

# Linux Disk Bölümleme NASIL

Yazan:  
**Anthony Lissot**

Yazan:  
**Kristian Koehntopp**

Çeviren:  
**Oğuz Yarımtepe**  
<oguz@comu.edu.tr>

11 Şubat 2007

## Özet

Bu küçük Linux NASIL belgesi, IDE ve SCSI disklerde bölümlemenin nasıl planlandığını ve yapıldığını öğretmektedir. Bölümleme terimlerini irdelemekte, büyüklük ve yer konularını ele almaktadır. Bölümleme tabloları oluştururken ve kurtarıırken fdisk bölümleme uygulamasının kullanımı da ele alınmıştır.

Bu belgenin en güncel sürümünü <http://lissot.net/partition> adresinde bulabilirsiniz.

## Konu Başlıkları

<b>1. Giriş</b>	5
1.1. Disk bölümü nedir?	5
1.2. Diğer Disk Bölümleme Yazılımları	5
1.3. İlgili NASIL'lar	5
1.4. Sistemdeki ek kaynaklar	6
<b>2. Aygıtlar</b>	6
2.1. Aygıt adları	6
2.1.1. İsimlendirme Kuralları	6
2.1.2. İsim Atama	7
2.1.3. Mantıksal Disk Bölümleri	7
2.2. Aygıt numaraları	8
<b>3. Disk Bölümü Türleri</b>	8
3.1. Disk Bölümü Türleri	8
3.2. Yabancı Disk Bölümü Türleri	8
3.3. Birincil Disk Bölümleri	9
3.4. Mantıksal Disk Bölümleri	9
3.5. Takas Bölümleri	9
<b>4. Bölümlendirme gereksinimleri</b>	9
4.1. Hangi Disk Bölümlerine İhtiyacım Var?	9
4.2. Durum Analizi	10
4.3. Dosya Sistemleri	10
4.3.1. Hangi dosya sistemleri kendi disk bölümlerine ihtiyaç duyar?	10
4.3.2. Disk bölümleme ölçütü olarak dosya ömrü ve yedekleme sıklığı	11
4.4. Takas Bölümleri	11
4.4.1. Takas alanı ne kadar büyük olmalıdır?	11

4.4.2. Takas alanım ne kadar büyük olabilir?	12
4.4.3. Takas alanımı nereye koymalıyım?	12
<b>5. fdisk ile diski bölümleme</b>	13
5.1. fdisk ile disk bölümü oluşturmak	13
5.1.1. fdisk kullanımı	13
5.1.2. Dört disk bölümü	14
5.1.3. Birincil ve mantıksal disk bölümleri karışık	15
5.1.4. Sunulan Örnekler	17
<b>6. Etiketler</b>	17
6.1. Aygıt Etiketleri	17
6.1.1. Basit Çalıştırma	17
6.1.2. Kullanım	18
6.2. Aygıt Etiketleri	18
<b>7. Bir ext2/3 disk bölümünü biçimlendirmek</b>	18
7.1. Basit Çalıştırma	18
7.2. Atanmış bloklar	19
<b>8. Silinmiş bir Disk Bölümünü Kurtarmak</b>	19
<b>9. Takas Alanını Ayarlamak</b>	20
9.1. Takas Dosyaları	20
9.2. Takas Dosyaları	21
9.3. Çoklu Takas Alanları	21
<b>10. Son Söz</b>	22
10.1. Disk Bölümlerinin Biçimlendirilmesi	22
10.2. Takas Alanının Etkinleştirilmesi	22
10.3. Disk Bölümlerinin Bağlanması	22
10.4. Saçılım ve Dosya sistemleri hakkında bazı gerçekler	23

**Bu çevirinin sürüm bilgileri:**

1.2	Nisan 2006	NBB
Çevirinin lisansı GPL yapıldı.		
1.1	Ocak 2006	OY
Güncelleme		
1.0	Ekim 2005	oy
İlk çeviri		

**Özgün belgenin sürüm bilgileri:**

3.5	26 Aralık 2005
Sayfa sıralaması düzenlendi. Takas alanı ayarlanamamasıyla ilgili sayfa eklendi. Disk bölümleri etiketleriyle ilgili sayfa eklendi. 4. bölümdeki en büyük takas alınıyla ilgili bilgi güncellendi. ext2/3 dosya sistemleri oluşturma adımları eklendi. Richard Calmbach tarafından bildirilen hatalı bağlar düzeltildi. XML sürüm hazırlandı.	
3.4.4	08 Mart 2004
HTML sürüm, XML sürümden üretildi. Lilo yerleştirilmesi ve takas alanı boyu konuları güncellendi.	
3.3	04 Nisan 2003
HTML sürüm SGML sürümden üretildi.	
3.3	10 Temmuz 2001
6. Bölüm'deki silindir numaralarının hesaplanması düzeltildi.	
3.2	1 Eylül 2000
2 Ekim 2000'de Dan Scott sgml sürümünü hazırladı. Giriş kısmı yeniden yazıldı. Mantıksal Aygıtlardaki aygıt isimleri kısmı yeniden yazıldı. Bölüm Türleri yeniden organize edildi. Bölümleme Gereksinimleri düzenlendi. Silinmiş bir bölüm tablosunun kurtarılması kısmı eklendi.	
3.1	12 Haziran 2000
Bölümleme Gereksinimlerindeki takas boyu sınırlaması düzeltildi, Giriş kısmındaki pek çok bağlantı güncellendi, fdisk ile Bölümleme Nasıl kısmına gönderilen örnek eklendi, Bölümleme Gereksinimleri kısmına dosya sistemi konusu eklendi.	
3.0	1 Mayıs 2000
Kristian Koehntopp tarafından yazılan "Linux Disk Bölümleme NASIL" (Linux Partition HOWTO) belgesine dayanılarak Tony Harris tarafından yazılmış ilk gözden geçirme..	
2.4	3 Kasım 1997
Kristian Koehntopp tarafından yapılan son gözden geçirme.	

Telif Hakkı © 2000–2004 Anthony Lissot ve Kristian Koehntopp – Özgün belge

Telif Hakkı © 2005 Oğuz Yarımtepe – Türkçe çeviri

**Yasal Açıklamalar**

Bu belgenin çevirisinin, *Linux Disk Bölümleme NASIL* 1.2 sürümünün **telif hakkı © 2005 Oğuz Yarımtepe'ye** aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Genel Kamu Lisansının 2. ya da daha sonraki sürümünün koşullarına bağlı olarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

BU BELGE "ÜCRETSİZ" OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI

OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ “OLDUĞU GİBİ”, AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

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

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

# 1. Giriş

## 1.1. Disk bölümü nedir?

Disk bölümleme, tek bir diski mantıksal pek çok sürücüye bölmek için kullanılan bir araçtır. Bir disk bölümü, bağımsız bir disk gibi ele alınan, bitişik blok kümeleridir. Bölümleme tablosu (bu NASIL belgesinin oluşturulma sebebidir) disk bölümleriyle fiziksel disk alanlarını ilişkilendirilen bir dizindir.

Neden birden çok disk bölümümüz olur?

- Verilerinizi diskin belirli bir bölgesiyle sınırlamış (encapsulate) olursunuz. Dosya sistemi ayrı bir disk bölümü üzerindeyse, bir kaza durumunda sadece bazı verilerinizi kaybedersiniz.
- Disk alanından beklenen verimini artırır. Disk bölümlerini, kullanımınıza göre, değişen blok boyutlarına göre biçimlendirebilirsiniz. Eğer verileriniz küçük boyuttaki (1k'dan daha az) çok sayıda dosyadan oluşuyorsa ve de disk bölümünüz 4k'lık bloklar kullanıyorsa, her dosya için 3k boşa harcıyorsunuzdur. Genelde, her dosya için ortamala olarak bir bloğun yarısını boşa harcarsınız, bundan dolayı eğer pek çok dosyanız varsa blok büyüklüğü ile ortalama dosya büyüklüğünü uyuşturmak çok önemlidir.
- Veri büyümesini önler. Denetimden çıkmış süreçler veya çılgın kullanıcılar, işletim sisteminin kaydını tuttuğu işlemler için artık yer kalmayacak şekilde çok fazla disk alanı harcayabilir. Bu bir felakete dönüşebilir. Disk alanını ayırarak, ayrılan disk alanı tükense dahi buradaki işlerin işletim sistemini etkilemeden öleceğini garantilemiş olursunuz.

## 1.2. Diğer Disk Bölümleme Yazılımları

- **sfdisk**: fdisk'in komut satırı sürümü.
- **cfdisk**: fdisk'in konsoldaki grafik tabanlı (curses-based) sürümü.
- **parted**<sup>(B3)</sup>: GNU disk bölümleme düzenleyicisi.
- **Partition Magic**<sup>(B4)</sup>: Verileri yok etmeden disk bölümlerini dönüştüren, birleştiren, yeniden boyutlandıran ve oluşturan ticari bir uygulama.
- **Disk Drake**<sup>(B5)</sup>: Disk bölümlerini silmek, yeniden boyutlandırmak ve oluşturmak için bir Perl/Gtk programı.

## 1.3. İlgili NASIL'lar

### İlgili NASIL'lar

Başlık	Yazar	Açıklama
<a href="#">İkili önyükleyici yükleme yöntemleri</a> <sup>(B6)</sup> (Dual boot install strategies)	Gjoen Stein <tobbyb (at) usa.net>	Dosya sisteminin farklı bölümleri için farklı boyut büyüklükleri ve hız gereksinimlerine nasıl karar vermelidir?
<a href="#">Linux Çoklu Disk Ayarlama</a> <sup>(B7)</sup> (Linux Multiple Disk System Tuning)	Gjoen Stein <sgjoen (at) mail.nyx.net>	Dosya sisteminin farklı kısımları için değişik hız ve boyut gereksinimleri nasıl değerlendirilmelidir.
<a href="#">Büyük Linux Diski</a> <sup>(B8)</sup> (Linux Large Disk)	Andries Brouwer <aeb (at) cwi.nl>	1024 silindirden daha fazlasına sahip disklerin tanıtımı ve görüşler.
<a href="#">Linux Kota</a> <sup>(B9)</sup> (Linux Quota)	Ralf van Dooren <r.vdooren (at) snow.nl>	Her kullanıcı için disk alanı kullanımını sınırlamayla ilgili ilk öğretiler.
<a href="#">Disk Bölümü Kurtarma küçük-NASIL Belgesi</a> <sup>(B10)</sup>	Jean-Daniel Dodin <jdaniell (at) dodin.net>	Bir Windows yüklenmesinden sonra silinen Linux disk bölümleri, verileri içeren kısım gözüküyorsa, nasıl kurtarılır.

Linux ADSM Yedekleme <sup>(B11)</sup> (Linux ADSM Backup)	Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de>	Linux'u bir IBM ADSM yedekleme ortamına uyarlamak için gerekli açıklamalar.
MSDOS ile Linux Yedekleme <sup>(B12)</sup> (Linux Backup with MSDOS)	Christopher Neufeld <neufeld@physics.utoronto.ca>	MSDOS kaynaklı Linux yedeklemeleriyle ilgili açıklamalar.
Linux NASIL Listesi (Linux HOWTO Index)	Tim Bynum <linux-howto@sunsite.unc.edu>	Bir NASIL belgesi yazma ve göndermenin adımları

## 1.4. Sistemdeki ek kaynaklar

- `/usr/src/linux/Documentation`
  - `/usr/src/linux/Documentation/ide.txt`: IDE sürücülerinizle ilgili bilgi
  - `/usr/src/linux/Documentation/scsi.txt`: SCSI sürücülerinizle ilgili bilgi

## 2. Aygıtlar

Bu bölümdeki açıklamaları anlamak için, Linux'un disk bölümlerinin gösteriminde kullandığı özel sınıflandırmayı bilmeniz gerekir.

Linux'ta, disk bölümleri aygıt dosyaları olarak temsil edilir. Bunlar `/dev` altına yerleştirilmiş sözde dosyalardır. Birkaçı şu şekildedir:

brw-rw----	1	root	disk	3,	0	May	5	1998	hda
brw-rw----	1	root	disk	8,	0	May	5	1998	sda
crw-----	1	root	tty	4,	64	May	5	1998	ttys0

Bir aygıt dosyası c (karakter aygıtı anlamında, tampon bellek kullanmayan aygıtlar) veya b (blok aygıt anlamında, tampon bellek kullanan aygıtlar) türünde bir dosyadır. Linux'ta, tüm diskler blok aygıtları olarak temsil edilir.

### 2.1. Aygıt adları

#### 2.1.1. İsimlendirme Kuralları

Uzlaşım olarak, IDE sürücülere `/dev/hda`'dan başlayıp `/dev/hdd`'ye giden aygıt isimleri verilir. A olarak geçen disk (`/dev/hda`) ilk disk sürücüsü ve C olarak geçerse (`/dev/hdc`) üçüncü disk sürücüsüdür.

#### IDE denetleyici isimlendirme kuralları

sürücü adı	sürücü denetleyici	sürücü numarası
<code>/dev/hda</code>	1	1
<code>/dev/hdb</code>	1	2
<code>/dev/hdc</code>	2	1
<code>/dev/hdd</code>	2	2

Tipik bir kişisel bilgisayarda, herbirine iki sürücü bağlanabilen iki IDE denetleyici vardır. Örneğin, `/dev/hda` ilk IDE denetleyicideki ilk sürücüdür (master) ve `/dev/hdd` ikinci IDE denetleyicideki ikinci sürücüdür (slave – bilgisayardaki dördüncü sürücü).

Bu aygıtlara doğrudan yazabilirsiniz (**cat** veya **dd** kullanarak). Bununla birlikte, bu aygıtlar ilk bloktan başlayarak tüm diski temsil ettiklerinden, yanlışlıkla, sürücünün kullanım dışı kalmasına sebep olabilir, MBR (master boot record) ve bölümleme tablosunun üzerine yazabilirsiniz.

## bölüm adları

sürücü adı	sürücü denetleyici	sürücü numarası	bölüm türü	bölüm numarası
/dev/hda1	1	1	birincil	1
/dev/hda2	1	1	birincil	2
/dev/hda3	1	1	birincil	3
/dev/hda4	1	1	takas	NA
/dev/hdb1	1	2	birincil	1
/dev/hdb2	1	2	birincil	2
/dev/hdb3	1	2	birincil	3
/dev/hdb4	1	2	birincil	4

Bir sürücü bir kere bölümlendirildi mi, bölümler isimlerin sonundaki numaralar olarak temsil edilir. Örneğin, ikinci disk sürücüsündeki ikinci disk bölümü `/dev/hdb2` olacaktır. Birincil bölüm türü, 3. Bölümdeki *Birincil Disk Bölümleri* (sayfa: 9) kısmına kadar açıklanmayacak olsa da yukarıda listelenmiştir.

## SCSI Sürücüler

sürücü adı	sürücü denetleyici	sürücü numarası	bölüm türü	bölüm numarası
/dev/sda1	1	6	birincil	1
/dev/sda2	1	6	birincil	2
/dev/sda3	1	6	birincil	3

SCSI sürücüler de benzer bir modeli takip ederler; 'hd' yerine 'sd' ile gösterilirler. Bundan dolayı da ikinci SCSI diskin ilk bölümü `/dev/sdb1` olacaktır. Yukarıdaki tabloda, sürücü numaraları rastgele 6 seçilerek SCSI ID numaralarının aygıt numaralarıyla bire bir uyuşmadığı gösterilmiştir.

## 2.1.2. İsim Atama

(Sun) Solaris ve (SGI) IRIX altında, SCSI sürücülere verilen aygıt isimlerinin onları nereye taktığınızla ilgisi vardır. Linux altında, sadece hayıflanma ve dış gıcırıtları vardır.

## Önce

SCSI ID #2	SCSI ID #5	SCSI ID #7	SCSI ID #8
/dev/sda	/dev/sdb	/dev/sdc	/dev/sdd

## Sonra

SCSI ID #2	SCSI ID #7	SCSI ID #8
/dev/sda	/dev/sdb	/dev/sdc

SCSI sürücüler 1'den başlayıp 15'e kadar uzanan ve ID olarak kullanılan sayılar kullanır. Düşük SCSI ID numaralılarına alfabetik sıradaki önce gelen harfler atanır. Örneğin eğer 2 ve 5 numaralı iki sürücünüz varsa, 2 numaralı sürücü `/dev/sda` ve 5 numaralı sürücü de `/dev/sdb` olacaktır. Eğer herhangi biri sistemden kaldırılırsa bu sürücülerden yüksek numaralı sürücülerin sistemin yeniden başlatılmasında yeniden isimlendirilecektir.

Eğer Linux makinanızda iki tane SCSI denetleyici varsa, `/bin/dmmsg` çıktısına bakarak her bir sürücüye hangi isim verildiğini görmek isteyebilirsiniz. Eğer bir veya iki denetleyiciyi çıkarırsanız, geri kalanların hepsi sürücü adlarını yeniden isimlendireceklerdir. Grrr...

İki tane alternatif çözüm vardır; her ikisi de bir program kullanarak her bir disk bölümüne bir etiket vermeyi içermektedir (bknz. *Etiketler* (sayfa: 17)). Verilen etiket kalıcıdır ve aygıt fiziksel olarak kaldırılırsa da erişilebilir. Daha sonra disk bölümüne doğrudan veya dolaylı olarak bu etiket ile erişebilirsiniz.

### 2.1.3. Mantıksal Disk Bölümleri

Mantıksal Disk Bölümleri

sürücü adı	sürücü denetleyici	sürücü numarası	bölüm türü	bölüm numarası
/dev/hdb1	1	2	birincil	1
/dev/hdb2	1	2	ek	yok
/dev/hda5	1	2	mantıksal	2
/dev/hdb6	1	2	mantıksal	3

Yukarıdaki tablo isim atamalarında gizemli bir atlamayı göstermektedir. Bu, daha sonra *Birincil ve mantıksal disk bölümleri karışık* (sayfa: 15) bölümünde anlatılacak olan ve daima 5 ile başlayan *Mantıksal Disk Bölümleri* (sayfa: 9) kullanımından kaynaklanır.

Linux disk aygıtlarını idare edebilmeniz için tüm bilmeniz gerekenler bunlardır. Bütünlüğün sağlanması için, aşağıdaki Kristian'ın aygıt numaralarıyla ilgili açıklamalarına bakınız.

## 2.2. Aygıt numaraları

Bir aygıt dosyasıyla ilgili tek önemli nokta, dosya boyu yerine ana ve alt aygıt numaralarının gösterilmesidir.

```
$ ls -l /dev/hda
```

Aygıt dosyalarının öznitelikleri

brw-rw—	1	root	disk	3,	0	Temmuz 18 1994	/dev/hda
izinler		sahibi	grubu	ana aygıt numarası	alt aygıt numarası	tarih	aygıt adı

Bir aygıt dosyasına erişirken, ana aygıt numarası giriş/çıkış işlemlerinde hangi aygıt sürücüsünün çağrılacağını belirtmek için kullanılır. Bu çağrı parametre olarak alt aygıt numarasının alınmasıyla gerçekleşir ve bu alt aygıt numarasının nasıl ifade edildiği tamamen sürücüye bağlıdır. Sürücü belgelendirmesi genelde bu alt aygıt numarasının nasıl kullanıldığını anlatmaktadır. IDE diskler için bu belgelendirme `file:/usr/src/linux/Documentation/ide.txt` altındadır. SCSI diskler için ise `file:/usr/src/linux/Documentation/scsi.txt` olduğu düşünülebilir ama orada değildir. Emin olmak için sürücü kaynak koduna bakılmalıdır (`file:/usr/src/linux/driver/scsi/sd.c:184–196`). Ne şansı ki, `file:/usr/src/linux/Documentation/devices.txt` içerisinde Peter Anvin'in aygıt numara ve isimleri listesi vardır; IDE'ler için 3, 22, 33, 34 ana aygıt numaralarına, SCSI aygıtlar için ise 8 ana aygıt numarasına bakınız. Ana ve alt aygıt numaraların her biri birer bayt yer kaplar, bu da her disk bölümündeki disk bölümü sayısının neden sınırlı olduğunu açıklamaktadır.

## 3. Disk Bölümü Türleri

### 3.1. Disk Bölümü Türleri

Bir disk bölümü belli bir dosya sistemine sahip olması için etiketlenir (disk etiketleri ile karıştırılmamalıdır. bkzn. *Etiketler* (sayfa: 17)). Böylesi bir dosya sistemi standart linux ext2 dosya sistemi veya linux takas alanı ve hatta (Microsoft) NTFS veya (Sun) UFS gibi yabancı dosya sistemleri de olabilir. Her bir disk bölümünün türüyle ilişkilendirilmiş sayısal bir kod vardır. Örneğin, ext2 için kod `0x83` ve linux takas alanı için kod `0x82`'dir. Disk bölümü türlerini ve karşılık gelen kodlarını görmek için `/sbin/sfdisk -T` komutunu çalıştırın.



### 3.2. Yabancı Disk Bölümü Türleri

Disk bölümü türü numaraları rasgele seçilmiştir (ne olmaları gerektiğini anlayamazsınız) ve ele alınan işletim sistemine özel olarak verilmişlerdir. Bu yüzden, aynı diskte iki işletim sistemi kullanıyorsanız, aynı kod farklı iki disk türünü belirlemek için kullanılabilir.

OS/2 ve Windows NT'nin NTFS'i disk bölümlerini `0x07` ile işaretler. MS-DOS, değişik çeşitlilikteki FAT dosya sistemi için değişik kodlar tahsis etmiştir: `0x01`, `0x04` ve `0x06` bilinenleridir. DR-DOS, korumalı FAT disk bölümlerini belirtmek için, Linux/Minix ile o zamanlarda fikir ayrılığı oluşturan `0x81`'i kullanırdı, fakat artık ne Linux/Minix ne de DR-DOS geniş çapta kullanılmaktadır.

### 3.3. Birincil Disk Bölümleri

Intel-tabanlı bir sistemde disk bölümü sayısı, işin başında sınırlandırılmıştır: Asıl disk bölümü tablosu önyükleyici bölümünün bir parçası olarak yüklenir ve de sadece 4 disk bölümü girişi için yer ayrılmıştır. Bu disk bölümleri birincil disk bölümleri olarak isimlendirilir.

### 3.4. Mantıksal Disk Bölümleri

Disk'in birincil bölümlerinden birisi alt bölümlere ayrılabilir. Bunlara mantıksal disk bölümleri denir. Bu durum da bize verimli bir şekilde, dört disk bölümü sınırını aşmamızı sağlar.

Mantıksal disk bölümlerini barındıran birincil disk bölümüne ek disk bölümü denir ve kendi dosya sistemi türü (`0x05`) vardır. Birincil disk bölümlerinin tersine, mantıksal disk bölümleri birbirini takip eder olmalıdır. Her bir mantıksal disk bölümü, mantıksal disk bölümü sayısının sınırsız olacağı sonucunu ifade eden, bir sonraki mantıksal disk bölümüne bir gösterici içerir. Yine de, Linux, disk üzerinde olası disk bölümü sayısında belli bir sınır kabul ettirmiştir, bu da sonuç olarak mantıksal disk bölümü sayısını sınırlamaktadır. Bu sayı SCSI diskleri için 15 disk bölümü iken IDE diskler için 63'tür.

### 3.5. Takas Bölümleri

Bilgisayarınızda çalışan her bir süreç RAM'de bir takım blokları kullanmaktadır. Bu bloklara sayfa adı verilir. Bellek içerisindeki, işlemci tarafından çok kısa süreler içerisinde kullanılacak olan sayfalara çalışma kümesi denir. Linux belleğe yapılan bu erişimleri tahmin etmeye çalışarak (en son kullanılan sayfaların yakın zamanda tekrar kullanılacağını varsayarak) eğer mümkünse bu sayfaların RAM'de tutulmasını sağlar.

Eğer makinanızda çalışan pek çok süreç varsa, çekirdek, sayfaları diske yazarak RAM'in boşaltılmasını sağlayacaktır. İşte takas alanı bu durumda devreye girer. Verimli bir şekilde var olan erişilebilir bellek alanınızı arttırır. Yine de, RAM'den okuyup RAM'e yazmaya kıyasla disk Giriş/Çıkışları çok yavaştır. Bunu ilave bir bellek olarak değil de acil durumlarda kullanılan bir bellek olarak düşünün.

Eğer, çekirdeğin bir başka süreç için çalışan bir sürecin sayfalarını kaldırması anlamına gelen, bellek sıkıntısı varsa, makinanız artık dökülmeye başlamış demektir. Okuyuculardan bazıları istemeyerek bu durumla karşılaşmış olabilir: disk çılgınlar gibi çalışıyor ama bilgisayar kullanılamayacak kadar yavaş. Takas alanı sahip olmanız gereken bir alandır, fakat RAM'in yerine geçecek bir alan değildir. İhtiyacınız olan takas alanının boyuyla ilgili olarak [Takas alanı ne kadar büyük olmalıdır?](#) (sayfa: 11) bölümündeki açıklamalara bakın.

## 4. Bölümlendirme gereksinimleri

### 4.1. Hangi Disk Bölümlerine İhtiyacım Var?

Önyükleyici Sürücüsü: Eğer işletim sisteminizi yeni bölümleyeceğiniz diskten açacaksanız, sahip olmanız gereken disk bölümleri şunlardır:

- Bir birincil disk bölümü
- Bir veya daha fazla sayıda takas alanı
- Sıfır veya daha fazla sayıda birincil veya mantıksal disk bölümü

Başka bir diskten açacaksanız:

- Bir veya daha fazla sayıda birincil veya mantıksal disk bölümü
- Sıfır veya daha fazla sayıda takas alanı

## 4.2. Durum Analizi

### Önyükleyici Disk Bölümü

Önyükleyici disk bölümünüzün mantıksal değil birincil disk bölümü olması gerekir. Bu durum herhangi bir felaket durumunda kurtarmayı sağlamaktadır, fakat teknik olarak gerekli değildir. 0x83 "Linux native" dosya türünde olmalıdır. Eğer 21–3 sürümünden önceki bir **lilo**<sup>(B23)</sup> kullanıyorsanız (örn., 1990lardan kalma), önyükleyici bölümünüz diskin ilk 1024 silindiri içinde yer almalıdır. (Tipik olarak, önyükleyicinin tek ihtiyacı çekirdek görüntüsünü içermesidir.)

Eğer birden fazla önyükleyici disk bölümünüz varsa (diğer işletim sistemlerinden olabilir), hepsini ilk 1024 silindir içerisinde tutun (*Tüm* DOS bölümleri ilk 1024 silindir içerisinde olmalıdır). Eğer modern bir lilo sürümü veya çekirdeğinizi yüklemeye yarayan başka araçlar kullanıyorsanız (örneğin, bir önyükleyici disket veya MS–DOS tabanlı çalışan **LOADLIN.EXE** Linux yükleyicisi), disk bölümü herhangi bir yerde olabilir. Ayrıntılı bilgi için [Büyük-disk \(Large-disk\) NASIL](#)<sup>(B24)</sup> belgesine bakabilirsiniz.

### Takas Bölümü

Dosyalara takas (bkznz. [Takas Dosyaları](#) (sayfa: 21)) yapmadığınız sürece ayrılmış bir takas bölümüne ihtiyacınız olacaktır. Bu bölüm 0x82 "Linux swap" tipinde olmalıdır. Disk üzerinde herhangi bir yerde konumlandırılabilir (siz yine de yerleştirme notlarına bakınız: [Takas alanımı nereye koymalıyım?](#) (sayfa: 12)). Birincil veya mantıksal disk bölümlerinden herhangi biri takas alanı olarak kullanılabilir. Bir disk üzerinde birden fazla takas alanı olabilir. Toplam 8 tanesine (sürücüler içerisinde) izin verilmiştir. Aşağıdaki takas alanı boyu notlarına bakınız ([Takas alanı ne kadar büyük olmalıdır?](#) (sayfa: 11)).

### Mantıksal Disk Bölümleri

Tek bir birincil disk bölümü mantıksal disk bölümleri içerecek şekilde (ek disk bölümü) kullanılmalıdır. Ek disk bölümü disk üzerinde herhangi bir yerde olabilir. Mantıksal disk bölümleri ardışık olmalıdır, fakat ek disk bölümünü doldurmaları gerekmemektedir.

## 4.3. Dosya Sistemleri

### 4.3.1. Hangi dosya sistemleri kendi disk bölümlerine ihtiyaç duyar?

Linux dosya sisteminizdeki herşey aynı (tek bir) disk bölümü altına yerleştirilebilir. Fakat, belli dosya sistemlerinin büyümesini sınırlamak isteyeceğiniz durumlar vardır. Örneğin, eposta kuyruğunuz kök dosya sisteminizle aynı dizindeyse, disk bölümündeki kalan alanı doldurması durumunda bilgisayarınızın çalışması sekteye uğrayacaktır.

`/var`

Bu dosya sistemi ileti ve yazıcılar için kuyruk dizinlerini içermektedir. Buna ek olarak, hata kayıt dosyalarını da içerir. Eğer makinanız bir sunucuysa ve sürekli bir hata üretiyorsa, bu iletiler disk bölümünü doldurabilir. Sunucu bilgisayarların `/var` dosya sisteminin kök / dosya sisteminin disk bölümünde değil farklı bir disk bölümünde bulundurmaları gerekmektedir.

### **/usr**

Bu kısım hemen her çalıştırılabilir ikiliğin bulunduğu yerdir. Ek olarak, çekirdek kaynak ağacı ve belgelerin çoğu burada bulunur.

### **/tmp**

Bazı programlar geçici dosyalarını buraya yazar. Genellikle, oldukça küçüktürler. Yine de, hesaba dayalı, fen ve mühendislik uygulamaları gibi, oldukça yoğun işler çalıştırırsanız, yüzlerle ifade edilen megabaytlara kısa süre içerisinde ihtiyaç duyulabilir. Bu durumda **/tmp**'yi kök **/** dosya sisteminin disk bölümünde değil farklı bir disk bölümünde tutun.

### **/home**

Burası kullanıcı ev dizinlerinin olduğu yerdir. Kullanıcılarınız için alan sınırlaması yapmazsanız, kendi disk bölümünde olmalıdır.

### **/boot**

Burası çekirdek bellekleşmelerinin bulunduğu yerdir. Eski sistemlerdeki yerleri için yukarılardaki açıklamalara bakınız.

## **4.3.2. Disk bölümleme ölçütü olarak dosya ömrü ve yedekleme sıklığı**

Ext2 ile disk bölümleme kararları, yedekleme ihtiyaçlarına bağlı olarak ve dosyaların farklı yaşam sürelerinde disk üzerine [saçılmalarından](#) (sayfa: 23) kaçınacak şekilde verilir.

Dosyaların birer ömrü vardır. Bir dosya oluşturulduktan sonra, sistemde belli bir süre kalacak ve daha sonra silinecektir. Dosya ömrü sistemin her yerinde farklılık gösterir ve kısmen dosyanın yoluna bağlıdır. Örneğin, **/bin**, **/sbin**, **/usr/sbin**, **/usr/bin** ve benzeri dizinler, aylar ve hatta daha uzun sürelerle ifade edilen oldukça uzun yaşam sürelerine sahiptir. **/home** dizini altındakiler ise haftalarla ifade edilen ömürlere sahiptir. **/var** altındaki dosyaların ise oldukça kısa yaşam süreleri vardır: **/var/spool/news** altındaki bir dosya bir kaç günden fazla kalmaz, **/var/spool/lpd** altındakilerinse ömürleri dakikalarla ölçülür.

Yedekleme sözkonusu olduğunda, günlük yapılan yedeklemelerin kullanılan tekil yedekleme aygıtının kapasitesinden daha küçük olması faydalıdır. Günlük yedekleme, sistemin tümünde yapılmış bir yedekleme olabileceği gibi arttırımlı bir yedekleme de olabilir.

Disk bölümü büyüklüklerinin bir yedekleme aygıtına sığacak şekilde yeterince küçük olmasına karar verebilirsiniz (günlük yedeklemeyi seçerseniz). Her halükarda bir disk bölümü günlük yapılan yedeklemelerde (değişikliğe uğramış tüm dosyalar ile) tek bir yedekleme aygıtına sığabilmelidir (arttırımlı yedeklemeyi seçin ve yedekleme aygıtının haftalık/aylık olarak dolacağını umarak değiştirin – gözetimsiz bir işlem olası değildir).

Yedekleme stratejiniz bu karara bağlıdır.

Disk alırken veya planlarken, yedekleme için de bir miktar para ayırmayı unutmayın! Yedeklenmemiş bilginin değeri yoktur! Verilerin yeniden oluşturulması hemen herkes için yedeklenmesinden daha fazla maliyet gerektirmektedir!

Yararlılık açısından farklı ömürlere sahip dosyaların farklı disk bölümlerinde bulundurulması faydalıdır. Bu şekilde "news" disk bölümündeki kısa ömürlü dosyalar çok güç saçılmaya uğrayabilir. Bunun **/** veya **/home** üzerindeki yararlılığı bir etkisi yoktur.

## **4.4. Takas Bölümleri**

### **4.4.1. Takas alanı ne kadar büyük olmalıdır?**

Genel kural RAM miktarınız kadar büyük olması şeklindedir.

Fakat bu genel geçerli bir kural değildir. Programların çalışan kümelerinin oldukça büyük veya oldukça küçük olduğu hayali senaryolar oluşturmak mümkündür (bknz. [Takas Bölümleri](#) (sayfa: 9)). Örneğin, çalışması sırasında rasgele erişim sağladığı geniş bir veri kümesine sahip olan bir simulasyon programı yerel olarak çalışan programının veri bloğundan bu kümeye her an işaretçiler barındırmayabilir. Bu durumda çalışan kümesi oldukça büyük olacaktır.

Diğer taraftan aynı açılmış pekçok JPEG formatlı resmi içeren bir grafik programının geniş bir veri bloğu olacaktır. Fakat aynı anda sadece tek resim üzerinde çalışılabileceğinden resim üzerindeki işlemlerde belleğin program tarafından kullanılan kısmının büyük kısmına erişilmez. Aynı durum pekçok penceresi düzenleme programları için de geçerlidir. Eğer bu programlar iyi tasarlanmışlarsa oldukça büyük yerel referans değerlerine sahiptir. Bu durum da onların büyük takas alanları kullanma zorunluluğunu ortadan kaldırır, sistem üzerinde ölümcül etkiler yapmalarını önler. Aynı nedenden programlarından hiç çıkmadan pekçok program çalıştıran bir kullanıcı daha çok takas alanı isteyecektir.

Sunucular masaüstü akranlarına göre daha fazla takas alanıyla ayarlanırlar Verilmiş olan takas alanı işlemleri için yeterli olsa da sunucu oldukça büyük sayfalamar yapması gereken iş yükleriyle karşılaşabilir. Bazı sistem yöneticileri bu durumun sunucuyu çökeltmesine göz yumarlar. Bu durumlarda takas alanının boyu bellekten kat kat büyük olabilir.

#### **4.4.2. Takas alanım ne kadar büyük olabilir?**

Takas alanının büyüklüğü günümüzde mimariye bağlıdır. i386, m68k, ARM ve PowerPC'ler için bu değer "resmi" olarak 2Gb'dir. Alpha makinalarda 128Gb, sparc işlemcilerde 1Gb ve sparc64 mimarilerinde 3Tb'tır. 2.6 çekirdekle çalışan bir opteron 16 Tb takas alanına yazabilir. Linux çekirdeğinin 2.1 ve önceki sürümlerinde sınır 129 Mb'dir. Takas alanı 128 MB'den daha büyük olabilir ama taşan kısım kullanılmaz. Çekirdeğin 2.1 ve önceki sürümleri için 128 MB'den daha büyük takas alanları kullanmak istiyorsanız birden fazla takas alanı oluşturmalsınız (en çok 9). 2.4 sürümden sonra 32 takas alanı oluşturabilmek mümkün hale gelmiştir. Takas alanı oluşturma detaylarına bakınız.<sup>(1)</sup>

#### **4.4.3. Takas alanımı nereye koymalıyım?**

Kısa cevap herhangi bir yer şeklindedir. Yine de, mümkün olduğunca çok hız elde etme istiyorsanız, iki temel strateji vardır (daha fazla RAM almak yerine).

- Takas alanını farklı sürücülere parçalayın ya da az sıklıkla yazdığınız sürücü üzerinde bu parçalamayı yapın.
- Her bir takas disk bölümünü dıştaki izlere yerleştirin.

Gözönüne alınması gerekenler:

- Eğer biri çok kafalı diğeri daha az disk kafalı olan iki diskiniz varsa ve diğer özellikler bakımından aynıysalar, çok disk kafası olan daha hızlı olacaktır. Farklı disk kafaları üzerinden veri okunması tamamen elektronik olduğundan daha hızlı olacaktır. Farklı izlerden veri okumak, kafanın hareketini gerektirdiğinden yavaştır.

Takasın başka bir diske yazılması, disk kafasının tek bir sürücü üzerinde ileri geri hareketinden daha hızlı olacağı sonucu çıkmaktadır.

- **Yerleşim:** Eski diskler tüm izlerde aynı sayıda sektör içermektedir. Bu disklerle, takas alanını ortaya yerleştirmek, disk kafası rasgele hareket sırasında takas alanından geçeceğinden, en hızlı olacaktır.
- Daha yeni diskler ZBR (zone bit recording: bölge biti kaydı) kullanmaktadır. Dış izlerinde daha çok sektörleri barındırırlar. Sabit bir dönme hızında, içteki izlere göre bu daha büyük bir performans gerektirir. Takas alanınızı hızlı olan izler üzerine koyunuz. (Genel olarak düşük silindir numaraları düşük disk bölüm numaralarıyla eşleşir. Gene de bu konuyla ilgili güncel yorumları <http://lissot.net/partition/mapping.html> adresinden inceleyiniz)

- *Kullanım:* Elbette, sizin disk kafanız rasgele hareket etmeyecektir. Eğer sürekli meşgul olan bir ev dizini disk bölümüyle, neredeyse hiç kullanılmayan bir arşiv disk bölümü arasında bir takas alanınız varsa, takas alanınızın ev dizini altında olması, disk kafasının kısa hareketler yapacak olmasından dolayı daha iyi bir yaklaşım olacaktır. Aslında, takas alanınız başka amaçla kullanılmayan bir diskte olursa, daha iyi olurdu.
- *Bölme:* Hız birden fazla takas alanını aynı anda kullanmakla artırılabilir. Aynı önceliğe sahip takas alanları RAID cihazlar gibi ele alınıp kullanılır. (bkz. [Çoklu Takas Alanları](#) (sayfa: 21))

**Özet:** Takas alanınızı pek çok disk kafası olan ve başka işler yapmakla meşgul olmayan hızlı bir diske yerleştirin. Eğer birden çok diskiniz varsa: takas alanınız bölerek bunu tüm disklere hatta farklı denetleyicilere dağıtın.

## 5. fdisk ile diski bölümleme

### 5.1. fdisk ile disk bölümü oluşturmak

Bu kısım aslında diskinizi **fdisk** uygulaması ile nasıl bölümlendireceğinizi anlatmaktadır. Linux sadece 4 tane birincil disk bölümüne izin vermektedir. Tek bir disk bölümünü alt bölümlere bölerek çok sayıda mantıksal bölüme sahip olabilirsiniz. Birincil disk bölümlerinden sadece bir tanesi alt bölümlere bölünebilir.

*Örnekler:*

1. Dört tane birincil disk bölümü: [Dört disk bölümü](#) (sayfa: 14)
2. Birincil ve mantıksal disk bölümleri karışık: [Birincil ve mantıksal disk bölümleri karışık](#) (sayfa: 15)

#### 5.1.1. fdisk kullanımı

**fdisk**, komut satırında (root olarak) **fdisk** *aygıt-adi* yazarak çalıştırılır. ([İsimlendirme Kuralları](#) (sayfa: 6)). Aygıt adı `/dev/hda` veya `/dev/sda` gibi bir şey olabilir. İhtiyacınız olacak temel fdisk komutları şunlardır:

<b>p</b>	disk bölümleme tablosunu göster
<b>n</b>	yeni bir disk bölümü oluştur
<b>d</b>	disk bölümü sil
<b>q</b>	değişiklikleri kaydetmeden çık
<b>w</b>	mevcut disk bölüm tablosunu yaz ve çık

Disk bölümü tablosunda yaptıklarınız **w** komutu ile yazana kadar etkisini göstermez. Aşağıda basit bir disk bölümleme tablosu bulunmaktadır:

```
Disk /dev/hdb: 64 kafa, 63 sektör/iz, 621 silindir
Birimler = silindir / 4032 * 512 bayt
```

Aygıt	Açılış	Başlangıç	Bitiş	BlokSayısı	Kml	Sistem
/dev/hdb1	*	1	184	370912+	83	Linux
/dev/hdb2		185	368	370944	83	Linux
/dev/hdb3		369	552	370944	83	Linux
/dev/hdb4		553	621	139104	82	Linux takas / Solaris

İlk satır diskin geometrisini göstermektedir. Fiziksel olarak tam doğru olmayabilir ama siz öyle olduğunu varsayabilirsiniz. Bu örnekteki disk her biri üzerinde bir disk kafası olan çift yönlü 32 tane plakadan oluşmuştur (muhtemelen doğru değil). Her bir plakanın eşmerkezli 621 izi bulunmaktadır. Üç boyutlu bir ize (tüm disk üzerindeki aynı izlere) silindir denir. Her bir iz 63 sektöre bölünmüştür. Her bir sektör 512 bayt veri içermektedir. Bundan dolayı disk bölümü tablosundaki blok sayısı 64 disk kafası \* 63 sektör \* 512 bayt / 1024'tür. (Bu problemin çözümü (sayfa: 20) bölümünde açıklanmıştır.) Başlangıç ve bitiş değerleri silindirlerdir.

### 5.1.2. Dört disk bölümü

*Gözden geçirme:* Takas alanınızın *boyuna* (sayfa: 11) ve nereye *yerleşmesi* (sayfa: 12) gerektiğine karar verin. Kalan alanı diğer üç disk bölümü için bölün.

Örnek:

**fdisk**'i komut satırından başlattım:

```
# fdisk /dev/hdb
```

Bu, IDE denetleyicinin üzerindeki ikinci sürücüyü kullandığım anlamına gelmektedir. (*Aygıt adları* (sayfa: 6) kısmına bakın.) (Boş) disk bölümleme tablosunu görüntülediğim zaman, sadece ayarlarla ilgili bilgi alırım.

```
Command (m for help): p
Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes
```

1.2GB disk kapasitem olduğunu biliyordum ama artık eminim:  $64 * 63 * 512 * 621 = 1281982464$  bayt. Bu alanın 128MB'ını takas için ayırmaya karar verdim, geriye 1153982464 kaldı. Eğer takas için birincil disk bölümlerinden birini kullanırsam ext2 disk bölümleri için geriye 3 tane kalmış olur. Eşit olarak bölün, bu da 384MB olacak şekilde her birini oluşturmak demektir. Artık şimdi çalışmaya başlayabilir.

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-621, default 1): <RETURN>
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-621, default 621): +384M
```

Daha sonra takas alanı için kullanacağım alanı ayarlıyorum:

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (197-621, default 197): <RETURN>
Using default value 197
Last cylinder or +size or +sizeM or +sizeK (197-621, default 621): +128M
```

Artık disk bölümleme tablosu şu şekilde gözüküyor:

Device	Boot	Start	End	Blocks	Id	System
/dev/hdb1		1	196	395104	83	Linux

/dev/hdb2	197	262	133056	83	Linux
-----------	-----	-----	--------	----	-------

Diğer kalan iki disk bölümünü de ilkin yaptığım gibi ayarlıyorum. Son olarak da ilk disk bölümünü önyüklenabilir yapıyorum:

```
Command (m for help): a
Partition number (1-4): 1
```

Ve ayrıca ikinci disk bölümünün türünü de takas alanı olarak ayarlıyorum:

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
Command (m for help): p
```

Sonuç şu şekilde gözüküyor:

```
Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1    *           1          196     395104+   83   Linux
/dev/hdb2             197          262     133056    82   Linux swap
/dev/hdb3             263          458     395136    83   Linux
/dev/hdb4             459          621     328608    83   Linux
```

Son olarak da **w** ile tabloyu diske yazdırıyorum.

İlgili konular:

- [Takas Alanının Etkinleştirilmesi](#) (sayfa: 22)
- [Disk Bölümlerinin Biçimlendirilmesi](#) (sayfa: 22)
- [Disk Bölümlerinin Bağlanması](#) (sayfa: 22)

### 5.1.3. Birincil ve mantıksal disk bölümleri karışık

*Gözden geçirme:* Tüm diğer alt disk bölümlerini içermesi için bir tane birincil disk bölümü oluşturun. Daha sonra onun içerisinde mantıksal disk bölümleri oluşturun. Diğer birincil disk bölümlerini mantıksal disk bölümleri oluşturmadan önce veya sonra oluşturabilirsiniz.

Örnek:

**fdisk**'i komut satırından çalıştırıyorum:

```
# fdisk /dev/sda
```

Bu, SCSI zincirimin ilk sürücüsü anlamına gelmektedir. ([Aygıt adları](#) (sayfa: 6) kısmına bakın.)

Öncelikle kaç tane disk bölümü istediğime karar veriyorum. Diskimin 183GB kapasitesi olduğunu biliyorum ve 26GB'lık disk bölümleri istiyorum (çünkü yaklaşık bu boyutta yedekleme teyplerim var)

$$183\text{GB} / 26\text{GB} = \sim 7$$

Dolayısıyla 7 disk bölümüne ihtiyacım var. **fdisk** disk bölümü boylarını MB ve kB olarak kabul etmesine rağmen, her bir disk bölümü sonundaki silindir numaralarını hesaplamaya karar veriyorum çünkü fdisk silindirlerin başlangıç ve bitiş silindir numaralarının bilgisini vermektedir. fdisk'e başladığım zaman 22800 silindir olduğunu görüyorum.



```
> The number of cylinders for this disk is set to 22800.  There is
> nothing wrong with that, but this is larger than 1024, and could in
> certain setups cause problems with:  1) software that runs at boot
> time (e.g., LILO)  2) booting and partitioning software from other
> OSs   (e.g., DOS FDISK, OS/2 FDISK)
```

Bu durumda, toplam 22800 silindirin 7 disk bölümünün her birine 3258 silindir düşer. Yani, her bir disk bölümü yaklaşık olarak 3258 silindir boyunda olacaktır. Önyükleme uyarısını dikkate almıyorum çünkü burası benim önyükleme amacıyla kullanacağım sürücüm (*Bölümlendirme gereksinimleri* (sayfa: 9)) değil.

4 tane birincil disk bölümüm olduğundan, bunlardan 3 tanesi 3258 uzunluğunda olabilir. Ek bölüm (4 \* 3258) yani 13032 silindir uzunluğunda olmalı, böylece 4 mantıksal disk bölümü buraya sığar.

İlk 3 birincil disk bölümünü oluşturmak için aşağıdaki komutları giriyorum (koyu ile yazdıklarım):

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
P
Partition number (1-4): 1
First cylinder (1-22800, default 1): <RETURN>
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-22800, default 22800): 3258
```

Son bölüm, ek disk bölümü:

```
Partition number (1-4): 4
First cylinder (9775-22800, default 9775): <RETURN>
Using default value 9775
Last cylinder or +size or +sizeM or +sizeK (9775-22800, default 22800): <RETURN>
Using default value 22800
```

Tabloyu göster komutunun çıktısı şöyle:

/dev/sda1	1	3258	26169853+	83	Linux
/dev/sda2	3259	6516	26169885	83	Linux
/dev/sda3	6517	9774	26169885	83	Linux
/dev/sda4	9775	22800	104631345	5	Extended

Sonra, ilk mantıksal disk bölümünden başlayarak, ek bölümü her biri 3258–silindirlik 4 alt bölüme ayırıyorum. Mantıksal disk bölümleri böylece /dev/sda5'ten başlıyor.

```
Command (m for help): n
First cylinder (9775-22800, default 9775): <RETURN>
Using default value 9775
Last cylinder or +size or +sizeM or +sizeK (9775-22800, default 22800): 13032
```

Sonuç şu şekilde:

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	3258	26169853+	83	Linux
/dev/sda2		3259	6516	26169885	83	Linux
/dev/sda3		6517	9774	26169885	83	Linux
/dev/sda4		9775	22800	104631345	5	Extended
/dev/sda5		9775	13032	26169853+	83	Linux
/dev/sda6		13033	16290	26169853+	83	Linux
/dev/sda7		16291	19584	26459023+	83	Linux
/dev/sda8		19585	22800	25832488+	83	Linux



Son olarak, **w** komutu ile tabloyu diske yazdırıyorum. Disk bölümlerinin kullanılabilir olması için onları [biçimlendircek](#) (sayfa: 22) ve kök dizine [bağlayacağım](#) (sayfa: 22).

#### 5.1.4. Sunulan Örnekler

Kendi disk bölümü yerleşimimi göstermek isterim, çünkü herhangi bir Linux dağıtımıyla iyi çalışabilmektedir (hatta büyük RPM tabanlı olanlarda bile). 10 GB olan tek bir sabit diskim var. Windows bunun 9.4 GB'tan fazlasını göremiyor, fakat Linux görüp kullanabiliyor. Aynı zamanda diskim 1024 silindirden fazlasını içeriyor.

**Bölüm yerleşim örneği**

Disk Bölümü	Bağlama noktası	Büyüklik
/dev/hda1	/boot	(15 MB)
/dev/hda2	windows 98 bölümü	(2 GB)
/dev/hda3	ek bölüm	(kullanışsız)
/dev/hda5	takas alanı	(64 MB)
/dev/hda6	/tmp	(50 MB)
/dev/hda7	/	(150 MB)
/dev/hda8	/usr	(1.5 GB)
/dev/hda9	/home	diskin geri kalanı

Yeni çekirdekleri USB diskleri bakımından deniyorum, bu da büyük **/boot** disk bölümlerini açıklıyor. LILO'yu MBR'ye kurdum ve öntanımlı olarak windows'un başlamasına ayarladım (bu bilgisayarı kullanan tek ben değilim).

Ayrıca, örneklerde GERÇEK disk bölümleme tablolarının kullanılmadığını da belirteyim, yeni başlayanlar için ÖZELLİKLE bir iki tane konulmasını öneririm<sup>(2)</sup>. Artık yeni başlayan statüsünde değilim ve disk bölümleme beni en zorlayan konulardan biriydi.

Valkor <[valkor \(at\) qx.net](mailto:valkor@qx.net)>

## 6. Etiketler

Linux kullanımında disk sürücülerini aygıtlar olarak düşünülür, bu aygıtlar da **/dev** altındaki sanal dosyalardır. Örneğin, en düşük numaralı SCSI sürücüsünün ilk disk bölümü **/dev/sdb1** olarak gösterilir. **/dev/sda** olarak işaret edilen sürücü sistemden kaldırılırsa **/dev/sda1** disk bölümünden sonraki bölüm bir sonraki sistem açılışında otomatik olarak yeniden isimlendirilecektir.

### 6.1. Aygıt Etiketleri

Aygıt etiketleri nereye ve başka nelerin bağlı olduğuna bakmaksızın disk bölümleri için isimlendirilme yapılmasını sağlar. Bir linux aygıtı için etiketler zorunlu değildir. Her biri en fazla 16 karakter uzunluğunda olabilir.

Aygıt etiketleri oluşturmak için kullanılan üç araç vardır: mke2fs, tune2fs ve e2label.

#### 6.1.1. Basit Çalıştırma

```
e2label /dev/hdb1 pubsw
```

```
tune2fs -L pubsw /dev/hdb1
```

Bu komutların her ikisi de ikinci sürücünün ilk disk bölümünü "pubsw" olarak isimlendirir. Bu isim sürücü başka bir kontrol birimine takılsa veya başka bir bilgisayar taşınsa bile değişmeden kalır.

```
mke2fs pubsw /dev/hdb1
```

```
mke2fs -L pubsw /dev/hdb1
```

Bu iki komut bu işi dosya sistemi oluşturduktan sonra yapması dışında yukarıdaki iki komutla aynı işi yapar. Bu aynı zamanda var olan disk bölümlerinin de sileceğini belirtir.

### 6.1.2. Kullanım

Aşağıda bir `fstab` içeriği görüntülenmiştir. Bu dosya, sistem kurulumu sırasında `/etc` altında oluşturulur. Hangi disk bölümlerinin nasıl bağlanacağı bilgisini içerir. Yeni aygıtlar eklenip çıkarıldığında çeşitli uygulamalar tarafından veya sizin tarafınızdan değiştirilebilir.

LABEL=/	/	ext3	defaults	1 1
LABEL=/boot	/boot	ext2	defaults	1 2
none	/dev/pts	devpts	gid=5,mode=620	0 0
none	/dev/shm	tmpfs	defaults	0 0
LABEL=HOME	/home	ext3	defaults	1 2
none	/proc	proc	defaults	0 0
none	/sys	sysfs	defaults	0 0
LABEL=/usr	/usr	ext3	defaults	1 2
/dev/hdc1	/k-space	ext3	defaults	1 2
/dev/hda6	swap	swap	defaults	0 0
/dev/hdd	/media/cdrecorder	auto	pamconsole,ro,exec,noauto,managed	0 0
/dev/fd0	/media/floppy	auto	pamconsole,exec,noauto,managed	0 0

En sol sütün aygıtları hemem onun karşısındaki sütunsa bağlanacakları noktaları gösterir. Bu örnek etiket ve bağlama noktalarını karma içeren bir örnektir. İkinci disk kontrolündeki ana sürücü daima `/k-space` dizinine bağlanır. "HOME" etiketli disk sürücüsü ise hangi sürücü olduğuna ya da disk bölümü numarasına bakmaksızın `/home` dizinine bağlanır. Bağlama noktaları da, `/usr` gibi, etiket olarak kullanılabilir.

## 6.2. Aygıt Etiketleri

**devlabel** aygıtlara sembolik bağlar oluşturmak için kullanılan bir betiktir. Örneğin,

```
devlabel -d /dev/hdb1 -s /dev/home
```

`/dev/hdb1`'den `/home` dizinine bir bağ oluşturur. `/dev/hdb1` üzerindeki tekil bir belirteci de belirtilen bağ ile birlikte `/etc/sysconfig/devlabel` dosyasında da saklar. Eğer donanım `/dev/hdc1` olarak taşınırsa ona ait olan tekil belirteç (`/usr/bin/partition_uuid` kullanılarak) `/etc/sysconfig/devlabel` dosyasındaki karşılığına bakılarak `/dev/home`'u işaret edecek şekilde ayarlanır.

## 7. Bir ext2/3 disk bölümünü biçimlendirmek

Bir sabit disk bölümlendirildiğinde, birtakım alanlara haritası çıkarılır ve bu alanların içeriği boştur. Tıpkı rafları yeni oluşturulmuş bir kütüphaneye benzer. Tek ihtiyacı olan kitap kataloğunun ve işaretlerin oluşturulmasıdır.

Disk bölümü içerisindeki organize bilgisine dosya sistemi denir. Linux iler standart dosya sistemi ext2 ve ext3'tür. ext3 dosya sistemi ext2 özelliklerine ilave olarak günlük olarak isimlendirilebilen disk kayıtlarını da içermektedir. Bu günlük ani güç kesilmelerinde ve başka durumlarda diskin kurtarılmasını sağlar.

ext2/3 dosya sistemi oluşturmak için kullanılan temel araç **mke2fs**'tir. Genelde `/sbin` altında bulunur. **mkfs.ext2** ve **mkfs.ext3** **mke2fs** üzerinde çalışan ve ona çeşitli seçenekler gönderen komutlardır.

### 7.1. Basit Çalıştırma

```
mke2fs /dev/hdb1
```

```
mkfs.ext2 /dev/hdb1
```

her ikisi de ikinci sürücünün ilk disk bölümünde ext2 dosya sistemi oluşturur. Ve,

```
mke2fs -j /dev/hdb1
```

```
mkfs.ext3 /dev/hdb1
```

bir ext3 dosya sistemi oluşturur.

## 7.2. Atanmış bloklar

`-m` seçeneği uzman olmayanların muhtemelen sıklıkla kullandığı bir özelliktir. Eğer dosya sistemi doldurulmuş ve yazılacak alan kalmadıysa temel olarak işletim sistemin sürekli yazmaya çalışmasından kullanım dışıdır. Varsayılan olarak disk bölümün yüzde beşi root kullanıcısının kullanımına ayrılır. Bu da root kullanıcısının diski üzerinde yönetimsel işlemler yapmasını veya gerekliyse bir takım verileri kaldırmasını sağlar. Yine de, bu durum disk bölümünün / veya home dizinlerini içermesi durumunda en kritik halini alır. Saf veri disk bölümleri için bu durum bir alan israfıdır. 250GB'lık alanın yüzde beşi 12.5 GB'tır. Büyük disk bölümlerinin olması durumunda ayrılan alanı en az tutmak gerekir, ki bu da yüzde birdir.

```
mkfs.ext3 -m 1 /dev/hdb1
```

root kullanıcısının kullanımına %1'lik bir ayrılmış alan oluşturarak dosya sistemini oluşturur. `tune2fs -m`, ayrılan blokların, diske verilerin yüklenmesinden sonra, ayarlanmasını sağlar.

## 8. Silinmiş bir Disk Bölümünü Kurtarmak

Aşağıda silinmiş bir disk tablosunu el ile kurtarmanın adımları anlatılmıştır. [gpart<sup>\(B47\)</sup>](#) veya [TestDisk<sup>\(B48\)</sup>](#) gibi bu işlemi oldukça kolay hale getiren uygulamalar vardır. Eğer bunu okuyor ve şansınız hiç kalmadıysa, yapacaklarınız:

1. İlk disk bölümünüzün büyüklüğü kadar bir disk bölümü oluşturun. Bu kısmı asıl disk bölümünden daha büyük de yapabilirsiniz. Eğer daha küçük yaparsanız, çok hayıflanıp dışarıdatacaksınız.

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-23361, default 1): <RETURN>
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-22800, default 22800): 13032
Command (m for help): w
```

2. `dumpe2fs`'i ilk disk bölümünde çalıştırarak blok sayısını bulun.

Örnek:

```
% dumpe2fs /dev/sda1 | grep "Block count:"
Block count:                41270953
```

Eğer bu değerden emin değilseniz, 1. Adımı daha büyük bir disk boyutu için tekrarlayın. Eğer blok sayısı değiştiyse, asıl disk bölümü blok sayısını iyi tahmin edememişsiniz demektir. 1. Adımı kararlı bir blok sayı bulana kadar sürdürün.

3. Yeni oluşturduğunuz disk bölümünü silin.

```
Command (m for help): d
Partition number (1-4): 1
```

4. Elde ettiğiniz blok sayısına bakarak yeni bir disk bölümü oluşturun. **fdisk**'e blok sayısı giremediğinizden, kaç tane silindir istendiğini hesaplamamız gerekmektedir. İşte formülü:

$$(\text{gerekli silindir sayısı}) = (\text{blok sayısı}) / (\text{blok büyüklüğü})$$

$$(\text{blok büyüklüğü}) = (\text{birim büyüklüğü}) / 1024$$

$$(\text{birim büyüklüğü}) = (\text{kafa sayısı}) * (\text{sektör/silindir sayısı}) * (\text{bayt/sektör sayısı})$$

1, 2, 4 ve 8 silindirlik 4 birincil disk bölümüne ayrılmış aşağıdaki disk bölümleme örneğine bakınız.

```
disk /dev/sda: 16 heads, 63 sectors, 23361 cylinders
Units = cylinders of 1008 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	2	976+	83	Linux
/dev/sda2		3	5	1512	83	Linux
/dev/sda3		6	10	2520	83	Linux
/dev/sda4		11	19	4536	83	Linux

**fdisk** çıktı başlangıcında ihtiyacım olan ayar bilgisini de sunmaktadır.

Birim büyüklüğü **516096**  
 = **16** disk kafası \* **63** sektör/silindir \* **512** bayt/sektör

Blok büyüklüğü **504**  
 = **516096** / **1024**

Dolasiyle ikinci disk bölümü için gerekli silindir sayısı **3**'tür (**1512** blok / **504**).

Disk bölümleme tablosu da bunun gerçekten de böyle olduğunu söylüyor: ilk silindir **3**, ikincisi **4** ve en sonuncu da **5**, toplam 3 silindir.

Üçüncü disk bölümü için gerekli silindir sayısı da benzer şekilde hesaplanır: **2520** blok / **504** = **5**, bunlar da bloklara karşı gelecek şekilde **6, 7, 8, 9, 10**'dur. Bu hesaplamanın ilk disk bölümünde çalışmayacağına dikkat edin (**1008** yerine **976**). Artı işareti tüm blokların fdisk uygulamasına katıldıklarını gösterir. Hesaplamayı denediğiniz zaman (**976** / **504**) **1.937** elde edersiniz. Silindir sayısının tam-sayı olması gerektiğini bilerek bunu yuvarlayabilirsiniz.

5. **e2fsck**'i çalıştırarak yeni disk bölümünün okunup okunamayacağına bakın.

6. 1'den 5'e kadar adımları kalan diğer disk bölümleri için de tekrarlayın.

Disk bölümünüzü tekrar bağlayın. Şaşırtıcı bir şekilde bilgilerinizi orada olduğunu göreceksiniz.

- jedi sistem yöneticisi ve MGH'nin en iyisi olan Mike Vevea'a, bu ipuçlarını verdiği için teşekkürler.

## 9. Takas Alanını Ayarlamak

### 9.1. Takas Dosyaları

Bir takas alanı oluşturmanın, ayarlamanın ve bunu /etc/fstab dosyasına eklemenin iki adımı vardır. /dev/hda6 olarak ayarlanmış bir takas alanının fstab dosyasındaki görüntüsü şu şekildedir:

```
/dev/hda6      swap      swap      defaults      0      0
```

Sistemi yeniden başlattığınızda yazılan bu takas alanı otomatik olarak etkin olacaktır.

Fakat, takas alanını hemen kullanmak isterseniz bunu elle yapmalısınız. root olarak şunları yazın:

```
mkswap -f /dev/hda6
swapon /dev/hda6
```

## 9.2. Takas Dosyaları

Takas alanının yetersiz kaldığı durumlar olabilir, böylesi durumlarda yeniden disk bölümlendirmek ya da yeni bir tane eklemek pek pratik değildir. Tek yapmanız gereken istediğiniz büyüklükte bir dosya oluşturmak

```
dd if=/dev/zero of=/var/my_swap bs=1024 count=131072
```

ve etkin kılmaktır

```
mkswap -f /var/my_swap
swapon /var/my_swap
```

Bu şekilde /var altında my\_swap isimli bir dosya oluşturulur. Boyutu başlangıçta 128 Mb'tir (128 x 1024=131072). İklendirme sırasında sıfırlar ile doldurulur. mkswap bu dosyayı takas alanı kullanılması için işaretler ve swapon da çekirdeğin bunu takas alanı olarak kulllanmaya başlamasını söyler. İşiniz bittikten sonra,

```
swapoff /var/my_swap
rm /var/my_swap
```

komutlarını çalıştırır.

## 9.3. Çoklu Takas Alanları

Tek bir sistem üzerinde birden fazla takas alanı kullanılabilir. Tek bir takas alanının olduğu bir fstab örneğine bakalım:

```
/dev/hda5      /          ext3      defaults      1      1
/dev/hda1      /boot      ext2      defaults      1      2
none          /dev/pts   devpts    gid=5,mode=620 0      0
none          /proc      proc      defaults      0      0
/dev/hda7      /usr       ext3      defaults      1      2
/dev/hda6      swap       swap      defaults      0      0
```

Takas alanı girdileri şu şekilde değiştirdiğimiz düşünelim:

```
/dev/hda6      none       swap      sw,pri=3      0      0
/dev/hdb2      none       swap      sw,pri=2      0      0
/dev/hdc2      none       swap      sw,pri=1      0      0
```

Bu ayarlama ile en yüksek önceliğe sahip olan (pri=3) /dev/hda6 çekirdek tarafından ilk olarak kullanılacaktır. En yüksek öncelik değeri 32767 en düşüğü ise 0'dır. Eğer ilk kullanılan yetersiz kalırsa sırasıyla önce /dev/hdb2 ve sonra da /dev/hdc2 kullanılacaktır. En yeni (en hızlı) sürücülere en büyük öncelik değerinin verildiğini düşünün. Bu durum takas alanının aşırı kullanımında hız kaybını en aza indirir.

Her üç bölüme de aynı anda yazmak mümkündür. Çekirdek bir RAID aygıtı yazar gibi her disk bölümüne aynı anda yazacaktır.

```
/dev/hda6 none swap sw,pri=3 0 0
/dev/hdb2 none swap sw,pri=3 0 0
/dev/hdc2 none swap sw,pri=3 0 0
```

Bu disk bölümlerinin farklı sürücülerde olduğunu ve bunun da hız artışı için yapıldığını düşünebilirsiniz.

## 10. Son Söz

### 10.1. Disk Bölümlerinin Biçimlendirilmesi

Komut satırında disk bölümlerim üzerindeki dosya sistemlerini oluşturmaya başlıyorum. *Birincil ve mantıksal disk bölümleri karışık* (sayfa: 15) örneğini takiben:

```
# mke2fs /dev/sda1
```

Bunu /dev/sda4 (ek disk bölümü) dışındakilerin hepsi için yapmalıyım.

Linux ext2 dışındaki dosya sistemlerini de destekler. /usr/src/linux/include/linux/fs.h dosyasının içeriğine bakarak çekirdeğinizin hangi dosya sistemi türlerini desteklediğini öğrenebilirsiniz.

Yaygın dosya sistemleri /sbin içerisindeki mk ile başlayan mkfs.msdos ve mke2fs gibi programlarla yapılabilir.

### 10.2. Takas Alanının Etkinleştirilmesi

Bir takas alanını yapılandırmak için:

```
# mkswap -f /dev/hda5
```

Takas alanını etkinleştirmek için:

```
# swapon /dev/hda5
```

Normalde, takas alanı önyükleme sırasında ilklendirme betikleri tarafından etkinleştirilir.

### 10.3. Disk Bölümlerinin Bağlanması

Disk bölümlerini bağlamak onları linux dosya sistemi ile ilişkilendirmek demektir. Bir linux disk bölümünü bağlamak için:

```
# mount -t ext2 /dev/sda1 /opt
```

-t ext2

Dosya sistemi türü. Kullanmak isteyebileceğiniz diğer dosya sistemi türleri:

- ext3 (ext2 tabanlı günlüklemeli (journaling) dosya sistemi)
- msdos (DOS)
- hfs (mac)
- iso9660 (CDROM)
- nfs (ağ dosya sistemi)

/dev/sda1

Aygıt adı. Kullanmak isteyebileceğiniz diğer aygıt dosyası adları:

- /dev/hdb2 (ikinci IDE sürücüsündeki ikinci disk bölümü)

- `/dev/fd0` (floppy disk A)
- `/dev/cdrom` (CDROM)

`/opt`

Bağlama noktası. Burası **ls /opt** yazdığınız zaman "görmek" istediğiniz yerdir, böylece `/dev/sda1` içerisinde ne varmış diye görebilirsiniz. Eğer `/opt` altında halihazırda dizinler ve/veya dosyalar varsa bağlama işleminden sonra bunlar görünmez olacaktırlar.

## 10.4. Saçılım ve Dosya sistemleri hakkında bazı gerçekler

Disk alanı işletim sistemi tarafından bloklar ve blok kayıtları olarak yönetilir. ext2'de, bloklar ve blok kayıtları aynı büyüklükte olmalıdır, böylece açıklamamızı bloklarla sınırlandırabiliriz.

Dosyalar herhangi bir büyüklükte olabilir. Blok sınırlarıyla sonlanmazlar. Dolayısıyla her bir dosya için bloğun bir kısmı boşa harcanır. Dosya boylarının rasgele olduğunu varsayarsak, her bir dosya için diskinizde yaklaşık yarım blok boşa harcanmaktadır. Tanenbaum "İşletim Sistemleri" ("Operating Systems") isimli kitabında buna "dahili saçılım" ("internal fragmentation") demektedir.

Kaç dosyanız olduğunu dosya düğümü (inode) sayısına bakarak tahmin edebilirsiniz. Benim diskimde

#	df -i				
Dosyasistemi	Dosyaaindexi	Dolu	Boş	Kull%	Bağlanılan yer
/dev/hda3	64256	12234	52022	19%	/
/dev/hda5	96000	43058	52942	45%	/var

/ üzerinde yaklaşık 12000, /var üzerinde ise yaklaşık 44000 dosya var. 1 kB'lık blok büyüklüğü için, yaklaşık  $6+22=28$  MB'lık disk alanı dosyaların blok sonlarındaki kısımlarında kaybedilir. 4 KB'lık blok büyüklüğü seçmiş olsaydım, bu büyüklüğün 4 katı kadarını kaybetmiş olacaktım.

Veri aktarımı tek ve kesintisiz bir veri tomarında daha hızlı gerçekleşir. Bu da, dosyalar büyüdükçe ext2'nin ardışık 8 blokluk birimler tahsis etmesinin sebebidir. Dosya kapatıldığında ayrıldığı halde kullanılmayan bloklar serbest bırakılır ve böylece boşa alan harcanmaz.

Dosyaların ardışık olmayan bloklardan oluşması performans açısından kötüdür, çünkü genelde dosyalara erişimde sıralı erişimle kullanılır. Bu durum da işletim sisteminin diske erişimini bölmeye zorlamakta ve de disk kafalarını hareketine sebep olmaktadır. Buna da harici saçılım ("external fragmentation") veya basitçe saçılım ("fragmentation") denir ki MS-DOS dosya yapılarındaki temel sorundur. MS-DOS'un kullandığı berbat önbelleğin birleşimi ile dosya saçılmalarının etkileri dikkat çekici boyutlardadır. DOS kullanıcıları disklerini birleştirmeyi bir kaç haftada bir yaptıkları mutad işler arasına almışlardır hatta bazıları disk birleştirmeye ilgili törenselle fikirler geliştirmiştir.

Bu alışkanlıklarının hiç birisinin Linux ve ext2 üzerinde işi yoktur. Linux dosya yapısı normal kullanımda ve azından %5'lik bir disk alanının boş olması durumunda hiçbir koşulda disk birleştirmeye gerek duymaz. ext2'de disk birleştirmek için defrag isimli bir araç vardır, fakat kullanıcılar dikkatsiz kullanımı konusunda uyarılmaktadır. Böylesi bir işlem sırasındaki güç kesintisi sisteminizi çökertebilir. Verilerinizi yedeklemeniz gerekeceğinden, kopyasından yazmak iş görecektir.

MS-DOS dosya yapısı aynı zamanda dahili saçılımdan dolayı büyük miktarda alan kaybetmesiyle bilinir. 256 MB'dan büyük disk kısımları için, DOS blok boyutu kullanılamayacak kadar büyür (Bu durum belli ölçülerde FAT32 için de geçerlidir). Ext2, 0.5 TB (terabayt, 1 TB= 1024 GB'tır) ve üstü büyüklükler dışında, sizi büyük dosya yapıları için büyük bloklar seçmeye zorlamaz. Bu durumlarda küçük blok büyüklükleri verimsiz hale gelmektedir. Dolayısıyla, DOS'takinin aksine büyük diskleri blok büyüklüğünü küçük tutmak için bir kaç bölüme ayırmanın gereği yoktur.

Eğer pek çok küçük dosyanız varsa 1 kB'lık blok büyüklükleri kullanın. Büyük disk bölümleri için 4 kB yeterlidir.

## Notlar

- a) Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- b) Konsol görüntüsünü temsil eden sarı zeminli alanlarda metin genişliğine sığmayan satırların sığmayan kısmı `▮` karakteri kullanılarak bir alt satıra indirilmiştir. Sarı zeminli alanlarda `▮` karakteri ile başlayan satırlar bir önceki satırın devamı olarak ele alınmalıdır.

(B3) <http://www.gnu.org/software/parted/parted.html>

(B4) <http://www.powerquest.com/partitionmagic/index.html>

(B5) <http://www.linux-mandrake.com/diskdrake>

(B6) <http://www.tldp.org/HOWTO/Install-Strategies>

(B7) <http://www.nyx.net/~sgjoen/disk.html>

(B8) <http://metalab.unc.edu/mdw/HOWTO/Large-Disk-HOWTO.html>

(B9) <http://metalab.unc.edu/mdw/HOWTO/mini/Quota.html>

(B10) <http://www.linuxdocs.org/HOWTOs/mini/Partition-Rescue.html>

(B11) <http://www.tldp.org/HOWTO/ADSM-Backup.html>

(B12) [http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html\\_single/Backup-With-MSDOS.html](http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Backup-With-MSDOS.html)

(B23) <http://tldp.org/HOWTO/LILO.html>

(B24) <http://www.tldp.org/HOWTO/Large-Disk-HOWTO.html>

- (1) **mkswap** kılavuz sayfasında belirtilmese de 2.4 sürüm çekirdek ile takas alanı sayısı her biri en büyük 64Gb olan 64 taneye kadar çıkmıştır. 2.6 çekirdek üzerinde çalışan 64 bitlik opteronda her biri 16 Tb olan 128 takas alanı oluşturmak mümkündür (Peter Chubb'a hesaplaması için teşekkürler).

- (2) Ç.N.: Linux Sistem Yöneticisinin Kılavuzu kitabının [Disk Bölümleri](#)<sup>(B46)</sup> bölümünde gerçekçi bir örnek bulabilirsiniz.

(B46) [../sag/sag.pdf#sag\\_disks-partiton](http://sag.sag.pdf#sag_disks-partiton)

(B47) <http://www.stud.uni-hannover.de/user/76201/gpart>

(B48) <http://www.cgsecurity.org/index.html?testdisk.html>

Bu dosya (partition-howto.pdf), belgenin XML biçiminin  $\text{\TeX}$ Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

11 Şubat 2007