

# Linux Açılış Disketi NASIL

Yazan:

**Tom Fawcett**

<Bootdisk-HOWTO (at) linuxdoc.org>

Çeviren:

**Yalçın Kolukısa**

<yalcin (at) belgeler.org>

Çeviri: v1.0, Ocak 2004

## Özet

Bu kılavuz, Linux için açılış/kurtarma disketlerinin nasıl oluşturulacağını anlatmaktadır. Bu disketler kurtarma disketi olarak veya yeni sistem elemanlarını sınamak için kullanılabilir.

Bu kılavuzda anlatılan açılış disketini oluşturmaya kalkmadan önce, sistem yönetim görevleri hakkında bilgi sahibi olmanız gerekmektedir. Şayet acil durumlar için bir kurtarma disketine ihtiyacınız var ise doğrudan [Hazır açılış disketleri](#) (sayfa: 31) bölümüne bakınız.<sup>(1)</sup>

## Konu Başlıkları

<b>1. Başlarken</b>	4
1.1. Sürüm notları	4
1.2. Yapılacak işler listesi	4
1.3. Geri besleme ve destekleyenler	4
1.4. Dağıtım politikası	5
<b>2. Giriş</b>	5
<b>3. Açılış disketleri ve önyükleme süreci</b>	6
3.1. Önyükleme süreci	6
3.2. Disk çeşitleri	7
<b>4. Bir kök dosya sisteminin oluşturulması</b>	7
4.1. Genel bakış	8
4.2. Dosya sisteminin oluşturulması	8
4.3. Dosya sisteminin doldurulması	9
4.3.1. /dev	10
4.3.2. /etc	11
4.3.3. /bin ve /sbin	12
4.3.4. /lib	13
4.4. PAM ve NSS'in sağlanması	14
4.4.1. PAM (Eklemlenebilen Kimlik Kanıtlama Modülleri)	14
4.4.2. NSS (İsim Servisi Değiştirici)	14
4.5. Modüller	15
4.6. Son olarak bazı ayrıntılar	15
4.7. Paketleme	15
<b>5. Çekirdek Seçimi</b>	16
<b>6. Herşeyi biraraya getirelim</b>	16

6.1. Çekirdeğin LILO ile kullanımı . . . . .	17
6.2. Çekirdeğin LILO'suz kullanımı . . . . .	18
6.3. ramdisk sözcüğünün yapılandırılması . . . . .	18
6.4. Kök dosya sistemini aktarılması . . . . .	19
<b>7. Sorun çözmek ya da Yenilginin Acısı</b> . . . . .	<b>20</b>
<b>8. Kök dosya sistemi boyutunun küçültülmesi</b> . . . . .	<b>21</b>
8.1. Disket kapasitesini artırın . . . . .	21
8.2. Ortak araçları BusyBox ile değiştirin . . . . .	22
8.3. Başka bir kabuk kullanın . . . . .	22
8.4. Kütüphaneleri ve çalıştırılabilirleri soyun . . . . .	22
8.5. Dosyaları uygulamalar disketine koyun . . . . .	22
<b>9. Çeşitli konular</b> . . . . .	<b>23</b>
9.1. Ramdisksiz kök dosya sistemleri . . . . .	23
9.2. Uygulamalar disketinin oluşturulması . . . . .	23
<b>10. Profesyoneller nasıl yapıyor</b> . . . . .	<b>24</b>
<b>11. Önyüklemeye yapabilen CD-ROMların hazırlanması</b> . . . . .	<b>24</b>
11.1. El Torito nedir? . . . . .	25
11.2. El Torito nasıl çalışır? . . . . .	25
11.3. Nasıl çalışır hale getireceğiz? . . . . .	25
11.4. Win9x Önyüklenebilir CD-ROMlarının yapımı . . . . .	26
<b>12. Sıkça Sorulan Sorular(SSS) listesi</b> . . . . .	<b>26</b>
<b>A. Özkaynaklar ve Gösterciler</b> . . . . .	<b>31</b>
A.1. Hazır açılış disketleri . . . . .	31
A.2. Kurtarma paketleri . . . . .	31
A.3. LILO — Linux yükleyicisi . . . . .	31
A.4. Ramdisk kullanımı . . . . .	32
A.5. Linux önyüklemeye süreci . . . . .	32
<b>B. LILO önyüklemeye hata kodları</b> . . . . .	<b>32</b>
<b>C. Kök dosya sistemi listesi örneği</b> . . . . .	<b>33</b>
<b>D. Uygulamalar disketi dizin listesi örneği</b> . . . . .	<b>39</b>
<b>Kavramlar Dizini</b> . . . . .	<b>40</b>

Telif Hakkı © 1995-2002 Tom Fawcett - İngilizce orijinalinin yazarı

Telif Hakkı © 2004 Yalçın Kolukısa - Türkçe çevirmeni

### Yasal Uyarı

Telif Hakkı © 1995–2002 Tom Fawcett ve Graham Chapman.

Türkçe çeviri: Telif Hakkı © 2004 Yalçın Kolukısa.

Bu belge [Linux Belgelendirme Projesi Kopyalama Lisansı<sup>\(B1\)</sup>](#)’nın kurallarına uymak şartıyla dağıtılabilir. Lisansı bulamadıysanız lütfen belge yazarlarına başvurunuz.

### Legal Notice

Copyright © 1995–2002 by Tom Fawcett and Graham Chapman.

Turkish translation: Copyright © 2004 by Yalçın Kolukısa.

This document may be distributed under the terms set forth in the [Linux Documentation Project Copying License<sup>\(B2\)</sup>](#). Please contact the authors if you are unable to get the license.

# 1. Başlarken



## Önemli

Elinizdeki sürüm, bu kılavuzun en güncel sürümü olmayabilir. Şayet elinizdeki sürümün tarihi, altı aydan daha eski ise, [BootDisk-HOWTO Ana Sayfasına](#) <sup>(B4)</sup> göz atınız.

Bu kılavuz, metin biçimli olarak da okunması kolay ve anlaşılır olmakla birlikte; pdf, postscript veya HTML biçimli olarak daha kolay okunabilir. Ne de olsa bu biçimlerde zengin görsel öğeler kullanılmaktadır.

## 1.1. Sürüm notları

İlk Açılış Disketi NASIL'ı Graham Chapman yazmış ve sürüm 3.1'e kadar güncellemeleri yapmıştır. Çekirdeğin v2. sürümünün çıktığı ilk zamanlarda, Tom Fawcett yardımcı yazar olarak belgeye destek vermeye başlamıştır. Bugün için ise Tom Fawcett, bu belgenin bakım sorumlusudur. Chapman'dan uzun süredir haber alınamamakta ve nerede olduğu bilinmemektedir.

Bu kılavuzdaki bilgiler *Intel* tabanlı Linuxlar için hazırlanmıştır. Buradaki bilgilerin pek çoğu diğer işlemciler için de geçerlidir, fakat bunların geçerliliği birinci elden kontrol edilmemiştir. Şayet diğer platformlar üzerinde açılış disketi oluşturmak hakkında deneyiminiz varsa, lütfen bunları bana bildirin.

## 1.2. Yapılacak işler listesi

1. <http://user-mode-linux.sourceforge.net/> adresindeki Kullanıcı Kipi Linux projesi, açılış disketlerini sınamak için sistemi tekrar başlatma zorunluluğunu ortadan kaldırması açısından oldukça yararlı görünüyor. Bu konu üzerinde şu ana kadar çalışma yapma şansım olmadı. Şayet birileri bunun için, ev yapımı açılış disketlerini kullanıyor ise lütfen bana da bilgi versin.
2. Dağıtımların açılış disketleri tekrar gözden geçirilecek ve [Profesyoneller nasıl yapıyor](#) (sayfa: 24) bölümü güncellenecek.
3. Init-getty-login işlemlerinin daha ne kadar basitleştirilebileceği araştırılacak. Bir kaç kişi; init'in doğrudan /bin/sh'a bağlanabileceğini söyledi. Şayet bu doğru ise ve fazladan sınırlamalar getirmeyecekse, başlangıç bölümü, bunu yapacak şekilde değiştirilecek. Böylece getty, login, gettydefs ve belki de bütün PAM ve NSS araçlarına olan bağımlılık ortadan kalkacak.
4. Çekirdeğin 2.4 sürümünün kaynak kodları incelenecek ve açılış süreci ve ramdisk yüklenmesi süreçleri hakkında ayrıntılı bir açıklama yazılacak (tabii ki öncelikle benim bu olayı daha ayrıntılı ve iyi bir şekilde öğrenmem gerekecek). Açılış aygıtlarının sınırları ve initrd hakkında henüz anlayamadığım bazı noktalar var.
5. Dağıtım açılış disketlerinin nasıl güncelleneceğini anlatan bölüm kılavuzdan çıkarılacak. Yarardan çok zararı var.
6. rdev komutları LILO anahtar kelimeleri ile değiştirilecek.

## 1.3. Geri besleme ve destekleyenler

Bu kılavuzun içeriği hakkındaki her türlü, iyi veya kötü, eleştirileri bana gönderebilirsiniz. Buradaki açıklama ve bilgileri, elimden gelenin en iyisini yapmaya çalışarak, anlaşılır ve doğru bir şekilde vermeye çalıştım fakat unutmayın ki ben de herşeyi bilmiyorum ve çekirdek gelişimi üzerine çalışmıyorum. Şayet herhangi bir hata veya eksiklik bulursanız lütfen beni düzeltin. Bana yazarken, elinizdeki kılavuzun sürüm numarasını belirtmeyi lütfen unutmayın.

Düzeltilmeler ve öneriler gönderen bütün herkese teşekkürlerimi sunarım. Onların sayesinde bu kılavuz, bizim tek başımıza yapabileceğimizden çok daha iyi olmuştur.

Yorum ve düzeltilmelerinizi yukarıda belirtilen, yazarın e-posta adresine gönderiniz. Çeviride bulduğunuz imla, yazım, sözdizimi, anlam hatalarını çeviren kısımdaki e-posta adresine gönderiniz. Sorularınızı sormadan önce *lütten [Sorun çözmek ya da Yenilginin Acısı](#) (sayfa: 20) bölümünü okuyunuz*. Hazırladığınız disket imajlarını bana göndermeyiniz.

## 1.4. Dağıtım politikası

Bu belgenin orjinal ingilizce sürümünün telif hakları © 1995–2002 Tom Fawcett ve Graham Chapman'a, türkçe tercümesinin telif hakları © 2004 Yalçın Kolukısa'ya aittir. Bu kılavuz [Linux Belgelendirme Projesi Lisansı](#)<sup>(B8)</sup> altında kullanılabilir. Şayet bu lisansa ulaşamıyorsanız lütten yazarlar ile iletişim kurunuz.

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.

## 2. Giriş

Linux açılış disketlerini kullanabileceğiniz pek çok durum vardır: yeni bir çekirdeği sınamak, hata veren bir diskı kurtarmak, açılmayan bir sistemi normale döndürmek veya çok önemli sistem dosyalarını (`libc.so` gibi) güven içinde güncellemek.

Açılış disketleri elde etmenin de çeşitli yolları vardır:

- Slackware gibi bir dağıtımın açılış disketini kullanmak. Bu disket ile en azından sistemi açabilirsiniz.
- Kurtarma disketleri üretmek için hazırlanmış bir kurtarma paketi kullanmak.
- Bu tür disketleri üretmek için neler gerektiğini öğrenmek ve kendi söküğünü dikmek.

Bazı insanlar son şıkkı tercih eder ve kendi disketlerini kendileri hazırlarlar. Bu yöntemde bir şeyler yanlış giderse, bunu düzeltmek için kendileri uğraşırlar. Ayrıca, bir Linux sisteminin nasıl çalıştığını öğrenmenin en iyi yolu da budur.

Bu kılavuz, sizin temel Linux yönetim mantığını ve komutlarını bildiğinizi kabul ederek yazılmıştır. Örneğin; dizinler, dosya sistemleri ve disketler hakkında bilgi sahibi olmanız beklenmektedir. Ayrıca `mount` ve `df` komutlarını kullanabilmeniz beklenmektedir. `/etc/passwd` ve `fstab` dosyalarının ne işe yaradığını ve bunların içeriklerinin neye benzediğini bildiğiniz kabul edilmektedir. Bu NASIL'da adı geçen komutların pek çoğunu root kullanıcı olarak çalıştırmanız gerektiğiniz bilmelisiniz.

Sıfırdan bir açılış disketi oluşturmak karışık bir iş olabilir. Şayet Linux SSS ve ilgili belgeleri okumadıysanız, kendi açılış disketlerinizi yapmaya kalkışmayın. Eğer tüm derdiniz çalışır vaziyette bir açılış disketi ise, hazır

yapılmış olanlarından birisini internetten indirmek çok daha kolaydır. Bunların listesi için [Hazır açılış disketleri](#) (sayfa: 31) bölümüne bakınız.

### 3. Açılış disketleri ve önyükleme süreci

Bir açılış disketi basitçe: bir disket içerisinde bulunan minik bir Linux sistemi diye tanımlanabilir. Büyük boyutlu Linux sistemlerinin gerçekleştirdiği işlevlerin pek çoğunu gerçekleştirmelidir. Bir açılış disketi yapmaya geçmeden önce, Linux açılış sürecini tam olarak anlamalısınız. Biz bu kısımda temel bilgileri anlatacağız, böylece kılavuzun geri kalanını anlamaya yetecek bilgiye sahip olacaksınız. Pek çok ayrıntı ve olası seçenekler, bu kılavuz içinde açıklanmamaktadır.

#### 3.1. Önyükleme süreci

Bütün PC sistemleri önyükleme sürecini ROM (BIOS) içindeki kodu çalıştırıp, önyükleme aygıtının 0.silindirin 0. sektöründen yükleyerek başlatır. Bu önyükleme aygıtı genellikle birincil disket sürücüdür (DOS için **A:** ve Linux için `/dev/fd0` ile ifade edilir). Daha sonra BIOS bu sektördeki kodu çalıştırmayı dener. Pek çok önyükleme yapılabilen diskte, 0.silindirin 0. sektörü şunlardan birini ihtiva eder:

- LILO gibi bir önyükleyicinin kodu (bu kod çekirdeği yükler ve düzgün olarak çalışmasını sağlar)
- Linux gibi bir işletim sistemi çekirdeğinin başlangıcını.

Şayet Linux çekirdeği bir diskete ham biçimde kopyalanmışsa, disketin ilk sektörü, Linux çekirdeğinin ilk sektörü olacaktır. Bu ilk sektör, önyükleme aygıtında bulunan çekirdeğin kalan kısımlarını yükleyerek önyükleme sürecinin devamını sağlar.

Çekirdek tamamen diskete/diske yüklenmişse, ilk sektör aygıt sürücülerini ve dahili veri yapısını ilklendirir. Bu işlem bir kere tamamlanınca, **ramdisk** diye adlandırılan özel bir yere yerleştirilen bir biteşleme sıra gelir. Buradan, kök dosya sistemini nerede ve nasıl bulunacağı bilgisi alınır. **Kök dosya sistemi** “/” şeklinde bağlanan bir dosya sistemidir. Çekirdeğe bu dosya sistemini nerede bulabileceği bilgisini verilmelidir. Şayet orada yüklenebilir bir biteşlem bulamazsa, sistem kapatılır.

Bazı önyüklemelerde — genellikle disketten yapılan önyüklemelerde — kök dosya sistemi bir ramdisk’e yüklenir. Bir ramdisk, disket muamelesine tabi tutulan bir RAM alanıdır. RAM, bir disketten çok daha hızlı olduğu için, ramdisk üzerinden yapılan sistem açılışları çok daha hızlı olmaktadır. Ayrıca, çekirdek sıkıştırılmış bir dosya sistemini ramdisk’e yükleyebilir ve orada açabilir. Bu sayede tek bir diskete çok daha fazla dosya sığdırılmış olur.

Kök dosya sistemi yüklenip, bağlandıktan sonra şuna benzer bir ileti göreceksiniz:

```
VFS: Mounted root (ext2 filesystem) readonly.
```

Sistem, kök dosya sistemini başarılı bir şekilde yükledikten sonra, **init** çalıştırılır (`/bin` veya `/sbin` içinden). **init**, `/etc/inittab` dosyasından ayarlarını okur, **sysinit**’i işaret eden bir satır arar ve daha sonra ismi geçen betiği çalıştırır. **sysinit** betiği genellikle, `/etc/rc` veya `/etc/init.d/boot` gibi birşeydir. Bu betik, temel sistem servislerini düzenleyen, — örneğin sabit disk üzerinde **fsck**’nın çalışmasını sağlayan — gerekli çekirdek modüllerini yükleyen, ağı ve takas alanını ilklendiren ve `/etc/fstab` içinde bahsi geçen dosya sistemlerini bağlayan kabuk komutları topluluğudur.

Bu betik genellikle, modüler başlangıçları sağlamak için, başka betikleri de çalıştırır. Örneğin; genel SysVinit yapısı içinde, `/etc/rc.d/` dizini, sistem servislerinin açılış ve kapanış durumlarını düzenleyen bilgileri içeren dosyaların bulunduğu pek çok alt dizin ihtiva eder. Bu dizin oldukça karışık bir yapıya sahiptir. Bununla birlikte, bir açılış disketindeki **sysinit** betiği oldukça basit bir yapıdadır.

**sysinit** betiğinin çalışması bitince, denetim tekrar **init**'e geçer. **inittab** içinde tanımlanmış olan **initdefault** anahtar sözcüğü ile belirtilen **öntanımlı çalışma seviyesine** girilir. Çalışma seviyesi satırı, genellikle **getty** gibi bir uygulamayı işaret eder. Bu uygulama konsollar ve tty'ler arası iletişimi kontrol etmekten ve düzenlemekten sorumludur. Sisteme girişiniz esnasında gördüğünüz "**login:**" isteminden **getty** sorumludur. Kullanıcının adını ve parolasını girmesinden sonra, bu bilgilerin geçerlilik ve tutarlılığını kontrol etmek için, **login** uygulaması çalıştırılır.

### 3.2. Disk çeşitleri

Temel açılış sürecine göz attıktan sonra, bahsi geçen disklerin çeşitlerini tanımlayabiliriz. Biz burada diskleri dörde ayıracağız. Aksi belirtilmediği sürece "disk" kelimesi disketleri tanımlamaktadır. Bunun yanında, burada belirtilen pek çok şey sabit diskler içinde geçerlidir.

#### önyükleme [İng: boot]

Açılış yapabilen bir çekirdek ihtiva eden bir disk. Bu disk önce çekirdeği yüklemek ve ardından başka bir diskten kök dosya sistemini yüklemek için kullanılabilir. Bir önyükleme disketindeki çekirdeğe, kök dosya sisteminin yolu bildirilmelidir.

Sıklıkla, önyükleme diskleri, kök dosya sistemlerini başka bir disketten yüklerler. Başka bir disketten yüklemek yerine, kök dosya sistemini sabit diskten yükleyecek şekilde ayar yapmak da mümkündür. Bu genellikle yeni bir çekirdek sürümünü denerken kullanılır (aslında "**make zdisk**" komutu, çekirdek kaynak kodlarını kullanarak bu tür bir açılış disketi yapmanızı sağlar).

#### kök [İng: root]

Bir Linux sisteminin çalışması için gereken dosyaları içeren bir dosya sisteminin bulunduğu disk. Bu tür diskler çekirdek veya önyükleyici içermek zorunda değildir.

Çekirdek bir kere yüklendikten sonra, kök disklerini diğer disklerle bağımlı olmadan kullanabilirsiniz. Genelde, kök diski ramdisk'e özdevinimli olarak kopyalanır. Bu sayede kök diske erişim çok daha hızlı olur ve disket sürücüyü uygulama disketlerini kullanabilmek için serbest bırakmış oluruz.

#### önyükleme ve kök [İng: boot/root]

Hem çekirdek hem de kök dosya sistemi içeren diskler. Başka bir deyişle: sabit disk olmadan bir Linux sistemini yükleyecek ve çalışmasını sağlayacak her şeyi içeren bir disket. Bu tür bir diskin yararı küçük boyutlu olmasıdır; gereken her şey tek bir disk üzerindedir. Bununla birlikte, bu yararı aynı zamanda zorluklarını da yanında getirir: tek bir diskete her şeyi sığdırmak, sıkıştırılmış olsa bile, oldukça zordur.

#### uygulamalar

Bir dosya sistemi içeren ama kök dosya sistemi olarak bağlanması tasarlanmamış olan diskler. İlave verilerin bulunduğu disk. Bu tür diskleri, kök diske sığdırmayı beceremediğimiz uygulama ve verileri taşımak için kullanacağız.

Genel olarak, "bir açılış disketi yapmak"tan bahsederken, hem önyükleme, hem de kök bölümleri yapmaktan bahsediyor olacağız. Bu iki bölüm tek bir diskette olabileceği gibi, iki ayrı diskette (önyükleme + kök diskleri) şeklinde de olabilir. Kurtarma diskleri için en esnek yapı, sanırım ayrı ayrı disklerin kullanılması olsa gerek. Gerekli uygulama ve veriler için de ek bir uygulama disketleri kullanabilirsiniz.

## 4. Bir kök dosya sisteminin oluşturulması

Bir kök dosya sisteminin oluşturulması, sistemin çalışması için gerekecek dosyaları seçmeyi zorunlu kılar. Bu bölümde, sıkıştırılmış bir kök dosya sisteminin nasıl yapılacağını göreceğiz. **Sıkıştırılmış dosya sistemini**,

doğrudan kök dosya sistemi olarak bağlanan bir disket üzerinde oluşturmanın daha az kullanılan bir diğer yöntemi [Ramdisksiz kök dosya sistemleri](#) (sayfa: 23) bölümünde anlatılmaktadır.

## 4.1. Genel bakış

Bir kök dosya sistemi, tam bir Linux sistemi desteklemek için gerekli olan herşeyi içermelidir. Bunu yapabilmek için, bu disket bir Linux sistemde gerekli olan asgari araçları içermelidir:

- temel dosya sistemi yapısını;
- en azından şu dizinleri: `/dev`, `/proc`, `/bin`, `/etc`, `/lib`, `/usr`, `/tmp`;
- temel araçları: `sh`, `ls`, `cp`, `mv`, vs.;
- ayar dosyalarının en temel olanlarını: `rc`, `inittab`, `fstab`, vs.;
- aygıtları: `/dev/hd*`, `/dev/tty*`, `/dev/fd0`, vs.;
- araçlar tarafından kullanılan temel işlevleri sağlayan çalışma anı kütüphanelerini.

Elbette ki, herhangi bir sistem, üzerinde bir şeyler çalıştırabildiğimiz sürece işe yarar. Bir kök disket aşağıdakileri veya bunlara benzer şeyleri yapabildiğiniz sürece bir anlam ifade eder:

- Başka bir sürücüdeki dosya sistemini kontrol edebilmeli. Örneğin, sabit diskinizdeki kök dosya sistemini denetleyebilmeli, sistemi başka bir sürücüden açtıktan sonra, sizin orjinal kök bölümünüzü, bağlı değil iken, `fsck` ile denetlemenizi sağlamalı.
- Arşivlenmiş ve/veya sıkıştırılmış orjinal kök dosya sisteminizin hepsini veya bir bölümünü `cpio`, `gzip`, `tar` veya `ftape` gibi araçlar kullanarak tekrar yükleyebilmeli.

**Sıkıştırılmış dosya sistemi** yapmayı burada açıklayacağız. Bu isimle anılmasını sebebi, sıkıştırılmış olması ve önyükleme işlemi sonrası ramdisk içine açılmasıdır. Sıkıştırılmış dosya sistemi kullanarak 6MB'ye yakın veriyi tek bir 1440k'lık bir diskete sığdırmak mümkündür. Dosya sistemi tek bir disketin boyutunda daha büyük olduğu için, dosya sistemini disket üzerinde hazırlamak mümkün değildir. Bu nedenle, dosya sistemini başka bir yerde hazırlamalı, orada sıkıştırıp daha sonra diskete kopyalamalıyız.

## 4.2. Dosya sisteminin oluşturulması

Bu tür bir kök dosya sistemi oluşturmak için, sıkıştırma öncesi bütün araç ve dosyaları içine alabilecek kadar büyük bir yere ihtiyacımız var. 4MB kadar kapasiteye sahip bir aygıtı ihtiyacımız var. Bunun için çeşitli seçenekler mevcuttur:

- Bir **ramdisk** (`AYGIT = /dev/ram0`) kullanın. Bu durumda, fiziksel bellek, bir disk sürücüsü gibi kullanılır. Ramdiskimiz yeterince büyük olmalıdır. Şayet LILO kullanıyorsanız, RAM içinde yaratılacak ramdiskin boyutunu sınırlayan bir satır (`RAMDISK=nnn` gibi) olup olmadığını `/etc/lilo.conf` ayar dosyası içinden kontrol edin. Öntanımlı değer 4096K'dır ve bu yeterli olacaktır. 8MB'den daha küçük RAM'i olan bir makine üzerinde böyle bir ramdisk yapmaya kalkışmayın. `/dev/ram0`, `/dev/ram` veya `/dev/ramdisk` gibi bir aygıtı sahip olduğunuzdan emin olun. Şayet bu tür bir aygıt sisteminizde yok ise `mknod` kullanarak `/dev/ram0` aygıtını oluşturun (`ana=1`, `alt=0`). Aygıt dosyaları yapmak için gerekli bilgiyi `man mknod` veya [Linux Sistem Yöneticisinin Kılavuzu](#)<sup>(B11)</sup> içinde bulabilirsiniz.
- Şayet yeterli boyutta kullanılmayan bir disk bölümüne sahipseniz, bu da kabul.
- Bir **geridönüş aygıtı** kullanın. Bu aygıt ile, bir disk dosyasına bir aygıtmış gibi davranabilirsiniz. Bir geridönüş aygıtı kullanarak, sabit diskiniz üzerinde, dosya sistemini yapmamıza yetecek kadar büyük bir dosya oluşturabilirsiniz (3MB civarı).



Geridönüş aygıtlar hakkında bilgi almak için: **man losetup**. Şayet sisteminizde **losetup** yok ise, **mount** ve **umount** sürümleri ile uyumlu olarak <ftp://ftp.win.tue.nl/pub/linux-«local/»utils/«linux/»> adresinden **util-linux** paketi içinde alabilirsiniz.

Şayet sisteminizde bir dönüş aygıtı yok ise (**/dev/loop0**, **/dev/loop1**, vs.), “**mknod /dev/loop0 b 7 0**” komutunu kullanarak bir tane oluşturmalsınız. Özel **mount** ve **umount** çalıştırılabilirlerini de yükledikten sonra, diskiniz üzerinde yeterli kapasiteye sahip geçici bir dosya oluşturun (örneğin **/tmp/fsfile**). **nnn** blokluk bir dosya oluşturmak için şu komutu kullanabilirsiniz:

```
dd if=/dev/zero of=/tmp/fsfile bs=1k count=nnn
```

Aşağıdaki **aygıt** kelimesinin yerine dosya ismini koyun. Bağlama işlemi yapacağınız zaman, **mount** komutu ile birlikte **-o loop** seçeneğini kullanmanız gerekmektedir. Böylece **mount** uygulamasına bağlanacak şeyin bir dönüş aygıt olduğu bildirilmiş olur.

Bu seçeneklerden birisini seçtikten sonra **aygıt**'ı hazırlayın:

```
dd if=/dev/zero of=aygıt bs=1k count=4096
```

Bu komut aygıtı sıfırlarla doldurur.



### Önemli

Bir aygıtın sıfırlarla doldurulması, dosya sistemi daha sonra sıkıştırılacağı için kritik bir öneme haizdir. Azami sıkıştırmayı sağlamak için kullanılmayan bütün bölümler sıfır ile doldurulmalıdır. Dosya sistemindeki bir dosyayı silerken veya taşıırken bunu aklınızda bulundurun. Dosya sistemi blokları doğru bir şekilde tekrar yerleştirir *ama onları tekrar sıfırla doldurmaz*. Şayet sıklıkla silme ve kopyalama yapıyorsanız, sıkıştırılmış dosya sisteminiz ihtiyaç duyduğumuzdan daha büyük olabilir.

Daha sonra dosya sistemini oluşturun. Linux çekirdeği, ramdiske özdevinimli olarak yüklenen, iki çeşit dosya sistemi kabul eder. Bunlar minix ve ext2'dir. ext2'yi tercih etmenizi öneririz. Şayet ext2 kullanırsanız, öntanımlı değerden daha fazla dosya düğümü oluşturmak için kullanılan **-N** seçeneğinin oldukça işe yarar olduğunu görürsünüz. **-N 2000** değerini kullanmanızı öneririm. Böylece dosya düğümü sıkıntısı yaşamazsınız. Ayrıca olarak, gereksiz **/dev** dosyalarını silerek de dosya düğümü tasarrufu sağlanabilir. **mke2fs**, öntanımlı olarak 1.44MB'lık bir diskette 360 adet dosya düğümü oluşturur. Benim yaptığım kök diskette 120 tane dosya düğümü yeterli olmuştu. Eğer **/dev** içindeki bütün aygıtları eklemeye niyetiniz varsa, rahatlıkla 360 düğüm sınırını geçebilirsiniz. Sıkıştırılmış bir dosya sistemi kullanmak, daha büyük bir dosya sistemine ve daha çok düğümüne izin verir fakat hala ya düğüm sayısını artırmanız ya da kullandığınız dosya sayısını sınırlamanız gerekmektedir.

Bu nedenle kullanacağımız komut şuna benzer olacaktır:

```
mke2fs -m 0 -N 2000 aygıt
```

(Şayet bir geridönüş aygıtı kullanıyorsanız, **aygıt** yazan yerde kullandığınız dosyanın adı olmalıdır.)

**mke2fs** kullanılabilir alanı kendiliğinden tespit eder ve gerekli ayarlamaları yapar. “**-m 0**” parametresi root kullanıcı için alan ayrılmasını önlemiş olur ve böylece diskette daha fazla alan bulunması sağlanmış olur.

Daha sonra aygıtı bağlayın:

```
mount -t ext2 aygıt /mnt
```

(Şayet **/mnt** bağlantı noktası sisteminizde yoksa, bunu oluşturmak zorundasınız.) Bölümün geri kalanında hedef dizin isimleri **/mnt**'ye göre ayarlanacaktır.

## 4.3. Dosya sisteminin doldurulması

Kök dosya sisteminiz için gerekli olan mümkün en az sayıda dizin şunlardan oluşabilir<sup>(2)</sup>:

- `/dev` — aygıt dosyaları, G/Ç'lar için gereklidir.
- `/proc` — proc dosya sistemi tarafından ihtiyaç duyulan dizin
- `/etc` — sistem yapılandırma dosyaları
- `/sbin` — Kritikal sistem çalıştırılabilirler
- `/bin` — sistemin parçası olarak kabul edilen çalıştırılabilirler
- `/lib` — çalışma zamanı desteği sağlayan paylaşımlı kütüphaneler
- `/mnt` — diğer disklerdeki bakım/onarım işleri için bağlantı noktası
- `/usr` — ilave araçlar ve uygulamalar

Bu dizinlerden üç tanesi boş olacaktır, bu nedenle sadece **mkdir** kullanarak bunları oluşturmak yeterli olacaktır. `/proc` dizini, proc dosya sisteminin yerleştiği bir dizindir. `/mnt` ve `/usr` dizinleri, sistem çalıştıktan sonra kullanılacak bağlantı noktalarıdır.

Geri kalan dizinler aşağıda ki bölümlerde açıklanmıştır.

#### 4.3.1. `/dev`

Bir Linux sistemi için gerekli olan bütün aygıtların dosyalarının bulunduğu bir dizindir. Dizin kendisi normal bir dizindir ve **mkdir** komutu ile normal bir şekilde oluşturulabilir. Aygıt özel dosyaları ise, **knod** komutu kullanılarak özel bir yöntemle oluşturulmalıdır.

Aynı zamanda bu işi yapmanın kısa bir yolu da vardır: sabit diskinizdeki aygıt dosyalarını disketin `/dev` dizinine kopyalayabilirsiniz. Bu işlem esnasında yapmanız gereken tek şey **-R** seçeneğini kullanmaktır. Böylece, dizin içindeki dosyaların içerikleri kopyalanmadan sadece dizin kopyalanır. Örneğin:

```
cp -dpR /dev/fd[01]* /mnt/dev
cp -dpR /dev/tty[0-6] /mnt/dev
```

Disketin `/mnt` dizinine bağlı olduğunu varsayarsak, **dp** seçenekleri sembolik bağların yine bağ olarak kopyalandığından, dosya özniteliklerinin korunduğundan ve iyelik bilgilerini saklandığından emin olmamızı sağlar.

Şayet bu işi zor yoldan halletmek isterseniz, istediğiniz aygıtın ana ve alt numaralarını görebilmek için **ls -l** komutunu kullanın. Daha sonra disket içerisinde **knod** komutunu kullanarak istediğiniz aygıtı oluşturun.

Aygıt dosyalarını oluştururken, ihtiyaç duyduğunuz özel aygıtların, diskette bulunduğundan emin olun. Örneğin; **ftape** dosyasını, teyp aygıtları kullanır ve şayet disketi kullanarak teyp sürücünüze erişmek istiyorsanız, teyp aygıtı dosyasını oluşturmalsınız.

Her bir özel aygıt için bir dosya düğümüne ihtiyacınız olduğunu ve disketlerde düğümlerin nadir bulunan değerli varlıklar olduğunu aklınızdan çıkarmayın. Öyle bol keseden dosya düğümü dağıtmayın. Ekleyeceğiniz aygıt dosyaları için seçici davranın. Şayet, sisteminizde SCSI diskler yok ise, `/dev/sd*` dosyalarını oluşturmayın. Seri portlardan iletişim sağlamayacaksanız, `/dev/ttyS*` aygıtlarına ihtiyacınız yok demektir.

Şayet, kök dosya sisteminizi oluştururken: aygıt içinde yer kalmadığı şeklinde bir hata iletisi görürseniz ama **df** komutunun çıktısı aygıtta yer olduğunu gösterirse, elinizdeki dosya düğümlerini bitirmişsiniz demektir. **df -i** komutu ile düğüm kullanım durumunu görebilirsiniz.



#### Önemli

Şu dosyaların bu dizin içinde bulunduğunda emin olun: `console`, `kmem`, `mem`, `null`, `ram0` ve `tty1`.

### 4.3.2. /etc

/etc dizini sistem yapılandırma dosyalarını barındırır. İçeriği, hangi süreçlerin çalıştıracağınıza bağlı olarak değişir. Pek çok sistemde, bunlar, üç gruba ayrılır:

1. Her zaman gerekli olanlar: `rc`, `fstab`, `passwd`, vs.
2. Gerekli olabilecek ama asla emin olamayacaklarınız. May be required, but no one is too sure.
3. İVIR ZIVIR.

Gerekli olmayan dosyalar aşağıdaki komut ile tespit edilebilir:

```
ls -ltr
```

Bu komut ile en son ulaşılan tarihten itibaren ters sıralama ile, bir dosya listesi görüntülenir. Böylece hiç erişim sağlanmamış dosyalar varsa, bunları gönül rahatlığı içinde yok sayabilirsiniz.

Benim kök disketimde, 15 taneden az ayar dosyam var. Böylece üç değişik türde dosya ile ilgilenmem gerekmekte ve yapmam gereken işler azalmakta:

1. Önyükleme ve kök sistemi için mutlaka yapılandırılmaları gerekenler:
  - a. `rc.d/*` — sistem başlatma ve çalışma seviyesi değiştirme betikleri
  - b. `fstab` — bağlanacak dosya sistemlerinin listesi
  - c. `inittab` — **init** süreci için parametreler, açılış sırasında çalıştırılan ilk süreçtir.
  - d. `gettydefs` — **login** süreci için parametreler, kullanıcının sistme girişinde çalıştırılır.
2. Önyükleme ve kök sistemi için düzenlemem gerekenler:
  - a. `passwd` — kullanıcıların, ev dizinlerinin, vs. bilgilerin bulunduğu önemli bir liste.
  - b. `group` — kullanıcı grupları.
  - c. `shadow` — kullanıcıların parolaları. Bunu dahil etmek zorunda değilsiniz.
  - d. `termcap` — uçbirim yetenekleri veritabanı.

Şayet güvenlik önemli ise `passwd` ve `shadow` dosyaları iyice budanmalı, kullanıcı parolalarının kopyalanması önlenmeli ve gereksiz girdiler çıkarılmalıdır. Böylece disketten açılış yaparken istenmeyen kullanıcı girişleri reddedilir.

`passwd` dosyasında en azından `root` kullanıcının bulunduğundan emin olun. Şayet diğer kullanıcıların da bağlantı yapmasını isterseniz, ev dizinlerinin ve kabukların sistemde var olduğundan emin olun.

`termcap`, uçbirim veritabanıdır ve yüzlerce kB'dır. Sizin önyükleme ve kök disketinizdeki sürümün, sadece işinize yarayacak olan uçbirimleri içerdiğinde emin olun: genellikle sadece `linux` veya `linux-console` girdisi yeterlidir.

3. Ve geriye kalanlar. O an çalışıyorlardı, bu yüzden onlara dokunmadım.

Bunların haricinde, gerçekte ayarlamam gereken sadece iki dosya var ve onlarda hayret verici bir şekilde küçükler:

- `rc` şunları içermeli:

```
#!/bin/sh
/bin/mount -av
/bin/hostname Kangaroo
```

Çalıştırma izni olduğunda emin olun, tepesinde "#" işareti bulunduğundan emin olun ve dosya isimlerinin doğru olduğundan da emin olun. Aslında **hostname**'i çalıştırmanız gerekmekte —yaparsanız hoş olur, yapmazsanız hiçbir şey olmaz.

- **fstab** en azından şunları içermeli:

/dev/ram0	/	ext2	defaults
/dev/fd0	/	ext2	defaults
/proc	/proc	proc	defaults

Çalışan sisteminizdeki **fstab** girdilerini buraya kopyalayabilirsiniz, fakat sabit diskinizdeki bölümleri özdevinimli olarak bağlamaya kalkışmayın. **noauto** anahtar kelimesini bunların yanında kullanmayı unutmayın. Açılış disketini kullanmak zorunda kaldığınız durumlarda, sabit diskinizin kafası biraz karışmış olabilir.

**inittab** içindeki **sysinit** satırı, **rc** veya hangi temel betik kullanılacak ise, ona göre düzenlenmelidir. Ayrıca seri portlar üzerindeki kullanıcıların bağlanamayacaklarından emin olmak isterseniz, sonlarında **ttys** veya **ttys** aygıtları içeren bütün **getty** satırlarının başına yorum işareti "#" getirin. **tty** portlarına dokunmayın ki, konsoldan bağlanabilesiniz.

En küçük yapıda **inittab** dosyası şuna benzer:

```
id:2:initdefault:
si::sysinit:/etc/rc
1:2345:respawn:/sbin/getty 9600 tty1
2:23:respawn:/sbin/getty 9600 tty2
```

Bu **inittab** dosyası, sistemin değişik seviyelerde çalışabileceğini/başlatılabileceğini, çoklu kullanıcı kipine geçebileceğini, vs.. tanımlar. **inittab** içindeki isimleri dikkatlice kontrol edin. Şayet **init**, bu isimleri bulamazsa, sistem açılışı gerçekleşmez ve hatta bir hata iletisi bile alamayabilirsiniz.

Unutmayın ki bazı uygulamalar olmazsa olmazdır ve başka uygulamalar bunları yerlerini bilerek çağırdığından yerleri değiştirilemez. Örneğin; benim sistemimde, **/etc/shutdown** **/etc/reboot**'u çalıştırdığından şayet **reboot**'u **/bin/reboot**'a taşırsam, bir **shutdown** komutu, **reboot**'u bulamayacağı için çalışmayacaktır.

İşin geri kalan kısmı, **/etc** dizinindeki tüm metin dosyalarını ve gerekli olacağından emin olmanız bile **/etc** dizinindeki bütün çalıştırılabilir dosyaları kopyalamaktan ibarettir. Yardımcı olması açısından **Kök dosya sistemi listesi örneği** (sayfa: 33) içindeki örnek listeye göz atabilirsiniz. Muhtemelen buradaki dosyalar yeterli olacaktır ama sistemler arası farklılıklar olduğu için, buradaki dosyaların, sizin sisteminizdekiler ile aynı olduğunda emin olamazsınız. Emin olmanın tek yolu **inittab** ile başlamak ve daha nelerin gerektiğini bulmaktır.

Pek çok sistem, değişik çalışma seviyelerine geçiş sağlayan kabuk betiklerini saklamak için **/etc/rc.d/** dizinini kullanırlar. Asgari gereksinim tek bir **rc** betiğidir ama **inittab** ve **/etc/rc.d** dizinindekileri doğrudan diskete kopyalamak ve daha sonra bir disket ortamında işi olamayacak kabuk betiklerini temizlemek daha basit olabilir.

#### 4.3.3. /bin ve /sbin

**/bin** dizini, temel işlemleri (**ls**, **mv**, **cat** ve **dd** gibi) yapmak için gerekli olan ilave araçları saklamak için elverişli bir dizindir. **/bin** ve **/sbin** dizininde bulunması gereken dosyaların örnek bir listesi için **Kök dosya sistemi listesi örneği** (sayfa: 33) bölümüne bakınız. Yedekleri geri yüklemek için gereken araçların (**cpio**, **tar** ve **gzip** gibi) hiçbirisi bu listede yoktur. Çünkü ben bu araçları başka bir uygulama disketine yerleştirdim. Amacım önyükleme ve kök disketlerinde yer tasarrufu sağlamaktır. Önyükleme ve kök disketinden açılış yapıldıktan sonra, bunların içindekiler ramdiske yüklenir ve disket sürücü başka disketler için boşalmış olur. Ben bu uygulama disketini genellikle **/usr** altına bağlarım.

Uygulama disketi yapımı [Uygulamalar disketinin oluşturulması](#) (sayfa: 23) içinde anlatılmaktadır. Yedekleme yapmak için, yedekleme araçlarının aynı sürümdeki birer kopyasını kullanmak iyi bir fikir olacaktır. Bu sayede sizin yedeklerinizi okuyamayan araçların yeni sürümlerini yüklemek ile vakit kaybetmezsiniz. Sisteminizdeki yedekleri hangi araçlar ile yaratmış iseniz, aynı araçların aynı sürümlerini kullanarak bir uygulama disketi oluşturun.



### Önemli

Şu uygulamaları yüklediğinizden emin olun: **init**, **getty** veya eşdeğeri, **login**, **mount**, rc betiklerini çalıştırabilecek bir kabuk ve **sh**'dan bu kabuğa bir bağ.

#### 4.3.4. /lib

Bu dizine gerekli olacak paylaşımlı kütüphaneleri ve yükleyicileri koyabilirsiniz. Şayet paylaşımlı kütüphaneler **/lib** dizini altında bulunamazsa, sistem açılmayacaktır. Şayet şanslı iseniz, bunun sebebini söyleyen bir hata iletisi ile karşılaşabilirsiniz.

Hemen hemen bütün uygulamalar **libc** kütüphanesine ihtiyaç duyarlar; **libc.so.N**: buradaki **N** geçerli sürüm numarası anlamına gelir. **/lib** dizininizi kontrol edin. Tam sürüm adını taşıyan bir dosya ismine verilmiş olan sembolik bir bağ olarak bir **libc.so.N** dosyası vardır:

```
# ls -l /lib/libc*
-rwxr-xr-x 1 root root 4016683 Apr 16 18:48 libc-2.1.1.so*
lrwxrwxrwx 1 root root 13 Apr 10 12:25 libc.so.6 -> libc-2.1.1.so*
```

Bu durumda **libc-2.1.1.so**'yu kullanmamız gerekecektir. Yüklemeniz gereken diğer kütüphaneleri bulmak için, yüklemeyi planladığınız çalıştırılabilirlerin bağımlılıklarını **ldd** ile kontrol ediniz. Örnek:

```
# ldd /sbin/mke2fs
libext2fs.so.2 => /lib/libext2fs.so.2 (0x40014000)
libcom_err.so.2 => /lib/libcom_err.so.2 (0x40026000)
libuuid.so.1 => /lib/libuuid.so.1 (0x40028000)
libc.so.6 => /lib/libc.so.6 (0x4002c000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Sağ tarafta bulunan bütün dosyalar gereklidir. Dosya bir sembolik bağ olabilir.

Bazı kütüphanelerin oldukça büyük olduğunu ve kök dosya sisteminize sığmayabileceğini aklınızda bulundurun. Örneğin; yukarıda listelenen **libc.so** yaklaşık olarak 4MB'dır. Kütüphaneleri kök dosya sisteminize kopyalarken, muhtemelen onları soymanız gerekecek. Ayrıntılar için [Kök dosya sistemi boyutunun küçültülmesi](#) (sayfa: 21) bölümüne bakınız.

**/lib** içine, kütüphaneleri yükleyecek bir yükleyici de yerleştirmelisiniz. Yükleyici ya **ld.so** (A.OUT kütüphaneleri içindir ve güncelliğini yitirmiştir) ya da **ld-linux.so** (ELF kütüphaneleri için) olmalıdır. **ldd**'nin yeni sürümleri hangi yükleyiciye ihtiyacınız olduğunu söyleyecektir –yukarıdaki örnekte görülebileceği üzere– fakat eski sürümler bunu yapmazlar. Şayet hangisine ihtiyaç duyduğunuzdan emin değilseniz, kütüphane için **file** komutunu çalıştırın:

```
# file /lib/libc.so.4.7.2 /lib/libc.so.5.4.33 /lib/libc-2.1.1.so
/lib/libc.so.4.7.2: Linux/i386 demand-paged executable (QMAGIC), stripped
/lib/libc.so.5.4.33: ELF 32-bit LSB shared object, Intel 80386, version 1, stripped
/lib/libc-2.1.1.so: ELF 32-bit LSB shared object, Intel 80386, version 1, not stripped
```

**QMAGIC**, **4.7.2**'nin A.OUT kütüphanesi olduğunu ve ELF de, **ELF** ile **2.1.1**'in ELF kütüphanesi olduğunu göstermektedir.

Gereken yükleyici(ler)yi, kök dosya sisteminize kopyalayın. Kütüphaneler ve yükleyicileri, içerildikleri çalıştırılabilirler karşı dikkatlice kontrol edilmelidir. Şayet çekirdek gereken bir kütüphaneyi yükleyemezse, herhangi bir hata verilmeksizin çekirdek askıda kalabilir. Alın başınıza bela.

#### 4.4. PAM ve NSS'in sağlanması

Sisteminiz, **ldd** ile görülemeyen `çzdev,ml`, yüklenen kütüphanelere ihtiyaç duyabilir. Şayet bunları sağlayamazsanız, sisteme bağlanma veya açılış disketini kullanma ile sorunlar yaşayabilirsiniz.

##### 4.4.1. PAM (Eklemlenebilen Kimlik Kanıtlama Modülleri)

###### (Pluggable Authentication Modules)

Şayet sisteminizde PAM kullanılıyor ise, açılış disketinizde bunun için bazı hazırlıklar yapmalısınız. Kısaca tanımlamak gerekirse, PAM: kullanıcıları doğrulama ve onların servislere erişimini denetleyen gelişmiş bir modüler yöntemdir denebilir. Sisteminizde PAM kullanılıp kullanılmadığını anlamanın en kolay yolu, **login** çalıştırılabilirli üzerinde **ldd** komutunu kullanmaktır. Şayet çıktıda **libpam.so** var ise, PAM'e ihtiyacınız var demektir.

Şans eseri, açılış disketleri için güvenlik bir sorun değildir: makinenize fiziksel erişimi olan bir kişi zaten istediği herşeyi yapabilir. Bu nedenle, kök dosya sisteminizde aşağıdakine benzer bir `/etc/pam.conf` dosyası oluşturup, PAM konusunu kapatabilirsiniz:

OTHER	auth	optional	/lib/security/pam_permit.so
OTHER	account	optional	/lib/security/pam_permit.so
OTHER	password	optional	/lib/security/pam_permit.so
OTHER	session	optional	/lib/security/pam_permit.so

Ayrıca `/lib/security/pam_permit.so` dosyasını kendi kök dosya sisteminize kopyalayın. Bu kütüphane sadece 8k'dır ve pek baş ağrıtmaz.

Bu ayarlar, herhangi bir kimsenin sisteminizdeki servis ve dosyalara erişimize izin verir. Özel sebeplerden dolayı, güvenlik, açılış disketiniz için önemli ise, sabit diskinizdeki PAM ayar dosyalarının çoğunu veya hepsini kök dosya sisteminize kopyalamak zorundasınız. PAM belgelerini dikkatlice okuyun ve `/lib/security` içindeki gerekli bütün kütüphaneleri kök dosya sisteminize kopyalayın.

Ayrıca `/lib/libpam.so` dosyasını da önyükleme disketinize kopyalamak zorundasınız. Siz bunu zaten biliyorsunuz, **ldd** komutunu `/bin/login` üzerinde çalıştırdınız değil mi? Böylece bağımlılıkların neler olduğunu öğrenmiş oldunuz.

##### 4.4.2. NSS (İsim Servisi Değiştirici)

###### (Name Service Switch)

Şayet glibc (nam-ı diğer libc6) kullanıyorsanız, isim servisleri için bazı düzenlemeler yapmalısınız yoksa sisteme bağlanamayabilirsiniz. `/etc/nsswitch.conf` dosyası, çeşitli servisleri yürüten veritabanlarını kontrol eder. Şayet ağ üzerinden servislere erişimi düşünmüyorsanız (örneğin, DNS veya NIS aramaları), aşağıdakine benzer basit bir `nsswitch.conf` dosyası hazırlamanız yeterli olacaktır:

```
passwd:    files
shadow:    files
group:     files
hosts:     files
services:  files
```

```
networks:  files
protocols: files
rpc:       files
ethers:    files
netmasks: files
bootparams: files
automount: files
aliases:   files
netgroup:  files
publickey: files
```

Burada bütün servislerin sadece yerel dosyaları yürütülebileceği belirtilmiştir. Ayrıca `/lib/libnss_files.so.X` dosyasını eklemeniz gerekmekte (burada `X` değeri glibc 2.0 için 1 ve glibc 2.1 için 2'dir) eklemeniz gerekmekte. Bu kütüphane, dosya aramalarını gerçekleştirmek için özdevimli yüklenecektir.

Şayet açılış disketiniz üzerinden ağa çıkmayı planlıyorsanız, daha ayrıntılı bir `nsswitch.conf` dosyası oluşturmanız gerekmekte. Tanımladığınız her bir `servis` için bir `/lib/libnss_service.so.1` dosyası eklemeniz gerekmektedir.

## 4.5. Modüller

Şayet modüler yapıda bir çekirdeğe sahipseniz, açılış disketi ile açılış yapıldıktan sonra hangi modüllerin yüklenmesini istediğinizi tespit etmelisiniz. Şayet yedekleriniz teyp üzerinde ise **ftape** ve **zftape** modüllerinin yüklenmesini isteyebilirsiniz veya SCSI aygıtlarınız varsa SCSI modüllerinin yüklenmesini isteyebilirsiniz. Acil durumlarda ağa erişim istiyorsanız, PPP veya SLIP modüllerini yüklemeniz kuvvetle muhtemeldir.

Bu modüller `/lib/modules` içinde yer alabilir. Ayrıca **insmod**, **rmmod** ve **lsmod**'u da eklemelisiniz. Özdevimli yüklenmesini istediğiniz modüllere bağlı olarak **modprobe**, **depmod** ve **swapout**'u da ekleyebilirsiniz. Şayet **kernel** kullanıyorsanız, onu da `/etc/conf.modules`'e ekleyin.

Bunların yanında, modülleri kullanmanın temel faydası, kritik öneme haiz olmayan modülleri bir uygulama disketi içinde tutabilmenizdir. Bu sayede kök diskette daha çok alan size kalır. Şayet pek çok değişik aygıt ile uğraşmanız gerekcek ise; bu yaklaşım, gömülü sürücülere sahip daha büyük bir çekirdek yapmayı tercih edilebilir kılar.



### Önemli

Sıkıştırılmış bir ext2 dosya sisteminden açılış yapabilmek için ramdisk ve ext2 desteğini çekirdeğe gömülü olarak vermeniz gerekir. BUNLARI MODÜL OLARAK KULLANAMAZSINIZ.

## 4.6. Son olarak bazı ayrıntılar

Bazı sistem uygulamaları, mesela **login**, `/var/run/utmp` dosyası ve `/var/log` dizini yok ise şikayet edip dururlar. Bu nedenle şu komutları verin:

```
# mkdir -p /mnt/var/{log,run}
# touch /mnt/var/run/utmp
```

En sonunda, gerekli bütün kütüphaneleri ayarladıktan sonra, kök dosya sistemindeki `/etc/ld.so.cache` dosyasını yeniden yapılandırmak için **ldconfig** komutunu kullanın. Bunun şu şekilde yapabilirsiniz:

```
# ldconfig -r /mnt
```

## 4.7. Paketleme



Kök dosya sistemi inşaatını bitirdimize göre, onu sistemden ayırılım ve bir dosyaya kopyalayıp sıkıştırılalım:

```
# umount /mnt
# dd if=aygit bs=1k | gzip -v9 > rootfs.gz
```

Bu işlemin sonunda `rootfs.gz` isimli bir dosyaya sahip olacaksınız. Bu sizin sıkıştırılmış dosya sisteminizdir. Bu dosyanın boyutunun diskete sığacağından emin olun. Şayet boyutu disket boyutundan büyük ise, geri dönmeli ve bazı dosyaları kaldırmalısınız. Kök dosya sistemini küçültmek hakkında bazı öneriler [Kök dosya sisteminin boyutunun küçültülmesi](#) (sayfa: 21) bölümünde bulunabilir.

## 5. Çekirdek Seçimi

Bulunduğunuz noktada, sıkıştırılmış bir kök dosya sistemine sahipsiniz. Şimdiki adımımız bir çekirdek seçmek veya derlemek. Pek çok durum için, kullandığınız sistemdeki çekirdeği diskete kopyalamak ve bununla açılış yapmak mümkün olabilir. Bununla birlikte, kendi çekirdeğinizi kendiniz oluşturmak isteyeceğiniz zamanlar da olabilir.

Birinci sebep: boyut. Şayet tek bir önyüklemeye ve kök disketi elde etmek istiyorsanız, çekirdek disketteki en büyük dosya olacak ve hali ile siz de bu boyutu küçültmek isteyeceksiniz. Çekirdek boyutunu küçültmek için, arzu edilen sistemde sadece gerekli olacak işlevleri yerine getirecek bir çekirdek düzenlemek yeterli olacaktır. Bunun anlamı, ihtiyacınız olmayan herşeyi atın demektir. Safra atma zamanı geldi millet. Ağ terkedilecek birinci şey olabilir. Kullanmayacağınız disk sürücülerini ve diğer aygıtları peşi sıra göndermek de iyi bir tercihtir. Daha önce de söylediğim gibi, ramdisk ve ext2 desteğini çekirdeğinize *gömmelisiniz*.

Gerekli olacak asgari gereksinimleri çekirdeğe gömdükten sonra, başka neler lazım olacak diye çalışmaya başlayabilirsiniz. Galiba, önyüklemeye ve kök disketi için en gerekli olan şey, bozulmuş dosya sistemlerini kontrol etmek ve yeniden düzeltmektir. Bunu yapabilmek için de çekirdek desteğine ihtiyacımız var. Örneğin; yedeklerinizi **ftape** kullanarak teyp sürücü üzerinde saklıyor iseniz ve kök sürücünüzü ve **ftape**'i içeren sürücünüzü kaybetmişseniz, yedeklerinizi teypten geri yükleyemezsiniz. Linux'u tekrar kurmalı, **ftape**'i tekrar indirip sisteme yüklemeli ve daha sonra yedeklerinize ulaşmayı denemelisiniz.

Buradaki önemli nokta; yedeklerinizi desteklemek için çekirdeğe eklediğiniz G/Ç desteğini, önyüklemeye ve kök disketindeki çekirdeğe de vermeniz gerektiğidir.

Çekirdeğin nasıl derleneceğini anlatan kılavuzlar, çekirdek ile birlikte dağıtılmaktadır. Buradaki adımları izlemek oldukça kolaydır, bu nedenle `/usr/src/linux` altına bakmanız faydalı olabilir. Şayet çekirdeği derlemek ile ilgili sorunlarınız varsa, önyüklemeye ve kök sistemi yapmaya kalkışmamanız gerektiği sonucunu çıkarabilirsiniz. Çekirdek yok ise, bu tür bir sistem yapmaya kalkmanın da bir anlamı yoktur. Derleme işleminde sonra **make zImage** ile çekirdeği sıkıştırmayı unutmayın.

## 6. Herşeyi biraraya getirelim

### *Disket(ler)in yapımı*

Şu anda elimizde bir çekirdek ve sıkıştırılmış bir dosya sistemi mevcut. Eğer önyüklemeye ve kök disketi yapacaksanız, bunların boyutunun tek bir diskete sığacağından emin olun. Şayet iki disketlik önyüklemeye + kök disketlerini yapacaksanız, kök dosya sisteminin tek bir diskete sığacak kadar küçülmüş olduğunda emin olun.

Önyüklemeye disketindeki çekirdeği yüklemek için LILO kullanıp kullanmayacağınıza karar vermek durumundasınız. Ayrıca isterseniz; çekirdeği doğrudan diskete kopyalayabilir ve LILO olmaksızın açılış yapabilirsiniz. LILO kullanmanın faydası, açılış esnasında bazı donanımları ilklendirecek özel parametreleri çekirdeğe verebilmektir (sisteminizdeki `/etc/lilo.conf` dosyasını kontrol edin. Şayet içerisinde `"append=..."` gibi bir satır varsa, muhtemelen bu özelliğe ihtiyacımız olacak demektir). LILO kullanmanın bir götürüsü de daha



karmaşık bir açılış disketi yapmak ve bunun biraz daha fazla bayta mal olmasıdır. **Çekirdek dosya sistemi** olarak adlandırabileceğimiz küçük bir dosya sistemi oluşturup buraya LILO'nun ihtiyaç duyacağı çekirdeği ve birkaç dosyayı koyacağız.

Şayet LILO kullanacaksanız, okumaya devam edin. Çekirdeği doğrudan diskete aktaracaksanız [Çekirdeğin LILO'suz kullanımı](#) (sayfa: 18) bölümüne atlayın.

## 6.1. Çekirdeğin LILO ile kullanımı

İlk olarak, LILO'nun en son sürümüne sahip olduğunuzdan emin olun.

LILO için küçük bir yapılandırma dosyası oluşturmanız gerekiyor:

```
boot      =/dev/fd0
install   =/boot/boot.b
map        =/boot/map
read-write
backup     =/dev/null
compact
image      = KERNEL
label      = Bootdisk
root       =/dev/fd0
```

Buradaki parametrelerin açıklamaları için LILO kullanıcı belgelerine bakınız. Ayrıca `append=...` şeklinde bir satır da eklemek isteyebilirsiniz (sabit diskinizdeki `/etc/lilo.conf` dosyasındaki satırı).

Bu dosyayı `bdlilo.conf` ismiyle kaydedin.

Şimdi, kök dosya sisteminden ayırt edebilmek için **çekirdek dosya sistemi** olarak adlandıracağımız küçük bir dosya sistemi oluşturacağız.

İlk olarak, bu dosya sisteminin boyutuna karar verin. `ls -s KERNEL` komutu ile çekirdeğin boyutunu blok cinsinden öğrenin ve buna 50 ekleyin. Bu 50 blok, diğer dosyalar ve düğümler tarafından kullanılacak blok sayısıdır. Bu sayıyı tam olarak hesaplamak mümkündür ama biz burada 50'yi kullanacağız. Şayet iki disketlik bir set yaparsanız, bu sayıyı abartmanızda bir mahsur yoktur. Nasıl olsa birinci disket sadece çekirdek tarafından kullanılacaktır. Bu sayıya **çekirdek\_blokları** adını verelim.

Sürücüye bir disket koyun (buna `/dev/fd0` diyeceğiz) ve bir ext2 çekirdek dosya sistemi oluşturun:

```
# mke2fs -N 24 -m 0 /dev/fd0 çekirdek_blokları
```

`“-N 24”`, bu dosya sistemi için ihtiyaç duyacağımız 24 tane dosya düğümünü belirtir. Daha sonra dosya sistemini bağlayın ve `lost+found` dizinini silin. Bunu ardından, LILO için `dev` ve `boot` dizinlerini oluşturun:

```
# mount -o dev /dev/fd0 /mnt
# rm -rf /mnt/lost+found
# mkdir /mnt/{boot,dev}
```

Daha sonra, `/dev/null` ve `/dev/fd0` dosyalarını oluşturun. Aygıt numaralarını aramak yerine, sabit diskinizden bunları kopyalayabilirsiniz (`-R` seçeneği ile):

```
# cp -R /dev/{null,fd0} /mnt/dev
```

LILO, kendi önyükleyicisinin bir kopyasına ihtiyaç duyar: `boot.b`. Bunu da sabit diskinizden kopyalayabilirsiniz. Bu dosya genellikle `/boot` dizini içindedir.

```
# cp /boot/boot.b /mnt/boot
```

Son olarak, geçen bölümde oluşturduğunuz LILO yapılandırma dosyasını ve çekirdeği kopyalayın. Her ikisi de kök dizine konabilir:

```
# cp bdlilo.conf KERNEL /mnt
```

LILO'nun ihtiyaç duyduğu herşey, artık çekirdek dosya sisteminde. Artık onu çalıştırmaya hazırız. LILo'nun `-r` seçeneği, onu başka kök sistemlere yüklemek için kullanılır:

```
# lilo -v -C bdlilo.conf -r /mnt
```

LILo hatasız olarak çalışmalı ve daha sonra çekirdek dosya sistemi şuna benzemeli:

```
total 361
 1 -rw-r--r-- 1 root root 176 Jan 10 07:22 bdlilo.conf
 1 drwxr-xr-x 2 root root 1024 Jan 10 07:23 boot/
 1 drwxr-xr-x 2 root root 1024 Jan 10 07:22 dev/
358 -rw-r--r-- 1 root root 362707 Jan 10 07:23 vmlinuz
boot:
total 8
 4 -rw-r--r-- 1 root root 3708 Jan 10 07:22 boot.b
 4 -rw----- 1 root root 3584 Jan 10 07:23 map
dev:
total 0
 0 brw-r----- 1 root root 2, 0 Jan 10 07:22 fd0
 0 crw-r--r-- 1 root root 1, 3 Jan 10 07:22 null
```

Dosya boyutları biraz farklı ise, bunu dert etmeyin.

Şimdi, disketi sürücüde bırakın ve [ramdisk sözcüğünün yapılandırılması](#) (sayfa: 18) bölümüne geçin.

## 6.2. Çekirdeğin LILo'suz kullanımı

Şayet LILo kullanmayacaksanız, çekirdeği önyükleme disketine **dd** kullanarak gönderin:

```
# dd if=KERNEL of=/dev/fd0 bs=1k
353+1 records in
353+1 records out
```

Bu örnekte **dd**, 353 tam kayıt + 1 kısmi kayıt yazdı. Bu nedenle, çekirdek disketin ilk 354 bloğu üzerine kurulmuş oldu. Bu sayıyı *çekirdek\_blokları* diye adlandıracağız.

Son olarak, kök aygıtı disket olarak ayarlayın ve burayı okunur/yazılır kipte yüklenecek şekilde düzenleyin:

```
# rdev /dev/fd0 /dev/fd0
# rdev -R /dev/fd0 0
```

İkinci **rdev** komutunda `-R` seçeneğini kullanırken dikkatli olun. Büyük harf ile yazdığınızdan emin olun.

## 6.3. ramdisk sözcüğünün yapılandırılması

Çekirdek biteşlemi içindeki **ramdisk** kelimesi<sup>(3)</sup> diğer seçeneklerin yanısıra kök dosya sisteminin nerede bulunacağını belirtir. Bu kelime **rdev** komutu ile ayarlanabilir. İçeriği aşağıda açıklandığı gibidir:

Bit alanı	Açıklama
0–10	1024 baytlık bloklar halindeki ramdisk'in başlangıç konumu
11–13	kullanılmadı
14	ramdiskin yükleneceğini gösteren bayrak
15	rootfs yüklenmeden önce bir istem geleceğini gösteren bayrak

Şayet 15 numaralı bit ayarlanmış ise, açılışta sürücüye yeni bir disket koymanızı söyleyen bir ekran çıkar. Bu iki disketlik setlerde gereklidir.

Tek veya çift disketlik bir set oluşturmanıza bağlı olarak iki ayrı durum söz konusudur:

1. Şayet tek bir disket oluşturmuşsanız, sıkıştırılmış dosya sistemi çekirdeğin ardı sıra yüklenir. Bu nedenle başlangıcı ilk boş blokta olacaktır (*çekirdek\_blokları* ile aynı olmalıdır). 14. bit 1, 15. bit ise 0 olacaktır.

Örneğin; diyelim ki tek disketlik bir sistem yaptınız ve kök dosya sistemi 253. bloktan (onluk olarak) başlıyor. Ramdiskin kelime değeri 14. bit 1, 15. bit 0 olmak üzere 253 olmalıdır. Değeri hesaplamak için ondalık değerleri toplamak yeterlidir:  $253 + (2^{14}) = 253 + 16384 = 16637$ . Şayet bu sayının nereden geldiğini anlamadıysanız, bilimsel bir hesap makinasına bunu yazın ve ikilik düzene çevirin.

2. Şayet iki disketlik bir set yaptıysanız, kök dosya sistemi 0. bloktan başlar, bu nedenle başlangıç konumu sıfır olacaktır. 14. ve 15. bit 1 olacaktır. Bu durumda ondalık değer ise  $2^{14} + 2^{15} = 49152$  olur.

Bu ramdisk sözcüğünün değerini dikkatlice hesapladıktan sonra, **rdev -r** ile bunu çekirdek biteşlemine yazın. Ondalık değerleri kullandığınızdan emin olun. Şayet LILO kullandıysanız, **rdev**'e verilecek argüman bağlanana çekirdeğin dosya yolu olmalıdır. Örneğin; `/mnt/vmlinuz`; şayet çekirdeği **dd** bunun yerine disket aygıtının ismi (örn, `/dev/fd0`) gelecektir.

```
# rdev -r çekirdek_yolu_veya_disket_aygıtı_değer
```

Şayet LILO kullandıysanız, şimdi disketi ayırın.



### Önemli

**rdev** veya **ramsize** man sayfasının ramdisk boyutu hakkında söylediklerine inanmayın. Bu kılavuz sayfası oldukça eski. Çekirdek 2.0 veya o civardakilerde, ramdisk kelimesi ramdisk boyutunu belirtmez, *ramdisk sözcüğünün yapılandırılması* (sayfa: 18) bölümünün başlangıcındaki tabloyu ifade eder. Ayrıntılı bilgi için, ramdisk.txt dosyasına veya <http://www.linuxhq.com/kernel/v2.4/doc/ramdisk.txt.html> adresine bakınız.

## 6.4. Kök dosya sistemini aktarılması

Son adımımız kök dosya sistemini aktarmaktır.

- Şayet kök dosya sistemi çekirdek ile aynı diskete konacaksa, **dd** komutunu **seek** seçeneği ile kullanarak aktarın. **seek** seçeneği kaç tane bloğun atlanacağını belirtir:

```
# dd if=rootfs.gz of=/dev/fd0 bs=1k seek=çekirdek_blokları
```

- Şayet kök dosya sistemi ikinci bir diskete konacaksa; birinci disketi çıkarın, ikinci disketi sürücüye takın ve daha sonra kök dosya sistemini aktarın:

```
# dd if=rootfs.gz of=/dev/fd0 bs=1k
```

Tebrikler, başardınız!



### Önemli

Yapmış olduğun açılış disketlerini, acil durumlar için kenara koymadan önce mutlaka sınavın. Şayet disket ile sistemi açamazsanız, okumaya devam edin.

## 7. Sorun çözmek ya da Yenilginin Acısı

Açılış disketleri yapmak için giriştiğiniz ilk bir kaç sefer başarısızlıkla sonuçlanabilir. Bir kök disket yapmak için genel yaklaşım; sisteminizdeki hazır parçaları bir araya toplamak, bunları disket tabanlı bir hale sokmak ve konsoldan açılış yapmaktır. Bir kere, sizinle iletişime geçimi, savaşın yarısını kazanmışsınız demektir. Sistem çalıştığı sürece, bağımsız sorunları çözmek mümkün olabilir. Şayet herhangi bir uyarı vermeden, sistem askıda kalırsa, bu problemi halletmek biraz zor olabilir. Sistemin sizinle iletişimi kestiği noktada meydana gelen problemleri bulmak için önerilen süreç şudur:

- Şuna benzer bir ileti alabilirsiniz:

```
Kernel panic: VFS: Unable to mount root fs on XX:YY
```

Bu genel bir sorundur ve birkaç sebebi vardır. İlk olarak, **XX:YY** aygıtının kodlarını `/usr/src/linux/Documentation/devices.txt` dosyasındaki aygıt kodları listesi ile karşılaştırın. Şayet yanlış ise, muhtemelen **rddev -R** yapmamışsınız demektir veya yanlış biteşlem üzerinde işlem yapmışsınız demektir. Eğer aygıt kodu doğru ise, çekirdek içinde gömülü aygıt sürücülerini kontrol edin; bir disket sürücü, ramdisk ve ext2 dosya sistemi desteğine sahip olduğundan emin olun.

- Şayet şuna benzer pek çok hata alırsanız:

```
end_request: I/O error, dev 01:00 (ramdisk), sector NNN
```

Bu ramdisk sürücüsünden gelen bir G/Ç hatasıdır. Genellikle çekirdeğin aygıt dışına yazmaya kalkıştığı durumlarda olur. Ramdisk, kök dosya sistemini taşımak için çok küçük gelmiştir. Açılış disketi çekirdek ilkendirme iletilerinde şuna benzer bir satır olup olmadığını kontrol edin:

```
Ramdisk driver initialized: 16 ramdisks of 4096K size
```

Bu boyutu *sıkıştırılmamış* kök dosya sistemi boyutu ile karşılaştırın. Şayet ramdisk yeterince büyük değilse, daha büyüğünü yapın.

- Kök diskette, olması gereken dizinlerin olup olmadığını kontrol edin. Yanlış bir yere kopyalamış olabilirsiniz. `/bin` yerine elinizde `/bootdisk/bin` diye bir dizin olabilir (Tıpkı benim yaptığım gibi :- ) –yazarın değil, çevirmenin yaptığı gibi).
- Sabit diskinizdeki `/lib` dizini içinde görünen bağ ile aynı adlı bir `/lib/libc.so` olduğundan emin olun.
- Sabit diskinizdeki `/dev` dizini altında bulunan sembolik bağların, kök dosya sisteminizde de olduğundan emin olun. Bu bağlar, sizin kök dosya sisteminize yerleştirdiğiniz aygıtlara atanmış sembolik bağlar olmak zorundadır. Özellikle `/dev/console` bağları pek çok durumda gerekliliktir.
- `/dev/tty1`, `/dev/null`, `/dev/zero`, `/dev/mem`, `/dev/ram` ve `/dev/kmem` dosyalarını diskete koyduğunuzdan emin olun.
- Çekirdek ayarlarınız kontrol edin — kullanıcı girişi anında gerekli olan bütün özkaynaklar, çekirdeğe gömülü olmalıdır, modüler yapıda değil. Bu nedenle *ramdisk ve ext2 desteği modüler olarak kullanılamaz*.
- Çekirdek kök aygıtı ve ramdisk ayarlarınızı kontrol edin.

Bu genel kaideleri geçtikten sonra, kontrol edebileceğiniz bazı belli başlı dosyalar vardır:

1. **init**'in `/sbin/init` veya `/bin/init` olarak yüklendiğinden ve çalıştırılabilir olduğundan emin olun.
2. **init**'in kütüphanelerini kontrol etmek için **ldd init** komutunu kullanın. Genelde bu sadece `libc.so` olur ama yine de kontrol etmekte fayda vardır.

3. Kütüphaneler için doğru yükleyicilere sahip olduğunuzdan emin olun — a.out için `ld.so` veya ELF için `ld-linux.so`.
4. Açılış disketi dosya sisteminizdeki `/etc/inittab`'ı **getty** (veya bazı **getty**-benzeri uygulamalar; mesela **agetty**, **mgetty** veya **getty\_ps**) çağrıları için kontrol edin. Sabit diskinizdeki `inittab` ile iki kere karşılaştırın. Kullandığınız uygulamaların man sayfalarını kontrol edin. Muhtemelen, `inittab` zincirin en zayıf halkasıdır: sözdizimi ve sistemin doğal yapısından dolayı. Bu sorunu çözmenin tek yolu `init`'in ve `inittab`'ın man sayfalarını okumak ve elinizdeki sistemin açılış esnasında ne yaptığını anlamaya çalışmaktır. `/etc/inittab` içinde sistem başlatma girdisi olup olmadığını kontrol edin. Sistem başlatma betiğini çalıştıracak bir satır olması gerekmektedir.
5. Tıpkı `init`'e yaptığımız gibi, `ldd` komutunu **getty** için de çalıştırın ve onun gereksinimlerini kontrol edin. Gerekli kütüphane ve yükleyicilerin kök dosya sisteminizde bulunduğundan emin olun.
6. rc betiklerini çalıştırma yeteneğine sahip bir kabuk programı eklediğinizden emin olun(örneğin **bash** veya **ash**).
7. Kurtarma diskinizde bir `/etc/ld.so.cache` dosyası varsa, bu dosyayı tekrar oluşturun.

Şayet `init` başlıyor ama bunun gibi bir ileti alıyorsanız:

```
Id xxx respawning too fast: disabled for 5 minutes
```

Bunu sebebi `init`'dir, genellikle **getty** veya **login**'in başlar başlamaz sonlandığı anlamına gelir. **getty** ve **login** çalıştırılabilir dosyalarını ve kütüphanelerini kontrol edin. `/etc/inittab` içinde belirtilmiş çağrıların doğru olduğunu kontrol edin. Şayet **getty**'den acayip iletiler alıyorsanız, bunu anlamı `/etc/inittab` içinden yapılan çağrıların hatalı olabileceğidir.

Şayet bir bağlantı ekranına kadar geliyor, geçerli bir isim giriyor ama sistem sizden başka bir isim girmenizi istiyorsa, problemin kaynağı PAM veya NSS olabilir. *PAM ve NSS'in sağlanması* (sayfa: 14) bölümüne göz atınız. Ayrıca gölgeli parolalar kullanıyor ama `/etc/shadow` dosyasını açılış disketinize kopyalamamış olabilirsiniz.

Şayet kurtarma diskinizdeki **df** gibi bazı çalıştırılabilir dosyaları deniyorsanız ve `df: command not found` gibi bir hata alıyorsanız; iki şeyi kontrol etmeniz gerekir:

1. `PATH` içinde, dosyanın içinde bulunduğu dizinin tanımlı olduğunu,
2. komut için gerekli olan kütüphane ve yükleyicilerin diskette bulunduğunu.

## 8. Kök dosya sistemi boyutunun küçültülmesi

Bu tür disket setleri yapmanın en zor tarafı, her şeyi bir veya iki diskete sığdırmak zorunda olmaktır. Dosyalar sıkıştırılmış bile olsalar, bu yine de zor bir iştir. Neden? Çünkü, Linux sistem araçları sürekli olarak artmakta. Aşağıda boyuttan tasarruf için kullanabileceğiniz bazı genel yöntemler açıklanmaktadır.

### 8.1. Disket kapasitesini artırın

Öntanımlı olarak, disketler 1440k olarak biçimlenir. Ama daha büyük disketler biçimlemek mümkündür. Daha büyük boyutlu disketler ile açılış yapabilmek BIOS'unuza bağlıdır. **fdformat** komutu, disketleri şu boyutlarda biçimleyebilir: 1600k, 1680k, 1722k, 1743k, 1760k, 1840k ve 1920k. Ayrıntılar için **fdformat** man sayfasına ve `/usr/src/linux/Documentation/devices.txt` dosyasına bakınız.

İyi de hangi disket boyutu/geometrisi sizin makinanızca desteklenecek? Aşağıda **fdutils**'in yazarı Alain Knaff'ın cevaplarını bulacaksınız (hafiften düzenlenmiş bir biçimde):

Bu olay, disket üzerine atılacak fiziksel biçimden ziyade BIOS ile alakalı bir durumdur. Şayet BIOS 18'den büyük sektör numaralarını bozuk olarak kabul ederse, yapılacak fazla birşey yok demektir. Gerçekte, deneme ve yanılma yöntemi elimizdeki tek yoldur. Bununla birlikte, BIOS ED diskleri destekliyorsa (ek yoğunluk: 36sektör/iz ve 2.88 MB), 1722k'lık disketler elde etme şansımız olabilir.

21 sektör/iz oranından daha fazlasına sahip süper biçimli disketler kendiliğinden açılabilir değildir. Aslında standart sektör boyutu yerine daha fazlasını kullanan disketlerin hiçbirisi sistem açılışı yapamazlar. Bununla birlikte, özel bir önyükleme sektörü yazmak mümkündür. Şayet doğru hatırlıyorsam, DOS 2m aracı böyle bir "canavara" sahipti, tıpkı OS/2'nin XDF aracı gibi.

Bazı BIOS'lar "yalancıkdan", 18'den büyük sektör numaralarının hata olması gerektiğini iddia ederler. 1722k'lık bir disket 21 sektör kullanır ve bu nedenle disket üzerinden açılış mümkün olmayacaktır. Bunu sınamanın en iyi yolu, bir DOS veya syslinus disketini 1722K olarak biçimlemek ve bu disketi önyükleme yapabilir yapmaktır. Şayet LILO kullanıyorsanız, **linear** seçeneğini kullanmayın. Bu seçenek öntanımlı olarak 18 sektör/iz kabul eder ve BIOS desteklese bile açılış işlemi başarısız olur.

## 8.2. Ortak araçları BusyBox ile değiştirin

Kök dosya sisteminin büyük kısmı GNU sistem araçları tarafından ortaklaşa kullanılır: **cat**, **chmod**, **cp**, **dd**, **df**, vs. BusyBox projesi, bu genel sistem araçlarını asgari gereksinimleri sağlayarak değiştirmek amacını gütmektedir. BusyBox, tek bir çalıştırılabilir dosya sağlar, `/bin/busybox`, yaklaşık 150kB civarındadır. Bu dosya, bu araçların görevlerini yerine getirir. Daha sonra, bu çalıştırılabilir dosyaya, değişik araçlardan sembolik bağlar yapabilirsiniz: **busybox** bunun nasıl adlandırıldığını görür ve doğru kodu çalıştırır. BusyBox basit bir kabuk bile içerir. BusyBox, pek çok dağıtım için, ikilik paketler halinde dağıtılmaktadır ve kaynak kodları [BusyBox yöresinden](#)<sup>(B24)</sup> elde edilebilir.

## 8.3. Başka bir kabuk kullanın

**bash** ve **tcsh** gibi bazı çok bilinen Linux kabukları oldukça büyüktür ve pek çok kütüphane isterler. Şayet BusyBox kabuğunu kullanmazsanız, bu kabuklar yerine bir başkasını kullanmayı düşünmelisiniz. Bazı hafif siklet kabuklar; **ash**, **lsh**, **kiss** ve **smash** oldukça küçük boyutludurlar ve çok az (ya da hiç) kütüphaneye ihtiyaç duyarlar. Bu kabukların pek çoğu <http://www.ibiblio.org/pub/Linux/system/shells/> adresinden elde edilebilir. Kullandığınız kabuğun, rc dosyasındaki komutları çalıştırma yeteneği olduğundan emin olun.

## 8.4. Kütüphaneleri ve çalıştırılabilirleri soyun

Pek çok kütüphane ve çalıştırılabilir hata ayıklama bilgileri ile birlikte dağıtılmaktadır. Bu dosyalar üzerinde **file** komutunu kullandığınızda, bu böyleyse "not stripped" şeklinde bunu belirtecektir. İkilik dosyaları kök dosya sisteminize kopyalarken, aşağıdaki komutu kullanmak oldukça faydalı olacaktır:

```
objcopy --strip-all bir_yerden bir_yere
```



### Önemli

Kütüphaneleri kopyalarken, **strip-all** yerine **strip-debug** kullanmalısınız.

## 8.5. Dosyaları uygulamalar disketine koyun

Şayet bazı ikilik dosyalar açılış veya sisteme giriş için gerekli değil ise, bunları başka bir uygulama disketine koyabilirsiniz. Ayrıntılar için [Uygulamalar disketinin oluşturulması](#) (sayfa: 23) bölümüne göz atınız. Tabii ki modülleri de bir uygulama disketi içine koymayı düşünebilirsiniz.

## 9. Çeşitli konular

### 9.1. Ramdisksiz kök dosya sistemleri

[Bir kök dosya sisteminin oluşturulması](#) (sayfa: 7) bölümünde ramdisk içine yüklenen bir sıkıştırılmış dosya sisteminin nasıl yapılacağı açıklanmıştır. Bu yöntemin çeşitli faydaları olduğu için, geniş çapta kullanıma sahiptir. Bununla birlikte, küçük boyutlu RAM'lere sahip bazı sistemler bunu gerçekleştiremezler. Bu nedenle kök dosya sistemi doğrudan disketten bağlanmalıdır.

Bu tür dosya sistemlerini oluşturmak, sıkıştırılmış dosya sistemlerini oluşturmaktan daha kolaydır. Çünkü başka aygıtlar üzerinde uğraşmaktansa, bu dosya sistemi doğrudan disket üzerinde oluşturulabilir ve sıkıştırılmak zorunda değildir. Biz burada, daha önceki anlatımımızdan farklı yönlerini açıklayarak ana hatları ortaya koyacağız. Şayet bu yöntemi tercih ederseniz, elinizde daha *kısıtlı* bir alan bulunacağını aklınızdan çıkarmayın.

1. Kök dosyaları için ne kadar alana sahip olduğunuz hesaplayın. Şayet tek bir önyükleme ve kök disketi yapıyorsanız, tek bir diskete çekirdek ve kök dosya sistemi için gereken blokları sığdırmak zorundasınız.
2. **mke2fs** kullanarak disket üzerinde uygun boyutlu bir kök dosya sistemi oluşturun.
3. [Daha önceden tarif edildiği şekilde](#) (sayfa: 9) dosya sistemini doldurun.
4. Bu işlem bittiğinde dosya sistemini ayırıp, onu diskete *sıkıştırılmamış* bir şekilde aktarın.
5. Çekirdeği diskete kopyalayın. Ramdisk kelime değerini hesaplarken, *14. biti sıfıra ayarlayın* ve dosya sisteminin ramdiske yüklenmeyeceğini belirtin. Daha önce tarif edildiği şekilde **rdev**'i çalıştırın.
6. Kök dosya sistemini daha önceki gibi aktarın.

Kullanabileceğiniz çeşitli kısa yollar mevcuttur. Şayet iki disketlik bir set yapıyorsanız, kök dosya sistemini doğrudan bir disket üzerinde oluşturabilir ve sabit diskinizdeki bir dosya ile disket arasında taşımacılık yapmak zorunda kalmazsınız. Ayrıca, hem tek disketli sistem yapacak hem de LILO kullanacaksanız, bütün disket üzerinde çekirdek, LILO dosyaları ve kök dosyaları içeren tek bir dosya sistemi oluşturabilir ve daha sonra son adım olarak LILO komutunu çalıştırabilirsiniz.

### 9.2. Uygulamalar disketinin oluşturulması

Bir uygulama disketi yapmak nispeten daha kolaydır — biçimlenmiş bir diskette bir dosya sistemi oluşturun ve dosyaları oraya kopyalayın. Bu disketi bir önyükleme disketi ile kullanmak için, sistem açılışından sonra, uygulama disketini kök dizine bağlamanız yeterlidir.

Daha önceki açıklamalarda da belirttiğimiz gibi, uygulama disketi **/usr** altına bağlanabilir. Bu durumda, ikilik dosyaları diskette **/bin** dizini altına yerleştirdiyseniz **PATH** ortam değişkenindeki **/usr/bin** tanımı sayesinde burası erişilebilir bir alan olur. Bu uygulamalar tarafından ihtiyaç duyulan kütüphaneler de uygulama disketi içinde **/lib** dizini altına yerleştirilebilir.

Bir uygulamalar disketi yaparken şu noktaları aklınızdan çıkarmayın:

1. Kritik öneme haiz sistem ikilikleri veya kütüphaneleri uygulamalar disketine koymayın. Unutmayın ki sistem açılana kadar, bu disketin içindekileri sisteme bağlamanız mümkün değildir.
2. Bir diskete ve disket sürücüsüne takılan bir teyp aynı anda erişim sağlayamazsınız. Yani, uygulama disketi sisteme bağlı iken teyp sürücüyü erişim mümkün olmayacaktır.
3. Uygulama disketindeki dosyalara erişim nispeten yavaş olacaktır.



*Uygulamalar disketi dizin listesi örneği* (sayfa: 39) bölümünde uygulama disketindeki dosyalar için örnek bir liste bulabilirsiniz. Faydalı olabilecek bir kaç fikir: diskleri araştırma ve yönetme programları (**format**, **fdisk**) ve dosya sistemleri (**mke2fs**, **fsck**, **debugfs**, **isofs.o**), hafif sıklet bir metin düzenleyici (**elvis**, **jove**), sıkıştırma ve arşivleme araçları (**gzip**, **bzip**, **tar**, **cpio**, **afio**), teyp araçları (**mt**, **ftmt**, **tob**, **taper**), iletişim araçları (**ppp.o**, **slip.o**, **minicom**) ve aygıtlar için araçlar (**setserial**, **mknod**).

## 10. Profesyoneller nasıl yapıyor

Slackware, RedHat veya Debian gibi ana dağıtımlar tarafından kullanılan açılış disketlerinin, bu kılavuzda anlatılardan daha karmaşık yapılarda olduğunu görmüş olabilirsiniz. Profesyonel dağıtımların kurtarma/açılış disketleri burada anlatılan ana hatlar üzerine kurulmuştur ama çeşitli ilave hünerleri daha vardır. Çünkü bu açılış disketleri ilave gereksinimlere ihtiyaç duyarlar. Birinci olarak, oldukça geniş çaplı donanım üzerinde çalışmalıdır, bu nedenle kullanıcı ile etkileşimli olmalı ve değişik aygıt sürücülerini yükleyebilmelidir. İkinci olarak, bu disketler değişik yükleme seçeneklerini, çeşitli otomatik seviyelerde desteklemelidir. Son olarakta, dağıtım açılış disketleri yükleme ve kurtarma kabiliyetinin birleşimine sahip olmalıdır.

Bazı açılış disketleri **initrd** (birincil ramdisk/initial ramdisk) diye bilinen bir özellik kullanırlar. Bu özellik çekirdeğin 2.0.x sürümü aşamasında devreye girmiş ve çekirdeğe iki aşamalı olarak yüklenme esnekliğini sağlamıştır. Bu birincil ramdisk, gerçek kök dosya sistemi yüklenmeden önce çalışan bir yazılım içeren bir kök dosya sistemidir. Bu yazılım, genellikle, ortamı denetler ve/veya kullanıcıdan çeşitli açılış seçeneklerinde birisini seçmesini ister (örneğin gerçek kök dosya sisteminin yükleneceği aygıtı). Ayrıca, çekirdek içine gömülmemiş ilave modülleri de yükler. Bu birincil yazılımın işi bitince, çekirdek gerçek kök biteşlemini yükler ve açılma işlemi normal olarak devam eder. **initrd** hakkında daha ayrıntılı bilgi elde edebilmek için sisteminizdeki <file:/usr/src/linux/Documentation/initrd.txt> dosyasına ve <ftp://elserv.fhm.fgan.de/pub/linux/loadlin-1.6/initrd-example.tgz> adresine bakabilirsiniz.

Aşağıda dağıtımların yükleme disketlerinin nasıl çalıştığının özetlerini bulacaksınız. Bu açıklamalar, kaynak kodlarına ve dosya sistemlerinde yapılan incelemelere göre çıkarılmıştır. Bu bilgilerin tam ve doğru olduğunun garantisi yoktur. Kılavuzumuzun bu sürümü yazıldığından beri bu bilgilerin değişmiş olması ihtimalini göz önünde bulundurmanız yararlı olur.

Slackware (v.3.1), *Çekirdeğin LILO ile kullanımı* (sayfa: 17) bölümünde tarif edildiği gibi basit LILO açılışını kullanmaktadır. Slackware açılış disketi LILO'nun **message** parametresini kullanarak bir açılış iletisi basar ("Welcome to the Slackware Linux bootkernel disk!/Slackware Linux çekirdek açılış disketine hoş geldiniz!"). Bu açıklamanın ardından kullanıcıdan, gerekli ise, açılış parametreleri girmesini ister. Açılıştan sonra, kök dosya sistemi ikinci bir disketten yüklenir. Kullanıcı, yüklemeyi başlatan **setup** betiğini çalıştırır. Slack, çeşitli çekirdek seçenekleri sunarak, modüler bir çekirdek kullanmak yerine, kullanıcının kendi ihtiyacına göre bir çekirdek seçmesine olanak verir.

Red Hat (v.4.0) de LILO açılışını kullanır. Birinci diskete sıkıştırılmış bir ramdisk yükler ve buradan özel bir **init** yazılımı çalıştırılır. Bu yazılım sürücülerini tarar ve şayet gerekli ise başka bir diskten ek dosyaları yükler.

Debian (v.1.3), muhtemelen en karışık başlangıç diski setine sahiptir. Değişik yükleme seçeneklerini düzenlemek için SYSLINUX yükleyicisi kullanır ve daha sonra yükleme işleminde kullanıcıya yol göstermek için bir **initrd** biteşlemi kullanır. Özelleştirilmiş bir kabuk ve **init** kullanmışlar gibi görünüyor.

## 11. Önyükleme yapabilen CD-ROMların hazırlanması



Bilgi



Bu bölüm Rizwan Mohammed Darwe ([rizwan AT cloverttechnologies dot com](mailto:rizwan AT cloverttechnologies dot com)) tarafından hazırlanmıştır.

Bu bölümde, Linux'da CD yazma süreci hakkında bilginiz olduğu kabul edilmiştir. Burada anlatılanları, yazdığınız CD'leri önyüklemeye yapabilir hale getirmek için hızlı referans olarak kabul edin. CD yazma hakkında daha ayrıntılı bilgi elde etmek için [Linux'da CD Yazımı](#)<sup>(B33)</sup> belgesine göz atınız.

### 11.1. El Torito nedir?

x86 platformlarında, pek çok BIOS önyüklemeye yapabilen CD'leri desteklemektedir. **mkisofs** için yazılan yamalar "El Torito" standardını temel alır. Basitçe anlatmak gerekirse; El Torito, CD'lerin kendiliğinden sistem açılışı gerçekleştirebilmeleri için nasıl biçimlenmesi gerektiğini gösteren bir belirtimdir.

"El Torito" belirtimi, BIOS desteklediği sürece herhangi bir CDROM sürücünün çalışması gerektiğini söyler (SCSI veya IDE). Bugüne kadar, bu olay sadece EIDE CD-ROMlar ile sınılanabildi, çünkü SCSI kontrolcülerin hiçbiri El Torito'yu desteklemiyordu. Anakartlar El Torito desteğine mutlaka sahip olmalıdır. Peki anakartınızın El Torito'yu desteklediğini nasıl anlarsınız? Şayet anakartınız sabit diskten, disketten, CD-ROM'dan veya ağ üzerinden başlatma seçeneğine sahip ise, El Torito'yu destekliyor demektir.

### 11.2. El Torito nasıl çalışır?

El Torito standardı, CD sürücüyü, BIOS'a normal disket sürücü gibi gösterir. Bu yöntemde, basitçe, bir disket boyutlu biteşlem, ISO dosya sistemine yerleştirilir (1440k'lık disket için 1440k'lık bir biteşlem dosyası). ISO dosya sisteminin başına bu biteşlemi gösteren bir gösterici yerleştirilir. BIOS bu biteşlemi, CD'den bulur ve sanki disketten açılış yapıyormuşçasına devam eder. Bu sayede çalışan bir LILO önyüklemeye diski elde edilmiş olur.

Kabaca söylersek; CD'nin ilk 1.44 (veya destekliyorsa 2.88) MB'lık kısmı bir disket biteşlemi içerir. Bu biteşlem, BIOS'a kendini disket sürücü diye yutturur ve sistemin açılması işlemini başlatır. Bu açılma işlemi sırasında birinci disket sürücünüz (A veya `/dev/fd0`) erişilmez durumda olur ama bu problemi `/dev/fd1` kullanarak aşabilirsiniz.

### 11.3. Nasıl çalışır hale getireceğiz?

İlk olarak bir dosya oluşturacağız, diyelim ki "boot.img". Bu dosya disket biteşlemi boyutlarında olmalı: 1.44 MB. Disketin 1. sürücünde olduğunu varsayarak aşağıdaki komut işiniz görür:

```
# dd if=/dev/fd0 of=boot.img bs=10k count=144
```

Bu dosyayı iso9660 dosya sistemi hiyerarşisi içine yerleştirin. Açılış ile ilgili bütün dosyaları ilgili dizin altına (mesela iso9660fs'in kökü altında `boot` dizinine) yerleştirmek iyi bir fikir olabilir.

Bir yetersizlik — Açılış disketiniz LILO üzerinden birincil ramdiski yükleyebilmelidir, çekirdek ramdisk sürücüsünü değil. Çünkü Linux çekirdeği yüklenmeye başladı mı, BIOS'a CD'yi disket olarak yutturma işlemi düşecek ve açılış gerçekleşmeyecektir. LILO, BIOS disk çağrılarını kullanarak, birincil ramdiski yükleyecek ve böylece yutturmaya tasarlandığı biçimde devam edebilecektir.

El Torito belirtimi bir "açılış kataloğu" gerektirir. Bu kullanım amacı dışında hiçbir şey ile ilgisi olmayan 2048 baytlık bir dosyadır. **mkisofs**'nin yazarı tarafından yapılan yama, özdevinimli olarak açılış kataloğu yapılmasını sağlayacaktır fakat bu kataloğun iso9660 dosya sistemi içinde nerede bulunacağını tanımlamak zorundasınız. Genellikle, bu kataloğu önyüklemeye biteşlemi ile aynı yere koymak iyi bir fikirdir. `boot.catalog` şeklinde bir isim ise oldukça uygun olacaktır.

Böylece `boot.img` dosyası içinde önyüklemeye biteşlemini elde etmiş olduk ve bunu iso9660 dosya sistemi içindeki `/boot` dizinine koyduk. Kataloğumuz da, `boot.catalog` adı ile aynı dizine yerleştirdik. `bootcd.iso` içinde iso9660 dosya sistemini oluşturmak için gereken komut:

```
# mkisofs -r -b boot/boot.img -c boot/boot.catalog -o bootcd.iso .
```

**-b** seçeneği kullanılacak önyükleme biteşlemini belirtir (dosya yolu diskteki yerine bağlı olarak değişebilir) ve **-c** seçeneği önyükleme kataloğunun dosyası içindir. **-r** seçeneği uygun dosya yetkileri ve kipleri (**mkisofs** man sayfasına bakınız) oluşturacaktır. En sondaki "." işaret ise kaynakların geçerli dizinden alınacağını belirtmektedir.

Daha sonra CD'yi alışılmış yöntemler ile yakın; "önyükleme yapabilen" CD'niz artık hazır.

## 11.4. Win9x Önyüklenebilir CD-ROMlarının yapımı

İlk adım önyükleme biteşleminin kaynak CD'den alınmasıdır. Ancak Linux altında CD'yi sisteme bağlayıp **dd** ile ilk 1440k'ı bir diskete ya da **boot.img** gibi bir dosyaya alamazsınız. Bunun yerine basitçe CD-ROM'dan açılışı yapın.

Win98 CD ile açılış yapıldıktan sonra aslında bir ramdisk olan A: istemine düşün. DOS'un **diskcopy** komutu ile A: biteşlemini o an için B: olarak görünen birinci disket yuvasındaki diskete kopyalayın. Bunu şu şekilde yapabilirsiniz:

```
diskcopy A: B:
```

Bu komut **dd** gibi çalışır. Bu yeni oluşturduğunuz diskten önyükleme yaptırarak kaynak CDRom'dan yapılan bir açılış gibi bir açılış yapıldığını deneyerek görebilirsiniz. Bu disketten **dd** ile **boot.img** gibi bir dosyaya önyükleme biteşlemini aktardıktan sonra yapılacaklar yukarıda anlatılanlar ile aynıdır.

## 12. Sıkça Sorulan Sorular(SSS) listesi

- 12.1. *Önyükleme ve kök disklerimden açılış yapıyorum ama hiçbir şey olmuyor. Ne yapmalıyım?*
- 12.2. *Slackware/Debian/RedHat açılış diskleri nasıl çalışır?*
- 12.3. *Büyük boyutlu diskleri (1440K'dan büyük) nasıl kullanabilirim? Kendi disket sürücüm ile hangi boyutta disketin çalışacağını nasıl anlayabilirim?*
- 12.4. *Ramdiskimin boyutunu nasıl artırırım?*
- 12.5. *Önyükleme yapabilen CDRom'ları nasıl oluşturabilirim?*
- 12.6. *Önyükleme yapabilen LS-120 disklerini nasıl oluşturabilirim?*
- 12.7. *Bir XYZ sürücüsü ile bir açılış diskini nasıl yapabilirim?*
- 12.8. *Kök disketimi yeni dosyalarla nasıl güncellerim?*
- 12.9. *Sistemi DOS'tan tekrar başlatabilmek için LILO'yu nasıl kaldırabilirim?*
- 12.10. *Şayet hem çekirdeği hem de açılış disketimi kaybedersem, sistemi nasıl açabilirim?*
- 12.11. *Önyükleme ve kök disketlerimin kopyalarını nasıl alabilirim?*
- 12.12. *Her seferinde "ahaxxxx=nn,nn,nn" yazmadan nasıl açılış yapabilirim?*
- 12.13. *Açılış esnasında, bir hata alıyorum: "A: B'yi çalıştıramıyor". Niye?*
- 12.14. *Çekirdeğim ramdisk desteğine sahip ama ramdisk başlangıç boyutu olarak 0 kB. Neden?*

---

12.1. *Önyükleme ve kök disklerimden açılış yapıyorum ama hiçbir şey olmuyor. Ne yapmalıyım?*  
*Sorun çözmek ya da Yenilginin Acısı* (sayfa: 20) bölümüne bakınız.

12.2. *Slackware/Debian/RedHat açılış diskleri nasıl çalışır?*  
*Profesyoneller nasıl yapıyor* (sayfa: 24) bölümüne bakınız.

12.3. *Büyük boyutlu diskleri (1440K'dan büyük) nasıl kullanabilirim? Kendi disket sürücüm ile hangi boyutta disketin çalışacağını nasıl anlayabilirim?*

*Kök dosya sistemi boyutunun küçültülmesi* (sayfa: 21) bölümüne bakınız. Orada Alain Knaff'in bu konudaki yorumlarını bulacaksınız. Kendisi bu konu hakkında, benim tanıdığım en otoriter kişidir.

#### 12.4. Ramdiskimin boyutunu nasıl artırırım?

Bu konu metin içinde açıklansa daha iyi olurdu ama başlangıç olarak buraya kısa bir yanıt koyuyorum.

İlk olarak; bunu yapmak için `rdev` veya `ramsize` komutlarını **kullanmayın**. Kılavuzların ne söylediğini boşverin. ramdisk kelimesi, artık ramdisk boyutunu tayin etmek için kullanılmamaktadır.

İkinci olarak; ramdisklerin gerçekte dinamik yapıda olduğunu unutmayın. Bir ramdisk ayarladığınız zaman, bellekte yer ayırmıyorsanız, sadece ramdiskin ne kadar genişliyebileceğine karar vermiş oluyorsunuz. Ramdisk boyutunu büyük tutmaktan korkmayın (8 MB veya 16 MB gibi). Fiziksel bellek alanı, gerekli olmadığı sürece kullanılmaz veya boş tutulmaz. Bu ayarı çeşitli şekillerde yapabilirsiniz.

#### 12.5. Önyükleme yapabilen CDROM'ları nasıl oluşturabilirim?

*Önyükleme yapabilen CD-ROMların hazırlanması* (sayfa: 24) bölümüne bakınız.

#### 12.6. Önyükleme yapabilen LS-120 disklerini nasıl oluşturabilirim?

Bir LS-120 sürücüyü sahip olmadığım için, aşağıdaki bilgileri Linux Router Projesi'nden [Dave Cienne tarafından sağlanan bilgilerden](#)<sup>(B38)</sup> derledim.

LS-120 bir IDE disket sürücüsüdür. Hem 3.5" lik hem de 120MB'lık yeni diskler ile uyumludur. Linux 2.0.31'den beri tam desteğe sahiptir. Bu diskler ile sistem açılışı yapabilmek için, bu diskleri sürücü 0 (IDE sürücüler normalde 80'den başlar) olarak kabul eden BIOS'lara ihtiyacınız var. Şayet BIOS desteğiniz yok ise, bu sorunu aşmak için Promise Technologies'den küçük bir IDE FloppyMAX kartı satın alabilirsiniz.

Çekirdek önyükleyicisi LS-120'yi sevmez ve anında ölür. Ayrıca 2M diskleri de onu sevmez ve önyükleme yapmaz. 1.44MB ile 1.74MB arası diskler normal çalışacaktır. SYSLINUX, sürüm 1.32'den beri 120MB'lık diskler ile anlaşabilmektedir. MS-DOS uyumu gerekmediği sürece, SYSLINUX kullanmak yerine, diskten yeniden bölümlenmek ve ext2 veya minix kullanmak daha uygundur.

LILLO 120MB'lık diskler ile çalışabilir. Burada örnek bir `lilo.conf` dosyası var:

```
boot=/dev/hda
compact
disk=/dev/hda bios=0
install=/floppy/boot.b
map=/floppy/map
image=/floppy/linux
label=Linux
append="load_ramdisk=1"
initrd=/floppy/root.bin
ramdisk=8192
```

"`disk=/dev/hda bios=0`" satırı LS-120'den önyükleme yapılabilmesini sağlayan hileyi gerçekleştirir.

#### 12.7. Bir XYZ sürücüsü ile bir açılış diskini nasıl yapabilirim?

En kolay yol, en yakınınızdaki bir Slackware yansıısından bir Slack çekirdeği indirmektir. Slack çekirdekleri, mümkün olan en fazla sürücüyü destek vermeye çalışan genel çekirdeklerdir. Şayet bir SCSI veya IDE kontrolcüsünün sürücüsüne ihtiyacınız var ise Slack çekirdeği içinde bu sürücüyü bulma ihtimaliniz göreceli olarak yüksektir.

`al` dizinine gidin ve sahip olduğunuz denetleyiciye göre IDE veya SCSI çekirdeği seçin. İsteddiğiniz sürücünün, bu çekirdek içinde bulunup bulunmadığını anlamak için `xxxxkern.cfg` dosyasını inceleyin. Şayet aradığınız sürücü bu çekirdek içinde ise, bu çekirdek sizin sisteminizi açabilmelidir. `xxxxkern.tgz` dosyasını indirin ve daha önce anlatıldığı şekilde açılış disketinize kopyalayın.

Daha sonra **rdev zImage** komutunu kullanarak çekirdekteki kök aygıtı kontrol etmelisiniz. Şayet bu sizin istediğiniz kök aygıtı değilse, değiştirmek için **rdev**'i kullanın. Örneğin; düzenlemeye çalıştığım çekirdek için `/dev/sda2` idi ama benim kök SCSI bölümüm `/dev/sda8` idi. Bir kök disketini kullanabilmek için **rdev zImage /dev/fd0** komutunu kullanmanız gerekir.

Şayet bir Slack kök disketinin nasıl yapıldığını bilmek istiyorsanız, bu konu bizim NASIL'ımızın alanı dışındadır. Linux Install Guide'ı incelemenizi veya bir Slackware dağıtımı almanızı öneririm.

### 12.8. Kök disketimi yeni dosyalarla nasıl güncellerim?

En kolay yol, kök disketteki dosya sistemini, daha önce kullandığımız *aygıt*'a (*Dosya sisteminin oluşturulması* (sayfa: 8)) kopyalamaktır. Daha sonra dosya sistemini bağlayın ve değişiklikleri yapın. Kök dosya sisteminizin nerede başladığını ve kaç bloktan oluştuğunu hatırlamanız gerekmektedir:

```
# dd if=/dev/fd0 bs=1k skip=kök_başlangıcı count=blok_sayısı | gunzip > aygıt
# mount -t ext2 aygıt /mnt
```

Değişiklikleri yaptıktan sonra, daha önce tarif edildiği şekilde (*Paketleme* (sayfa: 15)) kök dosya sistemini diskete aktarın. Yeni kök dosya sisteminin başlangıç konumunu değiştirmemişseniz, çekirdeği tekrar aktarmanız ve ramdiski tekrar hesaplamanızda gerekmeyecektir.

### 12.9. Sistemi DOS'tan tekrar başlatabilmek için LILO'yu nasıl kaldırabilirim?

Aslında bu bir açılış disketi konusu değil ama oldukça sık sorulmakta. Linux'ta iken şunu yapın:

```
# /sbin/lilo -u
```

Ayrıca, LILO tarafından açılış sektörüne kaydedilmiş yedeği kopyalamak için **dd** komutunu kullanabilirsiniz. Bunu yapmak istiyorsanız LILO belgelerine bakınız.

DOS ve Windows içinde bu DOS komutunu kullanabilirsiniz:

```
FDISK /MBR
```

MBR, Master Boot Record sözcüklerinin kısaltmasıdır ve ana önyükleme kaydı anlamına gelir. Bu komut önyükleme sektörünü temiz bir tanesi ile değiştirir. Bölümlene tablosunu etkilemez. Bazı titiz insanlar bu konuda hem fikir değillerdir ama LILO yazarı Werner Almesberger bu yöntemi önerir. Oldukça kolay ve işe yarar bir yöntemdir.

### 12.10. Şayet hem çekirdeği hem de açılış disketimi kaybedersem, sistemi nasıl açabilirim?

Şayet elinizde bir açılış diskeri yok ise, muhtemelen en kolay yol; disk denetleyicinize uygun bir Slack çekirdeği elde etmektir. Daha sonra "XYZ sürücüsü ile bir açılış diskini nasıl yapabilirim?" sorusundaki yöntemi izleyin. Bu çekirdek ile sistemi açabilir ve hasar gören ne ise onu düzeltebilirsiniz.

İndirdiğiniz çekirdek size uyan bir kök bölüm veya kök tipi içermeyebilir. Genel olarak Slack SCSI çekirdeğinde kayıtlı kök aygıtı `/dev/sda2`'dir. Benim ki ise `/dev/sda8` idi. Bu durumda çekirdek içinde kayıtlı kök aygıtı değiştirilmelidir.

Bütün sahip olduğunuz bir çekirdek veya başka bir işletim sistemi olsa bile (örneğin DOS), çekirdek içinde kayıtlı kök aygıtı ayarını değiştirme imkanınız hala var.

**rdev**, çekirdek dosyası içindeki sabit bayt konumlarındaki verileri değiştirerek, çekirdek ayarlarını değiştirir. Bu yöntemle, elinizdeki sistem ne olursa olsun bir onaltılık düzenleyici ile istediğiniz değişiklikleri yapabilirsiniz. Örneğin DOS altında Norton Utilities Disk Editor'ü kullanabilirsiniz. Daha sonra gerekli değişiklikleri çekirdek içinde yapın:

16'lık Onluk Açıklama			
0x01F8	504	RAMDISK sözcüğünün düşük baytı	
0x01F9	505	RAMDISK sözcüğünün yüksek baytı	
0x01FC	508	Kök aygıtın alt aygıt numarası - aşağıya bakınız	
0X01FD	509	Kök aygıtın ana aygıt numarası - aşağıya bakınız	

Ramdisk kelimesinin yorumu daha önce [ramdisk sözcüğünün yapılandırılması](#) (sayfa: 18) içinde yapılmıştı.

Bağlamak istediğiniz aygıtın minör ve major aygıt numaraları uygun şekilde değiştirilmelidir. Aşağıda bazı yararlı değerleri bulabilirsiniz:

AYGIT	ANA	ALT	
/dev/fd0	2	0	1. disket yuvası
/dev/hda1	3	1	1. IDE sürücünün 1. bölümü
/dev/sda1	8	1	1. SCSI sürücünün 1. bölümü
/dev/sda8	8	8	1. SCSI sürücünün 8. bölümü

Bu değerleri düzenledikten sonra, bu dosyayı Norton Utilities Disk Editor veya **rawrite.exe** yazılımını kullanarak diskete yazabilirsiniz. Bu yazılım bütün dağıtımların içinde bulunur. Bu yazılım, dosya sistemi yerine açılış sektöründen başlayarak "ham" disk üzerine yazabilen bir DOS yazılımıdır. Şayet Norton araçlarını kullanıyorsanız, dosyayı fiziksel diske, diskin başlangıç noktasından başlayarak yazmanız gerekir.

### 12.11. Önyükleme ve kök disketlerimin kopyalarını nasıl alabilirim?

Manyetik araçlar zamanla bozulduğu için, kurtarma setinizin bir kopyasını saklamak iyi bir fikir olacaktır.

Bunu yapmanın en kolay yolu, disketleri **dd** komutu ile diskteki bir dosyaya kopyalamak ve daha sonra aynı komutla diskten diskete kopyalamaktır. Bu işlem esnasında disketi bağlamak zorunda değilsiniz, çünkü **dd** ham aygıt arayüzünü kullanır.

Disketi diske kopyalamak için:

```
# dd if=aygit_adi of=dosya_adi
```

Burada *aygit\_adi* disketin aygıt adı ve *dosya\_adi* disketin içeriğinin kopyalanacağı sabit diskteki dosyanın adıdır. Komut **count** parametresi olmaksızın kullanılarak **dd**'nin bütün disketi olduğu gibi kopyalaması sağlanır.

Yeni bir diskete kopyalamak için, disket sürücüyü boş bir disket yerleştirin ve:

```
# dd if=dosya_adi of=aygit_adi
```

Bu açıklamalar, sadece bir adet disket sürücünüz olduğu varsayılarak yapılmaktadır. Şayet iki adet disket sürücünüz varsa:

```
# dd if=/dev/fd0 of=/dev/fd1
```

### 12.12. Her seferinde "ahaxxxx=nn,nn,nn" yazmadan nasıl açılış yapabilirim?

Bir disk aygıtı özdevinimli olarak saptanamadığı için çekirdeğe aygıt parametrelerini gösteren bir dizge verilmesi gereklidir:

```
aha152x=0x340,11,3,1
```

Bu parametre dizgesi LILO içine çeşitli şekillerde yazılabilir:

- Sistem LILO üzerinde açılırken, komut satırına her seferinde yazarak. Oldukça sıkıcı olsa gerek.
- LILO'nun **lock** anahtar sözcüğünü kullanarak bu satırı öntanımlı komut satırı haline getirerek. Böylece LILO her açılışta bu parametreyi kullanacaktır.

- LILO ayar dosyası içinde `append="ifade"` şeklinde kullanarak. `ifade`'nin çift tırnak ile kapatılmış olduğuna dikkat ediniz.

Örneğin, yukarıdaki parametre dizgesini kullanan bir komut satırı şu şekilde olurdu:

```
zImage aha152x=0x340,11,3,1 root=/dev/sda1 lock
```

Bu, aygıt parametre dizgesini geçecekti ve ayrıca çekirdeğin kök aygıtı `/dev/sda1` olarak ayarlamasını isteyecek ve bütün satırı kaydedip, sonraki açılıшта tekrar kullanmasını sağlayacaktır.

Örnek bir `APPEND` ifadesi:

```
APPEND = "aha152x=0x340,11,3,1"
```

EK BİLGİ: parametre dizgesi komut satırında çift tırnak alınmaz ama `APPEND` için kullanırken mutlaka çift tırnağı kullanmanız gerekmektedir.

Ayrıca yürütülen parametre dizgesi için, çekirdekte uygun türde sürücü bulunmalıdır. Şayet bulunamaz ise, parametre dizgesini sözünü dinleyecek hiçbir şey olmayacak demektir. Bu nedenle çekirdeği, bu sürücü desteği ile tekrar derlemeniz gerekecektir. Çekirdek derlemenin ayrıntıları için `/usr/src/linux` içine bakınız ve README'yi okuyunuz. Linux SSS'yi ve Kurulum NASIL'ı okuyun. Bundan başka, istediğiniz sürücüye sahip genel bir çekirdek alabilir ve onu yükleyebilirsiniz.

LILO'yu yüklemeye kalkmadan önce kullanıcıların LILO belgelerini okumaları şiddetle önerilir. Hatalı kullanımlar bölümlere zarar verebilir.

### 12.13. Açılış esnasında, bir hata alıyorum: “A: B’yi çalıştıramıyor”. Niye?

Yerleri çeşitli araçlara sabit kodlanmış uygulamalar bu soruna sebep olabilir. Bu her zaman olmayabilir ama sisteminizde görebildiğiniz halde çalışmayan bir uygulamanın çalışmama sebebini bu şekilde açıklamak mümkün olabilir. İstenilen yazılımın başka bir tanesine sabit kodlanmış olup olmadığını görmek için: çıktısı `grep`'e borulanan `strings` komutu işe yarayabilir.

Yerlerinin değişmeyeceği varsayılan uygulamalardan örnekler:

- `shutdown` bazı sürümlerde `/etc/reboot`'a sabit kodlanmıştır. Bu durumda `reboot` `/etc` dizini altında olmak zorundadır.
- `init` en azından bir kişi için, sorun çıkardı, çekirdek `init`'i bulamadı.

Bu sorunları aşmak için ya dosyaları olmaları gereken yerlere taşıyın ya da ayar dosyalarını (örneğin: `inittab`) düzeltin. Şayet iki arada bir derede kalmışsanız, bu dosyaları sabit diskinizde bulundukları yerlere koyun ve sabit diskinizdeki `inittab` ve `/etc/rc.d` dosyalarını kullanın.

### 12.14. Çekirdeğim ramdisk desteğine sahip ama ramdisk başlangıç boyutu olarak 0 kB. Neden?

Bunun olduğu yerde, çekirdek yüklenirken aşağıdakine benzer bir çekirdek iletisi görüntülenir:

```
Ramdisk driver initialized: 16 ramdisks of 0K size
```

Bunu muhtemel sebebi, önyükleme sırasında çekirdek parametreleri tarafından ramdisk boyutunun 0 kB olarak ayarlanmış olmasıdır. LILO yapılandırma parametrelerinde gözden kaçırdığımız bir şey olması da muhtemeldir:

```
ramdisk= 0
```

Bazı eski dağıtımların LILO ayar dosyalarında bu satır bulunmaktadır. Bu satır daha önceden verilmiş çekirdek parametrelerini yoksayar. Buna benzer bir satır ayar dosyasında var ise, o satırı silin.

EK BİLGİ: Şayet 0 boyutlu bir ramdisk kullanmaya kalkarsanız, bu davranış manasız olacağı için, çekirdek panik verebilir.



## A. Özkaynaklar ve Göstericiler

Bir paketi alırken, geçerli bir sebebiniz olmadığı sürece, en son sürümü aldığınızdan emin olun.

### A.1. Hazır açılış disketleri

Dağıtımların açılış disketleri için kaynaklar mevcuttur. *Bu makinelere aşırı yük binmemesi için, lütfen yansılardan birisini kullanın.*

- Slackware açılış disketleri<sup>(B42)</sup>, kök disketleri<sup>(B43)</sup> ve Slackware yansıları<sup>(B44)</sup>
- Red Hat açılış disketleri<sup>(B45)</sup> ve Red Hat yansıları<sup>(B46)</sup>
- Debian açılış disketleri<sup>(B47)</sup> ve Debian yansıları<sup>(B48)</sup>
- Mandrake ftp yansıları<sup>(B49)</sup>

Bu dağıtım açılış disklerine ek olarak, aşağıdaki kurtarma diski bíteşlemleri de mevcuttur. Aksi belirtilmediği sürece bunlara şu dizin içinde ulaşabilirsiniz: <http://www.ibiblio.org/pub/Linux/system/recovery/!INDEX.html>

- **RIP** bir açılış vekurtarma setidir ve çeşitli sürümleri vardır: birisi 1.44M'lik disketler, bir diğeri ise CD-ROM içindir. Geniş bir dosya desteğine ve disk bakım ve kurtarma için pek çok uygulama aracına sahiptir. ext2, ext3, iso9660, msdos, ntfs, reiserfs, ufs ve vfat desteği vardır. RIP setini bulabileceğiniz bir yer: <http://www.tux.org/pub/people/kent-robotti/loopleftlinux/rip/index.html>
- Tom Oehser tarafından yapılmış olan **tomsrtbt** tek bir disket üzerine kurulmuş olan Çekirdek 2.0'a sahip önyüklem ve kök disketidir. Geniş bir desteğe ve uygulama araçlarına sahiptir. IDE, SCSI, teyp ağ bağdaştırıcıları, PCMCIA, vs. desteği vardır. Tamir tadilat maksatlı 100 kusür uygulama ve araç disket içinde mevcuttur. Paket içinde ayrıca bíteşlemi parçalara ayırmak ve yeniden oluşturmak için gerekli betikler de mevcuttur. Bu sayede, gerekli olduğu durumlarda, yeni araçlar ekleme olanağı elde edilmektedir.
- 1.3.84 çekirdeği üzerine kurulmuş **rescue02**, John Comyns tarafından yapılmıştır. Kurtarma diskidir. IDE, Adaptec 1542 ve NCR53C7,8xx desteği vardır. ELF ikilik dosyalarını kullanır ama başka sistemler üzerinde de kullanılacak yeterince komuta sahiptir. Diğer SCSI kartlar için, açılış işlemi sonrasında kullanılabilecek modüllere sahiptir. 3MB civarı ramdisk kullandığı için, 4MB'lik RAM'e sahip sistemlerde çalışmayabilir.
- **resque\_disk-2.0.22**, Sergei Viznyuk tarafından yapılmıştır, Çekirdek 2.0.22 üzerine kurulu yetenekli bir önyüklem ve kök disketidir. Gömülü olarak IDE, pek çok değişik SCSI kontrolcüsü ve ELF/A.out desteği mevcuttur. Ayrıca pek çok modül ve sabit disk tamir-tadilatı için gerekli araç gereç içerir.
- **cramdisk bíteşlemleri**<sup>(B52)</sup>, çekirdek 2.0.23 üzerine kurulu 4MB ve 8MB'lik sistemler içindir. Matematik emülatörü ve ağ (PPP ve dialin betiği, NE2000, 3C509) ve paralel port ZIP sürücüsü desteği içerirler. Bu disket bíteşlemleri 4MB'lık bir 386 üzerinde çalışır. MSDOS desteği vardır, bu sayede internetten bir DOS bölüme indirebilirsiniz.

### A.2. Kurtarma paketleri

[www.ibiblio.org](http://www.ibiblio.org) adresinde, kurtarma disketleri yapmak için çeşitli paketler mevcuttur. Bu paketler ile açılış disketlerinin yapımına dosyaları dahil etmek ve bu işi özdevinimli yapmak için çeşitli setler tanımlayabilirsiniz. Ayrıntılı bilgi için: <http://www.ibiblio.org/pub/Linux/system/recovery/!INDEX.html>. Dosya tarihlerini kontrol etmeyi unutmayın. Bu paketlerin bazıları yıllardan beri güncellenmedi ve ramdiske yüklenen sıkıştırılmış kök dosya sistemi desteklenmemektedir. Bildiğimiz kadarı ile bunu yapacak tek paket **Yard**<sup>(B54)</sup>'dır.

### A.3. LILO — Linux yükleyicisi

Werner Almesberger tarafından yazılmıştır. Mükemmel bir önyükleyicidir ve belgeleri önyükleme sektörü içeriği ve önyükleme sürecinin ilk safhaları hakkında bilgi içermektedir.

Ftp'den adresinden erişilebilir. Ayrıca Metalab ve yansılarında da bulunmaktadır.

### A.4. Ramdisk kullanımı

Ramdiskin nasıl çalıştığını anlatan mükemmel belgeler, çekirdek belgeleri arasında bulunabilir. `/usr/src/linux/Documentation/ramdisk.txt` dosyasına göz atınız. Paul Gortmaker tarafından yazılmıştır ve sıkıştırılmış bir ramdisk yapmanın anlatıldığı bir bölüm de vardır.

### A.5. Linux önyükleme süreci

Linux önyükleme süreci hakkında bazı kaynaklar:

- [Linux Sistem Yöneticisinin Kılavuzu](#)<sup>(B56)</sup> önyükleme süreci hakkında bir bölüm içermektedir.
- [LILO "Technical overview"](#)<sup>(B57)</sup> alt seviyede eksiksiz bir teknik açıklamaya sahiptir.
- Kaynak kodları mükemmel bir kaynaktır. Aşağıda önyükleme süreci ile ilgili bazı çekirdek dosyaları bulunmaktadır. Şayet Linux çekirdek kaynak kodlarına sahipseniz, kendi makinenizde `/usr/src/linux` altında bu dosyaları bulabilirsiniz. Ayrıca olarak, Shigio Yamaguchi (shigio at tamacom.com)'nun çekirdek kaynak kodlarını okumak için çok hoş bir [hypertext çekirdek tarayıcısı](#)<sup>(B58)</sup> vardır. İlgili belgeler:

`arch/i386/boot/bootsect.S` ve `setup.S`

Önyükleme sektörünün sembolik makina dili kodlarını içerir.

`arch/i386/boot/compressed/misc.c`

Sıkıştırılmış çekirdeği açan kodlar bulunur.

`arch/i386/kernel/`

Dizin çekirdek ilkendirme kodlarını içerir. ramdisk sözcüğü `setup.c` dosyasında tanımlanmıştır.

`drivers/block/rd.c`

Ramdisk sürücüsünü içerir. `rd_load` ve `rd_load_image` süreçleri bir aygıttaki blokları ramdiskin içine yükler. `identify_ramdisk_image` süreci ne tür bir dosya sisteminin bulunduğunu ve sıkıştırılmış olup olmadığını saptar.

## B. LILO önyükleme hata kodları

Bu sorular USENET üzerinde o kadar çok soruldu ki bunları amme hizmeti olarak buraya koymayı uygun bulduk. Bu özet Werner Almsberger'in [LILO kullanıcı belgelerinden](#)<sup>(B59)</sup> alıntıdır.

LILO kendini yüklediği zaman **LILO** kelimesi ekranda görünür. Her bir harf belirli bir işlemin gerçekleşmesi öncesinde veya sonrasında ekrana yazılır. Şayet LILO bir noktada çakılırsa, o ana kadar ekrana gelmiş harfler sorunu aydınlatmak için kullanılabilir.

Çıktı	Sorun
(çıktı yok)	LILO'nun hiçbir parçası yüklenmemiş demektir. Ya LILO kurulmamıştır ya da içinde bulunduğu disk bölümünün açılış bayrağı etkin değildir.
L	LILO yüklenirken ilk aşama. LILO başlamış ama ikinci aşamaya geçememiş. İki haneli hata kodu sorunu tanımlar ("Disk hata kodları" bölümüne bakınız). Bir aygıt bozukluğu veya disk geometri hatasını belirtir (hatalı disk parametreleri gibi).



LI	Önyükleyicinin ilk evresi ikinci aşamayı yüklemiş fakat onu çalıştırmadan hata vermiş demektir. Ya disk geometri hatası söz konusudur ya da <code>/boot/boot.b</code> 'yi eşlem yükleyiciyi çalıştırmadan taşımışsınız demektir.
LIL	İkinci evre başlatılmış, fakat map dosyasından tanımlama tablosu yüklenmesi gerçekleştirilememiş. Aygıt hatası veya disk geometrisi uyumsuzluğu söz konusu olabilir.
LIL?	İkinci evrede önyükleyici yanlış adrese yüklendi. Bu genelde, farkedilmesi güç bir geometri hatasına veya <code>/boot/boot.b</code> 'nin eşlem yükleyicisi çalıştırılmadan taşındığına dehalettir.
LIL-	Tanımlama tablosu bozuk. Ya disk geometrisi uyumsuzluğundan ya da <code>/boot/map</code> 'in eşlem yükleyicisi çalıştırılmadan taşınmasından meydana gelmiştir.
LILO	LILO başarılı bir şekilde yüklendi.

LILO önyükleme biteşlemini yüklemeye çalışırken, BIOS hata sinyali verirse, kendi hata kodu görüntülenir. Bu kodlar `0x00`'dan `0xbb`'ye kadardır. Bunların açıklamaları için LILO Kullanıcı Kılavuzuna bakınız.

## C. Kök dosya sistemi listesi örneği

```
/:
drwx--x--x  2 root    root      1024 Nov  1 15:39 bin
drwx--x--x  2 root    root      4096 Nov  1 15:39 dev
drwx--x--x  3 root    root      1024 Nov  1 15:39 etc
drwx--x--x  4 root    root      1024 Nov  1 15:39 lib
drwx--x--x  5 root    root      1024 Nov  1 15:39 mnt
drwx--x--x  2 root    root      1024 Nov  1 15:39 proc
drwx--x--x  2 root    root      1024 Nov  1 15:39 root
drwx--x--x  2 root    root      1024 Nov  1 15:39 sbin
drwx--x--x  2 root    root      1024 Nov  1 15:39 tmp
drwx--x--x  7 root    root      1024 Nov  1 15:39 usr
drwx--x--x  5 root    root      1024 Nov  1 15:39 var

/bin:
-rwx--x--x  1 root    root      62660 Nov  1 15:39 ash
-rwx--x--x  1 root    root       9032 Nov  1 15:39 cat
-rwx--x--x  1 root    root     10276 Nov  1 15:39 chmod
-rwx--x--x  1 root    root      9592 Nov  1 15:39 chown
-rwx--x--x  1 root    root     23124 Nov  1 15:39 cp
-rwx--x--x  1 root    root     23028 Nov  1 15:39 date
-rwx--x--x  1 root    root     14052 Nov  1 15:39 dd
-rwx--x--x  1 root    root     14144 Nov  1 15:39 df
-rwx--x--x  1 root    root     69444 Nov  1 15:39 egrep
-rwx--x--x  1 root    root        395 Nov  1 15:39 false
-rwx--x--x  1 root    root     69444 Nov  1 15:39 fgrep
-rwx--x--x  1 root    root     69444 Nov  1 15:39 grep
-rwx--x--x  3 root    root     45436 Nov  1 15:39 gunzip
-rwx--x--x  3 root    root     45436 Nov  1 15:39 gzip
-rwx--x--x  1 root    root      8008 Nov  1 15:39 hostname
-rwx--x--x  1 root    root     12736 Nov  1 15:39 ln
-rws--x--x  1 root    root     15284 Nov  1 15:39 login
-rwx--x--x  1 root    root     29308 Nov  1 15:39 ls
-rwx--x--x  1 root    root      8268 Nov  1 15:39 mkdir
-rwx--x--x  1 root    root      8920 Nov  1 15:39 mknod
-rwx--x--x  1 root    root     24836 Nov  1 15:39 more
-rws--x--x  1 root    root     37640 Nov  1 15:39 mount
-rwx--x--x  1 root    root     12240 Nov  1 15:39 mt
-rwx--x--x  1 root    root     12932 Nov  1 15:39 mv
-r-x--x--x  1 root    root     12324 Nov  1 15:39 ps
```

```

-rwx--x--x 1 root root 5388 Nov 1 15:39 pwd
-rwx--x--x 1 root root 10092 Nov 1 15:39 rm
lrwxrwxrwx 1 root root 3 Nov 1 15:39 sh -> ash
-rwx--x--x 1 root root 25296 Nov 1 15:39 stty
-rws--x--x 1 root root 12648 Nov 1 15:39 su
-rwx--x--x 1 root root 4444 Nov 1 15:39 sync
-rwx--x--x 1 root root 19712 Nov 1 15:39 touch
-rwx--x--x 1 root root 395 Nov 1 15:39 true
-rws--x--x 1 root root 19084 Nov 1 15:39 umount
-rwx--x--x 1 root root 5368 Nov 1 15:39 uname
-rwx--x--x 3 root root 45436 Nov 1 15:39 zcat

/dev:
lrwxrwxrwx 1 root root 6 Nov 1 15:39 cdrom -> cdu31a
brw-rw-r-- 1 root root 15, 0 May 5 1998 cdu31a
crw----- 1 root root 4, 0 Nov 1 15:29 console
crw-rw-rw- 1 root uucp 5, 64 Sep 9 19:46 cua0
crw-rw-rw- 1 root uucp 5, 65 May 5 1998 cua1
crw-rw-rw- 1 root uucp 5, 66 May 5 1998 cua2
crw-rw-rw- 1 root uucp 5, 67 May 5 1998 cua3
brw-rw---- 1 root floppy 2, 0 Aug 8 13:54 fd0
brw-rw---- 1 root floppy 2, 36 Aug 8 13:54 fd0CompaQ
brw-rw---- 1 root floppy 2, 84 Aug 8 13:55 fd0D1040
brw-rw---- 1 root floppy 2, 88 Aug 8 13:55 fd0D1120
brw-rw---- 1 root floppy 2, 12 Aug 8 13:54 fd0D360
brw-rw---- 1 root floppy 2, 16 Aug 8 13:54 fd0D720
brw-rw---- 1 root floppy 2, 120 Aug 8 13:55 fd0D800
brw-rw---- 1 root floppy 2, 32 Aug 8 13:54 fd0E2880
brw-rw---- 1 root floppy 2, 104 Aug 8 13:55 fd0E3200
brw-rw---- 1 root floppy 2, 108 Aug 8 13:55 fd0E3520
brw-rw---- 1 root floppy 2, 112 Aug 8 13:55 fd0E3840
brw-rw---- 1 root floppy 2, 28 Aug 8 13:54 fd0H1440
brw-rw---- 1 root floppy 2, 124 Aug 8 13:55 fd0H1600
brw-rw---- 1 root floppy 2, 44 Aug 8 13:55 fd0H1680
brw-rw---- 1 root floppy 2, 60 Aug 8 13:55 fd0H1722
brw-rw---- 1 root floppy 2, 76 Aug 8 13:55 fd0H1743
brw-rw---- 1 root floppy 2, 96 Aug 8 13:55 fd0H1760
brw-rw---- 1 root floppy 2, 116 Aug 8 13:55 fd0H1840
brw-rw---- 1 root floppy 2, 100 Aug 8 13:55 fd0H1920
lrwxrwxrwx 1 root root 7 Nov 1 15:39 fd0H360 -> fd0D360
lrwxrwxrwx 1 root root 7 Nov 1 15:39 fd0H720 -> fd0D720
brw-rw---- 1 root floppy 2, 52 Aug 8 13:55 fd0H820
brw-rw---- 1 root floppy 2, 68 Aug 8 13:55 fd0H830
brw-rw---- 1 root floppy 2, 4 Aug 8 13:54 fd0d360
brw-rw---- 1 root floppy 2, 8 Aug 8 13:54 fd0h1200
brw-rw---- 1 root floppy 2, 40 Aug 8 13:54 fd0h1440
brw-rw---- 1 root floppy 2, 56 Aug 8 13:55 fd0h1476
brw-rw---- 1 root floppy 2, 72 Aug 8 13:55 fd0h1494
brw-rw---- 1 root floppy 2, 92 Aug 8 13:55 fd0h1600
brw-rw---- 1 root floppy 2, 20 Aug 8 13:54 fd0h360
brw-rw---- 1 root floppy 2, 48 Aug 8 13:55 fd0h410
brw-rw---- 1 root floppy 2, 64 Aug 8 13:55 fd0h420
brw-rw---- 1 root floppy 2, 24 Aug 8 13:54 fd0h720
brw-rw---- 1 root floppy 2, 80 Aug 8 13:55 fd0h880
brw-rw---- 1 root disk 3, 0 May 5 1998 hda
brw-rw---- 1 root disk 3, 1 May 5 1998 hda1
brw-rw---- 1 root disk 3, 2 May 5 1998 hda2

```

```

brw-rw----- 1 root    disk      3,    3 May  5 1998 hda3
brw-rw----- 1 root    disk      3,    4 May  5 1998 hda4
brw-rw----- 1 root    disk      3,    5 May  5 1998 hda5
brw-rw----- 1 root    disk      3,    6 May  5 1998 hda6
brw-rw----- 1 root    disk      3,   64 May  5 1998 hdb
brw-rw----- 1 root    disk      3,   65 May  5 1998 hdb1
brw-rw----- 1 root    disk      3,   66 May  5 1998 hdb2
brw-rw----- 1 root    disk      3,   67 May  5 1998 hdb3
brw-rw----- 1 root    disk      3,   68 May  5 1998 hdb4
brw-rw----- 1 root    disk      3,   69 May  5 1998 hdb5
brw-rw----- 1 root    disk      3,   70 May  5 1998 hdb6
crw-r----- 1 root    kmem       1,    2 May  5 1998 kmem
crw-r----- 1 root    kmem       1,    1 May  5 1998 mem
lrwxrwxrwx   1 root    root              12 Nov  1 15:39 modem -> ttyS1
lrwxrwxrwx   1 root    root              12 Nov  1 15:39 mouse -> psaux
crw-rw-rw-   1 root    root          1,    3 May  5 1998 null
crwxrwxrwx   1 root    root        10,    1 Oct  5 20:22 psaux
brw-r----- 1 root    disk          1,    1 May  5 1998 ram
brw-rw----- 1 root    disk          1,    0 May  5 1998 ram0
brw-rw----- 1 root    disk          1,    1 May  5 1998 ram1
brw-rw----- 1 root    disk          1,    2 May  5 1998 ram2
brw-rw----- 1 root    disk          1,    3 May  5 1998 ram3
brw-rw----- 1 root    disk          1,    4 May  5 1998 ram4
brw-rw----- 1 root    disk          1,    5 May  5 1998 ram5
brw-rw----- 1 root    disk          1,    6 May  5 1998 ram6
brw-rw----- 1 root    disk          1,    7 May  5 1998 ram7
brw-rw----- 1 root    disk          1,    8 May  5 1998 ram8
brw-rw----- 1 root    disk          1,    9 May  5 1998 ram9
lrwxrwxrwx   1 root    root              4 Nov  1 15:39 ramdisk -> ram0

```

\*\*\* Sadece kendi kullandığım IDE bölümdeki aygıtları ekledim.

\*\*\* Şayet siz SCSI aygıtlar kullanıyorsanız, bunlar yerine

\*\*\* /dev/sdXX aygıtlar kullanın.

```

crw----- 1 root    root          4,    0 May  5 1998 tty0
crw-w----- 1 root    tty          4,    1 Nov  1 15:39 tty1
crw----- 1 root    root          4,    2 Nov  1 15:29 tty2
crw----- 1 root    root          4,    3 Nov  1 15:29 tty3
crw----- 1 root    root          4,    4 Nov  1 15:29 tty4
crw----- 1 root    root          4,    5 Nov  1 15:29 tty5
crw----- 1 root    root          4,    6 Nov  1 15:29 tty6
crw----- 1 root    root          4,    7 May  5 1998 tty7
crw----- 1 root    tty          4,    8 May  5 1998 tty8
crw----- 1 root    tty          4,    9 May  8 12:57 tty9
crw-rw-rw- 1 root    root          4,   65 Nov  1 12:17 ttyS1
crw-rw-rw- 1 root    root          1,    5 May  5 1998 zero

```

/etc:

```

-rw----- 1 root    root        164 Nov  1 15:39 conf.modules
-rw----- 1 root    root        668 Nov  1 15:39 fstab
-rw----- 1 root    root         71 Nov  1 15:39 gettydefs
-rw----- 1 root    root       389 Nov  1 15:39 group
-rw----- 1 root    root       413 Nov  1 15:39 inittab
-rw----- 1 root    root         65 Nov  1 15:39 issue
-rw-r--r-- 1 root    root       746 Nov  1 15:39 ld.so.cache
-rw----- 1 root    root         32 Nov  1 15:39 motd
-rw----- 1 root    root       949 Nov  1 15:39 nsswitch.conf

```

```

drwx--x--x  2 root    root          1024 Nov  1 15:39 pam.d
-rw-----  1 root    root           139 Nov  1 15:39 passwd
-rw-----  1 root    root           516 Nov  1 15:39 profile
-rwx--x--x  1 root    root           387 Nov  1 15:39 rc
-rw-----  1 root    root            55 Nov  1 15:39 shells
-rw-----  1 root    root           774 Nov  1 15:39 termcap
-rw-----  1 root    root            78 Nov  1 15:39 ttytype
lrwxrwxrwx  1 root    root            15 Nov  1 15:39 utmp -> ../var/run/utmp
lrwxrwxrwx  1 root    root            15 Nov  1 15:39 wtmp -> ../var/log/wtmp

/etc/pam.d:
-rw-----  1 root    root           356 Nov  1 15:39 other

/lib:
-rwxr-xr-x   1 root    root          45415 Nov  1 15:39 ld-2.0.7.so
lrwxrwxrwx   1 root    root            11 Nov  1 15:39 ↵
ld-linux.so.2 -> ld-2.0.7.so
-rwxr-xr-x   1 root    root        731548 Nov  1 15:39 libc-2.0.7.so
lrwxrwxrwx   1 root    root            13 Nov  1 15:39 ↵
libc.so.6 -> libc-2.0.7.so
lrwxrwxrwx   1 root    root            17 Nov  1 15:39 ↵
libcom_err.so.2 -> libcom_err.so.2.0
-rwxr-xr-x   1 root    root          6209 Nov  1 15:39 libcom_err.so.2.0
-rwxr-xr-x   1 root    root        153881 Nov  1 15:39 libcrypt-2.0.7.so
lrwxrwxrwx   1 root    root            17 Nov  1 15:39 ↵
libcrypt.so.1 -> libcrypt-2.0.7.so
-rwxr-xr-x   1 root    root          12962 Nov  1 15:39 libdl-2.0.7.so
lrwxrwxrwx   1 root    root            14 Nov  1 15:39 ↵
libdl.so.2 -> libdl-2.0.7.so
lrwxrwxrwx   1 root    root            16 Nov  1 15:39 ↵
libext2fs.so.2 -> libext2fs.so.2.4
-rwxr-xr-x   1 root    root          81382 Nov  1 15:39 libext2fs.so.2.4
-rwxr-xr-x   1 root    root          25222 Nov  1 15:39 libnsl-2.0.7.so
lrwxrwxrwx   1 root    root            15 Nov  1 15:39 ↵
libnsl.so.1 -> libnsl-2.0.7.so
-rwx--x--x   1 root    root        178336 Nov  1 15:39 libnss_files-2.0.7.so
lrwxrwxrwx   1 root    root            21 Nov  1 15:39 ↵
libnss_files.so.1 -> libnss_files-2.0.7.so
lrwxrwxrwx   1 root    root            14 Nov  1 15:39 ↵
15:39 libpam.so.0 -> libpam.so.0.64
-rwxr-xr-x   1 root    root          26906 Nov  1 15:39 libpam.so.0.64
lrwxrwxrwx   1 root    root            19 Nov  1 15:39 ↵
libpam_misc.so.0 -> libpam_misc.so.0.64
-rwxr-xr-x   1 root    root           7086 Nov  1 15:39 libpam_misc.so.0.64
-r-xr-xr-x   1 root    root          35615 Nov  1 15:39 libproc.so.1.2.6
lrwxrwxrwx   1 root    root            15 Nov  1 15:39 ↵
libpwdb.so.0 -> libpwdb.so.0.54
-rw-r-r---   1 root    root        121899 Nov  1 15:39 libpwdb.so.0.54
lrwxrwxrwx   1 root    root            19 Nov  1 15:39 ↵
libtermcap.so.2 -> libtermcap.so.2.0.8
-rwxr-xr-x   1 root    root          12041 Nov  1 15:39 libtermcap.so.2.0.8
-rwxr-xr-x   1 root    root          12874 Nov  1 15:39 libutil-2.0.7.so
lrwxrwxrwx   1 root    root            16 Nov  1 15:39 ↵
libutil.so.1 -> libutil-2.0.7.so
lrwxrwxrwx   1 root    root            14 Nov  1 15:39 ↵
libuuid.so.1 -> libuuid.so.1.1
-rwxr-xr-x   1 root    root           8039 Nov  1 15:39 libuuid.so.1.1

```

```

drwx--x--x  3 root    root        1024 Nov  1 15:39 modules
drwx--x--x  2 root    root        1024 Nov  1 15:39 security

/lib/modules:
drwx--x--x  4 root    root        1024 Nov  1 15:39 2.0.35

/lib/modules/2.0.35:
drwx--x--x  2 root    root        1024 Nov  1 15:39 block
drwx--x--x  2 root    root        1024 Nov  1 15:39 cdrom

/lib/modules/2.0.35/block:
drwx-----  1 root    root        7156 Nov  1 15:39 loop.o

/lib/modules/2.0.35/cdrom:
drwx-----  1 root    root       24108 Nov  1 15:39 cdu31a.o

/lib/security:
-rwx--x--x  1 root    root        8771 Nov  1 15:39 pam_permit.so

*** Directory stubs for mounting

/mnt:
drwx--x--x  2 root    root        1024 Nov  1 15:39 cdrom
drwx--x--x  2 root    root        1024 Nov  1 15:39 floppy

/proc:

/root:
-rw-----  1 root    root        176 Nov  1 15:39 .bashrc
-rw-----  1 root    root        182 Nov  1 15:39 .cshrc
-rwx--x--x  1 root    root        455 Nov  1 15:39 .profile
-rw-----  1 root    root       4014 Nov  1 15:39 .tcshrc

/sbin:
-rwx--x--x  1 root    root       23976 Nov  1 15:39 depmod
-rwx--x--x  2 root    root      274600 Nov  1 15:39 e2fsck
-rwx--x--x  1 root    root       41268 Nov  1 15:39 fdisk
-rwx--x--x  1 root    root        9396 Nov  1 15:39 fsck
-rwx--x--x  2 root    root      274600 Nov  1 15:39 fsck.ext2
-rwx--x--x  1 root    root       29556 Nov  1 15:39 getty
-rwx--x--x  1 root    root        6620 Nov  1 15:39 halt
-rwx--x--x  1 root    root       23116 Nov  1 15:39 init
-rwx--x--x  1 root    root       25612 Nov  1 15:39 insmod
-rwx--x--x  1 root    root       10368 Nov  1 15:39 kernel.d
-rwx--x--x  1 root    root      110400 Nov  1 15:39 ldconfig
-rwx--x--x  1 root    root        6108 Nov  1 15:39 lsmod
-rwx--x--x  2 root    root       17400 Nov  1 15:39 mke2fs
-rwx--x--x  1 root    root        4072 Nov  1 15:39 mkfs
-rwx--x--x  2 root    root       17400 Nov  1 15:39 mkfs.ext2
-rwx--x--x  1 root    root        5664 Nov  1 15:39 mkswap
-rwx--x--x  1 root    root       22032 Nov  1 15:39 modprobe
lrwxrwxrwx  1 root    root         4 Nov  1 15:39 reboot -> halt
-rwx--x--x  1 root    root        7492 Nov  1 15:39 rmmmod
-rwx--x--x  1 root    root       12932 Nov  1 15:39 shutdown
lrwxrwxrwx  1 root    root         6 Nov  1 15:39 swapoff -> swapon
-rwx--x--x  1 root    root       5124 Nov  1 15:39 swapon

```

```

lrwxrwxrwx    1 root    root           4 Nov  1 15:39 telinit -> init
-rwx--x--x    1 root    root        6944 Nov  1 15:39 update

/tmp:

/usr:
drwx--x--x    2 root    root        1024 Nov  1 15:39 bin
drwx--x--x    2 root    root        1024 Nov  1 15:39 lib
drwx--x--x    3 root    root        1024 Nov  1 15:39 man
drwx--x--x    2 root    root        1024 Nov  1 15:39 sbin
drwx--x--x    3 root    root        1024 Nov  1 15:39 share
lrwxrwxrwx    1 root    root         10 Nov  1 15:39 tmp -> ../var/tmp

/usr/bin:
-rwx--x--x    1 root    root       37164 Nov  1 15:39 afio
-rwx--x--x    1 root    root        5044 Nov  1 15:39 chroot
-rwx--x--x    1 root    root       10656 Nov  1 15:39 cut
-rwx--x--x    1 root    root      63652 Nov  1 15:39 diff
-rwx--x--x    1 root    root      12972 Nov  1 15:39 du
-rwx--x--x    1 root    root     56552 Nov  1 15:39 find
-r-x--x--x    1 root    root        6280 Nov  1 15:39 free
-rwx--x--x    1 root    root        7680 Nov  1 15:39 head
-rwx--x--x    1 root    root        8504 Nov  1 15:39 id
-r-sr-xr-x    1 root    bin         4200 Nov  1 15:39 passwd
-rwx--x--x    1 root    root      14856 Nov  1 15:39 tail
-rwx--x--x    1 root    root      19008 Nov  1 15:39 tr
-rwx--x--x    1 root    root        7160 Nov  1 15:39 wc
-rwx--x--x    1 root    root        4412 Nov  1 15:39 whoami

/usr/lib:
lrwxrwxrwx    1 root    root          17 Nov  1 15:39 ↵
libncurses.so.4 -> libncurses.so.4.2
-rw-r-r---    1 root    root    260474 Nov  1 15:39 libncurses.so.4.2

/usr/sbin:
-r-x--x--x    1 root    root      13684 Nov  1 15:39 fuser
-rwx--x--x    1 root    root       3876 Nov  1 15:39 mklost+found

/usr/share:
drwx--x--x    4 root    root        1024 Nov  1 15:39 terminfo

/usr/share/terminfo:
drwx--x--x    2 root    root        1024 Nov  1 15:39 l
drwx--x--x    2 root    root        1024 Nov  1 15:39 v

/usr/share/terminfo/l:
-rw-----    1 root    root        1552 Nov  1 15:39 linux
-rw-----    1 root    root        1516 Nov  1 15:39 linux-m
-rw-----    1 root    root        1583 Nov  1 15:39 linux-nic

/usr/share/terminfo/v:
-rw-----    2 root    root        1143 Nov  1 15:39 vt100
-rw-----    2 root    root        1143 Nov  1 15:39 vt100-am

/var:
drwx--x--x    2 root    root        1024 Nov  1 15:39 log
drwx--x--x    2 root    root        1024 Nov  1 15:39 run

```

```

drwx--x--x  2 root    root          1024 Nov  1 15:39 tmp

/var/log:
-rw-----  1 root    root              0 Nov  1 15:39 wtmp

/var/run:
-rw-----  1 root    root              0 Nov  1 15:39 utmp

/var/tmp:
```

## D. Uygulamalar disketi dizin listesi örneği

```

total 579
-rwxr-xr-x  1 root    root          42333 Jul 28 19:05 cpio
-rwxr-xr-x  1 root    root          32844 Aug 28 19:50 debugfs
-rwxr-xr-x  1 root    root        103560 Jul 29 21:31 elvis
-rwxr-xr-x  1 root    root          29536 Jul 28 19:04 fdisk
-rw-r--r---  1 root    root        128254 Jul 28 19:03 ftape.o
-rwxr-xr-x  1 root    root          17564 Jul 25 03:21 ftmt
-rwxr-xr-x  1 root    root          64161 Jul 29 20:47 grep
-rwxr-xr-x  1 root    root          45309 Jul 29 20:48 gzip
-rwxr-xr-x  1 root    root          23560 Jul 28 19:04 insmod
-rwxr-xr-x  1 root    root           118 Jul 28 19:04 lsmod
lrwxrwxrwx  1 root    root           5 Jul 28 19:04 mt -> mt-st
-rwxr-xr-x  1 root    root          9573 Jul 28 19:03 mt-st
lrwxrwxrwx  1 root    root           6 Jul 28 19:05 rmmod -> insmod
-rwxr-xr-x  1 root    root        104085 Jul 28 19:05 tar
lrwxrwxrwx  1 root    root           5 Jul 29 21:35 vi -> elvis
```

## Kavramlar Dizini

## A

Ana Önyükleme Kaydı (MBR).....	28
aygıt (dev) dizini .....	20
aygıt dizini (/dev).....	10
aygıt sürücülerini.....	28

## B

/bin dizini .....	12
BIOS .....	6
birincil ramdisk (initrd) .....	24

## Ç

çekirdek	
derleme .....	16
dosya sistemi .....	18
parametreleri .....	17, 29
seçimi .....	16

## D

dosya düğümleri	
ayırma .....	17
kök dosya sisteminde ayrılması .....	9
dosyaların budanması .....	11

## E

ELF .....	13
/etc dizini .....	11
/etc/rc.d/ .....	6

## F

fstab dosyası .....	12
---------------------	----

## G

geridönüş aygıtı .....	9
getty .....	13
gölge parolalar .....	11

## İ

init .....	6, 13
inittab .....	6
inittab dosyası .....	12

## K

kabuklar .....	21, 22
kök diski .....	7
kök dosya sistemi .....	6, 33
güncelleme .....	28

kullanıcı grupları .....	11
kütüphane (lib) dizini .....	13
kütüphaneler .....	13
soyulması .....	22

## L

ld.so.cache .....	15
ldconfig .....	15
libc.so .....	13
LILO	
hata kodları .....	32
kaldırılması .....	28
lilo dosya sistemi .....	17
lilo.conf .....	17
login .....	13

## M

modüller .....	15
----------------	----

## O

objcopy .....	22
---------------	----

## Ö

önyükleme aygıtı .....	6
Önyükleme sektörü .....	6
önyükleme süreci .....	6, 32
önyükleme ve kök .....	7

## P

passwd dosyası .....	11
----------------------	----

## R

ramdisk .....	6, 23, 30, 32
ramdisk sözcüğü .....	19, 23, 28
rc betiği .....	12
rdev .....	18, 19, 23, 28

## S

/sbin dizini .....	12
sh .....	13
shells dosyası .....	13
sıkıştırılmış dosya sistemi .....	6
sorunların çözülmesi .....	20
strip .....	22
sysinit .....	12
sysinit .....	6

## U

uygulamalar disketi .....	7, 23, 23, 39
---------------------------	---------------

## Y

yerleri değiştirilemeyenler .....	12, 30
yükleyiciler .....	13, 21



## Notlar

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

(B1) <http://en.tldp.org/copyright.html>

(B2) <http://en.tldp.org/copyright.html>

- (1) Ç.N: Bu kılavuz açılış/kurtarma disketlerini anlatmakla birlikte, bir Linux sisteminde açılış sürecinin ve çekirdek yüklenmesinin nasıl gerçekleştiği, gerekli olan temel bileşenlerin neler olduğunu anlamaya yönelik iyi bir başlangıç kılavuzu olarak kabul edilebilir. Kılavuz içerisinde bu konular–başlangıç seviyesi için ayrıntılı bir şekilde ele alınmış ve sistemin genel hatları göz önüne serilmiştir. Bu NASIL'ı, GNU/Linux sistemlerinin temel yapısını anlamaya yönelik sağlam bir adım olarak değerlendirmek gerektiği inancındayım. Bir yerden başlamak gerek, öyle değil mi?

(B4) <http://www.linuxlots.com/~fawcett/Bootdisk-«HOWTO/index.html>

(B8) <http://en.tldp.org/copyright.html>

(B11) [../sag/sag.pdf](http://sag.sag.pdf)

- (2) Burada anlatılan dizin yapısı, sadece kök disket kullanımı içindir. Gerçek Linux Sistemleri hangi dosyanın nerede bulunacağını belirlediği daha karmaşık ve daha disiplinli bir yapıya sahiptir: [Dosyasistemi Hiyerarşi Standardı](http://www.pathname.com/fhs/2.0/fhs-«toc.html) (<http://www.pathname.com/fhs/2.0/fhs-«toc.html>).

- (3) Ç.N: Burada kelime ya da sözcük ile ifade edilen 16 bitlik bir alandır (8 bit = bayt, 16 bit = kelime [word]).

(B24) <http://www.busybox.net/>

(B33) [../howto/cd-«yazimi.pdf](http://howto.cd-«yazimi.pdf)

(B38)

(B42) <http://distro.ibiblio.org/pub/Linux/distributions/slackware/bootdsk.144/>

(B43) <http://distro.ibiblio.org/pub/Linux/distributions/slackware/rootdsk/>

(B44) <http://www.slackware.com/getslack/>

(B45) <ftp://ftp.redhat.com/pub/redhat/linux/current/en/os/i386/images/>

(B46) <http://www.redhat.com/mirrors.html>

(B47) <ftp://ftp.debian.org/debian/dists/stable/main/disks-«i386/current/>

(B48) <ftp://ftp.debian.org/debian/README.mirrors.html>

(B49) <http://www.linux-«mandrake.com/en/ftp.php3>

(B52) <http://www.ibiblio.org/pub/Linux/system/recovery/images>

(B54) <http://www.linuxlots.com/~fawcett/yard/index.html>

---

(B56) `../sag/sag.pdf`

---

(B57) `http://www.ibiblio.org/pub/Linux/system/boot/lilo/lilo-t-21.ps.gz`

---

(B58) `http://www.tamacom.com/tour/linux/index.html`

---

(B59) `http://www.ibiblio.org/pub/Linux/system/boot/lilo/lilo-u-21.ps.gz`

---

Bu dosya (bootdisk-howto.pdf), belgenin XML biçiminin  
T<sub>E</sub>XLive ve belgeler-xsl paketlerindeki araçlar kullanılarak  
PDF biçimine dönüştürülmesiyle elde edilmiştir.

22 Ocak 2007