



KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

Linux Ağ Yönetimi

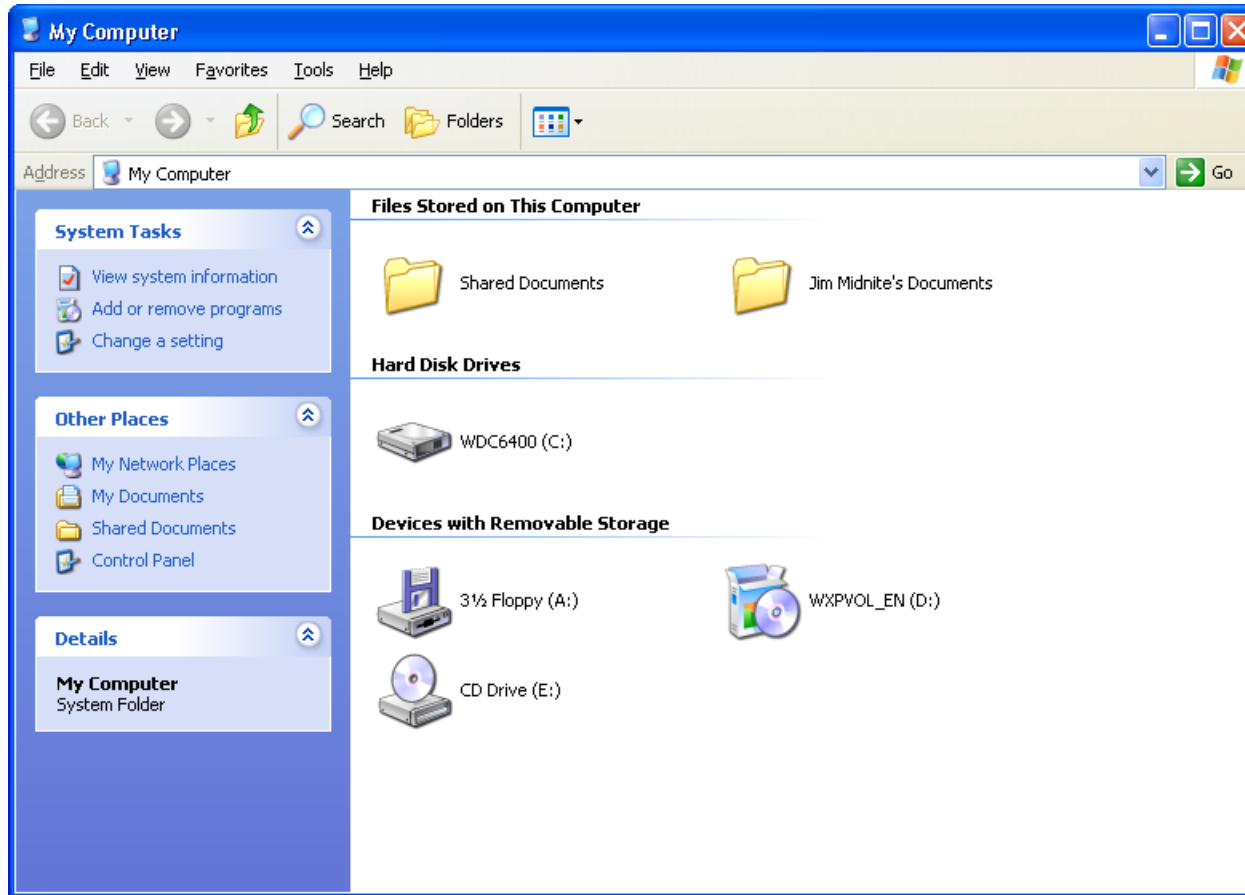
18. Hafta – Dosyalama Sistemi



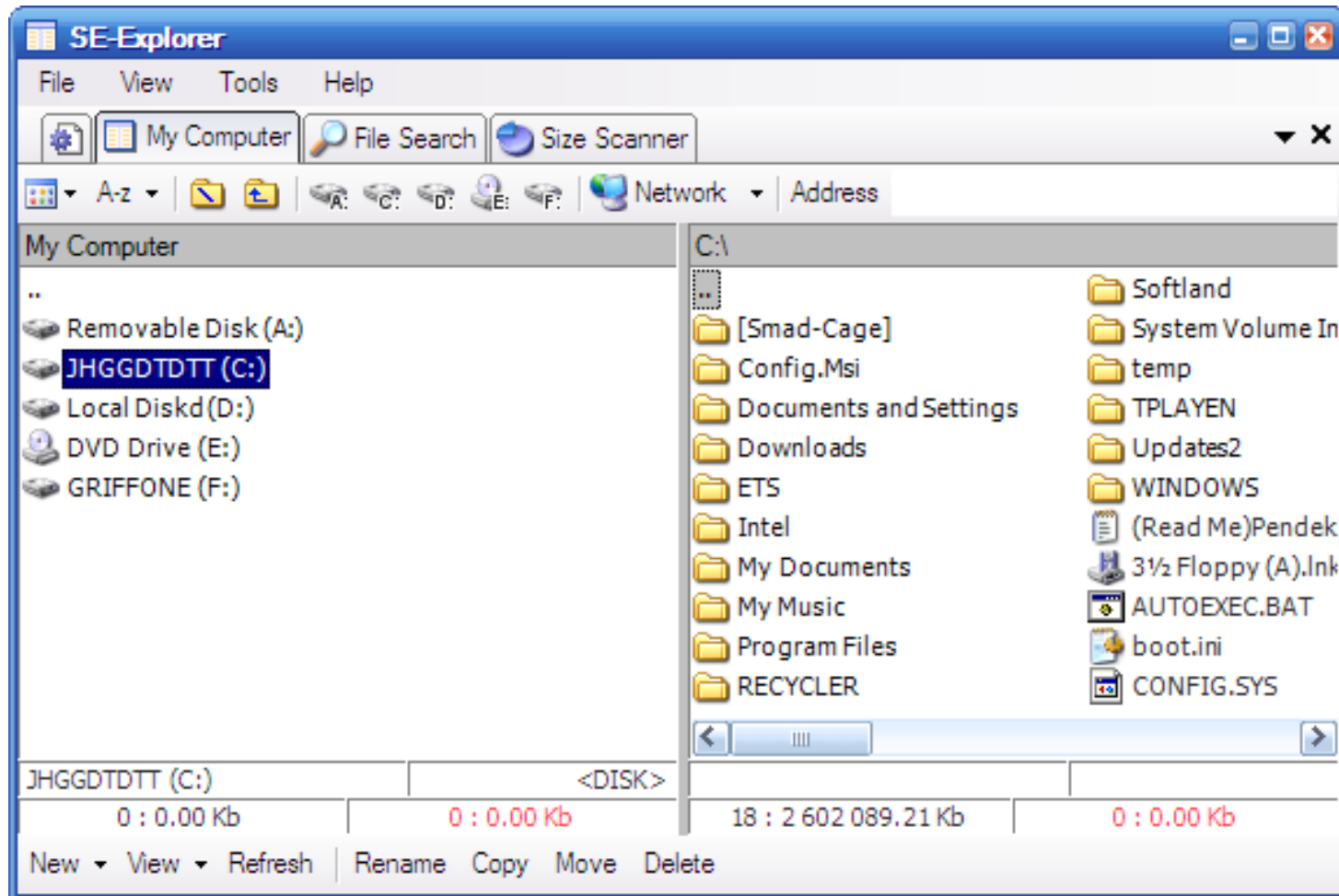
Yrd. Doç. Dr. A. Burak İNNER

Kocaeli Üniversitesi Bilgisayar Mühendisliği
Yapay Zeka ve Benzetim Sistemleri Ar-Ge Lab.
<http://yapbenzet.kocaeli.edu.tr>

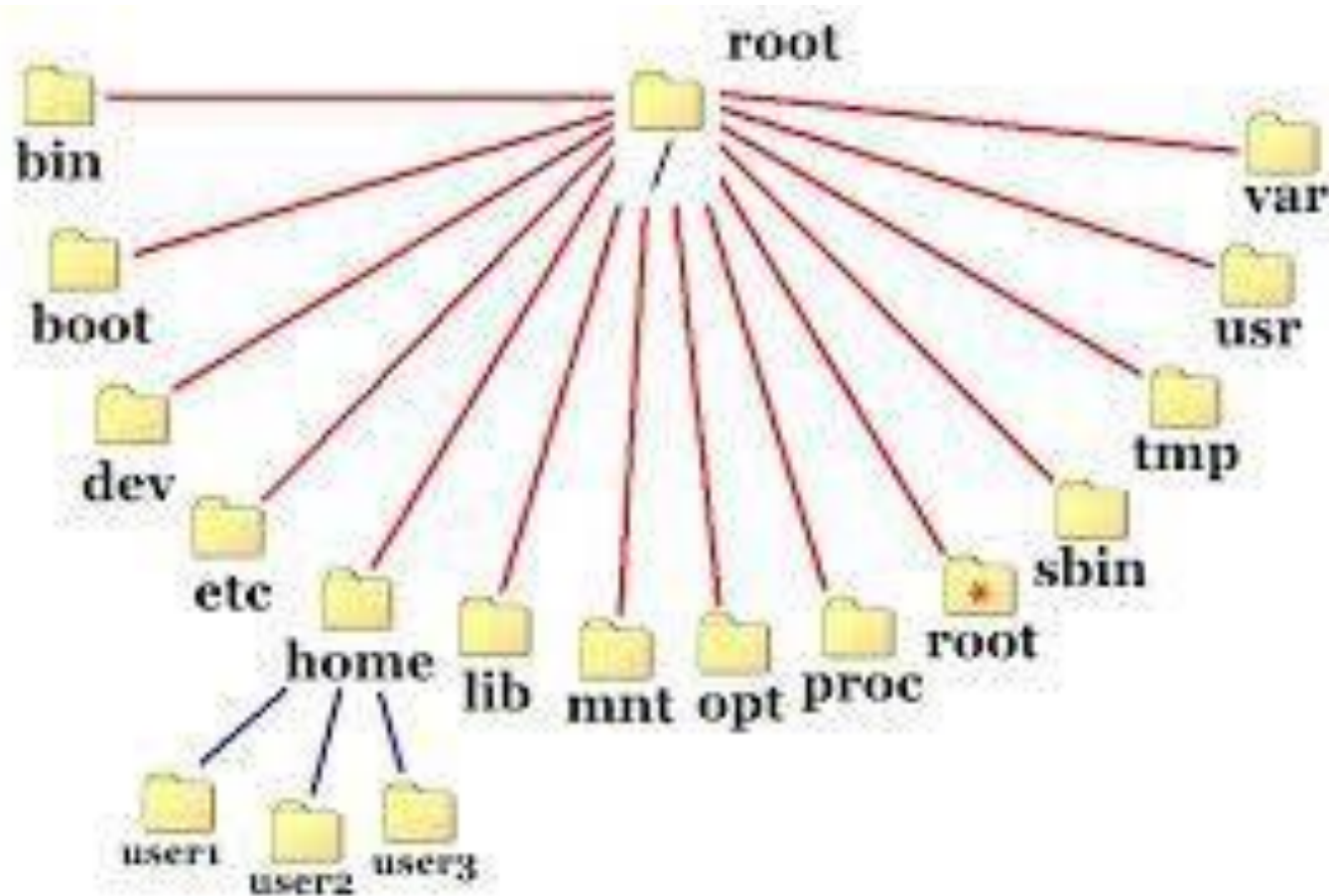
Dosya Sistemi Düzeni(Windows)



Dosya Sistemi Düzeni(Windows)



Dosya Sistemi Düzeni(Linux)



Linux'ta Dosya Hiyerarşisi

Windows'ta,

- Her disk bölümü kendi dosyalama ağacına sahiptir.
- Herhangi bir harici disk veya taşınabilir disk takıldığında yeni dosyalama ağacı oluşur

Linux ise birleşik dosya hiyerarşisi altyapısına sahiptir.

Bunun anlamı ne kadar farklı sayıda disk adeti olursa olsun tek bir dosya ağacı altyapısı vardır.

Dosyalama sisteminin kökü “/” ile başlar.

Birçok Linux dağıtımında bu hiyerarşik yapı temel teşkil eder.

Bu hiyerarşik altyapı çoklu kayıt dosyaları için alt dosyalama ağacı yapısını barındırır.

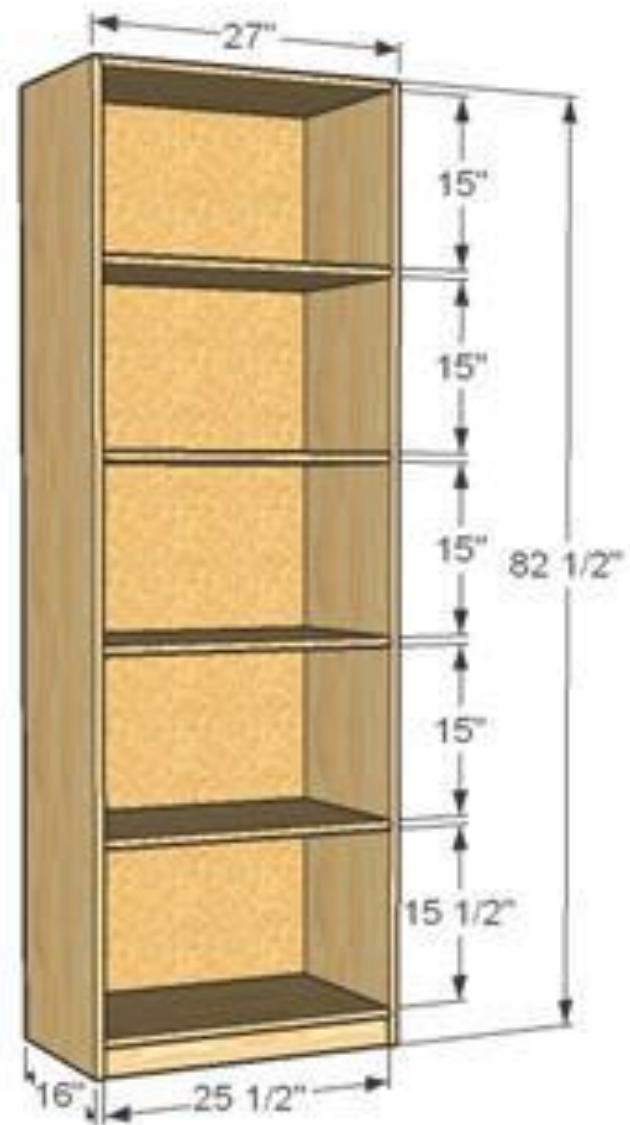
Örneğin : Hard Disk partitions , Flash Drives , CD/DVD Driver , SD Card

Depolama İşlemi Nasıl
Gerçekleştirilir....

Boş Depolama Alanı



Bölümleme



Dosya Sistemi Oluřturma ve Kurulumu



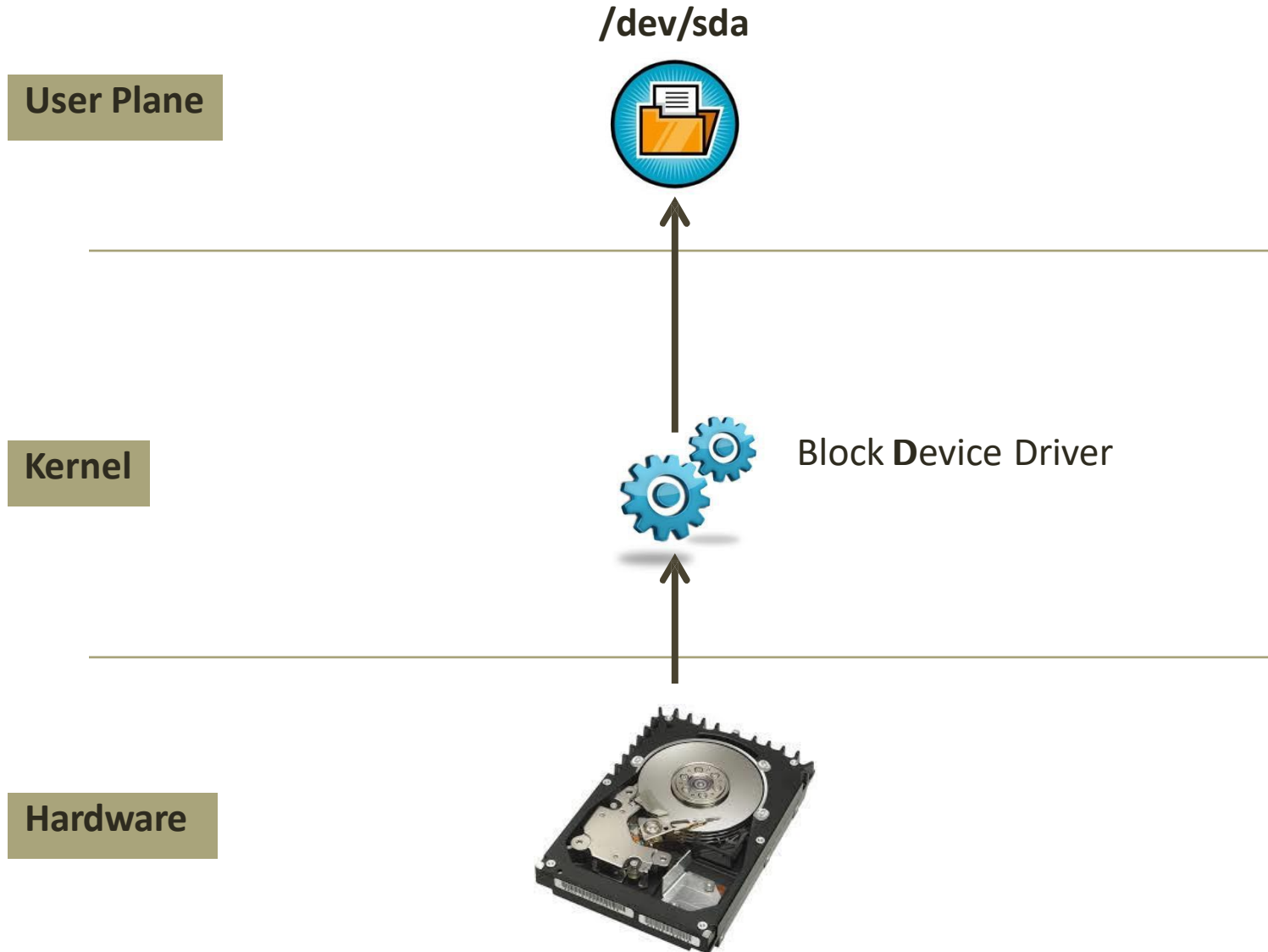
Anahtarları İstenilen Yerde Monte Edin



1. Adım: Bir Depolama Aygıtı Alın



Depolama Aygıtı, bir Donanım Aygıtıdır



2. Adım: Depolama Aygıtını Bölme

Ubuntu Masaüstü

root@ekin-VirtualBox: /home/ekin

ekin@ekin-VirtualBox:~\$ su

Parola:

su: Yetkilendirme hatası

ekin@ekin-VirtualBox:~\$ sudo passwd

[sudo] password for ekin:

Yeni parolayı girin:

Yeni parolayı tekrar girin:

passwd: şifre başarıyla güncellendi

ekin@ekin-VirtualBox:~\$ su

Parola:

root@ekin-VirtualBox:/home/ekin# su

root@ekin-VirtualBox:/home/ekin# fdisk -l

Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

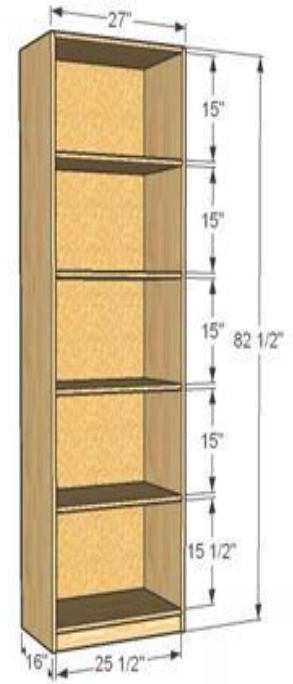
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0x3a3ea574

Aygıt	Açılış	Start	Son	Sektör	Size	Id	Türü
/dev/sda1	*	2048	48234495	48232448	23G	83	Linux
/dev/sda2		48236542	52426751	4190210	2G	5	Ek
/dev/sda5		48236544	52426751	4190208	2G	82	Linux takas / Solaris

root@ekin-VirtualBox:/home/ekin#



Disk Bölümlerini Yönetme

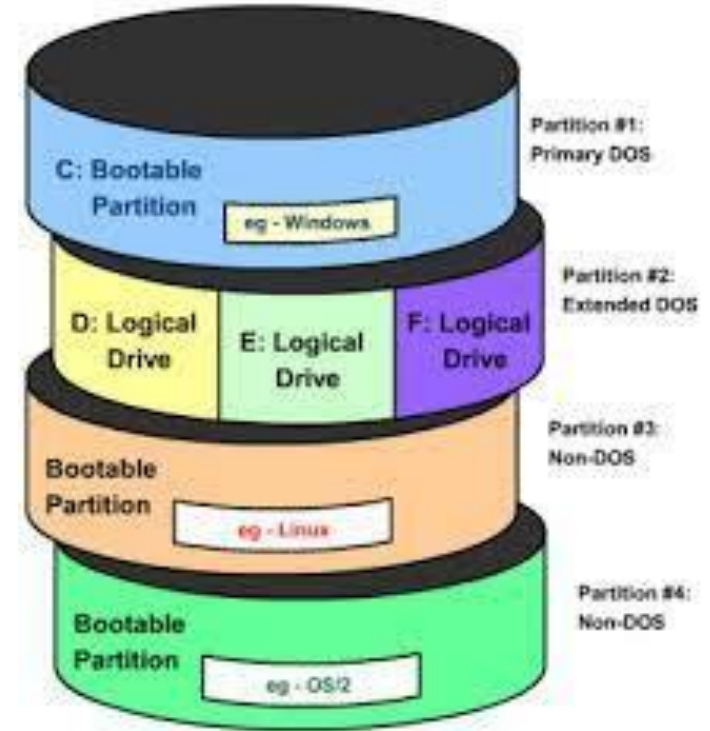
- Disk daha sonra bir veya daha fazla bölüme ayrılabilir. Her bölüm fiziksel olarak disk depolama alanının bir bölümünü kaplar . Bölmeler birbirinden izole edilir .
- Neden birden fazla bölüme sahip olmamız gerekiyor?
 - Bu, her bölümü farklı bir dosya sistemi türüne sahip olmamızı sağlar;
 - Bazı bölümler yalnızca okunur, bazıları yazılabilir olacak.
 - Bazı bölümler verileri sıkıştırılacak, bazıları da bunlara sahip olacak Sıkıştırılmadan .
 - Bazı bölümler, verilerinin şifreleneceği, diğerlerinin dosyalarının açık metni olacak

- Neden birden fazla bölüme sahip olmamız gerekiyor?
 - Bu, her bölümü farklı bir dosya sistemi türüne sahip olmamızı sağlar;
 - Bazı bölümler yalnızca okunur, bazıları yazılabilir olacak.
 - Bazı bölümler verileri sıkıştırılacak, bazıları da bunlara sahip olacak Sıkıştırılmadan .
 - Bazı bölümler, verilerinin şifreleneceği, diğerlerinin dosyalarının açık metni olacak
 - Farklı bölümlerdeki verilerin izolasyonu,
 - Bir bölümün bozulması diğer bölümleri etkilemez
 - Örneğin, günlük dosyası boyutlarının ve diğer biriktiricilerin patlamalarının sistem işlemini etkilemediğinden emin olmak için **/var** dizininin ayrı bir bölüme yerleştirilmesi
 - Disk daha sonra bir veya daha fazla bölüme ayrılabilir

Bölümlleme Kategorileri

Bölümler :

- **Birincil Bölmeler:**
 - Önyüklenebilir olarak ayarlanabilirler
 - Üzerinde önyüklenebilir bir işletim sistemi yükleyebilirsiniz
 - Sayım sınırlı
- **Mantıksal Bölmeler:**
 - Önyüklenebilir yapılamıyor
 - Üzerine hiçbir OS kurulamaz
 - Yalnızca veri taşımak için yararlıdır
 - Veri izolasyonu sağlar
- **Bölmeyi Değiştir (Swap Partition):**



SWAP Partition (Takas Bölümleme)

- SWAP bölümü, yalnızca sistem tarafından erişilebilen bir bölümdür
- Sistem fiziksel olarak genişlemek için bir veya daha fazla takas bölümü kullanır
- Yüksek bellek kullanımı durumunda sık erişilemeyen bellek sayfaları fiziksel bellekten takas bölümüne taşınabilir Bununla birlikte, erişim zamanı bellekten çok daha fazladır
- Kullanıcı bu bölüme erişemez veya kendi dosya sistemini buraya yerleştiremez

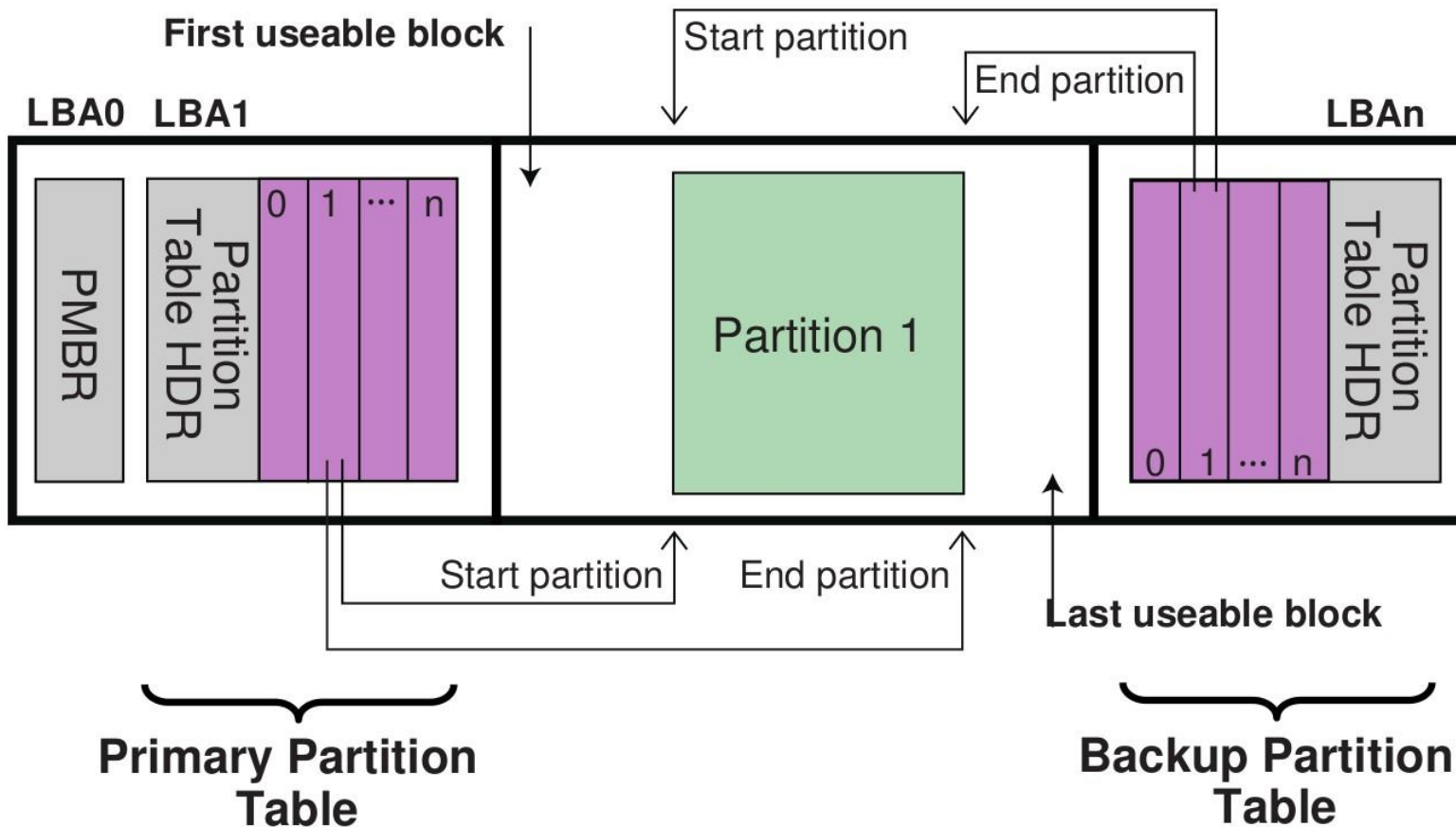
- Takas alanının fiziki miktarın iki katı olması önerilir
- Sistemdeki bellek 512 MB RAM'li bir sistem örneği:
 - 1. olasılık: 1 GB'lık bir takas bölümü
 - 2. olasılık: 512 MB'lık iki takas bölümü
 - 3. olasılık: iki sabit disk ile: her diskte 512 MB'lık 1 bölüm. Bu Son seçenek çok sayıda I / O beklendiğinde en iyi sonuçları verecektir
- Genel olarak, birden çok takas bölümü kullanmak erişim süresini hızlandıracaktır (özellikle farklı fiziksel depolama aygıtlarında bulunuyorsa)

Bölümleme Tablosu

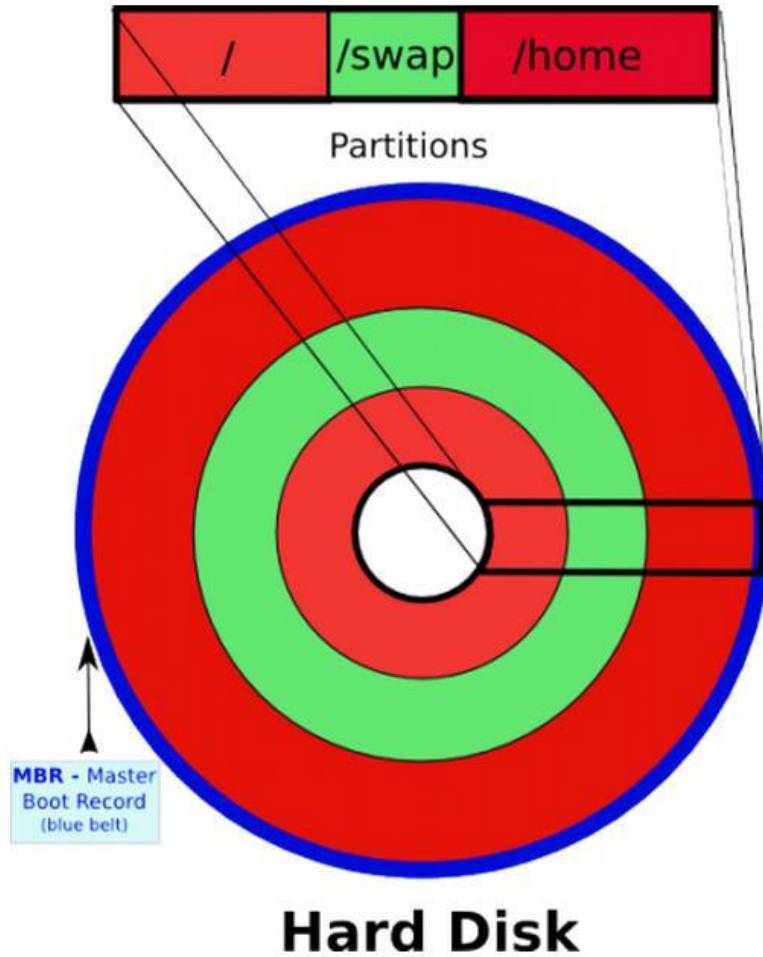
- Bölüm tablosu, diskteki bilinen bir yerde bulunan bir tablodur
- Diskteki bölümlerin tanımını içerir
 - Başlangıç Konumu
 - Uzunluk (veya Bitiş Konumu)
 - Tür (Birincil / Mantıksal)

- Diğer Bilgiler Bölüm tablosu için birden fazla format vardır, en popüler olanlar,
 - Ana Önyükleme Kaydı (MBR) .
 - 2 TB'a kadar diskleri destekler.
 - 4 ana bölüme kadar destekler (çoklu önyüklemeli ortamda maksimum 4 OS)
 - En yaygın formattır.
 - GUID Bölüm Tablosu (GPT)
 - Daha büyük boyutlu diskleri destekler
 - Daha fazla birincil bölümleri destekler (128'e kadar)

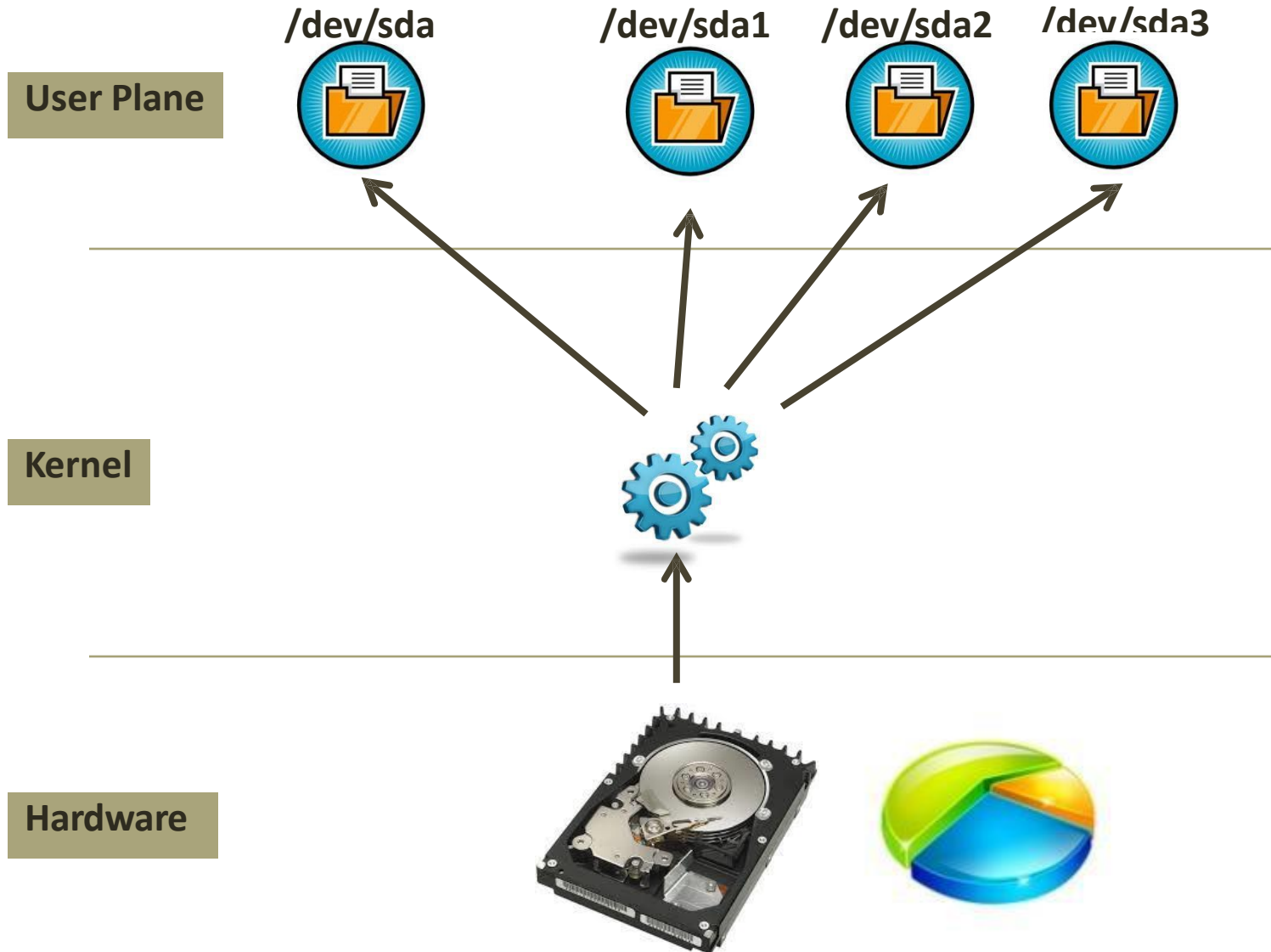
Bölme Tablosu Örneği (GPT)



Bölme ve Bölme Tablosu



Cihaz Bölümleme Dosyaları



Disk Bölümleme Yöneticisi (fdisk komutu)

\$ fdisk -l

\$ fdisk <device Name>

- Bu komut, disk görüntüleme ve yönetiminden sorumludur

\$ sudo fdisk -l

\$ sudo fdisk -l /dev/sda

\$ sudo fdisk /dev/sda

- Bu komutu kullanarak
 - Disk bölümlerini gösterme
 - Yeni bölüm oluşturma
 - Mevcut bölümleri silme
 - Mevcut bölümlerin boyutunu değiştirme
 - This is achieved by reading/writing in the partition table
- Bu, bölüm tablosunda okuma / yazma yoluyla elde edilir
- **fdisk** GPT bölüm tablosu biçimini desteklemediğini unutmayın; **parted** komutu kullanan

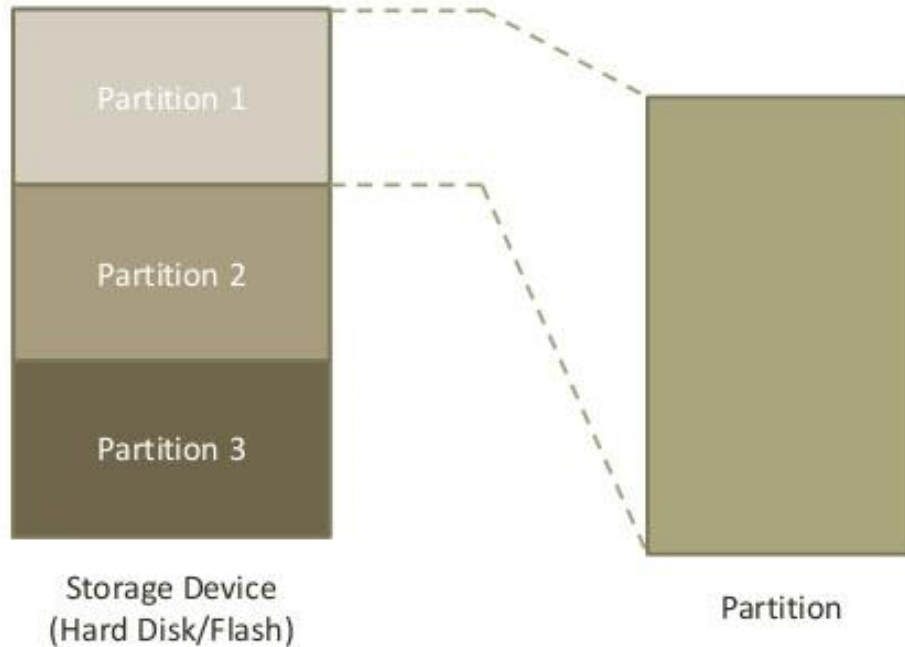
Disk Bölümleme Yöneticisi (fdisk komutu)

```
root@ekin-VirtualBox: /home/ekin
root@ekin-VirtualBox:/home/ekin# sudo fdisk -l
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x3a3ea574

Aygıt          Açılış      Start      Son      Sektör Size Id Türü
/dev/sda1      *           2048 48234495 48232448 23G 83 Linux
/dev/sda2          48236542 52426751 4190210 2G 5 Ek
/dev/sda5          48236544 52426751 4190208 2G 82 Linux takas / Solaris
root@ekin-VirtualBox:/home/ekin#
```

Dosyaları şimdi saklayabilir miyiz ??

Great... So can we store Files Now ??

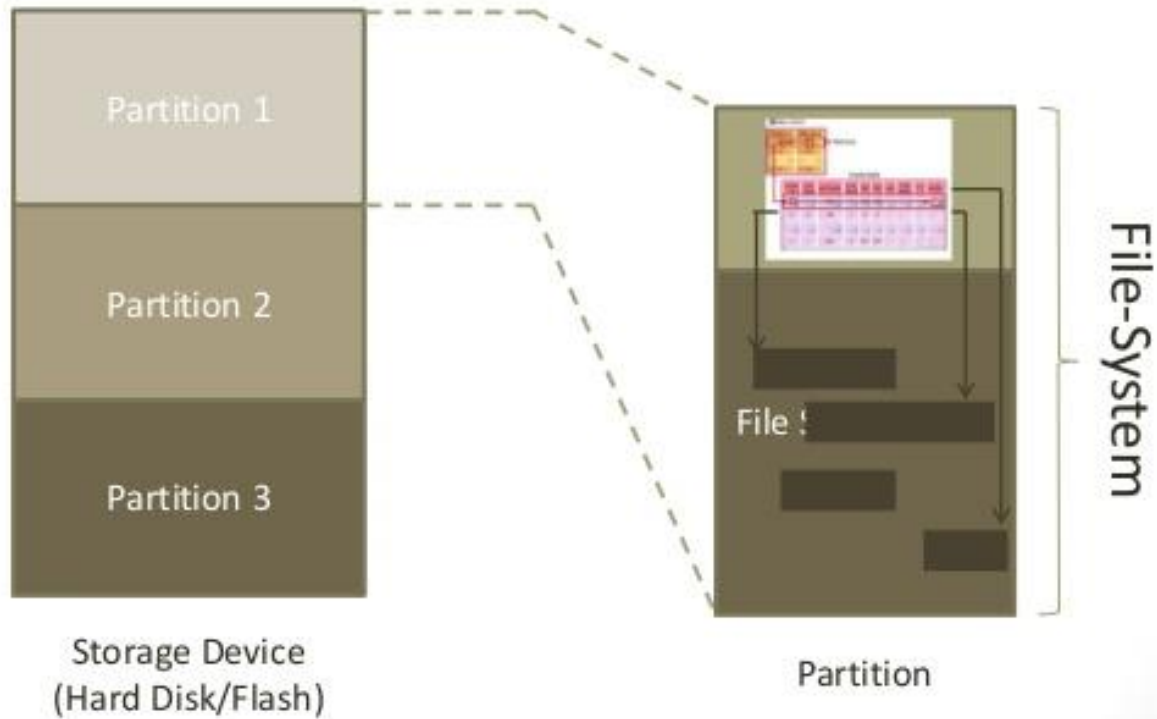


3. Adım: Dosya Sistemi Oluřturma ve Kurulum



Dosya Sistemi Kurma

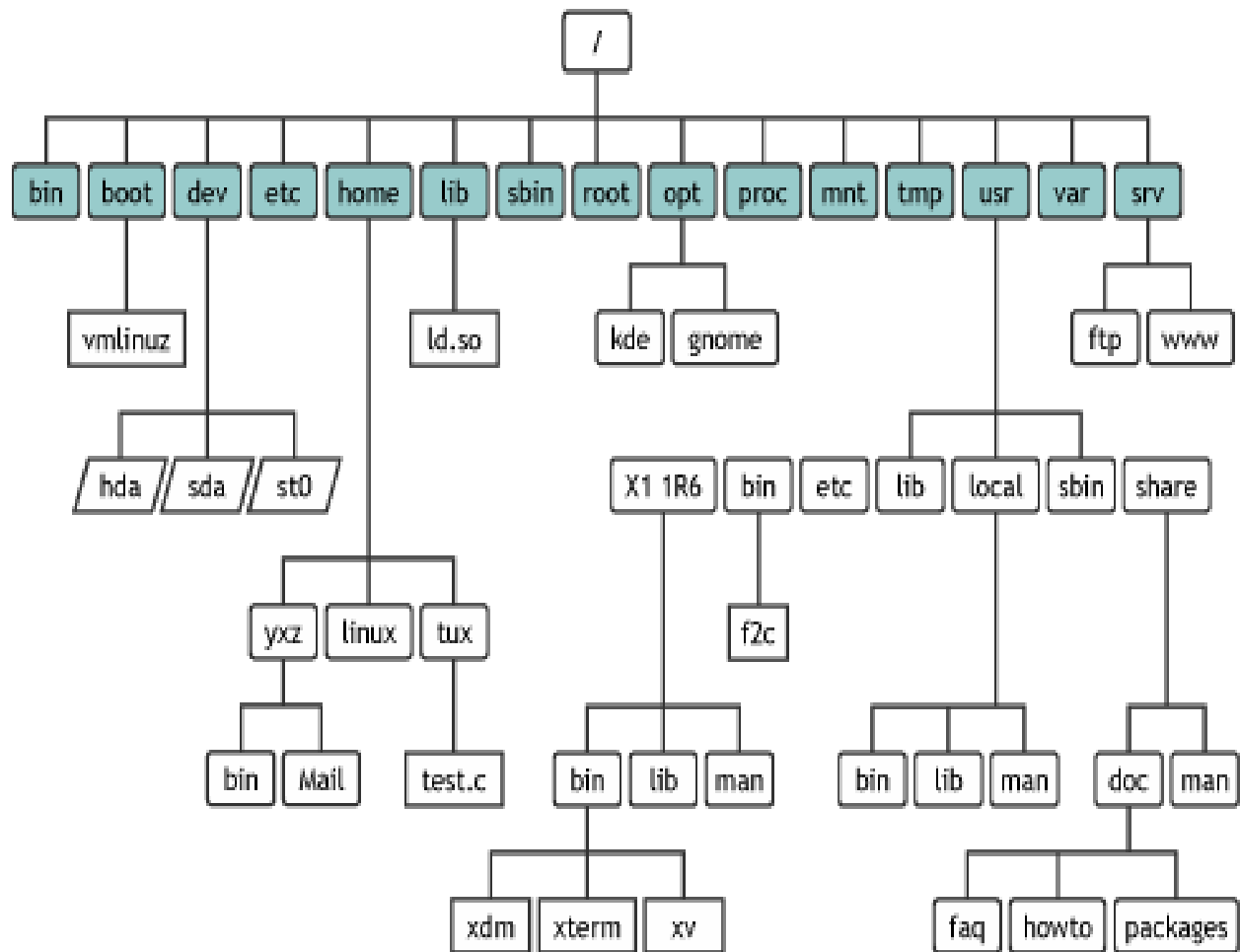
Install a File-System



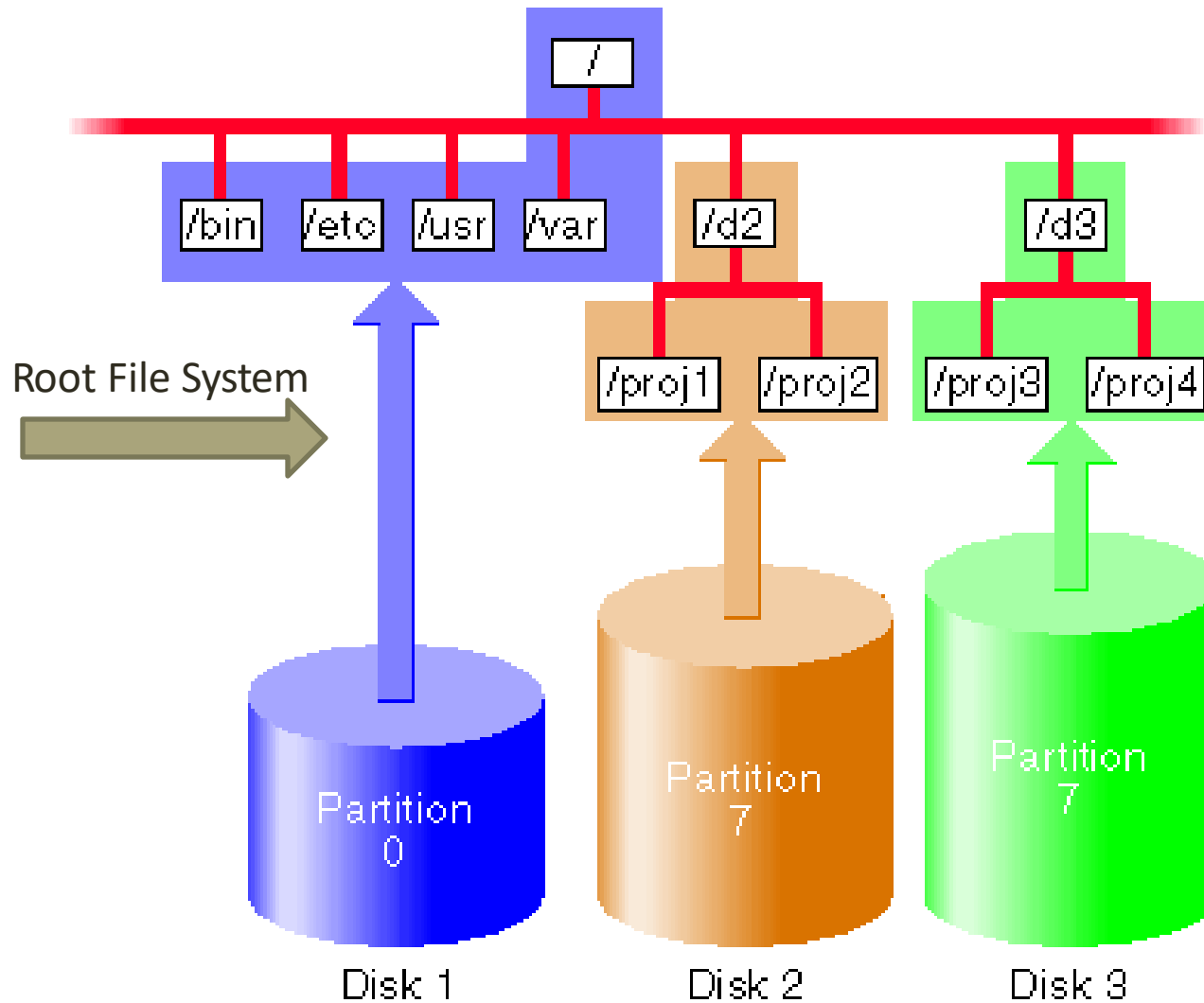
Dosya Sistemi Türleri

- Farklı dosya sistemleri türleri vardır
- İhtiyaçlarımız için en uygun türü seçeriz,
 - Depolama Ortamı türü (sabit disk, flash bellek, ağ, ...)
 - Okunur veya Okuma / Yazma
 - Desteklenen Dosya Boyutları
 - Performans için Optimize Edin Dosya Boyutları için iyileştirme (Sıkıştırma gerçekleştirir)
 - Güvenlik için En İyi Duruma Getirme (Şifreleme gerçekleştirir)
 - Arızalardan sonra veri kurtarma işlemini destekler (Günlüğe kaydetme)
- Diğer Kriterler Bazı özel dosya sistemleri (procfs, sysfs)
- Sık kullanılan dosya sistemi türlerinden bazıları şunlardır: (ext2, ext3, ext4, NTFS, FAT, JFSS2, NFS,

Adım 4: Dosya Sistemini Monte Edin



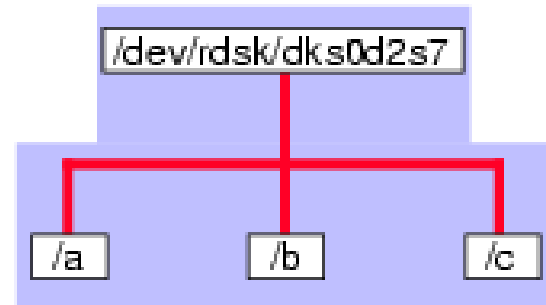
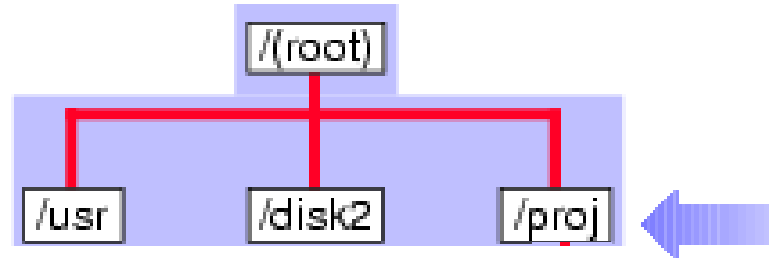
Dosya Sisteminin Bağlanması



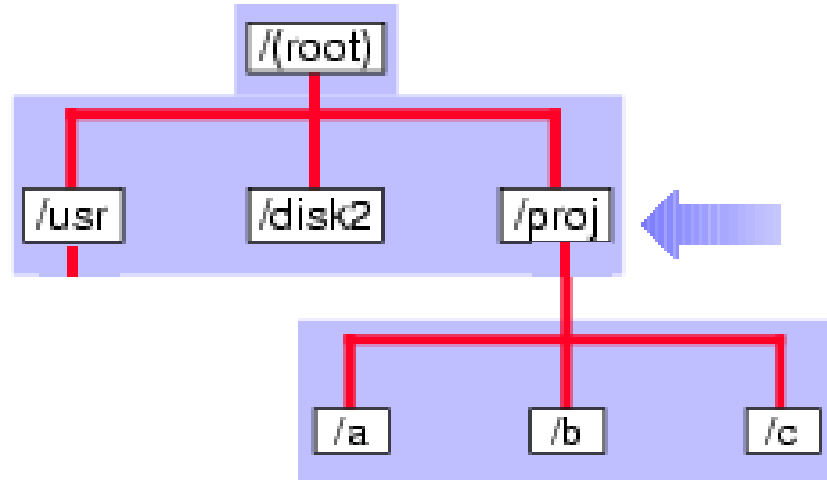
Kök Dosya Sistemi

- Root FileSystem, sistemi başlatmak ve çalıştırmak için gerekli tüm dosyaları içeren bir dosya sistemidir.
- Çekirdeğin başlatılmasında monte edilir
- Kök dosya sistemi için bağlama noktası `"/`
- Bu dosya sistemi, çalışma sırasında sökülemez
- Diğer Dosya sistemleri, kök dosya sistemi takıldıktan sonra monte edilebilir veya sökülebilir.

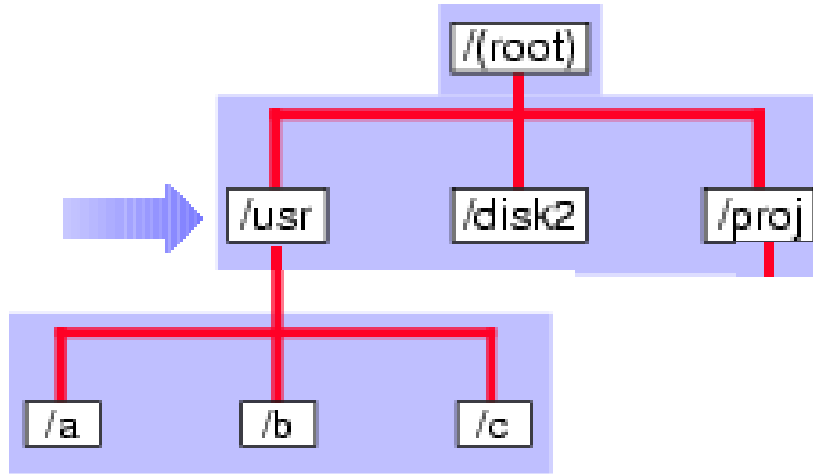
Bağlama Noktası Seçimi



Bağlama Noktası Seçimi



Bağlama Noktası Seçimi



Dosya Sisteminin Baęlanması



- Bir dosya sistemini baęladığımızda, baęlama noktasını belirtiriz
- Baęlama noktası, dosya sistemine erişmek istediğimiz dizindir.
- Baęlamayı gerçekleştirmeden önce, dizininin boş bir dizin olması gerekir

Bağlanmış Dosya Sistemlerini Görüntüleme (mount komutu)

\$ mount

\$ mount -l

\$ mount -l -t <fs Type>

- Bu komut, bağlanan dosya sistemlerini, kendi bağlama noktası ve ilişkili aygıtıda dahil olmak üzere kendi bilgileriyle birlikte görüntüler.

```
root@ekin-VirtualBox: /home/ekin
root@ekin-VirtualBox:/home/ekin# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1002776k,nr_inodes=250694,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=204792k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
```

mount komutu

\$ mount -t <fs Type> <device> <mount point>

- Bu komut, belirli bir blok aygıt dosyası için dosya sistemini (belirli bir türe ait) bir montaj noktasına
- Örnekler:

***\$ sudo mount -t ext4 /dev/sda1
/home/aelarabawy/project/***

\$ sudo mount -t iso9660 -o ro /dev/sr0 /mnt

Mount-umount komutları

\$ umount <device>

\$ umount <mount point>

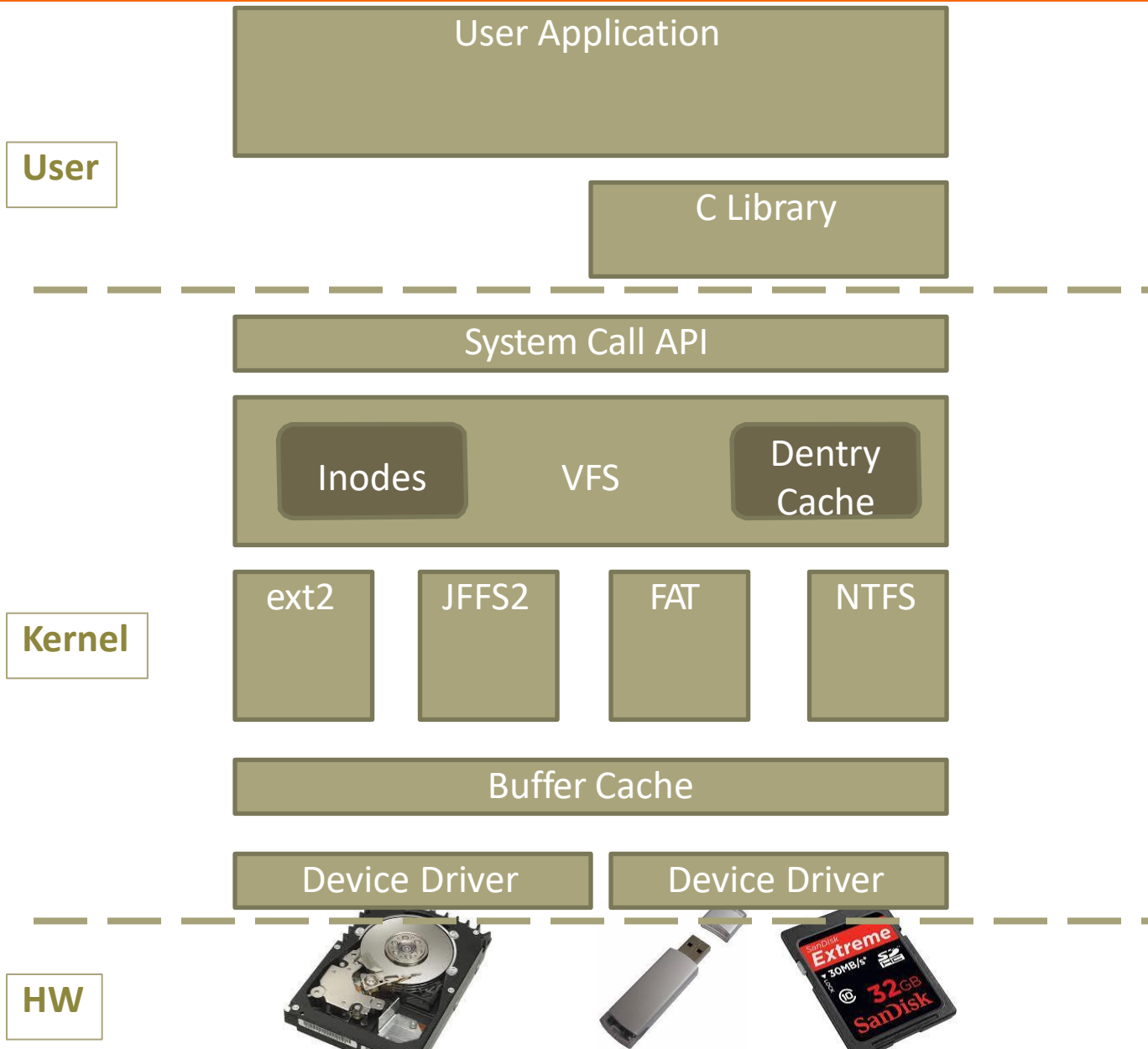
- Bu komut daha önce monte edilmiş dosya sistemini kaldırır. Dosya sistemi, tarafından tanımlanabilir,
 - İlişkili cihaz (Örnek */dev/sd1*)
 - İlişkilendirilmiş bağlantı noktası (Örnek */mnt/SdCard*)

/etc/fstab

- **etc / fstab** dosyası, açıklamaları ile birlikte bir dosya sistemi listesini içerir
 - Bağlantı noktası
 - İlişkili cihaz
 - Dosya sistemi tipi
- Bu dosya mount komutu tarafından kullanılabilir,
 - / Etc / fstab'de listelenen bir dosya sistemini bağlamak için, listelemek zorunda değilsiniz Tüm bilgileri, aygıt adını veya bağlantı noktası belirtmek
 - ***\$ sudo mount /dev/sd2***
 - ***\$ sudo mount /mnt/my-SD-Card***
 - Mount komutu geri kalan bilgileri okur.

\$ sudo mount -a

\$ sudo umount -a



Büyük Resim

- Kullanıcı uygulamaları standart bir API kullanarak dosyaları erişir
- Depolama ortamı, bölüm veya kullanılan dosya sistemi türü ne olursa olsun bu durum değişmez
- Kullanılan API, çekirdeğe doğru bir sistem çağrısı yapar (C-Library tarafından üretilir)
- Sistem çağrısı, Linux çekirdeğinin VFS (Sanal Dosya Sistemi) alt sistemi tarafından alınır
- Bu alt sistem, kavramlar yoluyla dosyaya birleşik bir erişim sağlar
-

- Dügümler ve süs bitkileri (Bir önceki ders ayrıntılı olarak VFS'yi kapsamıştır)
- VFS daha sonra bu dosyayı içeren kullanılan FileSystem ile iletişim kurar
- Dosya sistemi sırayla blok özetini kullanarak depolama ortamı aygıt sürücüsü ile iletişim kurar,
- Aygıt sürücüsü, depolama donanımı aygıtıyla etkileşim kurmaktan sorumludur

En çok kullanılan dosya
sistemi türleri

Linux Geniřletilmiř Dosya Sistemleri(ext2)

- Linux tarafından hem sabit disklerde hem de çıkarılabilir ortamlarda kullanılabilen yüksek bařarımı bir dosya sistemidir.
- Geniřletilmiř dosya sisteminin (**ext**) uzatması olarak geliřtirilmiřtir.
- Hem iřlemci kullanımı hem de hız aılarından Linux tarafından desteklenen dosya sistemleri arasında en yüksek bařarılısıdır.

Journaling

- Günlüğe kaydetmeyi destekleyen dosya sistemi, "Günlük dosyası" özel bir dosyaya sahip olacaktır.
- Dosya sisteminde her değişiklik olduğunda (bir yazma işlemi), bu değişiklik dergi dosyasında önce izlenir ve ardından dosya sistemindeki gerçek dosyaya karar verilir.
- Böylelikle, bir bozulma (dosya sistemi içeriği ve meta verileri arasındaki uyumsuzluk) durumunda geri dönebileceğimiz bir günlük defterine sahip olacağız.
- Ani bir kapatma / yeniden başlatma sonrasında bir önyükleme yapıldığında günlük dosyası izlenir ve dosya sisteminin içeriği ile karşılaştırılır
- Dergide yapılan değişiklikler, korumak için uygulanabilir veya kaldırılabilir.
- Veri tutarlılığı Çoğu yeni dosya sistemi, dosya sistemi içeriğinin veri bütünlüğünü korumak için Günlüğe Ekleme'yi desteklemektedir.

Linux Geniřletilmiř Dosya Sistemleri(ext3)

- Ext3 dosya sistemi, Journalling'i desteklemek iin ext2 dosya sisteminin bir uzantısıdır
- Ext2 ile ileri ve geri kelimelere uyumludur (ext2 dosya sistemini ext3'e, tersini de ext3'e dnřtrebiliriz)
- Ext2'de sistem aniden kapandıėında, bir sonraki nykleme uzun sryor nk dosya sistemi zerinde bir tutarlılık denetimi alıřıyor
- Ext3'te, tutarlılık denetimi gerekmez; deėiřiklikleri doėrulamak iin gnlk dosyası kontrol edilir, bu daha hızlı bir iřlemdir

Linux Geniřletilmiř Dosya Sistemleri(ext4)

- Ext4 dosya sistemi ext3 dosya sisteminin bir uzantısıdır řu anda, Linux için varsayılan dosya sistemidir.
- Ayrıca günlük kaydını da destekliyor Ext3'ün bazı sınırlamalarını ortadan kaldırır,
- Ext4 dosya sistemi ext3 için sınır olan 16 terabaytan fazla dosya sistemi boyutunu destekleyebilir
- Ext4 dosya sistemi, 1 Terabayt'a kadar dosya boyutlarını destekleyebilir

İkinci Nesil Günlük Dosyalama Sistemi

JFFS2 Dosya Sistemi

- Jffs2 dosya sistemi flash bellek depolama aygıtlarıyla birlikte kullanılır
- Gömülü sistemlerde flaş bellek depolama birimi kullanımı çok yaygındır
- Flash belleğin özellikleri:
 - Blok başına birden çok dosya: Blok boyutu büyüktür (onlarca ila yüzlerce kilobayt). Tipik bir değer 128KB'dir
 - Bu, aynı bloğa birden fazla dosya depolamanız gerektiği anlamına geliyor
 - Buna göre, flaş üzerindeki bir blok birkaç küçük dosya içerebilir
-

- Yavaş çekim işlemleri
 - Flaşın boş bir yerine bir değer yazılması, bir defada bir bayt (veya kelime) olarak yapılır (normal cihazlar gibi)
 - Bununla birlikte, tek bir bayt (sözcüğü) silmek (veya değiştirmek) Silme, tüm bloğun silinmesini gerektirir
 - Eğer küçük bir dosyayı değiştirmemiz gerekiyorsa, tüm blok silinecektir, daha sonra blok (veya başka bir tane boş olan) yeniden yazılır.
 - Flashki yazma işlemlerini diğer cihazlardan çok daha yavaş yapar
 - Bir yazma işlemi sırasında elektrik kesintileri nedeniyle daha yüksek bozulma olasılığına neden olur
- Flash Ömrü
 - Başka bir sınırlama olan flaş bellek sınırlı kullanım ömrüne sahiptir (yazma operasyonlarının sayısında belirtilmiştir)
 - Flash bellek ömrü, yazma işlemleri sayısıyla ölçülür
 - Flash zarar görmesini önlemek için yazım işlemlerini bile dışarıya çıkarmalıyız

İkinci Nesil Günlük Dosyalama Sistemi

JFFS2 Dosya Sistemi

- Jffs2 dosya sistemi flaşı aşağıdaki gibi işler,
 - Yavaş yazma işlemi nedeniyle, günlük kaydı çok önemlidir ve dolayısıyla JFFS2'de günlük kaydı desteklenmektedir
 - Yazma işlemlerini flaşa dağıtmak için blokların eşit olarak kullanılmasını sağlar.
 - Buna Aşınma Tesviyesi denir
- Güncellenmeyi sürdüreceğ araçların kullanımı ile özel bir dikkat gösterilmesine ihtiyaç vardır
 - Günlüğe kaydetme araçları (ex. Syslogd ve klogd) gibi dosyalar
 - Eğer başka depolama biçimlerimize sahipsek, çıktıyı flash bellekten uzakta yönlendirmek daha iyi olur
 - Önbellek kullanımının kullanılması, değişiklik sayısını azaltmak için yararlı olabilir
 - Sistem performansını ve flaş ömrünü etkiler

Cram Dosya Sistemi cramfs

- Bu dosya sistemi amacı bir dosya sistemini küçük bir **ROM'a sıkıştırmaktır**
- Bu salt okunur bir dosya sistemidir
- Dosya sistemindeki verilerin sıkıştırılmasını destekler
- Küçük gömülü sistemler için salt okunur verileri küçük bir ROM veya flaşla saklamada kullanışlıdır
- Önyükleme ROM'ları için ideal

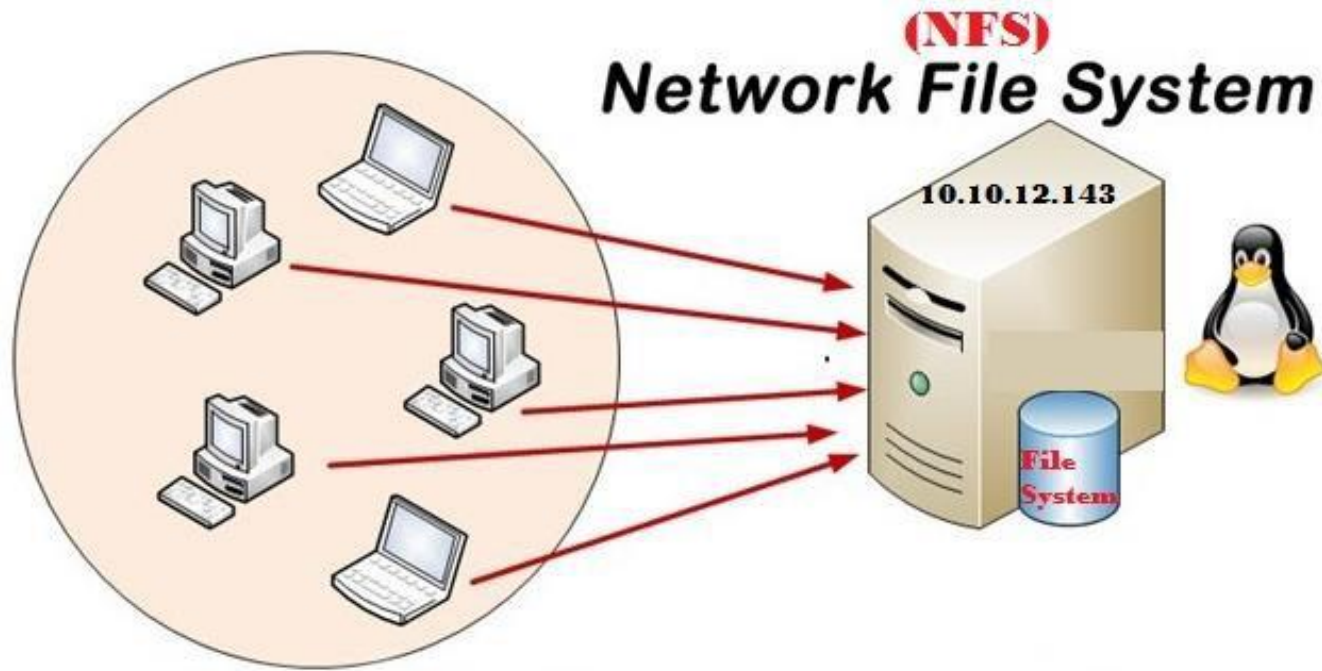
Bellekte Barındırılan Dosya Sistemleri ramfs

- Sistem belleğinde (RAM) yaşayan bir dosya sistemi
- Yüksek erişim hızı sağlar, ancak uçucudur (yeniden başlatma zamanında veya kapanış sırasında silinir)
- RamDisk'lerden farklıdır, ihtiyaca göre büyür ve küçülür.

Bellekte Barındırılan Dosya Sistemleri tmpfs

- Ramf gibi herşey sistem geçici bellekte (RAM) saklanır
- Buna göre, güç kesildiğinde veya yeniden başlatıldığında bu dosya sisteminin içeriği kaybolur
- Ramf'lerden farklı olarak dinamik olarak büyüyemez
- Ayrıca ram kullanamıyorsa takas da kullanabilir
- Normalde / tmp'ye bağlayın

Ağ Dosya Sistemi



Ağ Dosya Sistemi NFS

- Bu dosya sistemi uzaktaki bir makinede bulunacak ve ağ üzerinden erişilecek.
- Ağdaki klasörleri paylaşmak için kullanışlıdır
 - Merkezi bir NFS sunucusu dosya sistemi verilerini içerecektir
 - Verilere erişmesi gereken tüm makinelerin bir NFS İstemcisi içermesi gerekir

Gömülü Sistemlerde NFS



- NFS için bir diğer çok kullanışlı uygulama Gömülü geliştirmedir Sistemler
 - Gömülü hedef flash bellek boyutu, geliştirme sırasında kullanılan tüm araçları ve yardımcı programları tutamayabilir
 - Dolayısıyla, tüm araçlar ve yardımcı programlar uzaktaki bir makinede (NFS Sunucusu) bulunabilir ve hedef, ona erişebilmek için bir NFS dosya sistemi monte eder
 - Ayrıca, geliştirme süresince, yeni bir yapı oluşturduğumuzda, her ikisinin görüntüsünü yüklememiz gerekmez, bunun yerine ikili dosyayı geliştirme ana makine üzerinde tutuyoruz
 - Hedef, kök dosya sistemini NFS olarak monte edebilir, böylece hedef yalnızca önyükleme ve çekirdeği taşır. Geri kalan her şey uzaktaki makinede olacak

Sanal Dosya Sistemi procfs & sysfs

- Bu dosya sistemleri herhangi bir depolama aygıtında depolanmaz, ancak çekirdek tarafından yönetilirler.
- Bu dosya sistemindeki bir dosyadan okuma, çekirdeğe bir sorgu ile sonuçlanır
- Bir dosyaya yazmak, çekirdeğe bir miktar bilgi gönderilmesine neden olur
- Bu dosya sistemleri ayrı derslerde ayrıntılı olarak incelenecektir.

LINUX KOMUTLARI

Df komutu

\$ df

```
root@ekin-VirtualBox: /home/ekin
root@ekin-VirtualBox:/home/ekin# df
Dosyasistemi    1K-blok    Dolu      Boş    Kull%    Bağlanılan yer
udev            1002776    0         1002776    0%    /dev
tmpfs           204792    3640     201152    2%    /run
/dev/sda1       23606716  5647548  16736976  26%    /
tmpfs           1023956    252     1023704    1%    /dev/shm
tmpfs           5120        4         5116    1%    /run/lock
tmpfs           1023956    0         1023956    0%    /sys/fs/cgroup
tmpfs           204792    48         204744    1%    /run/user/1000
root@ekin-VirtualBox:/home/ekin#
```

Dosya sisteminde disk alanını göstermek için kullanılır.

df komutu

\$ df -i (Disk durumunu görmemizi saö)

```
root@ekin-VirtualBox: /home/ekin
root@ekin-VirtualBox: /home/ekin# df -l
Dosyasistemi    1K-blok    Dolu        Boş    Kull%    Bağlanılan yer
udev            1002776    0           1002776    0% /dev
tmpfs           204792    3640        201152    2% /run
/dev/sda1       23606716  5647548    16736976    26% /
tmpfs           1023956    252         1023704    1% /dev/shm
tmpfs           5120       4           5116      1% /run/lock
tmpfs           1023956    0           1023956    0% /sys/fs/cgroup
tmpfs           204792    48          204744    1% /run/user/1000
root@ekin-VirtualBox: /home/ekin#
```

Kullanılabilir disk alanı göstermek (du komutu)

\$ du <device>

- Bu komut, aygıt dosya adı tarafından belirtilen disk için işlem başına disk kullanımını gösterir

\$ du /dev/sda1

Kaynakça

- ☞ Ahmed ElArabawy, Linux for Embedded Systems for Arabs

Teşekkürler.



Dersin Sonu

Kocaeli Üniversitesi Bilgisayar Mühendisliği
Yapay Zeka ve Benzetim Sistemleri Ar-Ge Lab.
<http://yapbenzet.kocaeli.edu.tr/>