temasa geçti ve LTSP ve PXES gibi hazır çözümler olduğundan bahsetti ayrıca bu isteğin gerçekleştirilebilmesi için yardım isteğini de LKD dernek listesine aktardı.

Böyle bir isteğin varlığı, bir fikrin gelişmesine sebep oldu. Madem windows'larından vazgeçemiyorlardı; ki ben de Linux'u öğrenmeye başladığım zamanlar uzun süre vazgeçememiştim; o halde windows üzerinden Linux'un çalıştırılmasını sağlayacak bir çözüm geliştirebilirdim. Bu belge bu fikirden yola çıkılarak yapılan çalışmayı belgelendirir.

Konu Başlıkları

1. Giriş	3
2. Sunucu Makinaya Linux kurulması	3
2.1. DHCPD Yapılandırması	5
2.2. Samba ve Ağ Yapılandırması	6
2.3. Windows'ların Samba sunucusuna bağlanması	8
2.4. XFS yapılandırması	9
2.5. GDM yapılandırması	9
3. X Terminal Sisteminin Yapılması	14
3.1. initrd.img Dosyasının Oluşturulması	14
3.2. Çekirdeğin derlenmesi	17
3.3. Tek Başına (basit) Linux	18
3.4. Ağ Bağlı Linux Terminali	24
3.5. X Terminalinin Oluşturulması	27
4. Sonuç	38
A. Çekirdek Yapılandırma Dosyası (.config)	39
B. Busybox Yapılandırma Dosyası (.config)	51

Geçmiş

1.0
İlk sürüm

7 Temmuz 2004

NBB

Yasal Açıklamalar

Bu belgenin, *Windows Ağında Linux Terminalleri* 1.0 sürümünün **tefif hakkı © 2004 Nilgün Belma Bugüner**'e aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Genel Kamu Lisansının 2. ya da daha sonraki sürümünün koşullarına bağılı kalarak kopyalayabilir, dağıtabilir ve/veya değıştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

BU BELGE “ÜCRETSİZ” OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ “OLDUĞU GİBİ”, AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılması ona onay verildiği anlamında görülmemelidir.

1. Giriş

Önce projenin oturduğu zemini açıklamalıyım. Üzerlerinde Windows 98'ler kurulu zayıf PC'lerden oluşmuş bir ağ var. Bu ağı yönetenler, Linux'u da kullanmak istiyor ama Windows'ların çalışabildiği o zayıf makinalarda Linux'u verimli kullanamayacaklarını düşünüyorlar. Haklıdır, günümüz Linux sistemleri en az 256 MB bellek en az 1GHz'lik işlemci ile 3–5 GB disk alanı istiyor. Bu pahalı bir çözüm. Ama o yavaş PC'lerde günümüz yazılımlarını günümüz bilgisayarlarının hızında çalıştırmak da en ekonomik yoldan sadece Linux ile mümkün :–) Ne tezat değil mi?

Bu zemine oturan bu projenin, gerçekleşebilmesi için ağın sunucusunu kaçınılmaz olarak Linux yapmak gerekiyor, hem de oldukça hızlı bir işlemcisi (bir P4), en az 512MB belleği olan bir makina lazım. Linux sunucu Samba ile hem ağdaki Windows'ları yönetmek için hem de onlarda Linux çalıştırıldığında onlara X terminal sunucusu olmak için gerekli. Sunucu dışında diğer makinalara Linux kurulumu yapmayacağız. Her makinanın dosya sistemine 3 küçük dosya (toplamı 6 MB civarında) konacak ve bir makina üzerinde Linux çalıştırılmak istendiğinde Windows, Linux'u çalıştıracak. Olur mu demeyin. Nasıl olduğunu bu belgede anlatıyorum. Makinalarınızda – şu ucuz 10/100 Mbit ağ kartlarından varsa ve ağınız 100Mbit hızda çalışıyorsa, Linux makinaların sürat rekoru kıracağından endişeniz olmasın. 2.5 MHz AMD işlemcili, 1GB bellekli bir makinanın sunucu olarak kullanıldığı, 100–200 MHz işlemci ve 24–32 MB belleği olan 13 makinalı bir X terminalleri ağındaki makinalarda 262 sayfalık bir Excel dosyasının 1 saniyede açıldığından Linux listelerinde bahsedilmişti.

Bu projenin amacı sadece bu işlemi belgelendirmektir. Bir dağıtım yapılması amaçlanmamış, Linux'u kuracak ve üzerinde basit bir kaç işlem yapabilecek kadar Linux bilen biri tarafından uygulanabilecek bir işlemin belgelendirilmesi amaçlanmıştır. Bu belgeyi tamamen okuduktan sonra, başa dönüp işlemleri adım adım uygulamanız yeterli olacaktır. Yetmezse bana yazın, bilmeyenin ne bilmediğini bilmek gibi bir yeteneğim yok :–) . Belgeyi daha anlaşılır yapmak isterim, bunun için de sizden geribesleme almaya ihtiyacım var.

İşe başlarken önce sunucuyu yapılandıracağız. Sunucunun yapılandırılması hem ağdaki makinalardan donanım bilgilerinin toplanması hem de üzerlerinde çalışacak dosyaların hazırlanması için gerekli. Bu dosyalardan çekirdek (**vmlinux**) ve ramdisk eşlemini (**initrd.img**) kendimiz hazırlayacağız. **loadlin.exe** ise zaten Linux'u kurduğunuz CD'lerin ilkinde vardır. CD'den kurulan Linux'lar **loadlin.exe** ile kurulumu başlatılır. Yani yapacağımız işlem Linux kurulumu başlatan işlemin özel bir hali olacak. İşlemin sadece başlatma aşaması olan bellekte çalışan kısmı üzerinde çalışıp diğer kurulum adımlarına hiç girmeyeceğiz. Yani bizim Linux'umuz belleğe yüklenecek, hiç sabit diskle ilgilenmeyecek, hep bellekte çalışacak. Windows'un bu süreç içindeki rolü, sadece işlemi başlatmak. Bir kere Linux çalışmaya başladı mı, Linux tüm sistem kaynaklarını kullanmaya başlıyor, ama sabit disk, disket sürücüsü, CDROM devre dışı; onlara ihtiyacımız yok. Ancak, Linux ile işiniz bittiğinde Windows'a dönmek için makinaı tekrar başlatmaktan başka çare yok. Biz makina açıp kapatmaya alışkın değiliz ama Windows kullanıyorsanız bu kaçınılmaz. Ancak, Linux'u başlatmaya hazır olduğunuzda Linux'u başlatmak için Windows'u kapatmanız gerekmeyecek. Onu Linux sizin yerinize yapar.

Hiç böyle bir işleme ihtiyacınız olmasa bile bu belgede yazılanları bir kere uygulayın, bu kursu kaçırmayın derim.

2. Sunucu Makinaya Linux kurulması

Burada sabit diskimde hem Windows hem Linux olsun diyorsanız, bu konuya dahil değil. İnternete de bağlanacak bir makinada Linux çalışmalı. Virüslerle uğraşmak istemiyorsanız buna dikkat edin. Bu makinada sadece Linux olduğunu varsayıyorum. Sunucunuza kuracağınız dağıtım için bir kısıtlama yok. Ama ben Fedora Core kurduğunuzu varsayacağım ve ona göre anlatacağım. Dağıtımların dosya sistemi üzerindeki farklarını biliyorsanız herhangi bir dağıtımı kurabilirsiniz. Ek olarak elinizin altında bir Knoppix CD'si olmalı. Bilgi toplamak için ona ihtiyacımız olacak.

Linux'u kurarken kurulum size Linux kuracağınız disk bölümünü soracaktır. Kurulum bir disk yapılandırma aracı içerir. Bu disk yapılandırma aracını kullanarak kök disk bölümünü ve takas alanı olarak kullanılacak bölümü kendiniz ayarlayacaksınız. Bunu kurulum kendiliğinden yapabilir ama ona bırakmayın, çünkü bizim üzerinde

ramdisk'i yapılandıracağımız küçük bir disk bölümüne ihtiyacımız var. Kök disk bölümü için en az 10 GB yer ayırın, takas alanı olarak 1GB ayırın, özel disk bölümü için ise 100kB bile fazla gelir ama böyle yapmayın, ona da diskinizin boyutuna bağlı olarak büyükçe bir yer ayırın, başka amaçlar için de kullanabilirsiniz. Örneğin indirdiğiniz bazı paketleri ve dosyaları burada tutarsınız. Kök disk bölümü ve takas alanı kurulum sırasında yapılandırılacak. Ayırdığımız ek bölüme kurulum dokunmayacaktır, onu zamanı gelince biz yapılandıracağız. Sisteme kurulacak paketlerin seçimine sıra geldiğinde tam kurulumu seçin, çünkü ağınızda çalışan kullanıcıların hemen herşeye ihtiyacı olacaktır. Onları seçimlerinde özgür bırakmak için tam kurulum yapmalısınız. Kimi GNOME'u sever, kimi KDE'yi, kim çeviri yapar, kimi yazılım geliştirir.

Kurulum sırasında dil olarak türkçeyi seçin, Fedora ve bir çok dağıtım artık tamamen türkçe kurulumla sahip, kurulumdan sonra çoğu uygulama sizinle türkçe iletişim kuracak. Çeviri ekiplerimiz iyi çalışıyor, haklarını vermek ve türkçe kurulumu seçmek sonra da bir çeviri ekibine katılıp destek olmak lazım.

Ağ yapılandırmasında makinaların olması gerektiği gibi bir isim verin, yani `isim.alan.uzanti` biçiminde bir isim seçin. Ben belge içinde sunucu isminin `nilgun.buguner.home` olduğunu varsayacağım. Siz de bu ismi kendi verdiğiniz makina ismi ile değiştireceksiniz. Yine kurulum sırasında ağ yapılandırmasında makinaların ağ kartına sabit bir IP adresi atayın. Bu numarayı yerel ağlara ayrılmış olan ve internette kullanılmayan IP ailelerinden seçmelisiniz. Hatırlaması kolay olduğu için ben hep `10.0.0.0` ailesini tercih ederim. Ben sunucuyu makinasının iç ağa bakan ağ kartının IP adresinin `10.10.10.1`, ağ maskesinin (netmask) `255.255.255.0`, ağ geçidi adresinin (gateway address) `10.10.10.254` olduğunu ve `10.10.10.0` ağına (network address) hizmet verdiğini varsayacağım. Yani ağınızdaki makinalara `10.10.10.2` ile `10.10.10.253` arasındaki herhangi bir adresi atayabileceksiniz. Makinalarınızda dış ağa bağlanmak için başka bir ağ kartı varsa ona atanacak isim ve adres konumuz dışında. Kurulum bitip sistemi başlattıktan sonra sistem üzerinde bazı yapılandırmalarla uğraşacağız.



Uyarı

Burada önemli gördüğüm bir noktayı vurgulamadan geçmek istemiyorum. Makinaların iç işlemler için kullandığı IP adresi `127.0.0.1`'dir ve bu numara ile ilişkilendirilen isim `localhost.localdomain`'dir. Bu standart ismi değiştirmiyoruz. Ya da bu adresle ilişkilendirilecek bir isim ataması yapmıyoruz. Yaparsanız saçma sapan hatalarla ceremesini siz çekersiniz. Benden söylemesi. Hele bu numaraya isim nasıl atanır bilmiyorsanız, bunu bana hiç sormayın. Bir ağ sunucusunun ya da istemcisinin bu isme ihtiyacı olmaz.

Kurulumu bitirdiğinize göre sistemimizi yapılandırmaya hazırız demektir. Önce ağdaki makinalara IP adreslerini dağıtacak DHCP sunucusunu yani `dhcpd`'yi yapılandıracağız. Bunun ardından Samba'yı yapılandıracağız. Sonrada ağdaki windows'ların birinde Samba ile ilgili ayarlarını yapacağız. Bu makinaları sunucu makinaların yanına alıp istemcilerde çalışacak linux'u yapmaya başlayacağız. Bu makina üzerinde edindiğiniz deneyimle diğerleri için farklı bir yapılandırma gerekiyorsa onları artık "makinaları sunucunun yanına almadan" yapacaksınız.

Eğer sistem GNOME oturumuyla açılıyorsa (genelde böyledir), `root` dışında tanımladığınız diğer kullanıcı ismiyle oturumu açın. GNOME oturumu açıldıktan sonra bir konsol açın ve `su` komutunu verin. Parola isteyecektir. `root` kullanıcısı için tanımladığınız parolayı verin. Artık root yetkilerine sahipsiniz. Yani sistem üzerinde her türlü işlemi yapmaya yetkilisiniz, çok dikkatli olun, bir dosya silerseniz kurtarma şansı yok, gitti-gider. `mc` komutu ile Midnight Commander'ı çalıştırın. Bir Norton Commander klonudur. Acemilik çekmeyeceksiniz. `cd /etc` komutu ile `etc` dizine geçin ve `hosts.allow` dosyasını bulun. `F4` ile dosyayı düzenleme kipinde açın. İçini aşağıdaki dosyaya benzer şekilde düzenleyin:

```
# Bu dosyada yerel ag hizmetlerinden hangi makinalarin
# faydalanabilecegi belirtilir.
#
ALL: .buguner.home
ALL: 127
```

`.buguner.home` yerine makinalarınıza verdiğiniz ismin `.alan.uzanti` kısmını yazacaksınız. Başındaki noktaya dikkat! Gerekli değişikliği yaptıktan sonra **F2** ile kaydedin ve çıkın (**ESC ESC**). Artık dosya düzenlenecek dediğimde ne yapılacağını biliyorsunuz: `mc` çalıştırılacak, dizine geçilecek, kursor dosyanın üzerine getirilecek, **F4** ile dosya açılacak, değişiklikler yapılacak, **F2** ile kaydedilecek ve **ESC ESC** ile düzenleme kipinden çıkılacak.

2.1. DHCPD Yapılandırması

DHCP sunuculuğu yapan uygulamanın ismi **dhcpd** ve **dhcpd**'nin yapılandırma dosyası `/etc/dhcpd.conf`'dur. Alışlagelen, bir DHCP sunucunun bir DNS sunucusu ile birlikte çalışmasıdır. DNS sunucusu her Linux dağıtımında bulunur. Ama biz kullanmayacağız. Küçük Windows ağlarında istemci Windows'lar sunucularına kendi isimlerini kendileri gönderirler. Kuracağımız X terminali ağında ise Linux'ların sadece sunucuya bağlanana kadar ve sadece IP adresine ihtiyaçları olur, bağlandıktan sonra ise bu adresin önemi kalmaz, her türlü işlem için sunucu makinanın IP adresini ve ismini kullanırlar. Yani bir DNS sunucusuna asla ihtiyacımız yok. İsim çözümlemesi de gerekmiyor, yani `hosts` dosyası filan düzenlemeyeceğiz. Samba kendi ile ilgili kısmı kendisi zaten hallediyor olacak.

Bir önceki bölümde makinadan yararlanabilecek harici makinaların hangi ağdan gelebileceklerini belirlemiştik. Bunun dışında bir istek daha DHCP sunucuna gelmeden durdurulacağından, DHCP sunucuna gelen her isteğe sen kimsin nesin diye sormadan istediği IP adresini verebiliriz. `/etc/dhcpd.conf` dosyasını bu politika doğrultusunda düzenliyoruz. Eğer `/etc` dizininde bu dosya yoksa **touch dhcpd.conf** komutu ile dosyayı oluşturun. Eğer bir önceki bölümde anlatılan `hosts.allow` dosyasını düzenlemeden buraya geçtiyseniz bir önceki sayfaya geçip bu işlemi yapın. Bu, sistem güvenliği ile ilgili temel bir yapılandırma değildir.

```
allow booting;
allow bootp;
ddns-update-style none;

option domain-name "buguner.home";
max-lease-time 720;
default-lease-time 360;

subnet 10.10.10.0 netmask 255.255.255.0 {
    server-name "nilgun.buguner.home";
    pool {
        range 10.10.10.2 10.10.10.253;
        allow all clients;
    }
}
```

Şimdi **dhcpd**'yi başlatalım ve bakalım çalışıyor mu?

```
# /sbin/chkconfig dhcpd on
# /sbin/service dhcpd start
dhcpd başlatılıyor: [ TAMAM ]
# netstat -Nlp | grep dhcp
udp        0      0 0.0.0.0:bootps      0.0.0.0:*
    1181/dhcpd
raw        0      0 0.0.0.0:icmp        0.0.0.0:*
    1181/dhcpd
```

Çalışıyormuş :-). İlk komut, **dhcpd**'nin makinanın her açılışında çalışmasını sağlayacak düzenlemeyi yapar, ikinci komut ise hizmeti hemen başlatır. Makinamızı bu işlem için açıp kapatmak zorunda kalmadık. Eğer **dhcpd**'yi başlattığınızda yukarıdakinden farklı olarak bazı iletiler alıyorsanız `dhcpd.conf` dosyanızda

birşeyler yanlış olmuş demektir. Bu takdirde gerekli düzeltmeyi yapıp, birinci komut hariç işlemi tekrarlayın. `1181/dhcpd` sizin çıktınızda farklı olabilir. `1181`, `dhcpd`'nin süreç kimlik numarasıdır (PID).

2.2. Samba ve Ağ Yapılandırması

Samba'yı da yapılandırdıktan sonra sunucu makinamız, windows ağına hizmet vermeye hazır olacak. Diğer işlemler için acele etmenize gerek kalmayacak.

Samba yapılandırmasını anlatan çok belge var ama burada yapacağımız yapılandırmayı anlatan bir belge yok. Onun için bildiklerinizi bir kenara bırakıp buradaki adımları dikkatle yapın.

Önce `/etc/samba` dizindeki `smb.conf` dosyasını yapılandıracağız. Aşağıdaki satırları yeni açacağınız `smb.conf` dosyasına kopyalayın ve kaydedip çıkın.

```
[global]
    workgroup = LAB
    security = SHARE
    null passwords = Yes
    guest account = depocu
    passdb backend = guest
    domain logons = Yes
    mangle case = Yes

[netlogon]
    comment = Sistemle ilgili dosyalar
    path = /export/netlogon
    write list = root
    guest ok = Yes

[depo]
    comment = Dosya Deposu
    path = /export/depo
    write list = guest
    force user = depocu
    force group = users
    read only = No
    guest ok = Yes
```

Bu yapılandırmada Samba daha çok bir dosya sunucusu olarak çalışıyor. Kullanıcıların kimlikleri ile ilgilenmiyor. Samba'ya kullanıcılar değil Windows sisteminin kendisi oturum açıyor ve tüm paylaşımlar için yapay bir kullanıcı kimliği belirtiyor. Yani `[depo]` paylaşımına tüm Windows kullanıcıları dosya yazıp silebiliyor.

Ama bu yapılandırma dosyasında bir çelişki var. `security = SHARE` satırı ile `domain logons = Yes` satırı Samba'ya göre teorik olarak bir arada olamazlar. Yani bir paylaşım sunucusu aynı zamanda bir kullanıcının bir makina kimliği bile olsa oturum isteği yaptığı bir sunucu ile birlikte olamaz. `testparm -s` komutunu verdiğinizde Samba bunu size belirtecektir. Herşeye rağmen bizim bu çelişkiye ihtiyacımız var. Samba sunucusunu çalıştırmadan önce bu dosyada yazılı yapılandırmaların gereğini yapmamız lazım.

`[netlogon]` paylaşımını Linux için gereken dosyaları bulunduracağımız dizin olarak tasarladım. Windows kullanıcıları bu paylaşımındaki dosyaları değiştiremeyecekler ve silemeyecekler. Bu paylaşım ile erişilecek dizinleri oluşturalım:

```
# cd /
# mkdir export
# cd export
# mkdir netlogon
```

```
force user = depocu
force group = users
```

Satırlarına dikkatinizi çekerim. `depocu` kullanıcısını oluşturmamız lazım; ayrıca, bu kullanıcının parolasız bir kullanıcı olması lazım:

```
# /usr/sbin/useradd -g users -d /export/depo depocu
# passwd -l depocu
Locking password for user depocu.
passwd: Success
```

Windows'lara "domain logon" yaptıracağımız için, Samba'ya makina isimlerini de tanıtmamız lazım. Ben işlemi bir makina için yapacağım, siz ağınızdaki bütün makinalar için yapacaksınız.

```
# /usr/sbin/useradd -g users -d /dev/null -c "ws belma" -s /bin/false belma$
# passwd -l belma$
Locking password for user belma$.
passwd: Success
# smbpasswd -a -m belma
Added user belma$.
```

Makina hesapları makina isimlerinin sonuna `$` işareti eklenerek oluşturuluyor. Bir ev dizinleri yok. Onlara bir kabuk da açılmadığı gibi parolaları da yok. Bir sistem kullanıcısı olarak işlem görüyorlar. Burada benim hesap açtığım makinanın ismi `belma` ve `users` grubunun bir üyesi.

Yapılandırmamız bitti. Samba sunucusunu çalıştıralım ve neler yapmışız görelim:

```
# /sbin/chkconfig smb on
# /sbin/service smb start
Starting SMB services: [ TAMAM ]
done.
# smbclient -L localhost -U%
Server's Role (logon server) conflicts with share-level security
Domain=[LAB] OS=[Unix] Server=[Samba 3.0.4]

Sharename      Type      Comment
-----
netlogon       Disk      Sistemle ilgili dosyalar
depo            Disk      Dosya Deposu
IPC$           IPC       IPC Service (Samba 3.0.4)
ADMIN$         IPC       IPC Service (Samba 3.0.4)
Domain=[LAB] OS=[Unix] Server=[Samba 3.0.4]

Server          Comment
-----
NILGUN          Samba 3.0.4

Workgroup       Master
-----
LAB
```

Sunucumuz gayet sağlıklı çalışıyor. Malum çelişkili durum bu çıktıda da belirtilmiş ama onu bilerek öyle yaptım. Windows'ları yapılandırmaya geçmeden önce bunu neden yaptığımı açıklayayım. Üzerinde bu çalışmaları yaptığım w98, "domain logon" işlemi yapılmadığı zaman bazan kullanıcı girişlerinde Samba'ya bağlanamayabiliyordu. Ama "domain logon" işlemi yapmaya zorladığımda hiç kaçma şansı yok. Kendisi kullanıcıdan önce Samba'ya bağlanmak zorunda :-)... N'apalım. O mu yaman ben mi... Şimdi sesler gelecek bir yerlerden w98'ler "domain logon" yapmazlar diye, kim demiş, öyle de bir yaparlar ki... benim elime düşmemişler...

Neyse, latife bir yana w95 ve w98'ler için bir paket var onu şimdiden indirin bu işi w98'e yaptıran herşey o paketin içinde: <http://support.microsoft.com/default.aspx?kbid=142794> adresinden **nexus.exe**'yi indirin. 1 MB'lık küçük bir paket.

2.3. Windows'ların Samba sunucusuna bağlanması

İlk iş, Windows'ları "domain logon" işlemini olmadan Samba'ya bağlanmak olacak. Ondan önce, indirdiğiniz **nexus.exe** dosyasını `/export/depo` dizinine koyun, bağlandığınızda onu oradan alıp kuracaksınız ve ondan sonra "domain logon" işlemi yani ilk oturumu makinanın açtığı işlem devreye girebilecek.

Şimdi Windows kurulu makinalardan birinin başına geçin ve aşağıdaki işlemleri yapın. Benim kullandığım w98 ingilizce olduğundan yanlarında ingilizce terimlerin türkçe karşılıklarını da vereceğim ama Windows türkçesi ile tam uyumlu olmayabilir.

1. *Network Neighborhood* [Ağ Komşuları] simgesine farenin sağ tuşu ile tıklayın.
2. Açılan menüden *Properties*'e [Özellikler] tıklayın.
3. Açılan pencerede *Identification* [Kimlik Bilgileri] sekmesine tıklayın ve alanları aşağıdakine benzer şekilde doldurun:

```
Computer name [Makina ismi]: BELMA
Workgroup [Çalışma Grubu]: LAB
Computer Description [Açıklama]: Windows 98 workstation
```

4. *Access Control* [Erişim Denetimi] sekmesine tıklayın ve burada *Share-level access control* [Paylaşım seviyesinde erişim denetimi] düğmesini seçili duruma getirin.
5. *Configuration* [Yapılandırma] sekmesine tıklayın ve listeden *Client for Microsoft Networks* [Microsoft Ağları için İstemci] satırını seçin ve *Properties* [Özellikler] düğmesine tıklayın.
6. *Log on to Windows NT domain* [Windows NT alanına Bağlan] kutucuğunu işaretli duruma getirin ve *Windows NT domain* alanına LAB yazın. Windows NT alanına yani bizim sistemimiz için Samba sunucuya makinanın oturum açmasını sağlayacak olan bu bölümde yapılan işlemdir. Ancak **nexus.exe** kurulumunu yapıldıktan sonra bu yapılandırma bir işe yarayacak. Şimdilik bir hükmü yok.
7. *Network logon options* [Ağ oturumu seçenekleri] bölümünde ise *Logon and restore network connections* [Oturumu aç ve ağa bağlan] düğmesini seçili duruma getiriyoruz ve OK'e [TAMAM] basıp çıkıyoruz.
8. Yine listeden Ağ kartının TCP/IP ayarları satırını seçiyoruz ve *Properties* yani Özellikler düğmesine tıklıyoruz.
9. *Gateway* [Ağ Geçidi] sekmesinde *New gateway* [Yeni ağ geçidi] alanına 10.10.10.254 yazıp *Add* [Ekle] düğmesine tıklıyoruz.
10. *Wins Configuration* [Wins Yapılandırması] sekmesine tıklayıp, En alttaki *Use DHCP for WINS Resolution* [WINS çözümlemesi için DHCP kullanılsın] düğmesini seçili duruma getiriyoruz. Eğer bu seçenek erişilebilir değilse, *IP Address* sekmesinden *Obtain an IP address automatically* [IP adresini özdevinimli al] düğmesini seçili duruma getirdikten sonra geri dönüp DHCP ile bağlanma işlemi yapıyoruz.
11. Buradaki işimiz bu kadar. OK düğmesine tıklayıp buradan çıkıyoruz.
12. Son olarak *Primary Network Logon* [Birincil Ağ Oturumu] alanında listeden *Windows Logon*'u [Windows Oturumu] seçtikten sonra OK düğmesine tıklıyoruz. Bu noktada Windows sizden yeniden başlatılmayı talep edecektir. İzin verin, yapsın.

Sistem açıldığında kullanıcı ismi ve parola istenen alana bir isim yazın, parola isteğe bağlı. Yazacağınız ismin Linux üzerinde tanımlı olması gerekmiyor. Her kullanıcı Windows üzerinde kendine bir isim seçip onunla bağlanmakta özgür. Ancak tek yapMAMASı gereken *Cancel* [İptal] düğmesine tıklamak. O zaman kullanıcı Sambaya bağlanamayacaktır. OK düğmesine tıklayıp oturum açıldığında Samba üzerindeki paylaşımları Windows Explorer [Dosya Yöneticisi]nde Ağ komşularının altındaki NILGUN adı altında görüyor olacaksınız.

Eğer göremiyorsanız oturumu sonlandırıp, aynı kullanıcı ismiyle tekrar sisteme giriş yapın (makinaı yeniden başlatmayacaksınız sadece kullanıcının oturumdan çıkmasını sağlayacaksınız — Windows türkçesini bir türlü öğrenemedim...)

Daha önce Linux sistemindeki `/export/depo` dizinine koyduğumuz **nexus.exe** dosyası depo paylaşımı altında sizi bekliyor olacak. Onu `c:\temp` dizinine kopyalayıp orada çalıştırın. İçinden bir sürü dosya çıkacak. Sırayla *Start* [Başlat] → *Settings* [Ayarlar] → *Control Panel* [Denetim Masası] menü alanlarına tıklayın. *Add/Remove Programs* [Program Ekle/Kaldır]’a tıklayın. Açılan pencerede önce *Windows Setup* [Windows Kurulumu] sekmesine sonra da *Have Disk* [Diski Var] düğmesine tıklayın. Dosya araması ile `c:\temp` dizine gidin, oradaki tek **.inf** dosyası olan **Srvtools.inf** dosyasını kendi bulacak. Tamam’a tıkladığınızda kurulum gerçekleşecek. Bundan sonra makinaı yeniden başlattığınızda Samba’ya gayet hızlı bir biçimde bağlanacak ve paylaşımları göreceksiniz. **nexus.exe** paketinden çıkan uygulamalar bu yapılandırma altında beklendiği gibi çalışmayacaktır. Böylesi daha iyi aksi takdirde Linux makinasındaki tüm kullanıcılar (gizli sistem kullanıcıları dahil) görüntüleniyor.

Bu işlemleri her makinaı farklı isimler vererek tekrarlayın. İşlemler bittiğinde Windows ağıınız kullanıma hazır olacak.

Ağa sunucu olarak Linux üzerinde Samba’yı yerleştirip ağı tekrar çalışır duruma getirdiğinize göre artık, X terminallerinizi yapılandıracak işlemlere başlayabilirsiniz. Bunların o kadar acelesi olmayacaktır. Önce donanımları aynı olan makinalarla başlar, sonra diğer makinalara zaman içinde uygularsınız. Sunucu makinaı artık X terminallerine hizmet verecek şekilde yapılandırmaya başlayabiliriz.

2.4. XFS yapılandırması

xfs, X Pencere Sisteminin yazıtipi sunucusudur (XFS – X Font Server). İstemci makina üzerine kurulacak ramdiske X yazıtiplerini ve xfs sunucusunu koymayacağız. İstemci makinanın X Pencere Sistemi sunucu makinaadaki **xfs** sunucusunu kullanacak. Bunun için bu makinaadaki **xfs** sunucusunu dinlemeye almamız lazım. Hemen yapalım:

`/etc/X11/fs/config` dosyasının en alt satırına gidin,

```
# don't listen to TCP ports by default for security reasons
no-listen = tcp
```

son satırı iptal edeceğiz ki, **xfs** ağıdan gelen istekleri dinlesin:

```
# don't listen to TCP ports by default for security reasons
# no-listen = tcp
```

Eğer **xfs**’i kullanmıyorsanız başlatın, ben kullanıyordum, yeniden başlatıyorum ve dinlemeye geçmiş mi bakıyorum:

```
# /sbin/chkconfig xfs on
# /sbin/service xfs restart
xfs yeniden başlatılıyor:
xfs durduruluyor: [ TAMAM ]
xfs başlatılıyor: [ TAMAM ]
# netstat -Npl | grep xfs
tcp        0      0 0.0.0.0:xfs          0.0.0.0:*          LISTEN
   1249/xfs
unix       2      [ ACC ]     STREAM    LISTENING   9433    1249/xfs
   /tmp/.font-unix/fs7100
```

Bu da tamamdır.

2.5. GDM yapılandırması

GDM, X terminallerin sunucuya bağlanabilmesi için onlarda başlatılmış X oturumlarından gelecek istekleri dinler, bu isteklere GDM oturum açma ekranını gönderir ve kullanıcı girişi bekler. Verilen kullanıcı ismi ve parolası, üzerinde çalıştığı Linux sisteminde tanımlı kullanıcılardan biri ise üzerinde çalıştığı makinanın hemen tüm kaynaklarını oturum açmak isteyen kullanıcıya sunar. Bizim yapılandırmamızda ise kullanıcı 10 saniye içinde bir isim ve parola girmezse öntanımlı kullanıcı ismiyle sunucuya bağlanabiliyor. Ayrıca GDM, sadece terminalleri yönetiyor. Sunucu makina üzerinde bir X oturumu açılmak istenirse normal bir X oturumu olarak çalıştırılabilir.

gdm yapılandırması `/etc/X11/gdm/gdm.conf` dosyasında yapılıyor. Masaüstünde çalışan görsel bir GDM yapılandırma uygulaması olmasına rağmen biz onu kullanmayacağız. Çünkü bu yapılandırmayı yapabilmek için yeterli değil. `/etc/X11/gdm/gdm.conf` dosyasının içeriğini aşağıdaki örneğe göre düzenleyin:

```
[daemon]
# X terminallerinden bir kullanıcı 10 saniye içinde bir kullanıcı ismi ve
# parolası girmezse "talebe" kullanıcısı ile oturum açılacak.
# Bunu aşağıdaki seçeneklerle sağlıyoruz. "talebe" için makinada bir
# parolasız bir hesap tanımlanmış olmalı.
TimedLoginEnable=true
TimedLogin=talebe
TimedLoginDelay=10

# X terminallerinin bağlanması için kullanıcıya gösterilen oturum
# açma ekranını çalıştıran komut. Ağ yükü ağır gelirse daha basit
# görünümdeki gdmlogin ile değiştirebilirsiniz.
#RemoteGreeter=/usr/bin/gdmlogin
RemoteGreeter=/usr/bin/gdmgreeter

# gdm'nin sistem üzerinde çeşitli amaçlar için kullandığı kullanıcı ve
# grup ismi. Bunlara dokunmuyoruz.
User=gdm
Group=gdm

# gdm'nin günlük kayıtlarını tuttuğu dizin ve süreç kimliğini sakladığı dosya.
# Dokunmuyoruz, bilgi için kalsın.
LogDir=/var/log/gdm
PidFile=/var/run/gdm.pid

# X terminallerine oturum açmadan önce açarken ve açtıktan sonra
# yapılan eylemlerle ilgi yapılandırma. Dokunmuyoruz.
PostLoginScriptDir=/etc/X11/gdm/PostLogin/
PreSessionScriptDir=/etc/X11/gdm/PreSession/
PostSessionScriptDir=/etc/X11/gdm/PostSession/
DisplayInitDir=/etc/X11/gdm/Init

# X çökerse çalıştırılacak betik. Dokunmuyoruz.
XKeepsCrashing=/etc/X11/gdm/XKeepsCrashing

# Kimlik denetimi ile ilgili kayıtların tutulacağı dizin. Dokunmuyoruz.
ServAuthDir=/var/gdm

# Hangi masaüstünün çalıştırılacağını saptandığı betik.
BaseXsession=/etc/X11/gdm/Xsession

# Kullanıcı yetkilendirmesi ile ilgili X dosyaları için.
# Kullanıcının ev dizini yazılabilir değilse bu dizin kullanılacak.
UserAuthFBDir=/tmp
```

```
UserAuthFile=.Xauthority

# X sunucusu
StandardXServer=/usr/X11R6/bin/X

# X nest komutu
Xnest=/usr/X11R6/bin/Xnest -audit 0 -name Xnest

# Sistemde 6 tane kullanılabilir metin konsolu var (tty1-6).
# 7. konsolu bu makina üzerinde çalıştırılacak X oturumuna ayırıyoruz.
# Dolayısıyla bir terminalin alabileceği ilk konsol 8. olacak.
# /etc/securetty dosyanızdaki tty'ler tüm X terminallerinize yetmiyorsa
# sayısını arttırın.
FirstVT=8
VTAllocation=true

[security]
# X terminallerinden root kullanıcısının oturum açmasına izin vermiyoruz.
AllowRemoteRoot=false

# Terminallerden 10 saniye içinde giriş yapılmazsa oturumun "talebe"
# kullanıcısı ile açılmasına izin veren seçenek.
AllowRemoteAutoLogin=true

# yazma yetkisi: 0 kullanıcı, 1 grup, 2 herkes
# Bu yapılandırmada hiçbir önemi yok. Dokunmuyoruz.
RelaxPermissions=0

# gdm'nin terminallerden gelen istekleri dinlememesini sağlayan seçenek.
# Bu yapılandırma dinlemesini gerektiriyor.
DisallowTCP=false

[xdmcp]
# Standart xdmcp portu olan 177/udp portunun dinlenmesini sağlar.
Enable=true

[gui]
# Burada değiştireceğimiz bir şey yok.

[greeter]
# Oturum açma ekranı ile ilgil görsel ayarlar.

# Paneldeki saatin 24 saatlik olmasını isteriz.
Use24Clock=true

# Oturum açma ekranında kullanılan tema ve temaların bulunduğu yer.
#GraphicalTheme=circles
GraphicalThemeDir=/usr/share/gdm/themes/

[chooser]
# chooser, aşağıdaki diğer X terminal sunucularını listesini gösterir.
# Oturumun seçilen sunucuda açılmasını sağlar.
```

```
# Bu seçenekler çok sayıda X terminal sunucusu çalıştıracaksanız anlamlı.

#DefaultHostImg=/usr/share/pixmaps/nohost.png
# Directory with host images, they are named by the hosts: host or host.png
HostImageDir=/usr/share/hosts/
# Time we scan for hosts (well only the time we tell the user we are
# scanning actually, we continue to listen even after this has
# expired)
#ScanTime=4
# A comma separated lists of hosts to automatically add (if they answer to
# a query of course). You can use this to reach hosts that broadcast cannot
# reach.
Hosts=
# Broadcast a query to get all hosts on the current network that answer
Broadcast=true
# Set it to true if you want to send a multicast query to hosts.
Multicast=false
# It is an IPv6 multicast address.It is hardcoded here and will be replaced when
# officially registered xdmcp multicast address of TBD will be available
#Multicast_Addr=ff02::1
# Allow adding random hosts to the list by typing in their names
#AllowAdd=true

[debug]
# Hata ayıklama iletilerini isterseniz true yapın.
Enable=false

[servers]
# gdm'nin bu makina üzerindeki X oturumunu da yönetmesini istiyorsanız
# bu satırlardan ilkinin etkinleştirin. İkisini de etkinleştirebilirsiniz.
#0=Standard
#1=Standard

[server-Standard]
# servers bölümündeki ayarlara bağlı olarak buradaki seçenekler
# anlam kazanır.
name=Standard server
command=/usr/X11R6/bin/X -audit 0
flexible=true

[server-Terminal]
# Bu X terminal sunucusu.
name=Terminal server
command=/usr/X11R6/bin/X -audit 0 -terminate
# Make this not appear in the flexible servers (we need extra params
# anyway, and terminate would be bad for xdmcp choosing). You can
# make a terminal server flexible, but not with an indirect query.
# If you need flexible indirect query server, then you must get rid
# of the -terminate and the only way to kill the flexible server will
# then be by Ctrl-Alt-Backspace
flexible=false
# Not local, we do not handle the logins for this X server
handled=false

[server-Chooser]
# Diğer sunucuları seçmeyi sağlayan ekran.
name=Chooser server
```

```
command=/usr/X11R6/bin/X -audit 0
```

Yapılandırma dosyasında bir öntanımlı kullanıcı belirtmiştik. Şimdi o kullanıcı için sistemde bir hesap açalım. Bu kullanıcının parolası olmayacak:

```
# /usr/sbin/useradd -g users talebe
# passwd -l talebe
```

Bu komutlarla sisteme talebe kullanıcıasını tanımlamış olduk. Artık dosya sisteminde bir `/home/talebe` dizini var. X terminallerinden bir kullanıcı ismi belirtmeden oturum açan bütün kullanıcılar bu dizini ev dizini olarak ortak kullanacaklar. Bu hoş bir durum değil, kullanıcılara istedikleri takdirde bir kullanıcı ismi vermeniz daha iyi olur. Diğer yandan sahipsiz kalmış kullanıcı isimlerine ait ev dizinlerinin varolacak olması da sizin açınızdan hoş olmayabilir. Karar sizin. Her iki durum da bu yapılandırma ile mümkün.

`/home/talebe` dizininde `.dmrc` adıyla bir dosya açın ve içine aşağıdakileri yazın:

```
[Desktop]
Session=kde
Language=tr_TR.UTF-8
```

Bu dosyayı oturum açarken **gdm** okuyacak ve oturumu türkçe olarak `Session` satırında belirtilen masaüstü yöneticisi ile açacak.

Şimdi sıra `/etc/inittab` dosyasında. Bu makina üzerinde X oturumunu hiç başlatmaz, siz konsolda kalabilirsiniz ve isterseniz **startx** komutu ile bir X oturumu açabilirsiniz. Bir sunucu için en elverişli yöntem budur. `inittab` dosyasını buna göre yapılandırıyoruz ve sadece `initdefault` satırını 3. çalışma seviyesine ayarlıyoruz. Zaten böyleyse dokunmayın:

```
id:3:initdefault:
```

gdm bu makinada artalanda bir sunucu olarak çalışacağından sistem açılışında çalışmasını sağlamamız lazım. Bunun için `/etc/rc.d/rc.local` dosyasının sonuna:

```
/usr/bin/gdm
```

satırını ekliyoruz.

Sistemi yeniden başlattığınızda X sunucusu ağdan gelecek istekleri bekliyor olacak. **nmap** ile bakarsanız aşağıdakine benzer bir çıktı alacaksınız:

```
# nmap -sTU 127.0.0.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 3044 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
67/udp    open       dhcpserver
137/udp    open       netbios-ns
138/udp    open       netbios-dgm
139/tcp    open       netbios-ssn
177/udp    open       xdmcp
445/tcp    open       microsoft-ds
514/tcp    open       shell
901/tcp    open       samba-swat
6000/tcp   open       X11
7100/tcp   open       font-service

Nmap run completed -- 1 IP address (1 host up) scanned in 9 seconds
```

22/tcp open ssh

satırı sizde olmayabilir. Ama olacak. Yeri geldiğinde ondan bahsedeceğim.

3. X Terminal Sisteminin Yapılması

Bu kısım bu belgenin en önemli bölümü. Burada en temel Linux sistemini sıfırdan oluşturacağız. Gerekenlerin çoğunu sunucu makinaya zaten kurdunuz. Onlardan bize gerekenleri toplayacağız, biraz da internetten birşeyler toplayacağız, hepsini harmanlayıp, tek kullanıcı bir Linux yapacağız. Ancak, bu sistem terminal makinalarının sadece belleğinde varolacağından, mümkün olduğunca az yer kaplayan bir sistem olacak. İşimizi göreceğiz olan herşey olacak ama işimizi göreceğiz herhangi bir şeyi değil, en az yer kaplayanını kullanacağız. Önce bize neler lazım bir bakalım:

Ramdisk oluşturma araçları

Bunlar sisteminizde mevcut.

Çekirdek

Bir Linux sisteminin olmazsa olmaz parçası, herşeyi yöneten en temel parça. Herhangi bir Linux çekirdeği işimizi görür. Ben linux-2.4.22 çekirdeğini kullandım. Çekirdeği kendimiz derleyeceğiz. İçinde bize gerekli olmayan hiçbir şey olmamalı ki, bellekte fazla yer işgal etmesin.

init

Çekirdek kendini yükledikten sonra sistemin kullanıcıya hazırlanmasını sağlayan **init**'i çalıştırır. **init**'de yapılandırmasına bağlı olarak bazı betikleri çalıştırır.

Kabuk

init bazı betikleri çalıştırır dedik mi, o betikleri çalıştıracak bir kabuk gerektiği anlaşılır. Bu kabuk az yer kaplamalı. **init** ile **sh** kabuğunu ve bir sistem için gereken en temel komutları biraraya getiren bir uygulama var: **busybox**^(B3). Biz onu kullanacağız. Hemen indirin. Dağıtımlarla gelen bir paket değildir. Böyle çok özel uygulamalar için tasarlanmıştır.

DHCP istemcisi

Sunucu makinaya bağlanmak için ağ kartını DHCP sunucusundan aldığımız bilgilerle yapılandıracağız. Pek çok DHCP istemcisi var ama biz **busybox** içinde yerleşik olarak bulunan **udhcpd**'yi kullanacağız.

X sunucusu

X sunucusu; X'in ismi sunucu. İsmi sunucu da olsa burada X terminal istemcisi olarak çalışacak. X, kendi kütüphaneleri ve modülleri olan devasa bir sistem. Ama biz onu da küçülteceğiz! **xfs**'i baştan attık, dolayısıyla yazıtiplerinden kurtulduk. Sunucu kurulumunda hatırlarsanız **xfs**'i dinlemeye almıştık. Yazıtiplerini ondan yükleyeceğiz. Daha kullanmayacağımız çok şey bulacağız. Sırası var, sırası gelince ayrıntısına gireceğim.

Çekirdek dışındaki herşey ramdiski oluşturacak olan **initrd.img** dosyası içinde bulunacak. Bu dosya çok özel bir dosya; içinde bir dosya sistemi barındırıyor olacak. Önce bu dosyayı oluşturacağız. Sonra bize uygun bir çekirdek derleyeceğiz. Ardından basit bir Linux oturumu sağlayacak şekilde bir **initrd.img** oluşturup onu deneyeceğiz. En son aşamada ise **initrd.img** dosyasına X ile ilgili dosyaları yerleştirip X terminalimizi oluşturacağız.

3.1. **initrd.img** Dosyasının Oluşturulması

Burada, önce kurulum sırasında ayırdığımız ek disk bölümü üzerinde bir dosya sistemi oluşturacağız sonra da onu kök dosya sistemine bağlayacağız. Kök dosya sisteminizin `/dev/hda1` disk bölümünde olduğunu, takas alanının `/dev/hda2` bölümünde olduğunu, ayırdığınız ek disk bölümünün de `/dev/hda3` olduğunu varsayacağım. Hangi disk bölümlerinin ne için kullanıldığını **sfdisk** komutu ile öğrenebilirsiniz:

```
# /sbin/sfdisk -l

Disk /dev/hda: 4870 silindir, 255 kafa, 63 sektör/iz
birimler = 8225280 baytlık silindir, 1024 baytlık blok, 0'den başlayarak

          silindir   blok
Aygıt Önykl Balanç Bitiş sayısı   sayısı
/dev/hda1 *      0+   1200   1200-   9647001+  83  Linux
/dev/hda2      1201   1324   123-   987966   82  Linux takas
/dev/hda3      1325+  2525   1200-   9647001      Boş
...
```

Siz tam böyle olmasa da benzer bir liste alacaksınız. Şimdi `/dev/hda3` üzerinde bir ext3 dosya sistemi oluşturalım:

```
# /sbin/mke2fs -b 1024 -O sparse_super,filetype,has_journal /dev/hda3
```

Bu komut ile blok boyu 1024 bayt olan bir dosya sistemi oluşturduk. Blok boyunu özellikle 1024 bayt olarak seçtim. Çünkü çekirdek ramdiski 1024 baytlık bloklarla oluşturur. Ramdiskin bir örneğini bu disk bölümü üzerinde oluşturacağımızdan doğru boyut raporu alabilmek için 1024 bayt blok boyuna sahip bir dosya sistemine ihtiyacımız var. Kök disk bölümü genelde 4096 baytlık blok boyunda oluşturulur. Onun için kök disk bölümünde bu işi yapmamız doğru olmazdı.

Oluşturduğumuz dosya sisteminin kök dizine bağlamadan kullanamayız. Onun için `/a3` isminde bir dizin oluşturacağız, sonra da `/dev/hda3`'ü buraya bağlayacağız. Disk bölümünü bağlamadan önce bu disk bölümü ile ilgili bilgiyi `/etc/fstab` dosyasına ek bir satır olarak girelim ki, sistemi her yeniden başlattığınızda bu disk bölümü açılış sırasında `/a3` altına bağlansın:

```
# echo "/dev/hda3 /a3 ext3 defaults 1 2" >> /etc/fstab
# mount /a3
```

İlk komutta `/dev/hda3` ile ilgili bilgiyi `/etc/fstab` dosyasının sonuna bir satır olarak ekledik. Bu komutta dikkat edeceğimiz şey `/etc/fstab`'dan önce iki tane `>` konulması gerektiğidir. Bir tane `>` koyarsanız dosya silinir ve dosyaya sadece bu satır yazılır. Halbuki biz dosya sonuna ek bir satır girmek istiyoruz. Onun için `>` işaretinden peşpeşe 2 tane kullanmamız gerekiyor. Dikkatli olun!

İkinci satırda ise `/dev/hda3`'ü `/a3` dizinine bağladık. Nereden belli? **mount** komutu böyle tek parametrelili kullanıldığında diğer parametreleri almak için `/etc/fstab` dosyasına bakar. Bu komutu bir daha kullanmayacağız. Bundan sonra sistemi her açtığınızda `/a3` bağlı olacak.

`/a3` dizinine geçip, bir dizin açıyoruz (ramdisk dizini) ve ramdisk dizinine giriyoruz:

```
# cd /a3
# mkdir ramdisk
# cd ramdisk
```

Bu dizine **initrdyap** isminde bir dosya açıyoruz ve bu dosyayı çalıştırılabilir bir dosya yapıyoruz:

```
# touch initrdyap
# chmod +x initrdyap
```

Sonra da bu dosyanın içine aşağıdakileri yazıyoruz:


```
#!/bin/bash

compress=1
target="/export/netlogon/autoboot/initrd.img"
rc=0

IMAGESIZE=12000

MNTIMAGE="/a3/ramdisk/initrd"
IMAGE="/a3/ramdisk/initrd.rd"
MNTPOINT="/a3/ramdisk/initrd.mnt"

dd if=/dev/zero of=$IMAGE bs=1k count=$IMAGESIZE 2> /dev/null || exit 1

LODEV=$(echo findldev | /sbin/nash --quiet)

if [ -z "$LODEV" ]; then
    echo "loopback aygıtlarının hepsi kullanılmış." >&2
    exit 1
fi

/sbin/losetup ${LODEV} $IMAGE || exit 1

echo y | /sbin/mke2fs $LODEV $IMAGESIZE >/dev/null 2>/dev/null
/sbin/tune2fs -i0 $LODEV >/dev/null

if [ -n "$verbose" ]; then
    echo "$LODEV aygıtı kullanılıyor"
fi

mount -t ext2 $LODEV $MNTPOINT || {
    echo "Loopback aygıtı yok"
    exit 1
}

(cd $MNTIMAGE; tar cf - .) | (cd $MNTPOINT; tar xf -) || exit 1

umount $MNTPOINT
/sbin/losetup -d $LODEV

if [ -n "$compress" ]; then
    gzip -9 < $IMAGE > $target || rc=1
else
    cp -a $IMAGE $target || rc=1
fi

exit $rc
```

Bu betiği adından da anlaşılacağı üzere `initrd.img` dosyasını oluşturmakta kullanacağız. Şimdi bu betiğin çalışması için gereken izinleri oluşturalım:

```
# mkdir initrd
# mkdir initrd.mnt
```

Ramdiski oluşturacağımız dizin `initrd` dizini; `initrd.mnt` dizini boş kalacak. Orayı betik kullanacak. Betiği şimdi çalıştıracamız ve betik 12000 kB'lık `/a3/ramdisk/initrd.rd` dosyası ile bu dosyanın sıkıştırılmış hali olan `/export/netlogon/autoboot/initrd.img` dosyasını oluşturacak:

```
# ./initrdyap
```

Komut, herşey yolunda giderse konsolda hiçbir çıktı vermeyecek ve bahsettiğim 2 dosyayı oluşturacak. Konsola bir çıktı veriyorsa bu bir hata iletilidir. Birşeyleri eksik ya da yanlış yapmış olabilirsiniz. Bu betiği kendim yazmadım, `/sbin/mkinitrd` betiğinden bizim için gereksiz satırları temizledim ve kalan kodu biraz değiştirdim. (Amerika'yı yeniden keşfetmenin alemi yok). `/sbin/mkinitrd`'nin kullandığı yöntem bize `/a3/ramdisk/initrd` dizininde normal bir dizin gibi çalışma imkanı veriyor. Yani oraya yerleştireceğimiz ve kaldıracağımız dosyalar `initrd.img` dosyasında değişiklik bilgisi olarak görünmeyecek. Diğer yöntemlerde görünür ve dosya gereksiz yere şişer.

3.2. Çekirdeğin derlenmesi

Bu belgenin yazıldığı tarihte güncel çekirdek sürümleri 2.6.x idi. Ancak 2.6 çekirdekleri derlemek için sistemin bu çekirdeğe uygun olması gerekir. Benim elimdeki sistem buna uygun olmadığından ben 2.4 çekirdeklerle çalışmak zorundaydım. Yapacağımız iş için çekirdeğin 2.4 ya da 2.6 olmasının bir önemi yok. Ben zaten kullanmakta olduğum 2.4.22 çekirdeğinin kaynak kodunu kullandım. Siz de kurduğunuz sistemde bulunan çekirdeğin sürümü 2.4 serisinde ise onun kaynak kodunu kullanabilirsiniz. 2.6 çekirdeklerin derlenmesi biraz farklı sanırım (Hiç 2.6 çekirdek derlemedim). Yalnız çekirdeği derledikten sonra onu sisteme kuracağımızdan ismini değiştireceğiz. 2.4.22-x yapacağız ki, sistemde kurulu çekirdekle karıştırmayalım.

Çekirdeğin kaynak kodunun yeri dağıtımdan dağıtıma değişmez, yeri `/usr/src/linux` dizinidir. `/usr/src/linux` genelde kullanılan çekirdeğin kaynak kodunun bulunduğu dizini işaret eden bir sembolik bağ olarak düzenlenir. Bizim çalışacağımız dizin `/usr/src/linux-2.4.22` dizini olacak.

X terminalleri için kullanacağımız çekirdek kaynak kodunun sisteminizin çekirdek kaynak kodundan farklı bir yerde durması daha iyidir. `linux-2.4.22-x` diye bir dizin açıp `linux-2.4.22` dizininin içeriğini bu dizine kopyalayın. Daha önce bir çekirdek derlemişseniz yeni açtığınız dizine geçip `make mrproper` komutunu verin. Bu komut çekirdek kaynak kodunu sıfırlar. Yani onu hiç derleme yapılmamış duruma getirir. Kaynak kodunu derlememişseniz bu işlemi yapmanıza gerek yoktur. Ancak, birşeyler yanlış gider ve derleme işlemi yinelenmek isterseniz derlemeye başlamadan önce vereceğiniz ilk komut bu olmalı.

Eğer `/usr/src` altında bir linux dizini yoksa ve siz de X terminalleri için 2.4.22 çekirdeğini kullanmak isterseniz, bu çekirdeğin kaynak paketini <ftp://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.22.tar.bz2> adresinden `/usr/src` dizinine indirin ve aşağıdaki komutlarla paketi açın:

```
# bunzip2 linux-2.4.22.tar.bz2
# tar xfv linux-2.4.22.tar
```

Bu iki komut kaynak paketini `linux-2.4.22` dizini altında açacaktır. Dizine geçin ve Makefile dosyasını düzenlemek üzere açıp 4. satırı

```
EXTRAVERSION = -x
```

yapın. Bu derlenmiş çekirdeğin ismini `vmlinux-2.4.22-x` yapacak. Böylece bu çekirdeğin X terminalleri için derlediğimiz çekirdek olduğunu hatırlayacağız. Hatırlamamız lazım da, çünkü bu çekirdekle üzerinde çalıştığınız Linux sistemini açamayacaksınız. Bu çekirdek sadece X terminallerini çalıştırabilecek kadar yeteneğe sahip olacak. Örneğin ağ kartı, fare, ekran kartı ve makinanın belleği gibi temel donanım bilgilerinden başka donanım bilgisi içermeyecek. Sabit diskleri, disket sürücülerini, CDRom'ları ses kartlarını v.s. tanımayacak. Bir terminale bunlar gerekmez.

Linux çekirdeğini derlemek için iyi derecede İng. bilmek daha da ötesi iyi derecede sistem bilgisi gerekir. Bu derece ayrıntı bu küçük belgenin sınırlarını aşar. Bu bilgi ayrı bir kitap olur.

Ben sadece çekirdek yapılandırmasının sonucu olan `.config` dosyasının içeriğini vereceğim. Bunun yanında da neyi niçin yaptığımı hakkında kısa bir bilgi vereceğim. *Yapılandırma dosyasını* (sayfa: 39) eklerde bulabilirsiniz. Onu ayrı bir pencerede açıp burada yapacağımı açıklamaları oradan izleyebilirsiniz.

- 14. satırda başlayan bölümde çekirdeğe yüklenebilir modül desteği veriyoruz.
- 21. satırda başlayan bölümde çekirdeğin 386 ve sonraki işlemciler için geçerli olmasını sağlıyoruz. Böylece 386, 486, 586 ve 686 işlemcileri ve bunlarla uyumlu (AMD gibi) çekirdeklerle çekirdeğimiz çalışabilecek. Alt satırlarda bir 386 çekirdeğin desteklemediği hiçbir özelliği çekirdeğe vermiyoruz. 386SX'lerde bu çekirdek çalışmaz. En az 386DX gerekir. Bunların ne olduğunu bilmiyorsanız, boşverin. Karşılaşmamışsınız demektir (antika).
- 67. satırda başlayan genel ayarlar bölümünde çekirdeğe, 69. satırda ağ desteği, 70'de PCI desteği ve devamında sistemin genelini etkileyen bazı özel durumlara ilişki destekleri veriyoruz.
- 112. satırda başlayan blok aygıtları bölümünde sadece ramdisk ve initrd ile ilgili destekleri veriyoruz.
- 142. satırda başlayan ağ seçenekleri bölümünde bir linux ağı için gerekli ne varsa bunların desteğini veriyoruz.
- 242. satırda başlayan bölümde ağ aygıtları ile ilgili desteği açıyoruz ve 257. satırda başlayan 10/100 Mbitlik ethernet kartlarının tamamının modül olarak derlenmesini istiyoruz. Piyasadaki en ucuz, en kullanışlı ve en çok kullanılan ağ kartları bunlar. Sizininki bunlardan biri değilse biraz paraya kıyıp bunlardan temin edin ve ağınıza bunlara göre oluşturun. 100MBitlik bir ağın tadına doyum olmaz.
- 404. satırda başlayan bölümde klavye ve fare için donanımsal destek, 415. satırda başlayan bölümde seri portlarla ve konsol ile ilgili destek, 449. satırda başlayan bölümde ise fare türleri ile ilgili destek var.
- 496. satırda başlayan bölümde sadece donanım saati desteğini veriyoruz.
- 534. satırda başlayan bölümde ramdisk için gereken dosya sistemi ile ilgili destekleri veriyoruz.

Bu dosyayı `/usr/src/linux-2.4.22` dizini altına `.config` ismiyle kaydedin (başında `.` olacak).

```
# make menuconfig
```

Komutunu verin. Bir takım işlemlerden sonra bir pencere açılacak. Hiç orada dolaşmayın. `ESC` `ESC` ile çıkın ve ardından sırasıyla şu komutları verin:

```
# make dep
# make bzImage
# make modules
# make install
# make modules_install
```

Bu komutların herbiri biraz vakit alır, bu işlemler sürerken makinada oyun filan oynayabilirsiniz :-) Merak etmeyin işlemi öyle aman aman yavaşlatmaz.

Bu işlemler sonucunda `/boot` dizininde isminde `2.4.22-x` bulunan bazı dosyalar oluşacak. Bunlardan `vmlinuz-2.4.22-x` isimli dosya çekirdeğimizdir. Onu `/export/netlogon/autoboot` dizinine `vmlinuz` ismiyle kopyalayın. `vmlinuz-2.4.22-x` çekirdeğinin sistemin kendi çekirdeğine göre ne kadar küçük olduğuna dikkat edin (yarısı kadar).

Ayrıca derlenen modüller de `/lib/modules/2.4.22-x` dizini altındaki dizinlere konulmuş olacak. Bize `/lib/modules/2.4.22-x/kernel/drivers/net` dizini altındaki modüllerden biri gerekecek. Hangisi olacağını nasıl tespit edeceğiz? dersiniz ona da sıra gelecek.

3.3. Tek Başına (basit) Linux

Kullanacağımız çekirdek hazır. Artık ramdiskimizin içini doldurmaya başlayabiliriz. Önce bu bölümde en temel ve en basit bir konsol oturumu oluşturacağız. Bu sistem henüz sonucu makineye bağlanabilen bir yapı içermeyecek,

her türlü donanımla çalışabilen temel yapıyı oluşturup deneyeceğiz. Sonraki bölümde bunu çalıştıracığımız makinanın donanımına göre özelleştireceğiz ve sunucuya bağlanabilir hale getireceğiz ve tekrar deneyeceğiz. Son olarak da üzerine X ile ilgili dosyaları yerleştirip X terminalimizi oluşturacağız.

Sistemimiz bir pakete çok bağımlı, bu paket **busybox** kabuğunu içeriyor. **busybox**, **bash** gibi salt bir kabuk değil, içinde bize lazım olan hemen her sistem komutu var. Bu tür sistemler için özel olarak hazırlanmış bir paket. Önce onu indirip derleyeceğiz.

<http://www.busybox.net/downloads> adresinden **busybox**'ın son sürümünü indirin ve `/usr/src` dizinine taşıyın. Ben **busybox-1.00-pre7** sürümü ile çalıştım. Bu belgeyi hazırlarken indirdiğim **busybox-1.00-pre10** sürümünü derleyemedim. Komut geçmişi desteğini kaldırınca derleniyordu. Komut geçmişi olmadan da konsolda çalışmak işkence oluyor. Onun için size de **busybox-1.00-pre7** sürümünü öneririm.

busybox-1.00-pre7 paketini önce açın; çekirdek için yaptığımız gibi. **busybox-1.00-pre7** dizinine eklerde verdiğim *yapılandırma dosyasını* (sayfa: 51) `.config` ismiyle kopyalayın. **busybox**'da tıpkı çekirdek gibi bir yapılandırma arayüzüne sahip. **make menuconfig** komutundan sonra **make** sonra da **make install** komutlarını verdiğinizde derlenmiş **busybox**'ı `/usr/src/busybox-1.00-pre7/_install` dizinin altındaki dizinlerde bulacaksınız.

Bütün araçları tamamladığımıza göre `/a3/ramdisk/initrd` dizininin içeriğini oluşturmaya başlayabiliriz. `/a3/ramdisk/initrd` dizine geçeceğiz ve dizinin içinde bazı alt dizinler oluşturacağız. Bu dizin ağacı ramdisk içindeki kök dizinimizin ağacı olacağından, bir kök dizin için gereken bütün dizinleri değil, sadece bize gerekli olan dizinleri oluşturacağız. Çünkü bizim yerimiz az ve kıymetli. Aşağıdaki satırları `/a3/ramdisk` dizinine `kokyap` ismiyle kaydedin:

```
ilk='pwd'
cd initrd
kok='pwd'
mkdir bin dev etc lib loopfs proc sysroot tmp usr var
ln -s bin sbin
mkdir etc/X11 etc/init.d etc/X11/xserver
mkdir lib/X11 lib/modules
cd lib/modules
mkdir 2.4.22-x drivers extensions fonts input linux
mkdir 2.4.22-x/kernel 2.4.22-x/kernel/drivers 2.4.22-x/kernel/drivers/net
cd $kok
cd lib/X11
mkdir app-defaults locale xkb
ln -s ../../etc/X11/xserver xserver
cd $kok
cd etc/X11
ln -s ../../lib/X11/xkb xkb
cd $kok
cd usr
ln -s .. X11R6
ln -s ../bin bin
ln -s ../bin sbin
ln -s ../lib lib
cd $kok
cd var
mkdir lib lock log run lib/dhcp lib/xkb lock/subsys
cd ../dev
ln -s /proc/self/fd fd
ln -s fd/0 stdin
ln -s fd/1 stdout
```

```
ln -s fd/2 stderr
mkfifo -m 600 initctl
mknod -m 666 tty c 5 0
mknod -m 600 console c 5 1
mknod -m 660 initrd b 1 250
mknod -m 640 ram0 b 1 0
mknod -m 640 ram1 b 1 1
mknod -m 640 mem c 1 1
mknod -m 640 kmem c 1 2
mknod -m 666 null c 1 3
mknod -m 644 random c 1 8
mknod -m 640 apm_bios c 10 134
mknod -m 640 rtc c 10 135
mknod -m 600 tty1 c 4 1
mknod -m 600 tty2 c 4 2
mknod -m 600 tty3 c 4 3
mknod -m 600 tty4 c 4 4
mknod -m 600 tty5 c 4 5
mknod -m 600 ttyS0 c 4 64
mknod -m 600 ttyS1 c 4 65
mknod -m 640 psaux c 10 1
mknod -m 644 urandom c 1 9
mknod -m 444 zero c 1 5
ln -s ttyS0 mouse
ln -s ram1 ram
ln -s ram0 ramdisk
ln -s console tty0
cd $ilk
```

kokyap betiğini,

```
# . ./kokyap
```

komutunu vererek çalıştırın. Betiğin yaptığı işlemleri `initrd` dizinine geçip görebilirsiniz. Betik tarafından oluşturulan dizinlerin hepsini biz kullanmayacağız, bazı dizinleri çalıştırdığımız uygulamalar kullanacak. Yani bazı dizinleri boş bırakacağız. **kokyap** betiğinin içindeki komutları komut satırından tek tek vererek de aynı işlemi kendiniz yapabilirsiniz. Betikte kullanılan komutların pek çoğunun bir türkçe kılavuz sayfası var: [\(B8\)](#), [\(B9\)](#), [mkfifo^{\(B10\)}](#), [mknod^{\(B11\)}](#). [cd^{\(B12\)}](#) komutu ise, kabuğun [yerleşik komutlarından^{\(B13\)}](#) biridir.

Şimdi sistem açılışı için gerekli dosyaları yerleştireceğiz. Bir Linux sisteminin açılışında ne gibi işlemler yürütüldüğünü [\(B14\)](#) kılavuz sayfasında açıklanmıştır. Ayrıca, [\(B15\)](#), [\(B16\)](#) kılavuz sayfalarını da okursanız yapacağımız işlemleri niçin yaptığımızı, neden bazı şeyleri orada yazılanlardan biraz farklı yaptığımızı daha kolay anlayabilirsiniz.

[\(B17\)](#) kılavuz sayfasında yazdığı gibi önce `/linuxrc` betiğini yazacağız. Aşağıdaki betiği `/a3/ramdisk/initrd` dizinine `linuxrc` ismiyle kaydedin:

```
#!/bin/nash

echo /proc dosya sistemi baglaniyor
mount -t proc /proc /proc
echo Blok aygitlari olusturuluyor
mkdevices /dev
echo Kok aygiti olusturuluyor
mkrootdev /dev/root
echo 0x0100 > /proc/sys/kernel/real-root-dev
echo "Init sureci basliyor"
```

```
rm -f /linuxrc
```

Bu dosyayı oluşturduktan sonra,

```
# chmod +x linuxrc
```

komutuyla betiğe çalıştırılabilirlik özelliği verin.

Bu betiği biraz inceleyelim. İlk satır, betiğin `/bin` dizininde bulunan **nash** isimli bir betik yorumlayıcısı tarafından yorumlanacağını söylüyor. Yani bu betik içinde yazılan komutları bu uygulama çalıştıracak. **nash**, sadece linuxrc betiklerini çalıştırmak için tasarlanmış, küçük bir betik yorumlayıcıdır. Bir kabuk değildir. Sisteminizde **nash** vardır. Komut satırına **whereis nash** yazarsanız, **nash**'in sisteminizde nerede durduğunu öğrenebilirsiniz. Ben bu komutu verdiğimde:

```
# whereis nash
nash: /sbin/nash /usr/share/man/man8/nash.8 /usr/share/man/man8/nash.8.gz
```

şeklinde bir çıktı alıyorum. O halde yapacağımız işlem basit bir kopyalama işlemi:

```
# cp /sbin/nash /a3/ramdisk/initrd/bin/
```

İşimiz **nash**'i kopyalamakla bitmiyor. **nash**'ın içerdiği yerleşik komutlar için **nash**'a birer sembolik bağ yerleştireceğiz:

```
# cd bin
# ln -s losetup nash
# ln -s mkdevices nash
# ln -s makerootdev nash
# ln -s pivot_root nash
# ln -s showlabels nash
```

Böylece **linuxrc** betiğinin çalışması için gerek ve yeter şart sağlanmış oldu. Betik **echo 0x0100 > /proc/sys/kernel/real-root-dev** satırıyla çekirdeğe kök dosya sistemimizin ramdisk olduğunu söylüyor. Init süreci başladığında çekirdeğin kök dosya sistemi olarak nereyi kullanacağını belirtmiş olduk. **linuxrc** betiğinin tekrar tekrar çalıştırılmasını önlemek için son satırda betiği siliyoruz ve **linuxrc** betiğinin işi bitiyor, görevi **init** süreci devralıyor.

init bir uygulama ismi. Ama bizim sistemimizde henüz yok. Şimdi onu ve bir sistemde kullanılan bir çok komutu içeren **busybox**'i sistemimize yerleştireceğiz. Busybox'ı bu bölümün başlangıcında indirmiş ve derlemiştik. `/usr/src/busybox-1.00-pre7/_install/bin` dizinindeki herşeyi `/a3/ramdisk/initrd/bin` dizinine kopyalayın. `/usr/src/busybox-1.00-pre7/_install/` atındaki diğer dizinlerin içeriğini de `/a3/ramdisk/initrd/bin` dizinine kopyalayın. (Kopyalama için **mc**'den yararlanmıyorsanız, sembolik bağları kopyalarken **cp** komutunu **-d** seçeneği ile kullanmalısınız.) `/usr/src/busybox-1.00-pre7/_install/` dizini altındaki **linuxrc**'yi biz kullanmayacağız. Biz onu zaten yazmıştık. `/a3/ramdisk/initrd/bin` dizinine girdiğinizde **mc** kullanıyorsanız, bazı sembolik bağ dosyalarının hedeflerini bulamadıklarını göreceksiniz. Onların hedeflerini bulmalarını sağlamalısınız. İmleci onlardan birinin üzerine getirip CTRL tuşu basılıyken X ve S tuşladığınızda açılan pencereye **busybox** yazmanız yeterlidir. Hepsi için bu işlemi yapın.

`bin` dizinindeki sembolik bağlar arasında **init**'in de bulunduğunu göreceksiniz. Ama sistemin açılması için **init**'in varlığı yeterli değil. **init** çalıştığında önce `/etc/inittab` dosyasına bakacak ve oradaki satırları yorumlarak sistemi kullanıcı ile etkileşimli çalışacak duruma getirecek. O halde, `etc/inittab` dosyasını hazırlamamız lazım. **busybox** ile gelen **init**'in çalışacağı `inittab` dosyasının içeriği, (B18) kılavuz sayfasında anlatılan içerikten farklı. Bu konuda daha ayrıntılı bilgiyi `file:/usr/src/busybox-1.00-pre7/docs/BusyBox.html#item_init` adresinde bulabilirsiniz. Kullanacağımız `inittab` dosyası:

```

::sysinit:/etc/init.d/baslat

tty1::askfirst:/bin/busybox sh
tty2::askfirst:/bin/busybox sh

::ctrlaltdel:/bin/reboot -d 0
::restart:/sbin/init

```

Burada ilk satırda, sistemi ilklendirecek betiğin yerini belirttim. **init** bu satırı ve orada belirtilen betiği icra ettikten sonra diğer satırlara geçecek. Sonraki iki satırda iki konsolumuz (**tty1** ve **tty2**) olacağını belirttim. Bu konsolları etkinleştirmeden önce onay almasını istedim. Son iki satırda ise **CTRL-ALT-DEL** tuşlandığında ne işlem yapacağını ve yeniden başlatma durumunda ne yapacağını belirttim. Sistemi kapatmak için birşey belirtmek gerekmiyor. Konsolda komut olarak **poweroff** verildiğinde **busybox** sistemi sonlandırıyor. Normalde sistemi kapatması lazım ama bunun için ek modüller gerekiyor. Bu lükse ihtiyacınız varsa siz eklersiniz.

İki konsol belirttim. İlk konsolda açılış iletilerine bakmak ve bir takım komutlar orada çalıştırılabilsin istedim. İkinci konsolu daha sonra **X**'i çalıştırmak için kullanacağız. Şimdilik o da ilk konsol gibi bir konsol.

Şimdi **inittab**'ın ilk satırında belirttiğimiz betiği hazırlamalıyız. Betik:

```

#!/bin/sh
# Bu betik sistemi ilklendirir

PATH="/bin"

umask 022

# Biraz yer acalim
rmdir /lost+found

# Saatimiz dogru kalsin
hwclock -s

echo "lo arayuzu etkinlestiriliyor"
ip link set lo up
ip addr add 127.0.0.1 dev lo
ip route add 127.0.0.0/8 dev lo

exit 0

```

Çok basit bir betik. **rmdir /lost+found** satırında, ramdisk üzerinde dosya sistemini oluşturulurken **/lost+found** isminde bir dizin oluşur. Bu dizin bize lazım olmadığından yerimizi işgal etmesin diye siliyoruz. **hwclock -s** satırıyla çekirdeğin, sistem saatini donanım saatıyla eşzamanlamasını söylüyoruz. Böylece **date** komutu bize doğru zamanı gösterecek. Son olarak da sistemin yerel IP adresi ile ilgili yapılandırmayı gerçekleştiriyoruz.

Yalnız **busybox**'ın çalışması için bazı kütüphane dosyalarına ihtiyacı var. Bunların neler olduğunu öğrenmek için **ldd** komutunu kullanacağız:

```

# ldd /a3/ramdisk/initrd/bin/busybox
    libc.so.6 => /lib/i686/libc.so.6 (0x40030000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

```

Bu çıktıya göre **libc.so.6** ve **ld-linux.so.2** dosyalarını **lib** dizinimize yerleştirmemiz gerekli. Sistemin **/lib** dizine baktığımızda bu dosyaların sırasıyla **libc-2.3.2.so** ve **ld-2.3.2.so** dosyalarına sembolik bağ olduklarını görüyoruz. Yani 4 dosyayı da bizim **lib** dizinimize kopyalamamız lazım:


```
# cp -d /lib/libc.so.6 /a3/ramdisk/initrd/lib/
# cp /lib/libc-2.3.2.so /a3/ramdisk/initrd/lib/
# cp -d /lib/ld-linux.so.2 /a3/ramdisk/initrd/lib/
# cp /lib/ld-2.3.2.so /a3/ramdisk/initrd/lib/
```

Birinci ve üçüncü satırdaki `-d` seçeneğine dikkat edin. Dosya sistemimiz artık hazır, deneyebiliriz. `/a3/ramdisk/initrdyap` betiğini çalıştırıp `/export/netlogon/autoboot/initrd.img` dosyasını oluşturalım:

```
# /a3/ramdisk/initrdyap
```

`/export/netlogon/autoboot` dizininde bir önceki bölümde derlediğimiz çekirdeği yerleştirmiştik. Bu dizine Linux'u kurduğunuz CD'lerin ilkinde bulunan `loadlin.exe` dosyasını da kopyalayın. CD'yi CDRÖM sürücünüze yerleştirin ve `mount /mnt/cdrom` komutunu verin. CD'nin içeriğini `/mnt/cdrom` dizini altında bulacaksınız. `mc` ile CD'nin dizinlerinde `loadlin.exe` dosyasını bulun ve `F5` ile `/export/netlogon/autoboot` dizine kopyalayın. `/mnt/cdrom` dizininde çıkın ve `umount /mnt/cdrom` komutunu verdikten sonra CD'yi yuvasından çıkarın.

Şimdi bu 3 dosyadan yemek yapacağız. `/export/netlogon/autoboot` dizininde `linux.bat` ismiyle bir dosya açın (`touch linuxbat`) ve içine aşağıdaki satırı yazın:

```
loadlin autoboot\vmlinux initrd=autoboot\initrd.img ramdisk_size=12000
```

Bir üst dizine geçin ve `linuxyukle.bat` isminde bir dosya açın ve içine aşağıdakileri yazın.

```
net use z: \\NILGUN\netlogon /y
copy z:\autoboot\vmlinux c:\autoboot
copy z:\autoboot\initrd.img c:\autoboot
copy z:\autoboot\loadlin.exe c:\
copy z:\autoboot\linux.bat c:\
```

İlk satırdaki `\\NILGUN\` yerine makinalarınıza verdiğiniz ismi yazmalısınız. Benim için bu isim `nilgun.buguner.home` olduğundan `NILGUN` yazdım. Ayrıca, bu dosyayı windows altında çalıştıracığımız için dosyayı DOS dosya biçimine dönüştürmemiz ve dosyanın çalıştırılabilir bir dosya olmasını sağlamamız lazım. Bunun için aşağıdaki komutları çalıştırın:

```
# unix2dos linuxyukle.bat
# chmod +x linuxyukle.bat
```

Linux'da her satır LF (10 numaralı karakter)ile biterken, DOS'da CRLF (13 ve 10 numaralı karakterler) ile biter. Bu komut satır sonlarındaki LF'leri CRLF ile değiştirir.

Sunucu makina üzerindeki işimiz bitti. İstemci makinaya geçin ve Ağ Komşularında `netlogon` dizinine girin ve `linuxyukle.bat` dosyasını çalıştırın. Bu betiği çalıştırdığınızda `\\NILGUN\netlogon`'un karşılığı olarak `Z:` diski oluşacak ve `Z:` diskindeki dosyalar `C:` diskine kopyalanacak.

Şimdi windows masaüstünde bir kısayol oluşturacağız. `C:\linux.bat` dosyasının üzerindeyken farenin sağtuşu ile açılan menüden kısayol oluştur'u seçin ve oluşan kısayolu masaüstüne taşıyıp ismini `LINUX OL` olarak değiştirin. Simgenin üzerine sağ tıklayıp Özellikler'e girin. Program sekmesinde Gelişmiş düğmesine tıklayın. 2. ve 3. seçenekleri seçili hale getirin. MS-DOS kipi başlıklı bölümde, Mevcut MD-DOS ayarlarını kullan seçeneğini seçil duruma, MS-DOS kipine girerken uyar seçeneğini ise seçilmemiş duruma getirin ve Tamam'a tıklayıp çıkın. Şimdi sıra `config.sys` dosyasında, `C:\config.sys` dosyasında aşağıdakilerin bulunmasını sağlayın (bu dosyalar sizin sisteminizde farklı yerdeyse ona göre düzeltin):

```
DEVICE=C:\WINDOWS\HIMEM.SYS
```

```
DEVICE=C:\WINDOWS\EMM386.EXE
DOS=HIGH,UMB
```

Bundan sonra windows'u yeniden başlatın ve windows açılınca **LINUX OL** simgesine tıklayıp Linux sistemini başlatın. Ekran kararacak, önce çekirdek, ardından **initrd.img** yüklenecek. Çekirdek çalışacak, **init**'i başlatacak ve en son Enter'a basıp konsola düşeceksiniz. **ifconfig**, **ls**, **date**, **df** gibi bazı komutları deneyin. Bu konsolda yapabileceklerimiz bu gibi komut satırından bir takım komutlar girmekle sınırlı. Sistemi kapatmak için **halt** veya **poweroff** komutlarını girebileceğiniz gibi, makinaryı doğrudan da kapatabilirsiniz. Çünkü sistemin çalışması için herhangi bir sabit diski bağlamadık. Kaybedilecek ya da bozulacak bir bilgi yok. Bundan sonraki adım, iki makinaryı birbirine bağlamak.

3.4. Ağa Bağlanan Linux Terminali

Bu adımda, Linux terminalimizin DHCP sunucusundan IP adresi almasını, yani sunucu makinarya bağlanmasını sağlayacağız. Bunu yapmadan önce, istemci makina üzerinde bulunan ağ kartı için hangi çekirdek modülünün gerektiğini öğrenmemiz lazım. Bunu yapmışken, X yapılandırması ile ilgili dosyaları da alacağız. Nasıl mı? Hani belgenin başında elinizin altında bir Knoppix CD'si olsun demiştim. Şimdi onu kullanacağız. Knoppix, tamamen CD'den başlatılıp, sabit diske kurulmadan doğrudan CD'den çalışan bir Linux dağıtımdır. Eğer bir ağ bulur ve bu ağda da bir DHCP sunucusu algılasa kendine bir IP adresi alıp ağa bağlanır. Biz bu özelliğinden yararlanarak, onun çalışmak için oluşturduğu yapılanmayla ilgili dosyaları bu ağ bağlantısı üzerinden sunucumuza aktarabiliriz. Bunu yapabilmek için önce sunucu makina üzerinde SSH sunucusunu başlatmamız lazım. SSH bağlantısı kuracağız, çünkü **telnet** öntanımlı ayarlarıyla **root** kullanıcısının sisteme bağlanmasına izin vermez. SSH için böyle bir sorun yok. Sunucu makina üzerinde şu komutları girin:

```
# /sbin/chkconfig sshd on
# /sbin/service sshd start
```

Böylece sshd'nin her sistem açılışında çalışmasını da sağlamış olduk. İstemci makinanın CD sürücüsündeki Knoppix CD'sinden açılmasını sağlayın. İstemci makina benimki gibi yavaş bir makinarysa (P133) Knoppix'in açılması biraz zahmetli olacaktır. Böylece neden bir Knoppix CD'si alıp kullanın demediğimi daha iyi anlayacaksınız. Bizim oluşturacağımız sistem Knoppix'e göre sürat rekoru kırarak.

Knoppix'in masaüstü açıldığında **CTRL-ALT-F1** tuşu ile konsola düşün. Bu konsol bizim konsol gibi, yani parola filan sormuyor ve sistem root yetkileri ile kullanılıyor. **mc** var, isterseniz çalıştırın. Şu komutu girin:

```
# scp /proc/modules root@10.10.10.1:/a3/ramdisk/
```

Bu komuttan sonra **scp** size parola soracak. Sunucu makinadaki **root** kullanıcısı için kullandığınız parolayı girin. **/proc/modules** dosyası sunucu makina üzerindeki **/a3/ramdisk** dizinine kopyalanacak. **XF86Config** dosyası ile **XFree86.0.log** dosyasını da alalım:

```
# scp /etc/X11/XF86Config root@10.10.10.1:/a3/ramdisk/
# scp /var/log/XFree86.0.log root@10.10.10.1:/a3/ramdisk/
```

Bu komutları girmeden önce **/etc/X11** ve **/var/log** dizinlerine bakmayı ihmal etmeyin. Dosyaların isimleri biraz farklı olabilir, bu takdirde komutlardaki dosya isimlerini değiştirmelisiniz. Bu işlemden sonra Knoppix ile işimiz bitti. Bize gerekli bilgileri topladık. Sunucu makinarya geçip, yeni sistemi hazırlayabiliriz.

Knoppix'den kopyaladığımız **modules** dosyası, istemci makinadaki donanım için gerekli çekirdek modüllerini içerir. Bunlardan biri de ağ kartını sisteme tanıtan modüldür. **modules** dosyasındaki modül isimlerini makinanızdaki ağ kartları ile ilgili modül dosyalarının (**/lib/modules/2.4.22-x/kernel/drivers/net** dizinindeki modüller) isimleriyle karşılaştırın. En az bir eşleşme bulacaksınız, bazı modüller bir ek modüle daha ihtiyaç duyar, örneğin **8139too.o** modülü **mii.o** modülüne ihtiyaç duyar ve **modules** dosyasında bir modül bir başka modüle bağımlıysa bu bilgi dosyada vardır. Bunu göz önüne alarak **modules** dosyasını önce bir

inceleyin. Ağ kartı ilk algılanan donanımlardan biri olduğundan onunla ilgili modül ya da modüller **modules** dosyasının son satırlarındadır. Dosyayı sondan başa tararsanız işiniz çabuk biter.

Bu modül benim istemci makinam için **dmfe.o** ve ek bir modül gerektirmiyor. **/lib/modules/2.4.22-x/kernel/drivers/net/dmfe.o** dosyasını **/a3/ramdisk/initrd/lib/modules/2.4.22-x/kernel/drivers/net/** dizinine kopyalıyorum, siz de kendi modül ya da modülleriniz için aynı işlemi yapın:

```
# cp /lib/modules/2.4.22-x/kernel/drivers/net/dmfe.o \  
> /a3/ramdisk/initrd/lib/modules/2.4.22-x/kernel/drivers/net/
```

Ağ kartının çekirdeğe tanıtılmasını sağlayacak modülü sistemimize yerleştirdiğimize göre sistemi ağa bağlayacak işlemlere başlayabiliriz. **/etc/init.d/baslat** dosyasını aşağıdaki biçime getirin:

```
#!/bin/sh  
# Bu betik sistemi ilklendirir  
  
PATH="/bin"  
  
umask 022  
  
# Biraz yer acalim  
rmdir /lost+found  
  
# modulleri yoklarken cekirdek birsey basmasin  
echo "0" > /proc/sys/kernel/printk  
depmod -a  
modprobe -a -q *  
  
# Saatimiz dogru kalsin  
hwclock -s  
  
echo "lo arayuzu etkinlestiriliyor"  
ip link set lo up  
ip addr add 127.0.0.1 dev lo  
ip route add 127.0.0.0/8 dev lo  
  
echo "eth0 arayuzu etkinlestiriliyor"  
udhcpc -i eth0 -s /etc/init.d/dhcpc.sh  
  
exit 0
```

Önceki **baslat** betiğiyle karşılaştırıldığında hangi satırların eklendiğini gördünüz. **depmod** komutuyla modül bağımlılıkları tespit ediliyor ve hemen altındaki **modprobe** komutuyla gerekli modüller çekirdeğe ekleniyor. **depmod** ve **modprobe** komutlarını **bin** dizinimize eklemeliyiz. Ayrıca bir **/etc/modules.conf** dosyası oluşturup ağ kartı için gereken modülü bu dosyada belirtmeliyiz.

Dosyanın sonundaki **udhcpc** komutu ise **busybox** içindeki yerleşik komutlardan biri. DHCP sunucusundan ağ kartımız için IP adresinin alınmasını bu komut gerçekleştiriyor. Komut parametrelerinde **-i** seçeneği ile ağ arayüzünün ismini, **-s** seçeneği ile de komutun dönemsel olarak çalıştıracağı betiğin yerini belirtiyoruz. Bu betiği de biz yazacağız.

Şimdi bu işlemleri sırayla yapalım. **depmod** ve **modprobe** komutlarını sistemde bulunduğu yeri **whereis** ile tespit ettikten sonra bu dizine girip bu komutların dosyalarını buluyoruz. **modprobe** genelde **insmod** dosyasına bir sembolik bağıdır. **insmod** dosyasına sembolik bağ olarak çalışan başka komutlar da var. Hepsini **initrd/bin** dizinine kopyalıyoruz:

```
# cd /a3/ramdisk/initrd/bin
# cp /sbin/depmod ./
# cp /sbin/insmod ./
# cp -d /sbin/kallsyms ./
# cp -d /sbin/ksyms ./
# cp -d /sbin/lsmmod ./
# cp -d /sbin/modprobe ./
```

Sembolik bağları kopyalarken `-d` seçeneğini kullandığımıza dikkat edin. Bu seçeneği kullanmazsanız bağ dosyası değil, bağın hedefindeki dosya kopyalanır ki, bu bizim istemediğimiz bir durum.

depmod ve **insmod** komutlarının bağımlı olduğu kütüphanelerin neler olduğuna da bakmamız lazım:

```
# ldd ./insmod
    libc.so.6 => /lib/i686/libc.so.6 (0x40030000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
# ldd ./depmod
    libc.so.6 => /lib/i686/libc.so.6 (0x40030000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Bu dosyaları **busybox** için sistemimize dahi etmiştik. Farklı bir kütüphaneye ihtiyaç duymadıklarından birşey yapmamıza gerek yok.

Şimdi `etc` dizinine geçip oradaki işlemleri yapalım. Önce `etc` dizininde `modules.conf` dosyasını oluşturun ve içine aşağıdaki satırı sizin ağ kartınızın modül ismini `.o` uzantısız olarak içerecek şekilde yazın (benimki `dmfe`):

```
alias eth0 dmfe
```

`init.d` dizinine geçip `dhcpc.sh` dosyasını oluşturacağız ve dosyayı çalıştırılabilir yapacağız:

```
# touch dhcpc.sh
# chmod +x dhcpc.sh
```

Dosyanın içine aşağıdaki satırları kopyalayın:

```
#!/bin/sh

RESOLV_CONF="/etc/resolv.conf"
[ -n "$broadcast" ] && BROADCAST="broadcast $broadcast"
[ -n "$subnet" ] && NETMASK="netmask $subnet"

case "$1" in
    deconfig)
        ifconfig $interface 0.0.0.0
        ;;

    renew|bound)
        ifconfig $interface $ip $BROADCAST $NETMASK

        if [ -n "$router" ] ; then
            while ip route del default scope global nexthop dev $interface ; do
                :
            done 2>/dev/null

            for i in $router ; do
                ip route add default via $i dev $interface
            done
        fi
```

```

echo -n > $RESOLV_CONF
[ -n "$domain" ] && echo search $domain >> $RESOLV_CONF

for i in $dns ; do
    echo nameserver $i >> $RESOLV_CONF
done
;;
esac

exit 0

```

Bu betik tek başına çalışmaz, betiği **udhcpd** komutu bazı parametreler aktararak çalıştırır. Bu konu hakkında daha fazla bilgi almak için **busybox** paketindeki **networking/udhcp** dizinindeki README dosyalarındaki açıklamalara bakabilirsiniz.

initrdyap betiğini çalıştırıp **initrd.img** dosyasını oluşturduktan sonra istemci makineye geçip, **netlogon** dizinindeki betiği çalıştırıp dosyaları windows'a aktarın ve **LINUX OL** simgesine tıklayıp yaptıklarınızı deneyin.

İstemci makinedeki Linux sistemi açıldığında **ifconfig** komutunu vererek **eth0** yapılandırmasını görebilir. Sunucu makineye **ping** atabilir (**ping 10.10.10.1**), böylece ağ bağlantınızı sınayabilirsiniz.

İstemci makinenin Linux ile ağa bağlanmasını gerçekleştirdiğimize göre son adıma geçebilir, X terminalimizi oluşturmaya başlayabiliriz.

3.5. X Terminalinin Oluşturulması

Knoppix'i istemci makineye çalıştırdığımızda X yapılandırmasını içeren iki dosya kopyalamıştık. Biri X çalıştırıldığında X'in okuyup kendini yapılandığı **XF86Config** dosyası, diğeri ise X çalıştığında yaptığı işlemleri günlüklediği **XFree86.0.log** dosyası. **XF86Config** dosyasını biraz değiştirip kullanacağız. **XFree86.0.log** dosyasından ise X çalıştığında hangi dosyaları yüklediğini öğrenip bu dosyaları sistemimize yerleştireceğiz ki, bizim sistemimizde çalıştırıldığında bu dosyaları bulabilsin. Ancak bu dosyalarda gereken bütün dosyalara ilişkin bilgi yok. Bazılarını da X çalıştığında verdiği hatalardan öğrendim. Siz de X çalıştırıldığında benzer hatalar alabilirsiniz. En azından benim aldığım hataları almayacaksınız. Çünkü ben en azından bu sistemi daha önce çalıştırdığım için gerekli bazı dosyaları biliyorum. Hemen ürkmeyin öyle; siz bir hata yapmamışsanız, büyük ihtimalle hiç hata almayacaksınız.

X ile ilgili herşey başlıca iki dizin altında bulunur: **/usr/X11R6** ve **/etc/X11**. Bu iki dizinin yeri bütün dağıtımlarda aynıdır. Bu benzerliği bizde kullanıyoruz; bir farkla: **/usr/X11R6** dizinin içeriğine baktığımızda kök dizine benzediğini görüyoruz. Bu benzerlikten yararlanarak **/usr/X11R6** dizinini kök dizini gösteren bir bağ dosyası yapabiliyoruz. Böylece büyükçe ayrı bir dizin ağacı oluşturmak yerine kök dizinindeki dizin yapısından yararlanabiliyoruz. **initrd** dizinin içeriğini incelediğinizde bunun burada anlattığım şekilde oluşturulmuş olduğunu göreceksiniz. Yani, normal bir sistemde **/usr/X11R6** altında olan herşeyi **initrd** dizini içindeki dizinlere yerleştireceğiz.

Önce **/usr/X11R6/bin/X** sembolik bağı ile bu bağı hedefindeki dosyayı kendi sistemimize yerleştirelim:

```

# cp -d /usr/X11R6/bin/X /a3/ramdisk/initrd/bin/
# cp /usr/X11R6/bin/XFree86 /a3/ramdisk/initrd/bin/

```

Bunun ardından **XFree86**'nın hangi kütüphanelere bağımlı olduğunu tespit edip bunları da sistemimize yerleştirelim:

```

# ldd /a3/ramdisk/initrd/bin/XFree86
libz.so.1 => /usr/lib/libz.so.1 (0x40030000)

```

```
libm.so.6 => /lib/i686/libm.so.6 (0x4003e000)
libpam.so.0 => /lib/libpam.so.0 (0x40060000)
libdl.so.2 => /lib/libdl.so.2 (0x40068000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x4006b000)
libc.so.6 => /lib/i686/libc.so.6 (0x4006e000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Bu kütüphanelerden son ikisini zaten sistemimize daha önce kopyalamıştık. Diğerlerini de kopyalayalım:

```
# cd /a3/ramdisk/initrd/lib
# cp -d /usr/lib/libz.so.1 ./
# cp -d /lib/libm.so.6 ./
# cp -d /lib/libpam.so.0 ./
# cp -d /lib/libdl.so.2 ./
# cp -d /lib/libpam_misc.so.0 ./
```

Sembolik bağları kopyladık. Bu sembolik bağların hedefindeki dosyalar hatta bu sembolik bağlar bile sizin sisteminizde farklı olabilir. Ben benimkileri aşağıdaki gibi kopyladım, siz de sizinkilerin hedeflerindeki dosyaları kopyalayın:

```
# cp /usr/lib/libz.so.1.1.4 ./
# cp /lib/libm-2.3.2.so ./
# cp /lib/libpam.so.0.77 ./
# cp /lib/libdl-2.3.2.so ./
# cp /lib/libpam_misc.so.0.77 ./
```

X'in çalışması için gerekli en temel koşulları sağladık. X çalıştıktan sonra sistem yapılandırmasını öğrenmek için `/etc/X11/XF86Config` dosyasına bakar ve oradaki verilere göre bazı kütüphaneleri ve dosyaları yükler. Hangi kütüphaneleri yüklediğini ise günlük kayıt dosyası olan `/var/log/XFree86.0.log` dosyasına yazar. Bu dosyaların birer kopyasını Knoppix sayesinde edinmiştik. Önce Knoppix için anlamlı olan `XF86Config` dosyasını kendi sistemimiz için anlamlı olan biçime getirmeliyiz. Bunun için önce "Files" bölümünde `FontPath` satırını düzenleyeceğiz:

```
Section "Files"
    RgbPath      "/usr/X11R6/lib/X11/rgb"
    FontPath     "tcp/10.10.10.1:7100"
EndSection
```

Dosyanın görünümü sizinkinde farklı olabilir ama işe yarayan satırlar, bunlar. `FontPath` satırını yukarıdaki gibi düzenleyin. Bu satırdaki `10.10.10.1`, **xfs** sunucusunun çalıştığı makinanın IP adresi olmalı. Bu satır sayesinde X çalıştırıldığında gerekli yazıtiplerini bu adresteki **xfs** sunucusundan yükleyecek.

İkinci düzenleme klavye ile ilgili. Şüphesiz Türkçe karakterleri kullanmak isteyeceksiniz. Klavye ile ilgili bölüm fazlalıkları temizlenmiş olarak aşağıdaki gibi olmalı:

```
Section "InputDevice"
    Identifier   "Keyboard0"
    Driver       "keyboard"
    Option       "XkbRules" "xfree86"
    Option       "XkbModel" "pc105"
    Option       "XkbLayout" "tr_q,tr_f"
    Option       "XkbOptions" "caps:shift,grp:shift_toggle"
EndSection
```

Bu yapılandırma, her iki Shift tuşuna birlikte basıldığında Q ve F klavyeleri arasında geçiş yapar.



Uyarı

Bu yapılandırma standart X ile uyumlu değildir. Sistem bu özel duruma göre yapılandırılacaktır (Türkçe X klavyesinin yazarından bir hediye).

Bu düzenlemeleri yaptıktan sonra dosyayı `/a3/ramdisk/initrd/etc/X11` dizini altına kopyalayın.

İstemci makinalardaki fare, bir PS/2 fare ise `dev` dizinindeki `mouse` sembolik bağı `psaux` dosyasını gösterecek duruma getirmelisiniz (Normalde birinci seri portu gösterir).

Şimdi sıra `XFree86.0.log` dosyasına bakarak gerekli kütüphaneleri sistemimize yerleştirmeye geldi. `XFree86.0.log` dosyasına baktığınızda (II) `Loading` ile başlayan aşağıdakine benzer satırlar göreceksiniz:

```
(II) Loading /usr/X11R6/lib/modules/fonts/libbitmap.a
```

Bu satırlar sistemimize kopyalamamız gereken kütüphane dosyalarını belirtir. Bu satırlara bakarak, X'in aynı dosya yolunda aynı dosyayı bulabilmesini sağlamamız gerekir. Yukarıdaki dosya için bu işlem şöyle yapılmalıdır:

```
# cp /usr/X11R6/lib/modules/fonts/libbitmap.a /a3/ramdisk/initrd/lib/modules/fonts/
```

Bu işlemi benzer şekilde `XFree86.0.log` dosyasındaki diğer kütüphane yükleme satırları için yapın. `XFree86.0.log` dosyasından önce bu satırları seçiyorum:

```
# cat XFree86.0.log | grep "Loading /"
(II) Loading /usr/X11R6/lib/modules/fonts/libbitmap.a
(II) Loading /usr/X11R6/lib/modules/libpcidata.a
(II) Loading /usr/X11R6/lib/modules/extensions/libdbe.a
(II) Loading /usr/X11R6/lib/modules/extensions/libextmod.a
(II) Loading /usr/X11R6/lib/modules/linux/libfbdevhw.a
(II) Loading /usr/X11R6/lib/modules/extensions/libglx.a
(II) Loading /usr/X11R6/lib/modules/extensions/libGLcore.a
(II) Loading /usr/X11R6/lib/modules/extensions/librecord.a
(II) Loading /usr/X11R6/lib/modules/fonts/libfreetype.a
(II) Loading /usr/X11R6/lib/modules/fonts/libtype1.a
(II) Loading /usr/X11R6/lib/modules/extensions/libdri.a
(II) Loading /usr/X11R6/lib/modules/linux/libdrm.a
(II) Loading /usr/X11R6/lib/modules/drivers/s3vgrw_drv.o
(II) Loading /usr/X11R6/lib/modules/input/mouse_drv.o
(II) Loading /usr/X11R6/lib/modules/libvgahw.a
(II) Loading /usr/X11R6/lib/modules/libvbe.a
(II) Loading /usr/X11R6/lib/modules/linux/libint10.a
(II) Loading /usr/X11R6/lib/modules/libddc.a
(II) Loading /usr/X11R6/lib/modules/libi2c.a
(II) Loading /usr/X11R6/lib/modules/libfb.a
(II) Loading /usr/X11R6/lib/modules/libxaa.a
(II) Loading /usr/X11R6/lib/modules/libramdac.a
```

Bu liste büyük ihtimalle sizin için farklı olacaktır. Şimdi bu listeye göre bu dosyaları ramdisk sistemine kopyalayacağım:

```
# cd /a3/ramdisk/initrd/lib/modules
# cp /usr/X11R6/lib/modules/fonts/libbitmap.a ./fonts/
# cp /usr/X11R6/lib/modules/libpcidata.a ./
# cp /usr/X11R6/lib/modules/extensions/libdbe.a ./extensions/
# cp /usr/X11R6/lib/modules/extensions/libextmod.a ./extensions/
# cp /usr/X11R6/lib/modules/linux/libfbdevhw.a ./linux/
# cp /usr/X11R6/lib/modules/extensions/libglx.a ./extensions/
# cp /usr/X11R6/lib/modules/extensions/libGLcore.a ./extensions/
# cp /usr/X11R6/lib/modules/extensions/librecord.a ./extensions/
# cp /usr/X11R6/lib/modules/fonts/libfreetype.a ./fonts/
```



```
# cp /usr/X11R6/lib/modules/fonts/librecord.a ./fonts/
# cp /usr/X11R6/lib/modules/extensions/libdri.a ./extensions/
# cp /usr/X11R6/lib/modules/linux/libdrm.a ./linux/
# cp /usr/X11R6/lib/modules/input/mouse_drv.o ./input/
# cp /usr/X11R6/lib/modules/libvgahw.a ./
# cp /usr/X11R6/lib/modules/libvbe.a ./
# cp /usr/X11R6/lib/modules/linux/libint10.a ./linux/
# cp /usr/X11R6/lib/modules/libddc.a ./
# cp /usr/X11R6/lib/modules/libi2c.a ./
# cp /usr/X11R6/lib/modules/libfb.a ./
# cp /usr/X11R6/lib/modules/libxaa.a ./
# cp /usr/X11R6/lib/modules/libramdac.a ./
```

X bunlardan başka klavye için gereken bazı dosyaları da arar ve bunlar günlük dosyasında görünmezler. Şimdi bu dosyaları yükleyeceğiz. Yalnız bu dosyaları sadece türkçe klavye yüklenecek biçimde küçülteceğiz. Bu işlemleri burada anlattığım gibi aynen yapmalısınız. Önce gerekli dizinleri oluşturun:

```
# cd /a3/ramdisk/initrd/lib/X11/xkb
# mkdir compat geometry keycodes keymap rules semantics symbols symbols/pc types
# ln -s ../../../../var/lib/xkb compiled
```

xkbcomp dosyasını kopyalayalım ve hangi kütüphanelere ihtiyaç duyduğuna bakalım:

```
# cp /usr/X11R6/lib/X11/xkb/xkbcomp ./
# ldd ./xkbcomp
libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x40030000)
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x4003e000)
libc.so.6 => /lib/i686/libc.so.6 (0x4011d000)
libdl.so.2 => /lib/libdl.so.2 (0x40258000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Son 3 kütüphane zaten var. Diğerlerini kopyalamamız lazım:

```
# cd /a3/ramdisk/initrd/lib
# cp -d /usr/X11R6/lib/libXext.so.6 ./
# cp -d /usr/X11R6/lib/libX11.so.6 ./
# cp /usr/X11R6/lib/libXext.so.6.4 ./
# cp /usr/X11R6/lib/libX11.so.6.2 ./
# cd /a3/ramdisk/initrd/lib/X11/xkb
```

Bu dizinde 6 tane de dosya bulunacak. Bu dosyalar ve içerikleri aşağıdaki gibi olmalı. Ben basitçe dosyaları **cat** komutu ile aldığım konsol çıktılarını aşağıya kopyaladım, siz bunları dosya haline getirin. (Bu dosyalar sistem-inizde de vardır ama genel amaçlı olduklarından çok yer kaplarlar, bu dosyalar fazlalıklarından temizlenmiştir. Ama, sisteminizdeki dosyaları değiştirmeyin, sakın. Sabit diske kurulmuş bir sistemin böyle bir yer kazancına ihtiyacı yoktur.):

```
# cat compat.dir
-dp----- accessx(basic)
--p----- accessx(full)
-d----- basic(basic)
-d----- complete(complete)
-d----- default(default)
-d----- group_led(group_led)
-dp----- iso9995(default)
h-p----- keypad(overlay)
h-p----- keypad(overlay1)
h-p----- keypad(overlay2)
--p----- leds(scroll)
```

```
--p----- ----- leds (num)
--p----- ----- leds (caps)
-dp----- ----- misc (misc)
-dp----- ----- mousekeys (mousekeys)
-dp----- ----- norepeat (norepeat)
-dp----- ----- pc (pc)
-dp----- ----- xfree86 (basic)
-d----- ----- xtest (xtest)
# cat geometry.dir
-d----- ----- pc (pc101)
----- ----- pc (pc102)
----- ----- pc (pc104)
----- ----- pc (pc105)
# cat keycodes.dir
-d----- ----- xfree86 (xfree86)
----- ----- xfree86 (basic)
----- ----- xfree86 (102)
-d----- ----- xfree86.xkb-broken-capslock-fix-CVS11006 (xfree86)
----- ----- xfree86.xkb-broken-capslock-fix-CVS11006 (basic)
----- ----- xfree86.xkb-broken-capslock-fix-CVS11006 (102)
# cat keymap.dir
----- ----- xfree86 (tr_q)
----- ----- xfree86 (tr_f)
# cat symbols.dir
--p----- -m----- altwin (meta_alt)
--p----- -m----- altwin (meta_win)
--p----- -m----- altwin (left_meta_win)
--p----- -m----- altwin (super_win)
--p----- -m----- altwin (hyper_win)
--p----- -m----- altwin (menu)
--p----- -m----- compose (ralt)
--p----- -m----- compose (rwin)
--p----- -m----- compose (menu)
--p----- -m----- ctrl (nocaps)
--p----- -m----- ctrl (swapcaps)
--p----- -m----- ctrl (ctrl_ac)
--p----- -m----- ctrl (ctrl_aa)
--p----- -m----- group (switch)
--p----- -m----- group (lswitch)
--p----- -m----- group (win_switch)
--p----- -m----- group (lwin_switch)
--p----- -m----- group (rwin_switch)
--p----- -m----- group (toggle)
--p----- -m----- group (shift_toggle)
--p----- -m----- group (ctrl_shift_toggle)
--p----- -m----- group (caps_toggle)
--p----- -m----- group (ctrl_alt_toggle)
--p----- -m----- group (alt_shift_toggle)
--p----- -m----- group (menu_toggle)
--p----- -m----- group (lwin_toggle)
--p----- -m----- group (rwin_toggle)
--p----- -m----- group (ctrls_toggle)
--p----- -m----- group (alts_toggle)
--p----- -m----- group (lshift_toggle)
--p----- -m----- group (rshift_toggle)
--p----- -m----- group (lalt_toggle)
--p----- -m----- group (lctrl_toggle)
```

```
--p----- -m----- group(rctrl_toggle)
h-p----- am--g---- iso9995-3(full_common)
--p----- am--g---- iso9995-3(full)
--p----- am--g---- iso9995-3(full101)
h-p----- --k----- keypad(overlay)
h-p----- --k----- keypad(overlay1)
h-p----- --k----- keypad(overlay2)
--p----- -m----- level3(switch)
--p----- -m----- level3(ralt_switch)
--p----- -m----- level3(ralt_switch_multikey)
--p----- -m----- level3(menu_switch)
--p----- -m----- level3(win_switch)
--p----- -m----- level3(lwin_switch)
--p----- -m----- level3(rwin_switch)
h-p----- -m----- lock(shift)
h-p----- -m----- lock(caps)
h-p----- -m----- lock(group)
-dp----- -m----- pc104(altmeta)
----- pc104(compose)
--p----- -m----- pc104(menu)
--p----- --kf----- srvr_ctrl(xfree86)
--p----- --k----- srvr_ctrl(stdkeypad)
--p----- ---f----- srvr_ctrl(fkey2vt)
-dp----- ----- pc/latin(basic)
--p----- ----- pc/latin(type2)
--p----- ----- pc/latin(type3)
--p----- ----- pc/latin(type4)
--p----- ----- pc/latin(nodeadkeys)
--p----- ----- pc/latin(type2_nodeadkeys)
--p----- ----- pc/latin(type3_nodeadkeys)
--p----- ----- pc/latin(type4_nodeadkeys)
h-p----- am----- pc/pc(basic)
----- pc/pc(pc101)
----- pc/pc(pc102)
-d----- pc/pc(pc104)
----- pc/pc(pc105)
----- pc/pc(common)
-dp----- ----- pc/tr_q(basic)
--p----- ----- pc/tr_f(basic)
-dp----- ----- pc/tr.xkb-broken-capslock-fix-CVS11006(basic)
h-p----- am----- pc/pc.xkb-sysreq(basic)
----- pc/pc.xkb-sysreq(pc101)
----- pc/pc.xkb-sysreq(pc102)
-d----- pc/pc.xkb-sysreq(pc104)
----- pc/pc.xkb-sysreq(pc105)
----- pc/pc.xkb-sysreq(common)
--p----- -m----- level3.xkb-broken-capslock-fix-CVS11006(switch)
--p----- -m----- level3.xkb-broken-capslock-fix-CVS11006(menu_switch)
--p----- -m----- level3.xkb-broken-capslock-fix-CVS11006(win_switch)
--p----- -m----- level3.xkb-broken-capslock-fix-CVS11006(lwin_switch)
--p----- -m----- level3.xkb-broken-capslock-fix-CVS11006(rwin_switch)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(switch)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(ralt_switch)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(ralt_switch_multikey)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(menu_switch)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(win_switch)
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092(lwin_switch)
```

```
--p----- -m----- level3.xkb-ru-group-number-fix-CVS11092 (rwin_switch)
# cat types.dir
-d----- ----- basic(basic)
-dp----- ----- cancel(Shift_Cancels_Caps)
--p----- ----- caps(internal)
--p----- ----- caps(internal_nocancel)
--p----- ----- caps(shift)
--p----- ----- caps(shift_nocancel)
-d----- ----- complete(complete)
-d----- ----- default(default)
-dp----- ----- extra(default)
-dp----- ----- iso9995(default)
-dp----- ----- mousekeys(default)
-dp----- ----- nocancel(nocancel)
-dp----- ----- pc(default)
```

compat dizini için aşağıdaki işlemleri yapın:

```
# cp /usr/X11R6/lib/X11/xkb/compat/* ./compat/
# rm -f ./compat/README
```

geometry dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/xkb/geometry/pc ./geometry/
```

keycodes dizini için aşağıdaki işlemleri yapın:

```
# cp /usr/X11R6/lib/X11/xkb/keycode/aliases ./geometry/
# cp /usr/X11R6/lib/X11/xkb/keycode/xfree86 ./geometry/
```

keymap dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/xkb/keymap/xfree86 ./keymap/
```

rules dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/xkb/rules/xfree86 ./rules/
```

semantics dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/xkb/semantics/* ./semantics/
```

symbols dizinine, /usr/X11R6/lib/X11/xkb/symbols/ dizininden altwin, ctrl, iso9995-3, level3, lt_std, pc104, srvr_ctrl, us_group3, compose, group, keypad, lock, ralt, us_group2 ve us_intl dosyalarını kopyalayın.

symbols/pc dizinine, /usr/X11R6/lib/X11/xkb/symbols/pc dizininden latin ve pc dosyalarını kopyalayın. Aşağıdaki 2 dosyayı oluşturun:

```
# cat tr_q
// based on
// Turkish keyboard symbols for PC and xkb
// This file distributed without any expressed or implied warranty
// Nilgün Belma Bugüner <nilgun@superonline.com>, 2004
//
//Default Layout (Turkish Q)
default partial
xkb_symbols "basic" {

    include "pc/latin"
```

```

name[Group1]="Turkish_Q";

key <AE01> { [ 1, exclam, onesuperior, exclamdown ] };
key <AE02> { [ 2, apostrophe, sterling, twosuperior ] };
key <AE03> { [ 3, asciicircum, numbersign, threesuperior ] };
key <AE04> { [ 4, plus, dollar, onequarter ] };
key <AE06> { [ 6, ampersand, threequarters, VoidSymbol ] };
key <AE07> { [ 7, slash, braceleft, VoidSymbol ] };
key <AE08> { [ 8, parenleft, bracketleft, VoidSymbol ] };
key <AE09> { [ 9, parenright, bracketright, plusminus ] };
key <AE10> { [ 0, equal, braceright, degree ] };
key <AE11> { [ asterisk, question, backslash, questiondown ] };
key <AE12> { [ minus, underscore, division, VoidSymbol ] };

key <AD02> { [ w, W, VoidSymbol, VoidSymbol ] };
key <AD03> { [ e, E, EuroSign, VoidSymbol ] };
key <AD05> { [ t, T, trademark, VoidSymbol ] };
key <AD07> { [ u, U, ucircumflex, Ucircumflex ] };
key <AD08> { [ idotless, I, icircumflex, Icircumflex ] };
key <AD09> { [ o, O, ocircumflex, Ocircumflex ] };
key <AD10> { [ p, P, VoidSymbol, VoidSymbol ] };
key <AD11> { [ gbreve, Gbreve ] };
key <AD12> { [ udiaeresis, Udiaeresis, asciitilde, dead_macron ] };

key <AC01> { [ a, A, acircumflex, Acircumflex ] };
key <AC02> { [ s, S, section, VoidSymbol ] };
key <AC03> { [ d, D, VoidSymbol, VoidSymbol ] };
key <AC04> { [ f, F, ordfeminine, VoidSymbol ] };
key <AC05> { [ g, G, VoidSymbol, VoidSymbol ] };
key <AC06> { [ h, H, VoidSymbol, VoidSymbol ] };
key <AC08> { [ k, K, VoidSymbol, VoidSymbol ] };
key <AC09> { [ l, L, VoidSymbol, VoidSymbol ] };
key <AC10> { [ scedilla, Scedilla, acute, dead_acute ] };
key <AC11> { [ i, Iabovedot, apostrophe, dead_caron ] };
key <TLDE> { [ quotedbl, backslash, plusminus, degree ] };

key <BKSL> { [ comma, semicolon, grave, dead_grave ] };
key <LSGT> { [ less, greater, bar, brokenbar ] };
key <AB08> { [ odiaeresis, Odiaeresis, multiply, VoidSymbol ] };
key <AB09> { [ ccedilla, Ccedilla, periodcentered, division ] };
key <AB10> { [ period, colon, dead_abovedot, dead_abovedot ] };
key <SPCE> { [ space, space, nobreakspace, nobreakspace ] };

include "level3(ralt_switch_multikey)"
modifier_map Mod5 { <RALT> };
};
# cat tr_f
// based on
// Turkish keyboard symbols for PC and xkb
// This file distributed without any expressed or implied warranty
// Nilgün Belma Bugüner <nilgun@superonline.com>, 2004
//
//Default Layout (Turkish F)
default partial
xkb_symbols "basic" {

```

```

include "pc/latin"

name[Group1]="Turkish_F";

key <AE01> { [ 1, exclam, onesuperior, exclamdown ] };
key <AE02> { [ 2, quotedbl, twosuperior, VoidSymbol ] };
key <AE03> { [ 3, asciicircum, numbersign, threesuperior ] };
key <AE04> { [ 4, dollar, onequarter, VoidSymbol ] };
key <AE06> { [ 6, ampersand, threequarters, VoidSymbol ] };
key <AE07> { [ 7, apostrophe, braceleft, VoidSymbol ] };
key <AE08> { [ 8, parenleft, bracketleft, VoidSymbol ] };
key <AE09> { [ 9, parenright, bracketright, plusminus ] };
key <AE10> { [ 0, equal, braceright, degree ] };
key <AE11> { [ slash, question, backslash, questiondown ] };
key <AE12> { [ minus, underscore, division, VoidSymbol ] };

key <AD01> { [ f, F, at, VoidSymbol ] };
key <AD02> { [ g, G, VoidSymbol, VoidSymbol ] };
key <AD03> { [ gbreve, Gbreve, VoidSymbol, VoidSymbol ] };
key <AD04> { [ idotless, I, paragraph, VoidSymbol ] };
key <AD05> { [ o, O, ocircumflex, Ocircumflex ] };
key <AD06> { [ d, D, yen, VoidSymbol ] };
key <AD07> { [ r, R, registered, VoidSymbol ] };
key <AD08> { [ n, N, VoidSymbol, VoidSymbol ] };
key <AD09> { [ h, H, degree, VoidSymbol ] };
key <AD10> { [ p, P, sterling, VoidSymbol ] };
key <AD11> { [ q, Q, dead_diaeresis, dead_abovering ] };
key <AD12> { [ w, W, asciitilde, dead_breve ] };

key <AC01> { [ u, U, ucircumflex, Ucircumflex ] };
key <AC02> { [ i, Iabovedot, icircumflex, Icircumflex ] };
key <AC03> { [ e, E, EuroSign, VoidSymbol ] };
key <AC04> { [ a, A, acircumflex, Acircumflex ] };
key <AC05> { [ udiaeresis, Udiaeresis, ucircumflex, Ucircumflex ] };
key <AC06> { [ t, T, trademark, VoidSymbol ] };
key <AC07> { [ k, K, VoidSymbol, VoidSymbol ] };
key <AC08> { [ m, M, mu, VoidSymbol ] };
key <AC09> { [ l, L, VoidSymbol, VoidSymbol ] };
key <AC10> { [ y, Y, acute, dead_acute ] };
key <AC11> { [ scedilla, Scedilla, numbersign, dead_caron ] };
key <TLDE> { [ plus, asterisk, notsign, plusminus ] };

key <BKSL> { [ x, X, grave, dead_grave ] };
key <AB01> { [ j, J, guillemotleft, VoidSymbol ] };
key <AB02> { [ odiaeresis, Odiaeresis, guillemotright, VoidSymbol ] };
key <AB03> { [ v, V, leftdoublequotemark, VoidSymbol ] };
key <AB04> { [ c, C, cent, copyright ] };
key <AB05> { [ ccedilla, Ccedilla, rightdoublequotemark, VoidSymbol ] };
key <AB06> { [ z, Z, VoidSymbol, VoidSymbol ] };
key <AB07> { [ s, S, section, VoidSymbol ] };
key <AB08> { [ b, B, multiply, nobreakspace ] };
key <AB09> { [ period, colon, division, dead_abovedot ] };
key <AB10> { [ comma, semicolon, periodcentered, VoidSymbol ] };
key <SPCE> { [ space, space, nobreakspace, nobreakspace ] };
key <LSGT> { [ less, greater, bar, brokenbar ] };

include "level3(ralt_switch_multikey)"

```

```
modifier_map Mod5 { <RALT> };
};
```

types dizini için aşağıdaki işlemleri yapın:

```
# cp /usr/X11R6/lib/X11/xkb/types/* ./types/
# rm -f ./types/README
```

Klavye yapılandırması ile ilgili olarak /a3/ramdisk/initrd/lib/X11/locale dizininde de bazı işlemler yapacağız. Önce bize gereken dizinleri açalım:

```
# cd /a3/ramdisk/initrd/lib/X11/locale
# mkdir C en_US.UTF-8 iso8859-9 lib lib/common
```

Bu dizinde 3 dosya oluşturacağız. Bunlar:

```
# cat compose.dir
iso8859-1/Compose      C
iso8859-1/Compose      en_US.ISO8859-1
iso8859-9/Compose      tr_TR.ISO8859-9
#
en_US.UTF-8/Compose     en_US.UTF-8
en_US.UTF-8/Compose     tr_TR.UTF-8
# cat locale.alias
POSIX                  C
POSIX-UTF2             C
C_C.C                  C
C.en                   C
C.iso88591              en_US.ISO8859-1
Cextend                en_US.ISO8859-1
Cextend.en             en_US.ISO8859-1
English_United-States.437 C
en_US                  en_US.ISO8859-1
en_US.88591             en_US.ISO8859-1
en_US.88591.en          en_US.ISO8859-1
en_US.iso88591          en_US.ISO8859-1
en_US.iso885915         en_US.ISO8859-15
en_US.ISO-8859-1        en_US.ISO8859-1
en_US.ISO_8859-1        en_US.ISO8859-1

tr                      tr_TR.ISO8859-9
tr_TR                  tr_TR.ISO8859-9
tr_TR.iso88599          tr_TR.ISO8859-9
tr_TR.ISO-8859-9        tr_TR.ISO8859-9
tr_TR.utf8              tr_TR.UTF-8

tr:                     tr_TR.ISO8859-9
tr_TR:                  tr_TR.ISO8859-9
tr_TR.iso88599:         tr_TR.ISO8859-9
tr_TR.ISO-8859-9:       tr_TR.ISO8859-9
tr_TR.utf8:             tr_TR.UTF-8
# cat locale.dir
C/XLC_LOCALE           C
iso8859-1/XLC_LOCALE    en_US.ISO8859-1
iso8859-9/XLC_LOCALE    tr_TR.ISO8859-9
en_US.UTF-8/XLC_LOCALE  en_US.UTF-8
en_US.UTF-8/XLC_LOCALE  tr_TR.UTF-8
```


C/XLC_LOCALE:	C
iso8859-15/XLC_LOCALE:	en_US.ISO8859-15
iso8859-9/XLC_LOCALE:	tr_TR.ISO8859-9
en_US.UTF-8/XLC_LOCALE:	en_US.UTF-8
en_US.UTF-8/XLC_LOCALE:	tr_TR.UTF-8

C dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/locale/C/* ./C/
```

en_US.UTF-8 dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/locale/en_US.UTF-8/* ./en_US.UTF-8/
```

iso8859-9 dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/locale/iso8859-9/* ./iso8859-9/
```

lib/common dizini için aşağıdaki işlemi yapın:

```
# cp /usr/X11R6/lib/X11/locale/lib/common/* ./lib/common/
```

Bu dosyalardan başka X'in yüklediği bazı dosyalar var. Bunları bulamazsa hata verip çalışmasını durduruyor. Siz böyle bir durumda X terminaliniz çalışmazsa komut satırından **cat /var/log/XFree86.0.log** komutunu verip, hatanın sebebini öğrenip X'in aradığı dosyayı sisteminize eklemelisiniz.

Son işlemlerden olarak, aşağıdaki işlemleri yapın:

```
# cd /a3/ramdisk/initrd/lib/X11
# cp /usr/X11R6/lib/X11/app-defaults/Xmessage ./app-defaults/
# cp /usr/X11R6/lib/X11/Cards ./
# cp /usr/X11R6/lib/X11/Options ./
# cp /usr/X11R6/lib/X11/rgb.txt ./
# cp /usr/X11R6/lib/X11/XErrorDB ./
# cp /usr/X11R6/lib/X11/XKeysymDB ./
# ln -s /etc/X11/XF86Config
# cd /a3/ramdisk/etc/X11
# ln -s ../../bin/XFree86 X
# cp /etc/X11/xserver/SecurityPolicy ./xserver/
```

X ile ilgili olarak sistemin yapılandırılması bitti. Şimdi açılış sırasında X'in yüklenmesini sağlayacak işlemleri yapalım. **etc/inittab** dosyasını aşağıdaki biçime getirin:

```
::sysinit:/etc/init.d/baslat

tty1::askfirst:/bin/busybox sh
tty2::askfirst:/bin/X -query 10.10.10.1

::ctrlaltdel:/bin/reboot -d 0
::restart:/sbin/init
```

Bu dosyanın bir önceki inittab dosyasından farkı **tty2::askfirst:/bin/X -query 10.10.10.1** satırı. Bu satırla 2. konsola geçip enter tuşuna bastığımızda X çalıştırılacak. **-query 10.10.10.1** seçeneği ile de bağlanacağı X sunucusunun IP adresini belirtiyoruz.

etc/init.d/baslat betiğine de bazı açıklamalar koysak iyi olur. Dosyanın sonuna aşağıdaki satırları ya da siz ne uygun görürseniz yazın. Bu, sadece, X terminalinin kullanıcıya bilgi vermek için:

```
clear
```

```

echo ""
echo ""
echo ""
echo "          +-----+
echo "          | X oturumunu baslatmak icin: ALT-F2 ve ENTER |
echo "          |                                           |
echo "          | X Oturumunundan cikmak icin : CTRL-ALT-F1 |
echo "          |                                           |
echo "          | Sistemi kapatmak icin:                      |
echo "          | X oturumundan ciktikten sonra poweroff yazip |
echo "          | ENTER'a basin.                               |
echo "          +-----+

exit 0

```

Son olarak, **initrdyap** betiğini çalıştırıp **initrd.img** dosyasını oluşturduktan sonra istemci makinaya geçip, **netlogon** dizinindeki betiği çalıştırıp dosyaları windows'a aktarın ve **LINUX OL** simgesine tıklayıp yaptıklarınızı deneyin.

Sunucu makinaya bağlantı kurulduğunda karşınıza **gdm**'nin oturum açma ekranı gelecek. 10 saniye içinde bir kullanıcı ismi ve parolası vermezseniz **talebe** kullanıcısı ile size bir oturum açılacaktır. Bu noktadan sonra artık sunucu makinanın bir kullanıcısı olacaksınız ve ev dizininiz **/home/talebe** dizini olacaktır. Eğer tüm X terminali kullanıcılarının ayrı ev dizinleri olmasını isterseniz (hepsi **talebe**'nin ev dizinini kullanırsa kavga çıkabilir) onlar için sunucu makina üzerinde kullanıcı oluşturup, daha sonra değiştirmelerini söyleyerek bir parola vermelisiniz. Kullanıcılar sisteme kendi kullanıcı isimleriyle bağlandıktan sonra parolalarını **passwd** komutu ile değiştirebilirler. **talebe** kullanıcısı için bunu yapamazlar, çünkü yapmaları sistem tarafından engellenir.

/a3/ramdisk/initrd dizini altındaki sistemin ne kadar yer kapladığını **du /a3/ramdisk/initrd** komutunu vererek öğrenebilirsiniz. Çıktının son satırı dizinin sabit diskte kapladığı alanı kB cinsinden verir. Bu değer benim oluşturduğum sistem için 11279 kB idi. Eğer oluşturduğunuz sistem (12000 – 500) kB'ı aşıyorsa, **initrdyap** betiğindeki 12000 lik değeri arttırmalısınız. **linux.bat** dosyasında da bu değeri değiştirmelisiniz. X terminalinin konsolunda **df -k** komutu ile ramdiskin boyutunu ve ne kadarının kullanıldığı ile ne kadar yerin boş olduğunu öğrenebilirsiniz. Bu değerlere bakarak **initrd.rd** dosyasının boyutunun ne olması gerektiği hakkında bir fikir edinebilirsiniz. Ben 200 kB kadar bir boş alan kalmasına dikkat ettim ve bu dosyayı **initrdyap** betiği ile 12000 kB olarak oluşturdum. Çok sınır değerler için **initrdyap** betiği hata verecektir. Biraz bol tutmakta (ben ~700 kB kadar artı yer ayırdım) yarar var.

4. Sonuç

Bu belgedeki herşeyi adım adım uygulayarak yazdım. Ancak, elimde sadece 2 bilgisayar vardı. Daha fazla örnekleme imkanım yoktu. Eğer, uygulamada bir sorunla karşılaşırsanız, sorunun sizden kaynaklanmadığına iyice emin olduktan sonra bana bildirirseniz, geneli ilgilendiren bir sorunsu hem bir çözüm bulmaya çalışır hem de bu bilgiyi bu belgeye eklerim. Böylece bilgi paylaşılmış olur.

Umarım birilerine yararı olur.

A. Çekirdek Yapılandırma Dosyası (.config)

```
1 #
2 # Automatically generated by make menuconfig: don't edit
3 #
4 CONFIG_X86=y
5 # CONFIG_SBUS is not set
6 CONFIG_UID16=y
7
8 #
9 # Code maturity level options
10 #
11 CONFIG_EXPERIMENTAL=y
12
13 #
14 # Loadable module support
15 #
16 CONFIG_MODULES=y
17 # CONFIG_MODVERSIONS is not set
18 CONFIG_KMOD=y
19
20 #
21 # Processor type and features
22 #
23 CONFIG_M386=y
24 # CONFIG_M486 is not set
25 # CONFIG_M586 is not set
26 # CONFIG_M586TSC is not set
27 # CONFIG_M586MMX is not set
28 # CONFIG_M686 is not set
29 # CONFIG_MPENTIUMIII is not set
30 # CONFIG_MPENTIUM4 is not set
31 # CONFIG_MK6 is not set
32 # CONFIG_MK7 is not set
33 # CONFIG_MK8 is not set
34 # CONFIG_MELAN is not set
35 # CONFIG_MCRUSOE is not set
36 # CONFIG_MWINCHIP6 is not set
37 # CONFIG_MWINCHIP2 is not set
38 # CONFIG_MWINCHIP3D is not set
39 # CONFIG_MCYRIXIII is not set
40 # CONFIG_MVIAC3_2 is not set
41 # CONFIG_X86_CMPXCHG is not set
42 # CONFIG_X86_XADD is not set
43 CONFIG_X86_L1_CACHE_SHIFT=4
44 CONFIG_RWSEM_GENERIC_SPINLOCK=y
45 # CONFIG_RWSEM_XCHGADD_ALGORITHM is not set
46 CONFIG_X86_PPRO_FENCE=y
47 # CONFIG_X86_F00F_WORKS_OK is not set
48 CONFIG_X86_MCE=y
49 # CONFIG_TOSHIBA is not set
50 # CONFIG_I8K is not set
51 # CONFIG_MICROCODE is not set
52 # CONFIG_X86_MSR is not set
53 # CONFIG_X86_CPUID is not set
54 CONFIG_NOHIGHMEM=y
55 # CONFIG_HIGHMEM4G is not set
```

```
56 # CONFIG_HIGHMEM64G is not set
57 # CONFIG_HIGHMEM is not set
58 # CONFIG_MATH_EMULATION is not set
59 # CONFIG_MTRR is not set
60 # CONFIG_SMP is not set
61 # CONFIG_X86_UP_APIC is not set
62 # CONFIG_X86_UP_IOAPIC is not set
63 # CONFIG_X86_TSC_DISABLE is not set
64 CONFIG_X86_TSC=y
65
66 #
67 # General setup
68 #
69 CONFIG_NET=y
70 CONFIG_PCI=y
71 # CONFIG_PCI_GOBIO is not set
72 # CONFIG_PCI_GODIRECT is not set
73 CONFIG_PCI_GOANY=y
74 CONFIG_PCI_BIOS=y
75 CONFIG_PCI_DIRECT=y
76 CONFIG_ISA=y
77 CONFIG_PCI_NAMES=y
78 # CONFIG_EISA is not set
79 # CONFIG_MCA is not set
80 # CONFIG_HOTPLUG is not set
81 # CONFIG_PCMCIA is not set
82 # CONFIG_HOTPLUG_PCI is not set
83 CONFIG_SYSVIPC=y
84 CONFIG_BSD_PROCESS_ACCT=y
85 CONFIG_SYSCTL=y
86 CONFIG_KCORE_ELF=y
87 # CONFIG_KCORE_AOUT is not set
88 # CONFIG_BINFMT_AOUT is not set
89 CONFIG_BINFMT_ELF=y
90 CONFIG_BINFMT_MISC=m
91 # CONFIG_PM is not set
92 # CONFIG_ACPI is not set
93 # CONFIG_APM is not set
94
95 #
96 # Memory Technology Devices (MTD)
97 #
98 # CONFIG_MTD is not set
99
100 #
101 # Parallel port support
102 #
103 # CONFIG_PARPORT is not set
104
105 #
106 # Plug and Play configuration
107 #
108 # CONFIG_PNP is not set
109 # CONFIG_ISAPNP is not set
110
111 #
112 # Block devices
```

```
113 #
114 # CONFIG_BLK_DEV_FD is not set
115 # CONFIG_BLK_DEV_XD is not set
116 # CONFIG_PARIDE is not set
117 # CONFIG_BLK_CPQ_DA is not set
118 # CONFIG_BLK_CPQ_CISS_DA is not set
119 # CONFIG_CISS_SCSI_TAPE is not set
120 # CONFIG_BLK_DEV_DAC960 is not set
121 # CONFIG_BLK_DEV_UMEM is not set
122 CONFIG_BLK_DEV_LOOP=y
123 # CONFIG_BLK_DEV_NBD is not set
124 CONFIG_BLK_DEV_RAM=y
125 CONFIG_BLK_DEV_RAM_SIZE=4096
126 CONFIG_BLK_DEV_INITRD=y
127 # CONFIG_BLK_STATS is not set
128
129 #
130 # Multi-device support (RAID and LVM)
131 #
132 # CONFIG_MD is not set
133 # CONFIG_BLK_DEV_MD is not set
134 # CONFIG_MD_LINEAR is not set
135 # CONFIG_MD_RAID0 is not set
136 # CONFIG_MD_RAID1 is not set
137 # CONFIG_MD_RAID5 is not set
138 # CONFIG_MD_MULTIPATH is not set
139 # CONFIG_BLK_DEV_LVM is not set
140
141 #
142 # Networking options
143 #
144 CONFIG_PACKET=y
145 CONFIG_PACKET_MMAP=y
146 CONFIG_NETLINK_DEV=m
147 # CONFIG_NETFILTER is not set
148 CONFIG_FILTER=y
149 CONFIG_UNIX=y
150 CONFIG_INET=y
151 CONFIG_IP_MULTICAST=y
152 CONFIG_IP_ADVANCED_ROUTER=y
153 CONFIG_IP_MULTIPLE_TABLES=y
154 CONFIG_IP_ROUTE_NAT=y
155 # CONFIG_IP_ROUTE_MULTIPATH is not set
156 # CONFIG_IP_ROUTE_TOS is not set
157 # CONFIG_IP_ROUTE_VERBOSE is not set
158 # CONFIG_IP_ROUTE_LARGE_TABLES is not set
159 # CONFIG_IP_PNP is not set
160 # CONFIG_NET_IPIP is not set
161 # CONFIG_NET_IPGRE is not set
162 # CONFIG_IP_MROUTE is not set
163 # CONFIG_ARPD is not set
164 # CONFIG_INET_ECN is not set
165 CONFIG_SYN_COOKIES=y
166 # CONFIG_IPV6 is not set
167 # CONFIG_KHTTPD is not set
168 # CONFIG_ATM is not set
169 # CONFIG_VLAN_8021Q is not set
```

```
170 # CONFIG_IPX is not set
171 # CONFIG_ATALK is not set
172
173 #
174 # Appletalk devices
175 #
176 # CONFIG_DEV_APPLETALK is not set
177 # CONFIG_DECNET is not set
178 # CONFIG_BRIDGE is not set
179 # CONFIG_X25 is not set
180 # CONFIG_LAPB is not set
181 # CONFIG_LLC is not set
182 # CONFIG_NET_DIVERT is not set
183 # CONFIG_ECONET is not set
184 # CONFIG_WAN_ROUTER is not set
185 # CONFIG_NET_FASTROUTE is not set
186 # CONFIG_NET_HW_FLOWCONTROL is not set
187
188 #
189 # QoS and/or fair queueing
190 #
191 # CONFIG_NET_SCHED is not set
192
193 #
194 # Network testing
195 #
196 # CONFIG_NET_PKTGEN is not set
197
198 #
199 # Telephony Support
200 #
201 # CONFIG_PHONE is not set
202 # CONFIG_PHONE_IXJ is not set
203 # CONFIG_PHONE_IXJ_PCMCIA is not set
204
205 #
206 # ATA/IDE/MFM/RLL support
207 #
208 # CONFIG_IDE is not set
209 # CONFIG_BLK_DEV_IDE_MODES is not set
210 # CONFIG_BLK_DEV_HD is not set
211
212 #
213 # SCSI support
214 #
215 # CONFIG_SCSI is not set
216
217 #
218 # Fusion MPT device support
219 #
220 # CONFIG_FUSION is not set
221 # CONFIG_FUSION_BOOT is not set
222 # CONFIG_FUSION_ISENSE is not set
223 # CONFIG_FUSION_CTL is not set
224 # CONFIG_FUSION_LAN is not set
225
226 #
```

```
227 # IEEE 1394 (FireWire) support (EXPERIMENTAL)
228 #
229 # CONFIG_IEEE1394 is not set
230
231 #
232 # I2O device support
233 #
234 # CONFIG_I2O is not set
235 # CONFIG_I2O_PCI is not set
236 # CONFIG_I2O_BLOCK is not set
237 # CONFIG_I2O_LAN is not set
238 # CONFIG_I2O_SCSI is not set
239 # CONFIG_I2O_PROC is not set
240
241 #
242 # Network device support
243 #
244 CONFIG_NETDEVICES=y
245
246 #
247 # ARCnet devices
248 #
249 # CONFIG_ARCNET is not set
250 CONFIG_DUMMY=m
251 # CONFIG_BONDING is not set
252 # CONFIG_EQUALIZER is not set
253 # CONFIG_TUN is not set
254 # CONFIG_ETHERTAP is not set
255
256 #
257 # Ethernet (10 or 100Mbit)
258 #
259 CONFIG_NET_ETHERNET=y
260 # CONFIG_SUNLANCE is not set
261 CONFIG_HAPPYMEAL=m
262 # CONFIG_SUNBMAC is not set
263 # CONFIG_SUNQE is not set
264 CONFIG_SUNGEM=m
265 CONFIG_NET_VENDOR_3COM=y
266 CONFIG_EL1=m
267 CONFIG_EL2=m
268 CONFIG_ELPLUS=m
269 CONFIG_EL16=m
270 CONFIG_EL3=m
271 CONFIG_3C515=m
272 # CONFIG_ELMC is not set
273 # CONFIG_ELMC_II is not set
274 CONFIG_VORTEX=m
275 CONFIG_TYHOON=m
276 CONFIG_LANCE=m
277 CONFIG_NET_VENDOR_SMC=y
278 CONFIG_WD80x3=m
279 # CONFIG_ULTRAMCA is not set
280 CONFIG_ULTRA=m
281 # CONFIG_ULTRA32 is not set
282 CONFIG_SMC9194=m
283 CONFIG_NET_VENDOR_RACAL=y
```



```
284 CONFIG_NI5010=m
285 CONFIG_NI52=m
286 CONFIG_NI65=m
287 CONFIG_AT1700=m
288 CONFIG_DEPCA=m
289 CONFIG_HP100=m
290 CONFIG_NET_ISA=y
291 CONFIG_E2100=m
292 CONFIG_EWRK3=m
293 CONFIG_EEXPRESS=m
294 CONFIG_EEXPRESS_PRO=m
295 CONFIG_HPLAN_PLUS=m
296 CONFIG_HPLAN=m
297 CONFIG_LP486E=m
298 CONFIG_ETH16I=m
299 CONFIG_NE2000=m
300 CONFIG_NET_PCI=y
301 CONFIG_PCNET32=m
302 CONFIG_AMD8111_ETH=m
303 CONFIG_ADAPTEC_STARFIRE=m
304 CONFIG_AC3200=m
305 CONFIG_APRICOT=m
306 CONFIG_CS89x0=m
307 CONFIG_TULIP=m
308 CONFIG_TULIP_MWI=y
309 CONFIG_TULIP_MMIO=y
310 CONFIG_DE4X5=m
311 CONFIG_DGRS=m
312 CONFIG_DM9102=m
313 CONFIG_EEPRO100=m
314 CONFIG_EEPRO100_PIO=y
315 CONFIG_E100=m
316 # CONFIG_LNE390 is not set
317 CONFIG_FEALNX=m
318 CONFIG_NATSEMI=m
319 CONFIG_NE2K_PCI=m
320 # CONFIG_NE3210 is not set
321 # CONFIG_ES3210 is not set
322 CONFIG_8139CP=m
323 CONFIG_8139TOO=m
324 CONFIG_8139TOO_PIO=y
325 CONFIG_8139TOO_TUNE_TWISTER=y
326 CONFIG_8139TOO_8129=y
327 CONFIG_8139_OLD_RX_RESET=y
328 CONFIG_SIS900=m
329 CONFIG_EPIC100=m
330 CONFIG_SUNDANCE=m
331 CONFIG_SUNDANCE_MMIO=y
332 CONFIG_TLAN=m
333 CONFIG_TC35815=m
334 CONFIG_VIA_RHINE=m
335 CONFIG_VIA_RHINE_MMIO=y
336 CONFIG_WINBOND_840=m
337 # CONFIG_NET_POCKET is not set
338
339 #
340 # Ethernet (1000 Mbit)
```

```
341 #
342 # CONFIG_ACENIC is not set
343 # CONFIG_DL2K is not set
344 # CONFIG_E1000 is not set
345 # CONFIG_MYRI_SBUS is not set
346 # CONFIG_NS83820 is not set
347 # CONFIG_HAMACHI is not set
348 # CONFIG_YELLOWFIN is not set
349 # CONFIG_R8169 is not set
350 # CONFIG_SK98LIN is not set
351 # CONFIG_TIGON3 is not set
352 # CONFIG_FDDI is not set
353 # CONFIG_HIPPI is not set
354 # CONFIG_PLIP is not set
355 CONFIG_PPP=m
356 CONFIG_PPP_MULTILINK=y
357 CONFIG_PPP_FILTER=y
358 CONFIG_PPP_ASYNC=m
359 CONFIG_PPP_SYNC_TTY=m
360 CONFIG_PPP_DEFLATE=m
361 CONFIG_PPP_BSDCOMP=m
362 CONFIG_PPPOE=m
363 # CONFIG_SLIP is not set
364
365 #
366 # Wireless LAN (non-hamradio)
367 #
368 # CONFIG_NET_RADIO is not set
369
370 #
371 # Token Ring devices
372 #
373 # CONFIG_TR is not set
374 # CONFIG_NET_FC is not set
375 # CONFIG_RCPCI is not set
376 CONFIG_SHAPER=m
377
378 #
379 # Wan interfaces
380 #
381 # CONFIG_WAN is not set
382
383 #
384 # Amateur Radio support
385 #
386 # CONFIG_HAMRADIO is not set
387
388 #
389 # IrDA (infrared) support
390 #
391 # CONFIG_IRDA is not set
392
393 #
394 # ISDN subsystem
395 #
396 # CONFIG_ISDN is not set
397
```

```
398 #
399 # Old CD-ROM drivers (not SCSI, not IDE)
400 #
401 # CONFIG_CD_NO_IDESCSI is not set
402
403 #
404 # Input core support
405 #
406 CONFIG_INPUT=y
407 CONFIG_INPUT_KEYBDEV=y
408 CONFIG_INPUT_MOUSEDEV=y
409 CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
410 CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
411 CONFIG_INPUT_JOYDEV=m
412 CONFIG_INPUT_EVDEV=m
413
414 #
415 # Character devices
416 #
417 CONFIG_VT=y
418 CONFIG_VT_CONSOLE=y
419 CONFIG_SERIAL=y
420 # CONFIG_SERIAL_CONSOLE is not set
421 # CONFIG_SERIAL_EXTENDED is not set
422 CONFIG_SERIAL_NONSTANDARD=y
423 # CONFIG_COMPUTONE is not set
424 # CONFIG_ROCKETPORT is not set
425 # CONFIG_CYCLADES is not set
426 # CONFIG_DIGIEPCA is not set
427 # CONFIG_DIGI is not set
428 CONFIG_ESPSERIAL=m
429 # CONFIG_MOXA_INTELLIO is not set
430 # CONFIG_MOXA_SMARTIO is not set
431 # CONFIG_ISI is not set
432 # CONFIG_SYNCLINK is not set
433 # CONFIG_SYNCLINKMP is not set
434 # CONFIG_N_HDLC is not set
435 # CONFIG_RISCOM8 is not set
436 # CONFIG_SPECIALIX is not set
437 # CONFIG_SX is not set
438 # CONFIG_RIO is not set
439 # CONFIG_STALDRV is not set
440 CONFIG_UNIX98_PTYS=y
441 CONFIG_UNIX98_PTY_COUNT=256
442
443 #
444 # I2C support
445 #
446 # CONFIG_I2C is not set
447
448 #
449 # Mice
450 #
451 # CONFIG_BUSMOUSE is not set
452 CONFIG_MOUSE=y
453 CONFIG_PSMOUSE=y
454 # CONFIG_82C710_MOUSE is not set
```

```
455 # CONFIG_PC110_PAD is not set
456 # CONFIG_MK712_MOUSE is not set
457
458 #
459 # Joysticks
460 #
461 # CONFIG_INPUT_GAMEPORT is not set
462 # CONFIG_INPUT_NS558 is not set
463 # CONFIG_INPUT_LIGHTNING is not set
464 # CONFIG_INPUT_PCIGAME is not set
465 # CONFIG_INPUT_CS461X is not set
466 # CONFIG_INPUT_EMU10K1 is not set
467 # CONFIG_INPUT_SERIO is not set
468 # CONFIG_INPUT_SERPORT is not set
469 # CONFIG_INPUT_ANALOG is not set
470 # CONFIG_INPUT_A3D is not set
471 # CONFIG_INPUT_ADI is not set
472 # CONFIG_INPUT_COBRA is not set
473 # CONFIG_INPUT_GF2K is not set
474 # CONFIG_INPUT_GRIP is not set
475 # CONFIG_INPUT_INTERACT is not set
476 # CONFIG_INPUT_TMDC is not set
477 # CONFIG_INPUT_SIDEWINDER is not set
478 # CONFIG_INPUT_IFORCE_USB is not set
479 # CONFIG_INPUT_IFORCE_232 is not set
480 # CONFIG_INPUT_WARRIOR is not set
481 # CONFIG_INPUT_MAGELLAN is not set
482 # CONFIG_INPUT_SPACEORB is not set
483 # CONFIG_INPUT_SPACEBALL is not set
484 # CONFIG_INPUT_STINGER is not set
485 # CONFIG_INPUT_DB9 is not set
486 # CONFIG_INPUT_GAMECON is not set
487 # CONFIG_INPUT_TURBOGRAFX is not set
488 # CONFIG_QIC02_TAPE is not set
489 # CONFIG_IPMI_HANDLER is not set
490 # CONFIG_IPMI_PANIC_EVENT is not set
491 # CONFIG_IPMI_DEVICE_INTERFACE is not set
492 # CONFIG_IPMI_KCS is not set
493 # CONFIG_IPMI_WATCHDOG is not set
494
495 #
496 # Watchdog Cards
497 #
498 # CONFIG_WATCHDOG is not set
499 # CONFIG_SCx200_GPIO is not set
500 # CONFIG_AMD_RNG is not set
501 # CONFIG_INTEL_RNG is not set
502 # CONFIG_AMD_PM768 is not set
503 CONFIG_NVRAM=m
504 CONFIG_RTC=y
505 # CONFIG_DTLK is not set
506 # CONFIG_R3964 is not set
507 # CONFIG_APPLICOM is not set
508 # CONFIG_SONYPI is not set
509
510 #
511 # Ftape, the floppy tape device driver
```

```
512 #
513 # CONFIG_FTAPE is not set
514 # CONFIG_AGP is not set
515 CONFIG_DRM=y
516 # CONFIG_DRM_OLD is not set
517 CONFIG_DRM_NEW=y
518 # CONFIG_DRM_TDFX is not set
519 # CONFIG_DRM_R128 is not set
520 # CONFIG_DRM_RADEON is not set
521 # CONFIG_DRM_I810 is not set
522 # CONFIG_DRM_I810_XFREE_41 is not set
523 # CONFIG_DRM_I830 is not set
524 # CONFIG_DRM_MGA is not set
525 # CONFIG_DRM_SIS is not set
526 # CONFIG_MWAVE is not set
527
528 #
529 # Multimedia devices
530 #
531 # CONFIG_VIDEO_DEV is not set
532
533 #
534 # File systems
535 #
536 CONFIG_QUOTA=y
537 # CONFIG_QFMT_V1 is not set
538 # CONFIG_QFMT_V2 is not set
539 # CONFIG_QIFACE_COMPAT is not set
540 # CONFIG_AUTOFS_FS is not set
541 # CONFIG_AUTOFS4_FS is not set
542 # CONFIG_REISERFS_FS is not set
543 # CONFIG_REISERFS_CHECK is not set
544 # CONFIG_REISERFS_PROC_INFO is not set
545 # CONFIG_ADFS_FS is not set
546 # CONFIG_ADFS_FS_RW is not set
547 # CONFIG_AFFS_FS is not set
548 # CONFIG_HFS_FS is not set
549 # CONFIG_BEFS_FS is not set
550 # CONFIG_BEFS_DEBUG is not set
551 # CONFIG_BFS_FS is not set
552 # CONFIG_EXT3_FS is not set
553 # CONFIG_JBD is not set
554 # CONFIG_JBD_DEBUG is not set
555 # CONFIG_FAT_FS is not set
556 # CONFIG_MSDOS_FS is not set
557 # CONFIG_UMSDOS_FS is not set
558 # CONFIG_VFAT_FS is not set
559 # CONFIG_EFS_FS is not set
560 # CONFIG_JFFS_FS is not set
561 # CONFIG_JFFS2_FS is not set
562 # CONFIG_CRAMFS is not set
563 CONFIG_TMPFS=y
564 CONFIG_RAMFS=y
565 # CONFIG_ISO9660_FS is not set
566 # CONFIG_JOLIET is not set
567 # CONFIG_ZISOFS is not set
568 # CONFIG_JFS_FS is not set
```

```
569 # CONFIG_JFS_DEBUG is not set
570 # CONFIG_JFS_STATISTICS is not set
571 # CONFIG_MINIX_FS is not set
572 # CONFIG_VXFS_FS is not set
573 # CONFIG_NTFS_FS is not set
574 # CONFIG_NTFS_RW is not set
575 # CONFIG_HPFS_FS is not set
576 CONFIG_PROC_FS=y
577 # CONFIG_DEVFS_FS is not set
578 # CONFIG_DEVFS_MOUNT is not set
579 # CONFIG_DEVFS_DEBUG is not set
580 # CONFIG_DEVPTS_FS is not set
581 # CONFIG_QNX4FS_FS is not set
582 # CONFIG_QNX4FS_RW is not set
583 CONFIG_ROMFS_FS=m
584 CONFIG_EXT2_FS=y
585 CONFIG_SYSV_FS=m
586 # CONFIG_UDF_FS is not set
587 # CONFIG_UDF_RW is not set
588 # CONFIG_UFS_FS is not set
589 # CONFIG_UFS_FS_WRITE is not set
590 # CONFIG_XFS_FS is not set
591 # CONFIG_XFS_POSIX_ACL is not set
592 # CONFIG_XFS_RT is not set
593 # CONFIG_XFS_QUOTA is not set
594 # CONFIG_XFS_DMAPI is not set
595 # CONFIG_XFS_DEBUG is not set
596 # CONFIG_PAGEBUF_DEBUG is not set
597
598 #
599 # Network File Systems
600 #
601 # CONFIG_CODA_FS is not set
602 # CONFIG_INTERMEZZO_FS is not set
603 # CONFIG_NFS_FS is not set
604 # CONFIG_NFS_V3 is not set
605 # CONFIG_ROOT_NFS is not set
606 # CONFIG_NFSD is not set
607 # CONFIG_NFSD_V3 is not set
608 # CONFIG_NFSD_TCP is not set
609 # CONFIG_SUNRPC is not set
610 # CONFIG_LOCKD is not set
611 # CONFIG_SMB_FS is not set
612 # CONFIG_NCP_FS is not set
613 # CONFIG_NCPFS_PACKET_SIGNING is not set
614 # CONFIG_NCPFS_IOCTL_LOCKING is not set
615 # CONFIG_NCPFS_STRONG is not set
616 # CONFIG_NCPFS_NFS_NS is not set
617 # CONFIG_NCPFS_OS2_NS is not set
618 # CONFIG_NCPFS_SMALLDOS is not set
619 # CONFIG_NCPFS_NLS is not set
620 # CONFIG_NCPFS_EXTRAS is not set
621 # CONFIG_ZISOFS_FS is not set
622
623 #
624 # Partition Types
625 #
```

```
626 CONFIG_PARTITION_ADVANCED=y
627 # CONFIG_ACORN_PARTITION is not set
628 # CONFIG_OSF_PARTITION is not set
629 # CONFIG_AMIGA_PARTITION is not set
630 # CONFIG_ATARI_PARTITION is not set
631 # CONFIG_MAC_PARTITION is not set
632 CONFIG_MSDOS_PARTITION=y
633 # CONFIG_BSD_DISKLABEL is not set
634 # CONFIG_MINIX_SUBPARTITION is not set
635 # CONFIG_SOLARIS_X86_PARTITION is not set
636 # CONFIG_UNIXWARE_DISKLABEL is not set
637 CONFIG_LDM_PARTITION=y
638 # CONFIG_LDM_DEBUG is not set
639 # CONFIG_SGI_PARTITION is not set
640 # CONFIG_ULTRIX_PARTITION is not set
641 # CONFIG_SUN_PARTITION is not set
642 # CONFIG_EFI_PARTITION is not set
643 # CONFIG_SMB_NLS is not set
644 # CONFIG_NLS is not set
645
646 #
647 # Console drivers
648 #
649 CONFIG_VGA_CONSOLE=y
650 CONFIG_VIDEO_SELECT=y
651 # CONFIG_MDA_CONSOLE is not set
652
653 #
654 # Frame-buffer support
655 #
656 # CONFIG_FB is not set
657
658 #
659 # Sound
660 #
661 # CONFIG_SOUND is not set
662
663 #
664 # USB support
665 #
666 # CONFIG_USB is not set
667
668 #
669 # Bluetooth support
670 #
671 # CONFIG_BLUEZ is not set
672
673 #
674 # Kernel hacking
675 #
676 CONFIG_DEBUG_KERNEL=y
677 # CONFIG_DEBUG_STACKOVERFLOW is not set
678 # CONFIG_DEBUG_HIGHMEM is not set
679 # CONFIG_DEBUG_SLAB is not set
680 # CONFIG_DEBUG_IOVIRT is not set
681 CONFIG_MAGIC_SYSRQ=y
682 # CONFIG_DEBUG_SPINLOCK is not set
```



```
683 # CONFIG_FRAME_POINTER is not set
684
685 #
686 # Library routines
687 #
688 CONFIG_ZLIB_INFLATE=m
689 CONFIG_ZLIB_DEFLATE=m
690
```

B. Busybox Yapılandırma Dosyası (.config)

```
#
# Automatically generated make config: don't edit
#
HAVE_DOT_CONFIG=y

#
# General Configuration
#
# CONFIG_FEATURE_BUFFERS_USE_MALLOC is not set
CONFIG_FEATURE_BUFFERS_GO_ON_STACK=y
# CONFIG_FEATURE_BUFFERS_GO_IN_BSS is not set
CONFIG_FEATURE_VERBOSE_USAGE=y
CONFIG_FEATURE_INSTALLER=y
# CONFIG_LOCALE_SUPPORT is not set
# CONFIG_FEATURE_DEVFS is not set
# CONFIG_FEATURE_DEVPTS is not set
# CONFIG_FEATURE_CLEAN_UP is not set
# CONFIG_FEATURE_SUID is not set
# CONFIG_SELINUX is not set

#
# Build Options
#
# CONFIG_STATIC is not set
# CONFIG_LFS is not set
# USING_CROSS_COMPILER is not set
EXTRA_CFLAGS_OPTIONS=""

#
# Installation Options
#
# CONFIG_INSTALL_NO_USR is not set
PREFIX="._install"

#
# Archival Utilities
#
# CONFIG_AR is not set
# CONFIG_BUNZIP2 is not set
# CONFIG_CPIO is not set
# CONFIG_DPKG is not set
# CONFIG_DPKG_DEB is not set
# CONFIG_GUNZIP is not set
# CONFIG_GZIP is not set
# CONFIG_RPM2CPIO is not set
```

```
# CONFIG_RPM is not set
# CONFIG_TAR is not set
# CONFIG_UNCOMPRESS is not set
# CONFIG_UNZIP is not set

#
# Coreutils
#
CONFIG_BASENAME=y
# CONFIG_CAL is not set
CONFIG_CAT=y
# CONFIG_CHGRP is not set
# CONFIG_CHMOD is not set
# CONFIG_CHOWN is not set
CONFIG_CHROOT=y
# CONFIG_CMP is not set
CONFIG_CP=y
# CONFIG_CUT is not set
CONFIG_DATE=y
CONFIG_FEATURE_DATE_ISOFMT=y
# CONFIG_DD is not set
CONFIG_DF=y
# CONFIG_DIRNAME is not set
# CONFIG_DOS2UNIX is not set
CONFIG_DU=y
CONFIG_FEATURE_DU_DEFAULT_BLOCKSIZE_1K=y
CONFIG_ECHO=y
CONFIG_FEATURE_FANCY_ECHO=y
CONFIG_ENV=y
# CONFIG_EXPR is not set
CONFIG_FALSE=y
# CONFIG_FOLD is not set
CONFIG_HEAD=y
# CONFIG_FEATURE_FANCY_HEAD is not set
# CONFIG_HOSTID is not set
# CONFIG_ID is not set
# CONFIG_INSTALL is not set
# CONFIG_LENGTH is not set
CONFIG_LN=y
# CONFIG_LOGNAME is not set
CONFIG_LS=y
CONFIG_FEATURE_LS_FILETYPES=y
CONFIG_FEATURE_LS_FOLLOWLINKS=y
CONFIG_FEATURE_LS_RECURSIVE=y
CONFIG_FEATURE_LS_SORTFILES=y
CONFIG_FEATURE_LS_TIMESTAMPS=y
CONFIG_FEATURE_LS_USERNAME=y
CONFIG_FEATURE_LS_COLOR=y
# CONFIG_MD5SUM is not set
CONFIG_MKDIR=y
# CONFIG_MKFIFO is not set
CONFIG_MKNOD=y
CONFIG_MV=y
# CONFIG_OD is not set
# CONFIG_PRINTF is not set
CONFIG_PWD=y
# CONFIG_REALPATH is not set
```

```
CONFIG_RM=y
CONFIG_RMDIR=y
# CONFIG_SEQ is not set
# CONFIG_SHA1SUM is not set
CONFIG_SLEEP=y
# CONFIG_FEATURE_FANCY_SLEEP is not set
CONFIG_SORT=y
# CONFIG_STTY is not set
CONFIG_SYNC=y
CONFIG_TAIL=y
CONFIG_FEATURE_FANCY_TAIL=y
# CONFIG_TEE is not set
CONFIG_TEST=y

#
# test (forced enabled for use with shell)
#
CONFIG_TOUCH=y
CONFIG_TR=y
CONFIG_TRUE=y
CONFIG_TTY=y
CONFIG_UNAME=y
CONFIG_UNIQ=y
# CONFIG_USLEEP is not set
# CONFIG_UUDECODE is not set
# CONFIG_UUENCODE is not set
# CONFIG_WATCH is not set
CONFIG_WC=y
# CONFIG_WHO is not set
# CONFIG_WHOAMI is not set
CONFIG_YES=y

#
# Common options for cp and mv
#
CONFIG_FEATURE_PRESERVE_HARDLINKS=y

#
# Common options for ls and more
#
CONFIG_FEATURE_AUTOWIDTH=y

#
# Common options for df, du, ls
#
CONFIG_FEATURE_HUMAN_READABLE=y

#
# Console Utilities
#
CONFIG_CHVT=y
CONFIG_CLEAR=y
# CONFIG_DEALLOCVT is not set
# CONFIG_DUMPKMAP is not set
# CONFIG_LOADFONT is not set
# CONFIG_LOADKMAP is not set
# CONFIG_OPENVT is not set
```

```
CONFIG_RESET=y
# CONFIG_SETKEYCODES is not set

#
# Debian Utilities
#
# CONFIG_MKTEMP is not set
# CONFIG_PIPE_PROGRESS is not set
# CONFIG_READLINK is not set
# CONFIG_RUN_PARTS is not set
# CONFIG_START_STOP_DAEMON is not set
CONFIG_WHICH=y

#
# Editors
#
# CONFIG_AWK is not set
# CONFIG_PATCH is not set
CONFIG_SED=y
# CONFIG_VI is not set

#
# Finding Utilities
#
CONFIG_FIND=y
CONFIG_FEATURE_FIND_MTIME=y
CONFIG_FEATURE_FIND_PERM=y
CONFIG_FEATURE_FIND_TYPE=y
CONFIG_FEATURE_FIND_XDEV=y
# CONFIG_FEATURE_FIND_NEWER is not set
# CONFIG_FEATURE_FIND_INUM is not set
CONFIG_GREP=y
# CONFIG_FEATURE_GREP_EGREP_ALIAS is not set
# CONFIG_FEATURE_GREP_FGREP_ALIAS is not set
CONFIG_FEATURE_GREP_CONTEXT=y
# CONFIG_XARGS is not set

#
# Init Utilities
#
CONFIG_INIT=y
CONFIG_FEATURE_USE_INITTAB=y
CONFIG_FEATURE_INITRD=y
# CONFIG_FEATURE_INIT_COREDUMPS is not set
CONFIG_FEATURE_EXTRA_QUIET=y
CONFIG_HALT=y
CONFIG_POWEROFF=y
CONFIG_REBOOT=y
# CONFIG_MESG is not set

#
# Login/Password Management Utilities
#
CONFIG_USE_BB_PWD_GRP=y
# CONFIG_ADDGROUP is not set
# CONFIG_DELGROUP is not set
# CONFIG_ADDUSER is not set
```

```
# CONFIG_DELUSER is not set
# CONFIG_GETTY is not set
# CONFIG_LOGIN is not set
# CONFIG_PASSWD is not set
# CONFIG_SU is not set
# CONFIG_SULOGIN is not set
# CONFIG_VLOCK is not set

#
# Miscellaneous Utilities
#
# CONFIG_ADJTIMEX is not set
# CONFIG_CROND is not set
# CONFIG_CRONTAB is not set
# CONFIG_DC is not set
# CONFIG_DEVFSD is not set
# CONFIG_LAST is not set
# CONFIG_HDPARM is not set
# CONFIG_MAKEDEVS is not set
# CONFIG_MT is not set
# CONFIG_RX is not set
CONFIG_STRINGS=y
# CONFIG_TIME is not set
# CONFIG_WATCHDOG is not set

#
# Linux Module Utilities
#
# CONFIG_INSMOD is not set
# CONFIG_LSMOD is not set
# CONFIG_MODPROBE is not set
# CONFIG_RMMOD is not set

#
# Networking Utilities
#
# CONFIG_FEATURE_IPV6 is not set
# CONFIG_ARPING is not set
# CONFIG_FTPGET is not set
# CONFIG_FTPPUT is not set
CONFIG_HOSTNAME=y
# CONFIG_HTTPD is not set
CONFIG_IFCONFIG=y
CONFIG_FEATURE_IFCONFIG_STATUS=y
# CONFIG_FEATURE_IFCONFIG_SLIP is not set
CONFIG_FEATURE_IFCONFIG_MEMSTART_IOADDR_IRQ=y
CONFIG_FEATURE_IFCONFIG_HW=y
CONFIG_FEATURE_IFCONFIG_BROADCAST_PLUS=y
# CONFIG_IFUPDOWN is not set
# CONFIG_INETD is not set
CONFIG_IP=y
CONFIG_FEATURE_IP_ADDRESS=y
CONFIG_FEATURE_IP_LINK=y
CONFIG_FEATURE_IP_ROUTE=y
# CONFIG_FEATURE_IP_TUNNEL is not set
CONFIG_IPCALC=y
CONFIG_FEATURE_IPCALC_FANCY=y
```

```
# CONFIG_IPADDR is not set
# CONFIG_IPLINK is not set
# CONFIG_IPROUTE is not set
# CONFIG_IPTUNNEL is not set
# CONFIG_NAMEIF is not set
# CONFIG_NC is not set
CONFIG_NETSTAT=y
CONFIG_NSLOOKUP=y
CONFIG_PING=y
CONFIG_FEATURE_FANCY_PING=y
# CONFIG_ROUTE is not set
# CONFIG_TELNET is not set
# CONFIG_TELNETD is not set
# CONFIG_TFTP is not set
# CONFIG_TRACEROUTE is not set
# CONFIG_VCONFIG is not set
# CONFIG_WGET is not set

#
# udhcp Server/Client
#
# CONFIG_UDHCPD is not set
CONFIG_UDHCPC=y
# CONFIG_FEATURE_UDHCP_SYSLOG is not set
# CONFIG_FEATURE_UDHCP_DEBUG is not set

#
# Process Utilities
#
CONFIG_FREE=y
CONFIG_KILL=y
CONFIG_KILLALL=y
CONFIG_PIDOF=y
CONFIG_PS=y
# CONFIG_RENICE is not set
# CONFIG_TOP is not set
CONFIG_UPTIME=y
# CONFIG_SYSCTL is not set

#
# Another Bourne-like Shell
#
CONFIG_FEATURE_SH_IS_ASH=y
# CONFIG_FEATURE_SH_IS_HUSH is not set
# CONFIG_FEATURE_SH_IS_LASH is not set
# CONFIG_FEATURE_SH_IS_MSH is not set
# CONFIG_FEATURE_SH_IS_NONE is not set
CONFIG_ASH=y

#
# Ash Shell Options
#
# CONFIG_ASH_JOB_CONTROL is not set
CONFIG_ASH_ALIAS=y
CONFIG_ASH_MATH_SUPPORT=y
CONFIG_ASH_GETOPTS=y
CONFIG_ASH_CMDCMD=y
```

```
# CONFIG_ASH_MAIL is not set
CONFIG_ASH_OPTIMIZE_FOR_SIZE=y
# CONFIG_HUSH is not set
# CONFIG_LASH is not set
# CONFIG_MSH is not set

#
# Bourne Shell Options
#
# CONFIG_FEATURE_SH_EXTRA_QUIET is not set
CONFIG_FEATURE_SH_STANDALONE_SHELL=y
# CONFIG_FEATURE_SH_APPLETS_ALWAYS_WIN is not set
CONFIG_FEATURE_COMMAND_EDITING=y
CONFIG_FEATURE_COMMAND_HISTORY=15
CONFIG_FEATURE_COMMAND_SAVEHISTORY=y
CONFIG_FEATURE_COMMAND_TAB_COMPLETION=y
# CONFIG_FEATURE_COMMAND_USERNAME_COMPLETION is not set
CONFIG_FEATURE_SH_FANCY_PROMPT=y

#
# System Logging Utilities
#
# CONFIG_SYSLOGD is not set
# CONFIG_LOGGER is not set

#
# Linux System Utilities
#
# CONFIG_DMESG is not set
# CONFIG_FBSET is not set
# CONFIG_FDFLUSH is not set
# CONFIG_FDFORMAT is not set
# CONFIG_FDISK is not set
# CONFIG_FREERAMDISK is not set
# CONFIG_FSCK_MINIX is not set
# CONFIG_MKFS_MINIX is not set
# CONFIG_GETOPT is not set
# CONFIG_HEXDUMP is not set
CONFIG_HWCLOCK=y
CONFIG_FEATURE_HWCLOCK_LONGOPTIONS=y
# CONFIG_LOSETUP is not set
# CONFIG_MKSWAP is not set
# CONFIG_MORE is not set
# CONFIG_PIVOT_ROOT is not set
# CONFIG_RDATE is not set
# CONFIG_SWAPONOFF is not set
CONFIG_MOUNT=y
# CONFIG_NFSMOUNT is not set
CONFIG_UMOUNT=y
CONFIG_FEATURE_MOUNT_FORCE=y

#
# Common options for mount/umount
#
# CONFIG_FEATURE_MOUNT_LOOP is not set
# CONFIG_FEATURE_MTAB_SUPPORT is not set
```



```
#  
# Debugging Options  
#  
# CONFIG_DEBUG is not set
```

Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

(B3) <http://busybox.net/downloads/busybox-1.00-pre10.tar.bz2>

(B8) [../man/man1/man1-ln.pdf](#)

(B9) [../man/man1/man1-mkdir.pdf](#)

(B10) [../man/man1/man1-mkfifo.pdf](#)

(B11) [../man/man1/man1-mknod.pdf](#)

(B12) [../bashref/bashref.pdf#bashref_bourne.shell.builtins-cd](#)

(B13) [../bashref/bashref.pdf#bashref_commands.index](#)

(B14) [../man/man4/man4-initrd.pdf](#)

(B15) [../man/man8/man8-init.pdf](#)

(B16) [../man/man5/man5-inittab.pdf](#)

(B17) [../man/man4/man4-initrd.pdf](#)

(B18) [../man/man5/man5-inittab.pdf](#)

Bu dosya (windows-ve-linux.pdf), belgenin XML biçiminin \TeX Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

2 Şubat 2007