



KOCAELİ ÜNİVERSİTESİ  
**BİLGİSAYAR MÜHENDİSLİĞİ**

# Linux Ağ Yönetimi

## 17. Hafta – Dosyaları Arşivleme ve Sıkıştırma



Yrd. Doç. Dr. A. Burak İNNER

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

# Dosyaları Arşivleme

- Arşivleme, bir ağaç yapısında organize edilen bir grup dosyayı tek bir dosyaya birleştirmektir.
- Bu, yedekleme veya aktarma amaçları gibi durumlarda daha kolay taşıma olanağı sağlar.
- Bu, sıkıştırmadan ayrı bir prosedürdür.
- Dosyaları arşivlemek için kullanılan araç ise "tar"dır.



# Dosyaları Arşivleme (Tar Komutu)

**\$ tar <Ayarlar> <Hedef Arşiv Dosyası> <Arşivlenecek Dizin/Dosya>**

- Bu komut, bir grup dosya / dizini tek **bir tar dosyasına** arşivlemek için kullanılır
- Ayrıca bir tar dosyasının orijinal dizin ağacı yapısına arşivden çıkarmak için kullanılır.
- Bir grup dosya ve dizini arşivlemek için (tar)

**\$ tar cvf <arşivDosya.tar> <Bir dizi dosya veya dizin>**

- 'c' Yaratmak için
  - 'v' Doldurmak(Ayrıntılı Dosya) için
  - 'f' Dosya için
- Örnekler:  
***\$ tar cvf dosyalarım.tar ~/Dökümanlar/pdfs/***  
***\$ tar cvf seçilmisDosyalar.tar ~/Dosya1.txt***  
***~/Dökümanlar/fDosya2.txt ~/\*.pdf***

# Bir Tar Dosyası İle Çalışma(Tar Komutu)

- Şimdi tar dosyasına sahibiz diyelim ve onunla aşağıdakileri yapabiliriz,
  - Bir dosyayı tüm orijinal bileşenlerine ayıklamak (un-tar) için
    - **\$ tar xvf <arşivDosya.tar>**
      - 'x' çıkarmak için
      - Bir tar dosyasından bazı dosyaların / dirlerin çıkarılması
        - **\$ tar xvf <arşivDosya.tar> <Dosyalar/dir'den untar'a>**
    - Bir tar dosyasının içeriğini göstermek için
      - **\$ tar tvf <arşivDosya.tar>**
  - Örnekler:
    - ***\$ tar cvf tar-file.tar \*.pdf \*.txt***
    - ***\$ tar tvf tar-file.tar***
    - ***\$ tar xvf tar-file.tar \*.pdf***
    - ***\$ tar xvf tar-file.tar***

# Bir find komutu içinde tar kullanımı

- Diyelim ki ev dizininizdeki tüm pdf dosyalarını arşivlememiz lazım.
- Bu dosyaları aramak için **find** komutuna ihtiyacımız olacak.
- **Find** komutunun sonucu daha sonra tar komutuna geçirilir.
- Bu, bir boru(pipe) kullanılarak başarılabilir,

*\$ find ~ -isim '\*.pdf' | tar cvf dosya.tar*

- Aynı hedefe ulaşmanın diğer bir yolu, tar komutu, find komutu içerisinde çalıştırılacak kullanıcı tanımlı komut olarak kullanmaktır.

*\$ find ~ -isim '\*.pdf' -exec tar cvf dosya.tar '{}' '+'*

# Dosyaları Sıkıştırma

- Dosyaları sıkıştırmak, dosyadaki gereksiz verileri kaldıran bir araçla dosyanın boyutunu küçültme prosedürüdür.
- Sıkıştırılmış dosyalar için çoklu formatlar vardır,
  - **‘.gz’** → gzip ve gunzip araçlarını kullanma
    - \$ gzip <dosya>** dosya, dosya.gz içine sıkıştırılır
    - \$ gunzip <dosya.gz>** dosya.gz, dosyaya çıkarıldı
  - Daha iyi sıkıştırma **‘.bz2’** → **bzip2** ve **bunzip2** kullanma
    - \$ bzip2 <dosya>**
    - \$ bunzip2 <dosya.bz2>**
  - Daha da iyi sıkıştırma **‘.lzma’** → **lzma** ve **unlzma** kullanma
    - \$ lzma <dosya>**
    - \$ unlzma <dosya.lzma>**
- Bu metotlar için not olarak,
  - Bir dizin ağacında bir dosya grubunu sıkıştırabilmek için tek bir dosya ile değil, bir grup dosya ile uğraşılır, önce bunları arşivlemeniz gerekir.
  - Dosyayı, sıkıştırılmış / düzleştirilmiş haliyle değiştirirler. Orijinal dosya silindi

# Sıkıştırılan Metin Dosyalarına Erişme

- Bir metin dosyası bir .gz biçiminde sıkıştırılmışsa, yine de bir dizi araç (Z araçları) kullanarak dosyanın sıkıştırılmasına gerek kalmadan erişebiliriz.

***\$ gzip dosyam.txt***

Dosya içeriğini görüntülemek için zcat komutunu kullanıyoruz (sıkıştırılmamış metin dosyaları için cat komutuna benzer)

***\$ zcat dosyam.gz***

- Dosya uzunsa ve sayfayı sayfaya göre görüntülememiz gerekiyorsa, zmore ve zless komutlarını kullanabilirsiniz (sıkıştırılmamış metin komutları için daha fazla ve daha az komutlara benzer şekilde)

***\$ zless dosyam.gz***

***\$ zmore dosyam.gz***

- Sıkıştırılmış metin belgesinin içinde zgrep ve zegrep komutlarıyla da arama yapabiliriz (grep ve egrep'e benzer)

# Sıkıştırma Dosyaları hakkında daha fazla bilgi

- Sıkıştırmanın dosyaların izinlerini ve zaman damgasını koruduğunu unutmayın. Sıkıştırılmış dosyanın sıkıştırmasını açtıktan sonra, orijinal dosyanın aynı izinleri ve zaman damgaları ile sonuçlanır.
- Zaten sıkıştırılmış bir dosyayı sıkıştırmaya çalışmak yararlı değildir, çünkü muhtemelen boyutunu artıracaktır.
  - Azaltma zaten ilk seferlik olduğundan dosya boyutunda bir azalma yok
  - İkinci sıkıştırma zamanında ek yük verisi (meta veri) eklendi



# Arşiv ve Sıkıştırmanın Karıştırılması

- Arşivleme, bir ağaç yapısında düzenlenmiş birden fazla dosyayı tek bir dosyaya birleştirmek demektir.
- Sıkıştırma, tek bir dosyanın boyutunu küçültmektir. Dolayısıyla, her ikisini de gerçekleştirmek istersek bunu iki adımda yapabiliriz;

***\$ tar cvf pdfler.tar \*.pdf***

(Bu pfler.tar üretir)

***\$ gzip pdfler.tar***

(Bu pdfler.tar.gz üretecektir)

- Gzip'i başka herhangi bir sıkıştırma aracıyla değiştirebiliriz (istenen sıkıştırma formatına bağlı olarak)
- Her iki görevi de (arşivleme ve sıkıştırma) tek bir adımda gerçekleştirebiliriz

# Arşivlemek + Sıkıştırmak için tar kullanımı

**\$ tar <ayarlar> <sıkıştırılmış dosya> <dosya yada klasör>**

**\$ tar <ayarlar> <sıkıştırılmış dosya>**

- Tar komutu için doğru seçenekler grubuyla hem arşivleme hem de sıkıştırma (istediğiniz sıkıştırma formatına)

1. 'z' ekleyin **gzip** yada **gunzip**

2. 'j' ekleyin **bzip2** yada **bunzip2**

3. '--lzma' ekleyin **lzma** yada **unlzma** için

- Örnekler:

***\$ tar cvzf dosya.tar.gz ~/Dökümanlar/\*.doc***

***\$ tar xvzf dosya.tar.gz***

***\$ tar xvjf dosya.tar.bz2***

***\$ tar cvf --lzma dosya.tar.lzma ~/projem/***

***\$ tar tzvf dosya.tar.gz***

# Araçlar (rar Komutu)

RAR aracı, .rar biçim arşiv dosyasına arşivleme + sıkıştırma veya özütleme dosyaları çıkarma için kullanılabilir.

Önce arabirimin kurulması gerekir.

***\$ sudo apt-get install rar***

Rar aracı çok güçlü bir araçtır, yalnızca temellerini anlatacağız. Rar arşivine bir dosya (dosya grubu) eklemek için

***\$ rar a arşivim.rar ~/project/\*.pdf***

- Bu pdf dosyalarını arşivim.rar arşivine ekleyecektir. Arşiv yoksa, oluşturulacaktır Arşiv varsa, bu dosyalar eklenecektir
- Daha fazla dosya eklemeyi durdurmak ve arşivi kilitlemek için

***\$ rar k arşivim.rar***

# RAR Arşiv dosyasına erişim (rar Komutu)

Arşivin içeriğini listelemek için,

***\$ rar l arşiv.rar***

- Bir dosyayı arşivden silmek için:

***\$ rar d arşiv.rar dosya.pdf***

- Özgün hiyerarşiyi koruyamadan geçerli dizindeki dosyaları ayıklamak için (düz dosyalar dizisi, alt klasörler oluşturmadan)

***\$ rar e arşiv.rar***

***\$ rar e arşiv.rar file-1.pdf***

- Dosyaları doğru dosya hiyerarşisinde çıkarmak (dizin yapısını korumak için). Bu, geçerli dizinin içindeki alt dizinleri oluştur.

***\$ rar x arşiv.rar***

***\$ rar x arşiv.rar file-2.pdf***

# Arşivi Yenilemek (Rar Komutu)

- Tüm proje dosyalarınızı arşivlediğimizi varsayalım

***\$ rar a arşiv.rar ./my-project***

Ardından, arşivlenen bazı dosyaları değiştirdik.

Arşivi yeni değiştirilen dosyalarla güncellememiz gerekiyor.

***\$ rar f arşiv.rar*** (sadece yerel dosyaları yenilemek için)

***\$ rar f arşiv.rar \**** (Tüm alt dizinleri de yeniler)

# Diğer Arşiv + Sıkıştırma Araçları (zip / unzip aracı)

- 'Zip' ve 'unzip' araçlarının windows formatı '.zip'
- Bu araçların hem arşivleme hem de sıkıştırma gerçekleştirdiğini unutmayın

***\$ zip -r dosya.zip ~/my-documents/*** (recursive için -r,tüm alt dizinlerde ziplenmesi için -r gereklidir.)

***\$ unzip dosya.zip***

- Not:
- File.zip için bir dizinde 'zip' kullanırsak ve bu dosya zaten mevcutsa, file.zip güncellenecek,
- Yeni dosyalar eklendi
- Değiştirilen dosyalar güncellendi.

*\$ zip dosya.zip ~/my-documents/*

my-documents klasörünün alt dizinleri olmasına rağmen bu komutu çalışmaya çalışsak ne olur?

Herhangi bir zipleme yapılmaz -r kullanarak çalıştırırsak en alt dizinden seçilen en üst dizine kadar ziplenir.

# Dosya Bütünlüğü Denetimi(md5sum Komutu)

**\$ md5sum <Korunacak dosya> > <Bütünleşik Dosya>**

**\$ md5sum -c <Bütünleşik Dosya>**

- Eğer bir dosyamız varsa ve dosyanın zamanla yada transfer ettikten sonra modifiye edilmediğinden yada hasar görmediğinden emin olmamız gerekiyorsa biz onun bütünlüğünü checksum hesabıyla koruyup saklayabiliriz.

***\$ md5sum dosya.txt > checksum-dosya***

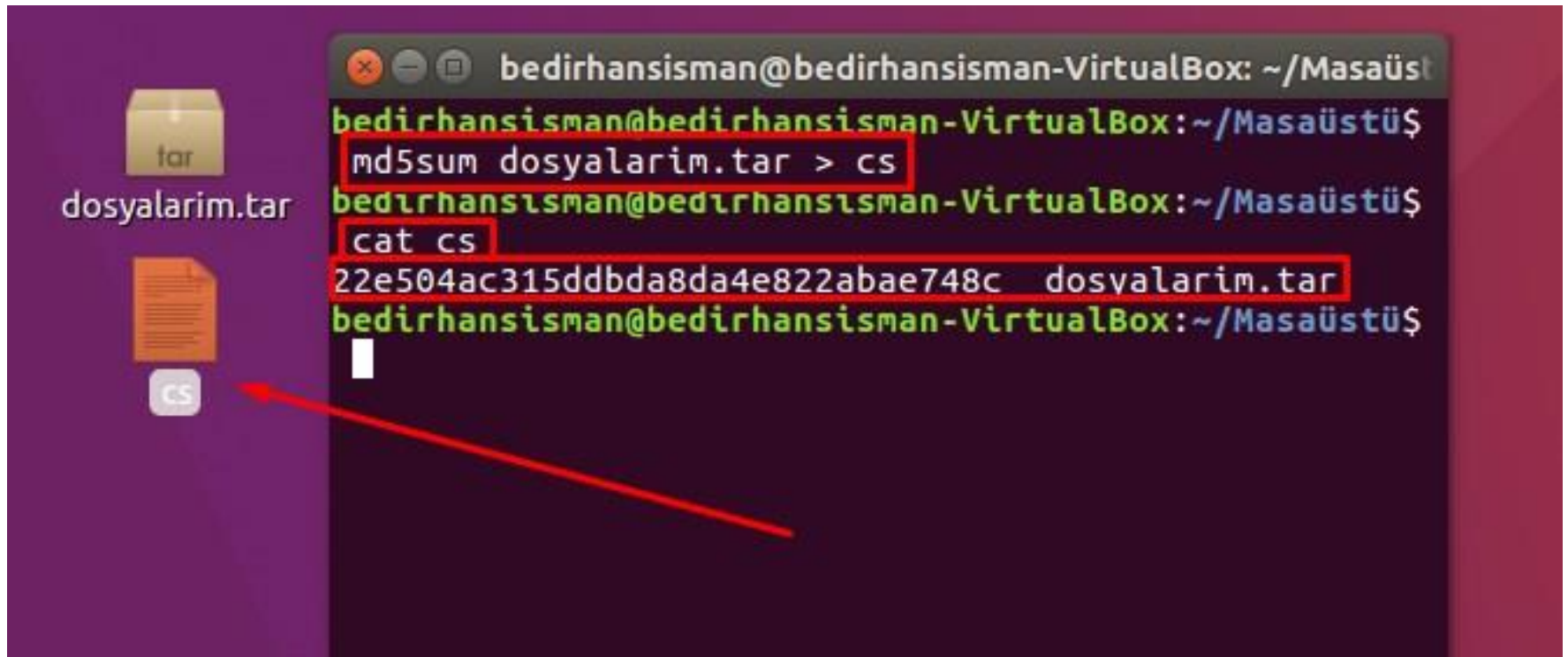
- Daha sonra, onu etkileyebilecek daha sonraki bir aşamada veya bazı prosedürlerden sonra, checksum u tekrar hesaplayabilir ve orijinalle karşılaştırırız

***\$ md5sum -c checksum-dosya***

- Checksum geçerli dosya ile eşleşirse bir durum almamız gerekir
- Sağlama toplamı dosyasının dosya içeriğinin 128 bit'lik bir ileti özeti içerdiğine dikkat edin



# Dosya Bütünlüğünü Denetleme (md5sum Komutu)



```
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü$  
md5sum dosyalarim.tar > cs  
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü$  
cat cs  
22e504ac315ddbda8da4e822abae748c dosyalarim.tar  
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü$
```

The image shows a Linux desktop environment. On the left, a file manager window displays a folder icon labeled 'tar' and a file icon labeled 'dosyalarim.tar'. Below the file icon is a small white icon with the letters 'cs'. A red arrow points from the terminal window's output to this 'cs' icon. The terminal window on the right shows the following commands and output:

# Dosya Bütünlüğünü Çoklu Dosyalar için Kontrol Etme

- Birden fazla dosya için bütünlüğü kontrol etmemiz gerekiyorsa, aşağıdaki yöntemlerden birini kullanabilirsiniz

Korunacak dizin yapısını arşivleyebilir ve arşiv dosyasında md5sum performansı sağlayabiliriz

```
$ rar a marşiv.rar ~/my-project/
```

```
$ md5sum arşiv.rar > csum
```

- Korunacak dosyaları seçmek için kabuk genişletme joker karakterlerini kullanabiliriz

```
$ md5sum * > csum
```

```
$ md5sum *.java > csum
```

- Koruma altındaki dosyaları seçmek için find komutunu kullanabiliriz.

```
$ find ~ -type f -exec md5sum {} + > csum
```

```
$ find ~ -type f -name \*.pdf | md5sum > csum
```

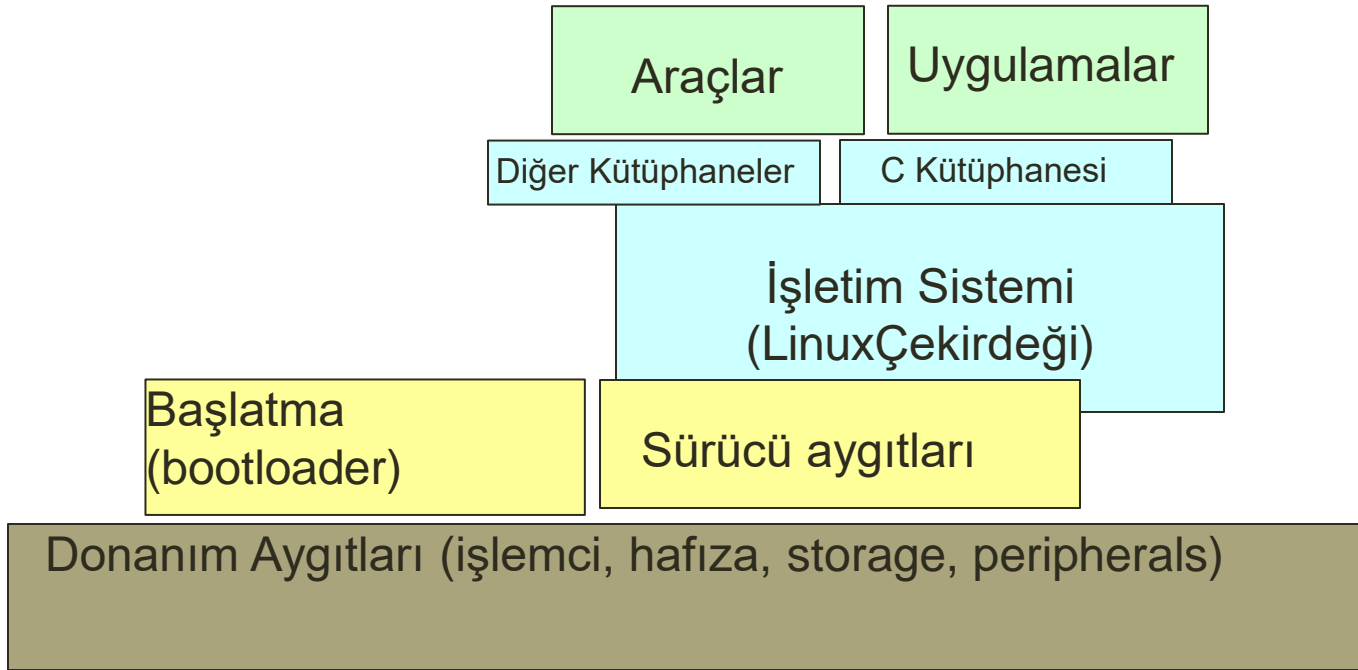
# Birden Çok Dosyanın Bütünlüğünü Kontrol Etme

```
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$ find . -type f -exec md5sum {} + > m
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$ cat m
d41d8cd98f00b204e9800998ecf8427e  ./m
7a069e83f536d0354f40bab5698a6336  ./cs
448bc4e75e813d4ab9bd3327e1481366  ./dosyalar.tgz
424f4e0390567097f8f29a5349cc37fb  ./dosya2
22e504ac315ddbda8da4e822abae748c  ./dosyalarim.tar
78b62ac309a90ebeca5842a9505f7fce  ./dosya1
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$
```



# Aygıtlar ve Aygıt Sürücüleri

# Aygıt Sürücüsü Nedir?



# Aygıt Sürücüsü Nedir ?

- Bir aygıt sürücüsü, bu donanım aygıtının ayrıntılarından işletim sistemini ve kullanıcı uygulamasını izole etmek için bir donanım aygıtının taşınmasından sorumlu bir **kernel modülüdür**.
- Bu şekilde, Linux Çekirdeği ve üstünde çalışan kullanıcı uygulamaları, donanım aygıtının ayrıntılarını bilmek zorunda değildirler.
- Örneğin, kullanıcı uygulaması, bir iç sabitleme işlemi tamamen farklı olmasına rağmen, sabit disk (herhangi bir türde) ve bir flash sürücüyü aynı şekilde ele alır.
- Bu, hergün yeni donanım aygıtlarının piyasaya sürülmesine yol açar ve Linux Çekirdeği, onları çekirdeği değiştirmeye gerek kalmadan uygun aygıt sürücüsünü kullanarak destekleyebilir.

# Doğru fakat Tam Değil....

- Bu tanım yalnızca bazı aygıt sürücülerini temsil eder , hepsini değil.
- Aygıt sürücülerini, sabit donanım desteğiyle sınırlı değildir.
- Bazı aygıt sürücülerini komple bir yazılım işlevi yerine getirir ve herhangi bir donanım aygıtını etkileşimini gerçekleştirmez
- Örnekler, bu aygıtları destekleyen aygıt sürücülerini,
  - Rastgele Sayı Üretici (`/dev/random` & `/dev/urandom`)
  - Sıfır Üretici (`/dev/zero`)
  - Çıkış Eyyesi (`/dev/null`)
- Aygıtların ve Aygıt Sürücüsünün daha genel tanımını anlamamız gerekir

# Linux'un Geniřletilebilir Mimarisi

- Linux çekirdeęi, farklı hedeflere sahip binlerce geliřtirici tarafından sürekli geliřtirilmektedir.
- Bir sürü zevk ve geliřmiř özellikleri destekliyor.
- Bu özelliklerin tümü, bir çok kullanıcı için gerekli deęildir.
- Ayrıca, tonlarca donanım aygıtı için destek vardır. (yazıcılar, tarayıcılar, kameralar, depolama aygıtları, vb.)
- Çekirdeęi tüm özellikleriyle inşa etmek ve donanım aygıtı desteęi çok büyük bir görüntü (gereksiz yere) ve çeliřkili işlemlere sahip bazı durumlarda sonuçlanacaktır.
- Böyle büyük bir görüntü ile Linux gömülü sistemler de dahil olmak üzere birçok sistem üzerinde çalışamayabilir.
- Ve çok fazla kaynaęa (yine gereksiz yere) ihtiyaç duyacak.



# Linux'un Geniřletilebilir Mimarisi

- Bu sorunun özümü, geliřmiř özelliklerin çoğunun ve donanım aygıtlarının işlevselliđi desteklediđi isteđe bađlı özellikler olarak kabul edilir
- ekirdek oluřtururken, her isteđe bađlı özellik için belirlemek için bir yapılandırma aşaması geçirdik:
  - **Etkinleřtir**: Bu özelliđi ekirdek imajına ekleyelim statik bir řekilde yapın
  - **Disable**: Bu özelliđin kodunu atlamak için ekirdeđi desteklemiyorsunuz
  - **Modül**: Kod ayrı bir modül olarak inşa edilmiř ve ayrı bir ikili (\*ko) Ayrıca, ekirdek modülleri, ekirdek yapısından bađımsız olarak geliřtirilebilir ve inşa edilebilir ve gelecekte yük / boşaltma için ekirdek ağacına eklenebilir.
- ekirdek görüntüsü daha küçük tutmak ve gerektiđinde bunları ekleme veya ıkarma esnekliđini eklemek için ekirdek özellikleri "ekirdek Yüklenebilir Modüller" olarak oluřturulabilir.

# Çekirdek Yüklenebilir Modül (KLM)

- Bir KLM, Linux çekirdeğini oluştururken ilave bir adım olarak veya tamamen ayrı bir prosedür olarak oluşturulabilen iki bitlik (\*.ko) dosyadır.
- KLM'nin Linux ağacında bulunduğu yer:  
**`/lib/modules/{kernel release}/`**  
KLM'ler çalışma zamanında Linux çekirdeğine yüklenebilir veya yüklenemez (Çekirdek restartından sonra)
- KLM'yi kullanmak, aşağıdakileri mümkün kılar:
  - Yeni bir KLM'nin her tanıtımında çekirdeğin yeniden oluşturulmasına gerek yok
  - KLM'yi her yükleyip / çıkarmamız gerektiğinde çekirdeği yeniden başlatmaya gerek yok
  - The KLM çekirdekte çalışır.

# Çekirdek Modüllerini Yükleme

- Çekirdek Modülleri aşağıdaki gibi etkinleştirilebilir;
  - Çekirdek modülleri kullanıcı komutlarına bağlı olarak çekirdekte yüklenebilir (bunu daha sonra bu derste anlatacağız)
  - Linux çekirdeği, yeni donanım aygıtlarının otomatik keşifini desteklemektedir ve bu, ilişkili çekirdek modüllerinin otomatik yüklenmesine neden olabilir (kullanarak Çekirdekte **udev subsystem** ).
- Örnek olarak, bir USB flash sürücü takarken kullanılır.
- Linux çekirdeği başlangıçta yüklemek için çekirdek modüllerinin bir listesini (bazı başlatma komut dosyaları aracılığıyla) içerir.
- Bazı Çekirdek Modülleri, Linux Çekirdeğinin önyükleme işleminde gereklidir ve bu nedenle statik olarak çekirdekte bulunurlar
- Diğer Modüller sistemin çalışması için gereklidir ve onları çekirdekte statik olarak inşa etmeye karar verilir (çekirdek resminin bir parçası haline geldikleri için kaldırılamazlar)

# Önyüklemede Başlayacak Modülleri Seçme (/ etc / modules)

- / Etc / modules dosyası, çekirdek başlangıçta başlatılacak modüllerin listesini içerir

```
bedirhansisman@bedirhansisman-VirtualBox: /etc
bedirhansisman@bedirhansisman-VirtualBox:/etc$ cat modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
bedirhansisman@bedirhansisman-VirtualBox:/etc$
```

# KLM Yardımcı Programları

# Bir Modül Ekleme (insmod Komutu)

**\$ insmod <modül Dosyası>**

**\$ insmod <modül Dosyası> <argümanlar>**

- Bu komut çekirdeğe bir modül ekler;

***\$ sudo insmod /lib/module/`uname -r`/my-driver.ko***

- Modülün tam yolunun belirtilmesi gerektiğini unutmayın.
- Modüle parametreleri iletebilirsiniz, bu parametreler modül kodu ile işlenir.

***\$ sudo insmod /lib/module/`uname -r`/my-device.ko debug\_enable=1***

- Bazen, bir çekirdek modülü başka bir modülün önce yüklenmesini ister, kullanıcı bu bağımlılığa dikkat etmeli, aksi takdirde insmod komutu başarısız olabilir.

# Bir Modülün Kaldırılması (rmmod Komutu)

**\$ rmmod <çekirdek modülü>**

- Bu komut, çekirdek modülünü çekirdekten kaldırır.

***\$ sudo rmmod my-driver***

- Artık Linux ağacındaki ikili dosyasını değil, çekirdeğin çekirdek modülünü ele aldığımız için hiçbir yol belirtilmedi. Sadece modül adı

# Bir Modülü Yükleme / Boşaltma (modprobe Komutu)

**\$ modprobe <kernel module name >**

**\$ modprobe -r <kernel module name>**

- Bu komut, hem insmod hem de rmmmod Komutlarının yerini alır

*\$ sudo modprobe my-driver*

*\$ sudo modprobe -r my-driver*

- Çok daha zengin bir komuttur, çünkü
  - Modül bağımlılıklarını anlama ve istenen modülü yüklemeye önce (önceden yüklenmemişse) tüm modülleri yükleme yeteneğine sahiptir.
  - Bir modülü kaldırırken, diğer modüller tarafından kullanılmayan bağımlılıklar da kaldırılacaktır.
  - Ayrıca, modülleri bulmak için bir araç vardır; bu nedenle, modül dosyasının tam yolunu belirtmeniz gerekmez. Modül adı yeterli olmalıdır
- Modprobe komutu ayrıca, yapılandırma dosyalarında varsayılan parametre seçeneklerini ayarlama ve belirtilen modülü yüklerken bu seçenekleri kullanma gibi diğer özel özellikleri de gerçekleştirir
  - Varsayılan modül parametreleri,
    - */etc/modprobe.conf*
    - */etc/modprobe.d/\*.conf*



# Yüklenen KLM'leri Listeleme (lsmod Komutu)

## \$ lsmod

```
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$ lsmod | more
```

Module	Size	Used by
btrfs	1056768	0
xor	24576	1 btrfs
raid6_pq	102400	1 btrfs
ufs	73728	0
qnx4	16384	0
hfsplus	106496	0
hfs	57344	0
minix	40960	0
ntfs	102400	0
msdos	20480	0
jfs	184320	0
xfs	1048576	0
libcrc32c	16384	1 xfs
vboxvideo	49152	2
snd_intel8x0	40960	2
crct10dif_pclmul	16384	0
crc32_pclmul	16384	0
snd_ac97_codec	131072	1 snd_intel8x0
joydev	20480	0
ttm	102400	1 vboxvideo
ghash_clmulni_intel	16384	0
ac97_bus	16384	1 snd_ac97_codec
aesni_intel	167936	0
drm_kms_helper	167936	1 vboxvideo
snd_pcm	110592	2 snd_ac97_codec,snd_intel8x0
drm	368640	5 vboxvideo,ttm,drm_kms_helper
aes_x86_64	20480	1 aesni_intel
input_leds	16384	0
serio_raw	16384	0
snd_seq_midi	16384	0
snd_seq_midi_event	16384	1 snd_seq_midi
snd_rawmidi	32768	1 snd_seq_midi
lrw	16384	1 aesni_intel
qlue helper	16384	1 aesni_intel

# Çekirdek Modülü bilgisini görüntüleme (modinfo Komutu)

## \$ modinfo <kernel module name>

- Bu komut, modül hakkında bilgi gösterir

*\$ modinfo my-driver*

- Şunları içerir,
  - **.ko** dosyası için tam yol
  - Yazar
  - Lisans Tipi
  - Modül işlevinin açıklaması
  - Diğer modüllere bağımlılıklar
  - Geçerli Parametreler

# modinfo

```
bedirhansisman@bedirhansisman-VirtualBox: ~/Masaüstü
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$ modinfo skge
filename:      /lib/modules/4.8.0-36-generic/kernel/drivers/net/ethernet/marvell/
skge.ko
version:      1.14
license:      GPL
author:       Stephen Hemminger <shemminger@linux-foundation.org>
description:  SysKonnect Gigabit Ethernet driver
srcversion:   725826B5E8C66128EE2D109
alias:        pci:v00001737d00001032sv*sd00000015bc*sc*i*
alias:        pci:v00001737d00001064sv*sd*bc*sc*i*
alias:        pci:v00001371d0000434Esv*sd*bc*sc*i*
alias:        pci:v000011ABd00005005sv*sd*bc*sc*i*
alias:        pci:v000011ABd00004320sv*sd*bc*sc*i*
alias:        pci:v00001186d00004302sv*sd*bc*sc*i*
alias:        pci:v00001186d00004C00sv*sd*bc*sc*i*
alias:        pci:v00001186d00004B01sv*sd*bc*sc*i*
alias:        pci:v00001148d00004320sv*sd*bc*sc*i*
alias:        pci:v00001148d00004300sv*sd*bc*sc*i*
alias:        pci:v000010B7d000080EBsv*sd*bc*sc*i*
alias:        pci:v000010B7d00001700sv*sd*bc*sc*i*
depends:
intree:       Y
vermagic:     4.8.0-36-generic SMP mod_unload modversions
parm:        debug:Debug level (0=none,...,16=all) (int)
bedirhansisman@bedirhansisman-VirtualBox:~/Masaüstü$
```

# Aygıt Sürücüleri Nedir?

- Bir aygıt sürücüsü bir çekirdek modülüdür.
- Çekirdek imajında statik olarak oluşturulabilir veya dinamik olarak yüklenebilir
  - **Udev** aracılığıyla otomatik olarak yüklenir
  - Kullanıcı komutlarıyla talep üzerine (modprobe)
  - Başlatma komut dosyalarında önyükleme zamanı
- Aygıt sürücüsü kendisini bir sayı ile sisteme kaydeder (**büyük numara** olarak adlandırılır)
- Aygıt sürücüsü çekirdekte çalışır ve gerekli işlevleri yerine getirir (donanım kesmeleri ile etkileşim kurar, donanım aygıtlarıyla iletişim kurar, Linux Çekirdek yapılarına erişir, diğer aygıt sürücüleri ile iletişim kurar, vb.)
- Aygıt sürücüsü bir veya daha fazla aygıtı destekler

# Aygıt Sürücülerini Nedir?

- Bir aygıt, / **dev**'de bulunan özel bir dosyadır (ayrıca aygıt düğümü olarak da adlandırılır)
- Dosya gerçek bir dosya değil. Diskte bulunmaz, yalnızca yer tutucudur (sadece bellekte yaşar)
- Her aygıt, aygıt oluşturma saatinde belirli bir aygıt sürücüsüne bağlanır. Kullanıcı düzleminin aygıt sürücüsüne normal dosya gibi erişmesine olanak tanıyan bir soyutlamadır.
- Bu, Linux çekirdeğinin "Her şey bir dosyadır" ifadesini oluşturmak için kullandığı yöntemdir, Böylece, şunları yapabilirsiniz:
- Aygıt sürücüsünden okuyarak aygıt sürücüsünden bilgi isteyin.
- Aygıt dosyasına yazarak bilgi gönderin.
- Bir veya daha fazla aygıt aynı aygıt sürücüsüyle bağlanabilir (aynı Binbaşı Numarasına sahip olmalı), ancak her birinin Benzersiz bir numara seçin (Küçük Numara denir)

# Major & Minor Sayılar

- Her Cihaz bir çift sayıya sahip olacak,
  - Major Sayı:
    - Bu, bu aygıtla ilişkili aygıt sürücüsünü tanımlayan sayıdır.
    - Ana sayı, çekirdekte yüklendiğinde aygıt sürücüsü tarafından saklanır ve kaydedilir
  - Minor Sayı:
    - Bu sayı aygıt dosyasıyla ilişkilendirilmiştir.
    - Bu aynı aygıt sürücüsüyle ilişkili farklı aygıt dosyalarını ayırt etme yöntemidir.
    - Ayrıca, hangi aygıtın onunla iletişim kurduğunu aygıt sürücüsü dosyasının tanıma biçimidir

# Major & Minor Sayılar

bedirhansisman@bedirhansisman-VirtualBox: ~

```
bedirhansisman@bedirhansisman-VirtualBox:~$ ls -l /dev | more
toplam 0
crw----- 1 root root      10  235 May 19 02:24 autofs
drwxr-xr-x  2 root root      620 May 19 02:24 block
drwxr-xr-x  2 root root       80 May 19 02:24 bsg
crw-rw----  1 root disk     10  234 May 19 16:42 btrfs-control
drwxr-xr-x  3 root root       60 May 19 02:39 bus
lrwxrwxrwx  1 root root        3 May 19 02:24 cdrom -> sr0
drwxr-xr-x  2 root root    3580 May 19 16:42 char
crw-----  1 root root        5   1 May 19 02:24 console
lrwxrwxrwx  1 root root       11 May 19 02:24 core -> /proc/kcore
drwxr-xr-x  2 root root       60 May 19 02:39 cpu
crw-----  1 root root      10  59 May 19 02:24 cpu_dma_latency
crw-----  1 root root     10  203 May 19 02:24 cuse
drwxr-xr-x  5 root root     100 May 19 02:24 disk
drwxr-xr-x  2 root root       80 May 19 02:24 dri
lrwxrwxrwx  1 root root        3 May 19 02:24 dvd -> sr0
crw-----  1 root root     10  61 May 19 02:24 ecryptfs
crw-rw----  1 root video    29  0 May 19 02:24 fb0
lrwxrwxrwx  1 root root      13 May 19 02:24 fd -> /proc/self/fd
crw-rw-rw-  1 root root        1   7 May 19 02:24 full
crw-rw-rw-  1 root root     10  229 May 19 02:24 fuse
crw-----  1 root root    247  0 May 19 02:24 hidraw0
crw-----  1 root root     10  228 May 19 02:24 hpet
drwxr-xr-x  2 root root        0 May 19 02:24 hugepages
crw-----  1 root root     10  183 May 19 02:24 hwrng
lrwxrwxrwx  1 root root      25 May 19 02:24 initctl -> /run/systemd/initctl/fifo
drwxr-xr-x  4 root root     320 May 19 02:24 input
crw-rw-rw-  1 root root        1   11 May 19 02:24 kmsc
```

# Cihaz Sınıflandırması

- Cihazların iki adet Ana sınıfı Var:
  - **Karakter Aygıtları**
    - Seri veri akışları
    - Okuma ve Yazma işlemleri seri bir şekilde yapılır
    - Örnekler:
      - Klavye
      - Mouse
      - */dev/ones*
      - */dev/tty1*
      - */dev/ttyS1*
    - "C" türündeki bir dosya tarafından temsil edilir
    - En yaygın cihazlar
  - **Blok Aygıtları**
    - Okuma ve Yazma, Bloklar
    - Bloklar adreslenebilir ve dolayısıyla sıralı olmayan bir şekilde okuyabilirler
    - Örnekler:
      - Depolama Aygıtları (hard disk, flash)
      - */dev/sda1*
      - "B" türündeki bir dosya tarafından temsil edilir



# Cihaz Sınıflandırması

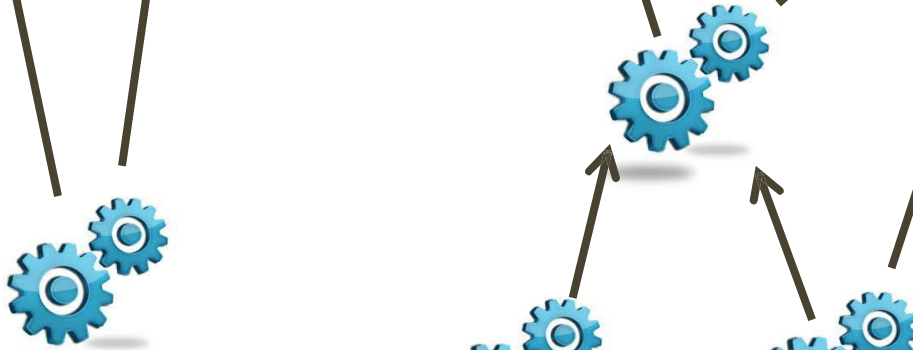
```
bedirhansisman@bedirhansisman-VirtualBox:~$ ls -l /dev | more
toplam 0
crw----- 1 root root      10, 235 May 19 02:24 autofs
drwxr-xr-x 2 root root      620 May 19 02:24 block
drwxr-xr-x 2 root root        80 May 19 02:24 bsg
crw-rw---- 1 root disk     10, 234 May 19 16:42 btrfs-control
drwxr-xr-x 3 root root        60 May 19 02:39 bus
lrwxrwxrwx 1 root root         3 May 19 02:24 cdrom -> sr0
drwxr-xr-x 2 root root    3580 May 19 16:42 char
crw----- 1 root root         5,  1 May 19 02:24 console
lrwxrwxrwx 1 root root        11 May 19 02:24 core -> /proc/kcore
drwxr-xr-x 2 root root        60 May 19 02:39 cpu
crw----- 1 root root     10,  59 May 19 02:24 cpu_dma_latency
crw----- 1 root root     10, 203 May 19 02:24 cuse
drwxr-xr-x 5 root root     100 May 19 02:24 disk
drwxr-xr-x 2 root root        80 May 19 02:24 dri
lrwxrwxrwx 1 root root         3 May 19 02:24 dvd -> sr0
crw----- 1 root root     10,  61 May 19 02:24 ecryptfs
crw-rw---- 1 root video    29,  0 May 19 02:24 fb0
lrwxrwxrwx 1 root root        13 May 19 02:24 fd -> /proc/self/fd
crw-rw-rw- 1 root root         1,  7 May 19 02:24 full
crw-rw-rw- 1 root root     10, 229 May 19 02:24 fuse
crw----- 1 root root    247,  0 May 19 02:24 hidraw0
crw----- 1 root root     10, 228 May 19 02:24 hpet
drwxr-xr-x 2 root root         0 May 19 02:24 hugepages
crw----- 1 root root     10, 183 May 19 02:24 hwrng
lrwxrwxrwx 1 root root        25 May 19 02:24 initctl -> /run/systemd/initctl/fifo
drwxr-xr-x 4 root root     320 May 19 02:24 input
crw-r--r-- 1 root root         1, 11 May 19 02:24 kmsg
drwxr-xr-x 2 root root        60 May 19 02:39 lightnvm
lrwxrwxrwx 1 root root        28 May 19 02:24 log -> /run/systemd/journal/dev-log
brw-rw---- 1 root disk         7,  0 May 19 02:24 loop0
brw-rw---- 1 root disk         7,  1 May 19 02:24 loop1
brw-rw---- 1 root disk         7,  2 May 19 02:24 loop2
brw-rw---- 1 root disk         7,  3 May 19 02:24 loop3
brw-rw---- 1 root disk         7,  4 May 19 02:24 loop4
brw-rw---- 1 root disk         7,  5 May 19 02:24 loop5
```

# Donanım Aygıtı, Aygıt Sürücüsü ve Aygıt Dosyası

Kullanıcı  
Düzlemi



Çekirdek



Donanım



# Bir Aygıt dosyası oluşturma (mknod Komutu)

**\$ mknod <aygıt dosya adı> <c | b> <Major> <Minor>**

**\$ mknod -m <izinler> <aygıt dosya adı> <c | b> <Major> <Minor>**

- Bu komut, bir aygıt dosyası oluşturur
- Aygıtın dosya adı oluşturulacak dosyadır. / Dev'de bulunmalıdır.
- Cihaz sınıfı ya bir "Karakter" ya da "Blok" aygıtıdır.
- Major Numarası, bağlanacak aygıt sürücüsünün numarasıdır. Mayor Numara, bu aygıt dosyasının numarasıdır (bağlı olan tüm aygıtlarda eşsiz olmalıdır. Bu aygıt sürücüsü)
- Cihazı normal bir dosya olarak ele alabilirsiniz
- Sadece okunabilir, sadece yazılabilir veya okunabilir-yazılabilir
- Herhangi bir dosya gibi kullanarak dosyadan okuyabilirsiniz (kedi kullanın veya giriş yönlendirme)
- Herhangi bir dosya gibi bir aygıt dosyasına yazabilirsiniz (örneğin, yankı veya çıktı yönlendirme kullanın)
- Bir aygıt düğümü oluşturmak için

***\$ mknod /dev/my-new-device c 235 1***

***\$ mknod -m 666 /dev/my-new-device c 235 1***

# Yaygın Kullanılan Cihazlara Örnekler

# SCSI Disk Sürücü Aygıtları

- Tüm SCSI Disk sürücü cihazları (SCSI Sabit diskler için kullanılanlar gibi), sd sürücüsüne bağlıdır ve sd sürücüsünün Major sayısı 8'dir.
- Cihaz dosyalarının adlandırılması:
- ***/dev/sd[a-z][1-15]***  
Sonunda ki numara bölüm numarasıdır  
Numarasız aygıt dosyası tüm diski belirtir•

```
bedirhansisman@bedirhansisman-VirtualBox: ~  
bedirhansisman@bedirhansisman-VirtualBox:~$ ls -l /dev/sd*  
brw-rw---- 1 root disk 8, 0 May 19 02:24 /dev/sda  
brw-rw---- 1 root disk 8, 1 May 19 02:24 /dev/sda1  
brw-rw---- 1 root disk 8, 2 May 19 02:24 /dev/sda2  
brw-rw---- 1 root disk 8, 5 May 19 02:24 /dev/sda5  
bedirhansisman@bedirhansisman-VirtualBox:~$
```

# TTY Terminalleri & Seri Arayüzler

- Tüm Linux sanal terminallerinin cihaz adları var
- `/dev/tty[n]` (örnek `/dev/tty1`)
- Tüm Linux Seri Arabirimleri, aygıt adlarını kullanır
- `/dev/ttyS[n]` (örnek `/dev/ttyS1`)

# Aygıt Yardımcı Programları

# Seri Arabirimlerin Konfigürasyonu (setserial Komutu)

**\$ setserial <options> <serial device file> <parameters to set>**

- Bu komut, seri portların konfigürasyonunu okumak / yazmak için kullanılır.
- Şunları İçerir,
  - Seri arabirim tarafından kullanılan G / Ç bağlantı noktasının ayarlanması
  - Seri arabirim tarafından kullanılan IRQ kanalının ayarlanması
  - Baud hızı
  - Diğer Bilgiler
- Örnek:

***\$ sudo setserial -a /dev/ttyS1***

***\$ sudo setserial /dev/ttyS1 baud\_base 115200***



# USB Donanım Aygıtlarını Listeleme (lsusb Komutu)

## \$ lsusb

- Bu komut, USB Donanım aygıtlarını ve bunlarla ilgili bilgileri listeler

**\$ lsusb** (USB aygıtlarını listele)

**\$ lsusb -a** (USB aygıtlarını ve bütün bilgilerini listele)

**\$ lsusb -t** (USB Aygıtlarını ağaç hiyerarşi yapısında listele)

bedirhansisman@bedirhansisman-VirtualBox: ~

```
bedirhansisman@bedirhansisman-VirtualBox:~$ lsusb -tv
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ohci-pci/12p, 12M
   |__ Port 1: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 12M
bedirhansisman@bedirhansisman-VirtualBox:~$
```

# PCI Donanım Aygıtlarını Listeleme (lspci Komutu)

## \$ lspci

- Bu komut, PCI Donanım aygıtlarını ve bunlarla ilgili bilgileri listeler

*\$ lspci*

*\$ lspci -tv* (Ayrıntılı bilgi içeren ağaç hiyerarşik yapısı)

```
bedirhansisman@bedirhansisman-VirtualBox:~$ lspci -tv
-[0000:00]-+-00.0 Intel Corporation 440FX - 82441FX PMC [Natoma]
      +-01.0 Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
        +-01.1 Intel Corporation 82371AB/EB/MB PIIX4 IDE
          +-02.0 InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
            +-03.0 Intel Corporation 82540EM Gigabit Ethernet Controller
              +-04.0 InnoTek Systemberatung GmbH VirtualBox Guest Service
                +-05.0 Intel Corporation 82801AA AC'97 Audio Controller
                  +-06.0 Apple Inc. KeyLargo/Intrepid USB
                    +-07.0 Intel Corporation 82371AB/EB/MB PIIX4 ACPI
                      \-0d.0 Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode]
bedirhansisman@bedirhansisman-VirtualBox:~$
```

# Verileri Kopyala ve Dönüştür (dd Komutu)

**\$ dd if=<source> of=<destination> <options>**

- Bu komut bir dosyadaki verileri kopyalar, gerekli tüm dönüşümleri gerçekleştirir ve başka bir dosyayı depolar
- Özellikle, aygıt dosyalarını kaynak, hedef ya da her ikisi olarak kullanabileceği çok kullanışlı bir komuttur
- Bu komutu kullanırken dikkatli olmanız gerekir. Makinenizdeki bütün bir bölümü veya sürücüyü silebilirsiniz
- Örnek:

***\$ dd if=/dev/urandom of=~ /random-data-file bs=4 count=1000***

***\$ dd if=/dev/sr0 of=/my-file.iso bs=2048 conv=noerror,sync***

***\$ dd if=/dev/sda of=~ /disk.img***

***\$ dd if=/dev/sda of=/dev/sdb***

# Dosya Dökümü (od komutu)

**\$ od <Options> <file>**

- Bu komut, dosyaları farklı biçimlerde döker.

- Örnekler:

***\$ od -x file.img*** (Dosyayı onaltılı biçimde çıktılar)

***\$ od -s file.img*** (Dosyayı ondalık biçimde çıktılar)

# 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/>