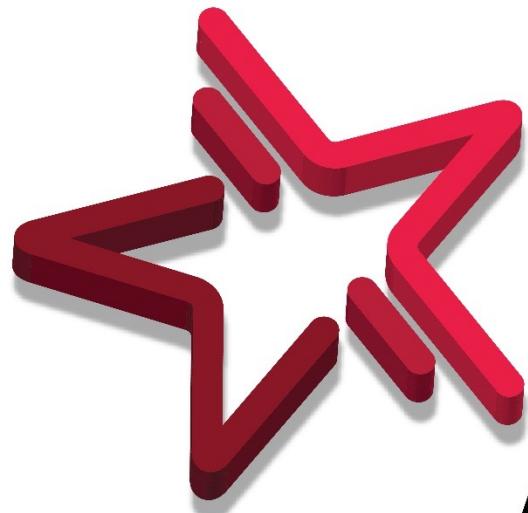




- Pardus



Duygu ÖLmez / PARDUS ve Ürün Ailesi Yazılım Geliştirme Takım Liderliği Yazılım Mühendisi



HAVELSAN, is an establishment of  
Turkish Armed Forces Foundation.

Bu belgenin içeriği HAVELSAN'ın fikri mülkiyetindedir. HAVELSAN'ın izni olmadan bu belgenin tamamını veya bir kısmını bir başkasına kopyalamak ve vermek yasaktır.

TASNİF DİŞİ

## Yazılımin Özgürlüğü Kavramı?

Yazılım kullanıcısının, yazılım üzerinde dağıtma, değiştirme, inceleme gibi haklara sahip olduğu yazılımdır.

- ★ **Özgürlük 0** - İstenen her amaca uygun olarak uygulamayı **çalıştırabilmek**,
- ★ **Özgürlük 1** - Programın çalışmasını inceleyebilmek ve kendi ihtiyaçlarına uygun biçimde **değiştirebilmek**,
- ★ **Özgürlük 2** - Programın kopyalarını **dağıtabilmek**,
- ★ **Özgürlük 3** - Program geliştirebilmek ve bu geliştirmeleri **halka açarak** tüm camianın faydalananabilmesini sağlamak.

Bir yazılım, eğer yazılımın kullanıcıları bu özgürlüklere sahipse özgür yazılımdır. Yani, kopyalarını değiştirerek ya da değiştirmeden, **Ücretli** ya da **Ücretsiz**, herhangi birine dağıtmaya serbest olmalısınız. Bu anlamda özgür olmak demek, tüm bunları yapmak için izin istemek zorunda olmamanız demektir.



## Yazılım Lisansları

- ★ Özgür yazılımların tek çeşit lisansı yok
- ★ Farklı kullanımlara özelleşmiş çeşitli lisanslar var
- ★ Çoğu lisans aslında birbirinine çok benzer
- ★ **GPL: (Genel Kamu Lisansı)**
  - Özgür yazılımların %66'sı tarafından tercih ediliyor.
  - Yazılımin özgür olmasının yanı sıra, özgürlüğünü sürdürmesini de garanti altına alır.
- ★ **LGPL: (Kısıtlı Genel Kamu Lisansı)**
  - GPL'den farklı olarak, yazılımin özgür olmayan yazılımlar tarafından da kullanılabilirliğine olanak tanıyor.
- ★ **Apache Lisansı**
  - GPL'e ek olarak, patentlere yönelik belirli kısıtlamalar içeriyor.
- ★ **Ve daha bir çok Lisans bulunmakta**  
<https://www.gnu.org/licenses/license-list.html>  
<http://ozgurlisanslar.org.tr/>

## ||||| *Özgür yazılım vs. Açık Kaynak*

- ★ Açık kaynak kodlu bir yazılım, özgür olmayabilir.
- ★ Özgür bir yazılım, açık kaynak kodludur.

### ★ **Açık kaynak**

Teknik avantajlara odaklanır. Yazılımı açık kaynak olarak geliştirmek teknik olarak daha başarılı bir model olduğu için tercih edilir.

### ★ **Özgür yazılım**

Teknik avantajların yanı sıra işin toplumsal boyutunu ve "özgürlük" kavramını da ele alır.

## ||||| *Programlama Dilleri ve Araçları*

### Programlama Dilleri



### Yazılım Geliştirme Araçları



**Jenkins**



## ||||| *Programlama Dilleri ve Araçları*

### ★ **Git**

Günümüzde yazılan bir uygulamanın sürüm takibi yazılım geliştirme sürecinin en önemli parçasını oluşturmaktadır. Bu noktada dağıtık bir sürüm takip sistemi olan git ve web tabanlı sunucu ortamı sağlayan GitLab yazılım geliştirme sürecinin en önemli aracı olmaktadır.

### ★ **RedMine**

Yazılım geliştirme sürecini takip etmek için bir çok araç ve yöntem bulunmaktadır. Bunların içinde iş takibi, ekip çalışması için büyük önem taşımaktadır. RedMine web tabanlı bir araç olarak bu konuda rüştünü ispatlamış yaygın kullanılan bir yazılımdır.

### ★ **Jenkins**

Her ne kadar derleme kavramı programlama dillerine ait olarak görünse de uygulama kodlarının sadece makine diline çevrilmesi değil, uygulamanın yüklenmeye hazır paket haline getirilmesini de kapsar. Bu işlemin parçaları içinde testlerin çalıştırılması, statik kod analizi yapılması, hatta test ortamlarına uygulamanın yüklenmesine kadar bir çok işlevi yerine getirebilen Jenkins günümüzde en yaygın kullanılan düzenli derleme aracı konumundadır.

### ★ **SonarQube**

Kod kalitesi ve uygulamanın sürdürülebilir olması açısından, yazılım ekibinin belirlenmiş yazım kurallarına uyması gerekmektedir. SonarCube ile bu yazım kuralları otomatik denetlenmekte ve uygunsuz durumlar önerilerle birlikte raporlanmaktadır.

## ||||| **Programlama Dilleri ve Araçları**

### ★ **Java**

Java programlama dili 1995 yılında, geliştirilen bir uygulamanın üzerinde uğraşılmadan farklı platformlarda çalışabilmesini ve yazılımcıların hata yapma ihtimalini azaltma amaçlarıyla geliştirilmiş, günümüzde ise en yaygın kullanılan programlama dillerinin başında gelmektedir. Mobil cihazlardan masaüstüne, sunuculardan özel geliştirilmiş cihazlara kadar bir çok farklı platformda ve işletim sisteminde çalışabilen java uygulamaları, günümüzde özellikle "kurumsal uygulamalar" için en çok tercih edilen programlama dilidir.

### ★ **PHP**

Web uygulamaları ve sayfaları için ilk kullanılan programlama dillerinden biridir. Hedefi web sayfalarının daha kolay yazılmasını sağlamak olan PHP, günümüzde özellikle web sayfaları ve uygulamaları için en çok tercih edilen programlama dili olma özelliğini korumaktadır.

### ★ **Python**

Genel amaçlı dinamik bir dil olan Python, güçlü matematik kütüphaneleri ile akademik çalışmalarında en çok tercih edilen dillerden biri olmasının yanı sıra, işletim sistemi bileşenleri ve web uygulamaları için de yaygın olarak kullanılmaktadır.

### ★ **JavaScript**

Tarayıcılar içinde çalışmak üzere geliştirilmiş bir betik dili olan JavaScript'in ( asıl adı EcmaScript ) zaman içinde gelişmesi ile günümüzde sunucu tarafında da kullanılmak üzere evrimleşmiştir. Özellikle Node.JS ve veri tabanlarının desteği nedeniyle, hızlı web servisleri geliştirmek için tercih edilen bir dil olmuştur.



## GNU (*GNU is Not Unix*) nedir?



- ★ AT&T Bell Laboratuvarları multics projesini bıraktıktan sonra AT&T Bell Laboratuvarları çalışanlarından Dennis M. Ritchie ve Ken Thompson yeni bir işletim sistemi çalışmalarına başladı. Bu işletim sistemine “Unics” ismini koydular. Daha sonra isim “UNIX” olmuştur.
- ★ UNIX işletim sistemi üniversitelerde duyulmaya başlandı. Böylece üniversitelerin katkıları oldu. 1980'li yılların başlarında AT&T şirketi para kazanma amacıyla UNIX işletim sistemini pazarlamaya başladı. UNIX'in gelişmesine yardımcı olanlar bu durumdan hoşlanmadı. **Richard Stallman** tarafından UNIX'ten bağımsız olan GNU projesi başlatmıştır. GNU araçları UNIX kodlarını içermemektedir.
- ★ 1984 yılında Stallman ve arkadaşları tarafından FSF projesi başlatmıştır. FSF'nin açılımı "Free Software Foundation" yani "Özgür Yazılım Derneği"dir. FSF'nin amacı özgürce dağıtımı yapabilecek ve kodları üzerinde değişiklik yapabilecek GNU projesine destek vermektedir.

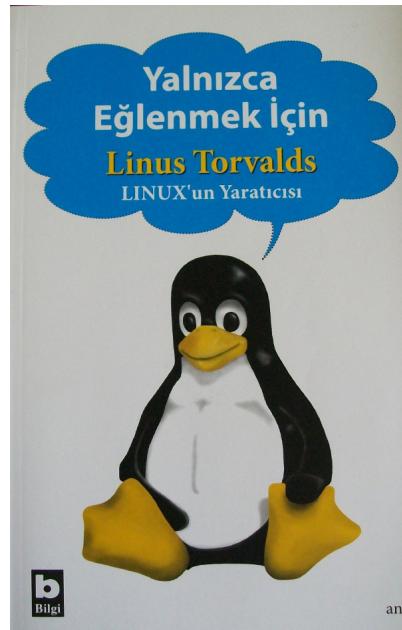
## Özgür Yazılımın Önemi

- ★ **Güvenilirlik** : Açık kaynak kodlu yazılımların belki de en büyük artısı olan güvenilir oluşları, kimi çevreler tarafından doğru olmayan bir mantıkla bir dezavantaj gibi gösterilmeye çalışılmakta ve topluma 'Açık kaynak kodlu bir yazılımın tam olarak nasıl çalıştığı kaynak koduna bakılarak bilinebildiği için yazılımın barındırdığı güvenlik problemleri saldırganlar tarafından kolaylıkla tespit edilip kullanılabilir' gibi bir düşünce yerleştirilmeye çalışılmaktadır. Fakat açık kaynak kodlu bir yazılım, doğası gereği geliştirilmesi sürecinden kullanımı zamanına degen onbinlerce göz tarafından denetlenmektedir. Var olması muhtemel bir güvenlik problemi ya da kötü niyetli bir programcı tarafından bilinçli olarak yerleştirilebilecek bir kod parçası yazılım sizin elinize ulaşmadan tespit edilmekte ve hızla düzeltilmekte ya da size ulaşması engellenmektedir. Ayrıca kimse, binlerce insanın kollektif bir şekilde geliştirdiği bir yazılımın güvenilmez olduğunu iddia ederken, sınırlı sayıda kişinin geliştirdiği ve kaynak kodunu kapattığı bir yazılımın güvenilirliğini kimin ve nasıl garanti edeceği sorusuna yanıt veremez.
- ★ **Sağlamlık** : Yine açık kaynak kodlu yazılımların doğal geliştirme süreci içerisinde çok sayıda insan tarafından denenmeleri ve kaynak kodlarının gözden geçirilmesi sonucunda ortaya yüksek kalitede, stabil ve kuvvetli yazılımlar çıkmaktadır.
- ★ **Esneklik** : Kaynak kodu açık bir yazılım hızla ve kolaylıkla yeni bir sistem üzerinde çalışacak şekilde yeniden yapılandırılabilir makte, bir kısmı çıkarılarak kapsamı daraltılabilir makte ya da eklenen yeni fonksiyonlarla kapsamı genişletilebilmektedir.
- ★ **Uygulama Desteği** : Çok geniş bir yelpazede ve çeşitlilikteki bir çok açık kaynak kodlu özgür yazılım her gün duyurulmaktadır. Herhangi bir konudaki ihtiyaç açık kaynak kodlu yazılımlarla hızlı ve kaliteli bir şekilde çözüme kavuşturulabilmektedir.

## ||||| Linux Nedir?

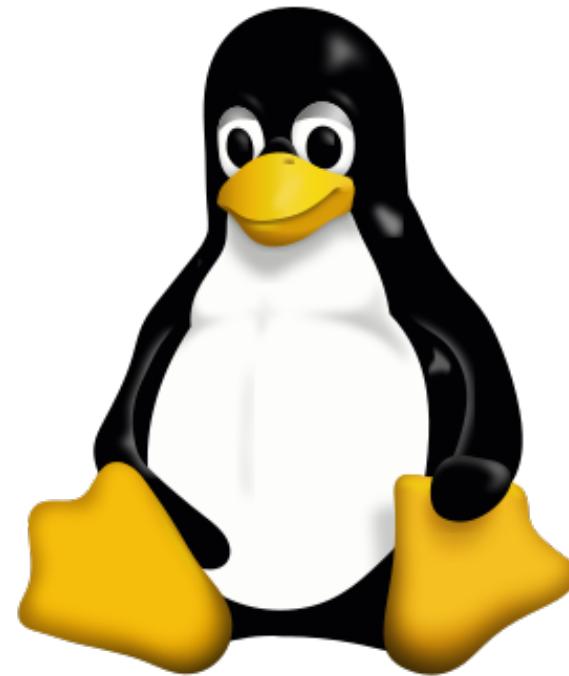
★ Helsinki Üniversitesi'nde bir öğrenci:  
**Linus Torvalds**  
**Revolution OS** filmi

★ Linus'un 25 Ağustos 1991 tarihinde minix mail listesine gönderdiği maili:  
“Minix kullanan dışarıdaki herkese merhaba. 386(486) AT kullanan insanlar için (serbest) işletim sistemi yapıyorum. (sadece hobi, gnu gibi büyük ve profesyonel olmayacak).”



## ||||| *Peki Linux tam olarak ne?*

- ★ Linux, aslında GNU/Linux işletim sistemlerinin sadece çekirdeğine (kernel) verilen isim.
- ★ Linux çekirdeğinin üzerine bir çok program eklenirse (ki bu programların bir kısmı GNU projesi araçlarını içermekte) bir Linux Dağıtıımı oluşturulmaktadır.
- ★ Örneğin bir dizindeki dosyaların listelenmesi
- ★ komutu olan “ls” bile çekirdeğin bir parçası değildir.



## ||||| Linux çekirdeği nerelerde kullanılıyor?

★ Kisaca *her yerde* kullanılıyor.

- ★ Sunucular
- ★ Süper bilgisayarlar
- ★ Kişisel bilgisayarlar
- ★ Cep telefonları
- ★ Tabletler
- ★ E-kitap okuyucular
- ★ Akıllı ocaklar
- ★ Saatler
- ★ Video oynatıcılar
- ★ ADSL Modemler
- ★ Apple PowerPC



Şu anda nerede ise tüm üniversite bilgi işlem merkezlerinin yanı sıra, NASA, IBM, HP, Boeing, HSBC, CityBank, SonyEricsson, Nokia, Siemens, Samsung, General Motors, Hyundai, Oracle, EToys gibi çok büyük kuruluşlar Linux kullanmakta ve bir şekilde özgür yazılımlara destek olmaktadır.



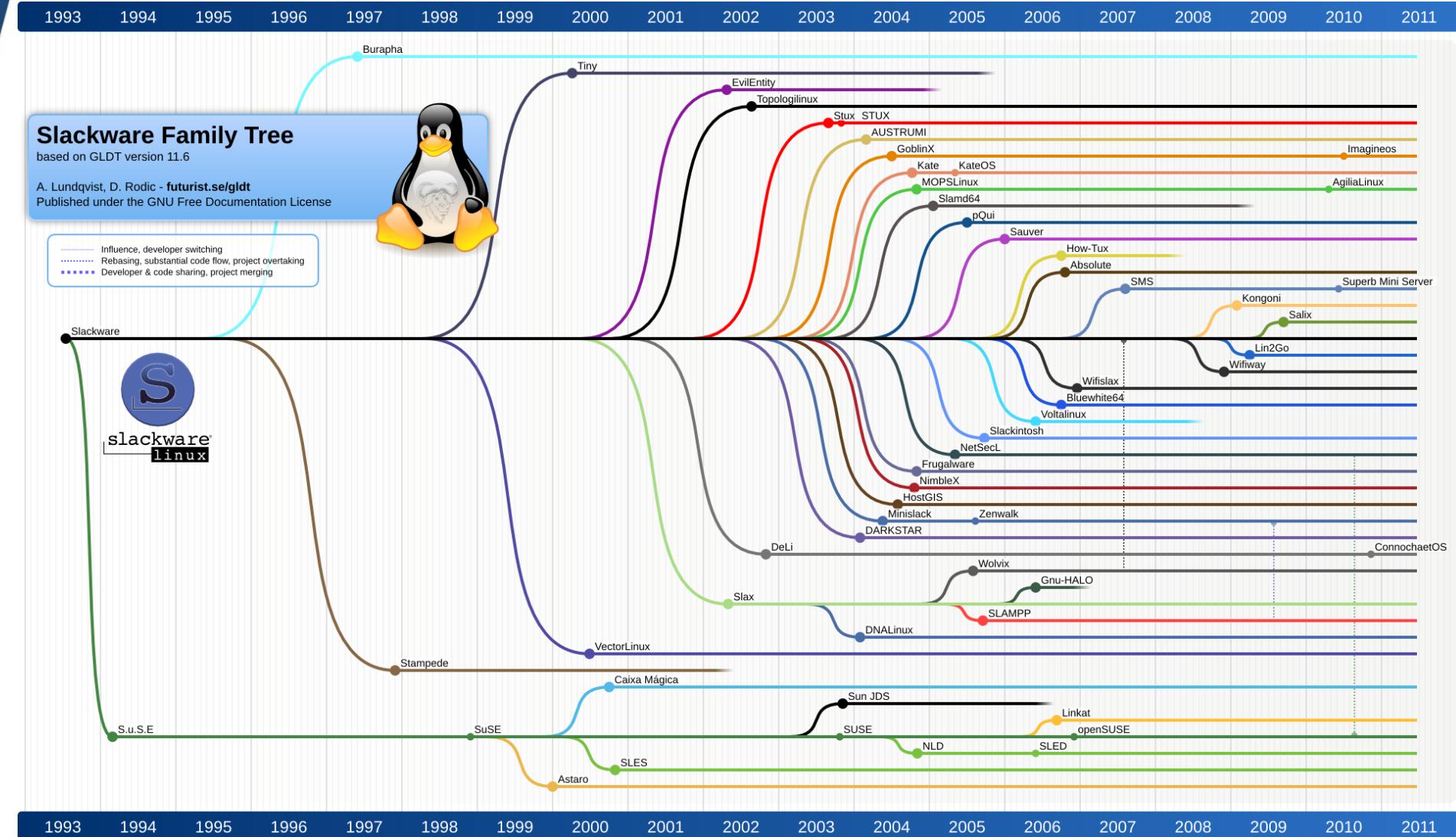
## GNU/Linux Dağılımı Nedir?

- ★ Linux çekirdeği, GNU araçları ve bir takım araçların bir araya gelmesiyle, bu birlikteliği sürdürülebilir şekilde yönetecek bir yapılandırma araçları seti, yazılım güncelleme araçları vb. ile oluşturularak tam teşekküllü bir işletim sistemi haline gelen uygulamalar bütündür.

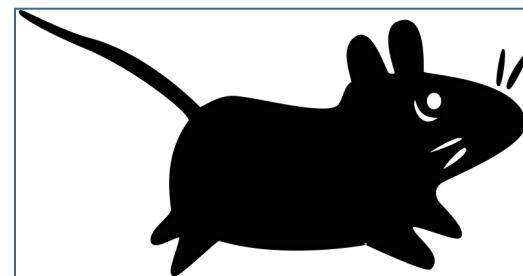




## GNU/Linux Dağılımı Nedir?



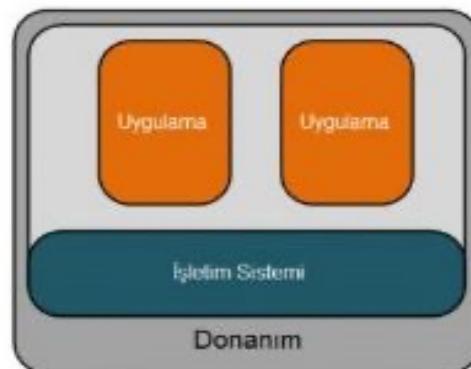
## ||||| Masaüstü Ortamları



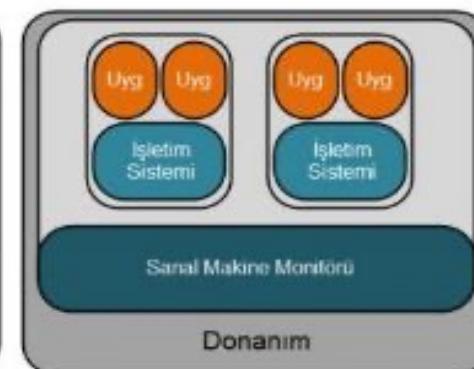
## ||||| Sanallaştırma Kavramı

- ★ Sanallaştırma, işletim sistemleri ve donanımların herhangi birisinin aslını kullanmak yerine sanallaştırılmış olarak kullanmaktadır.
- ★ Bu sayede bir işletim sistemi üzerinde aynı donanımlarla birden fazla işletim sistemi çalıştırılabilir olmuştur.
- ★ Bu teknoloji yönetim maliyetleri ve işlemci gücünden tasarruf sağlamaktadır.
- ★ 2005 yılından itibaren sanallaştırma uygulamaları hayatımızda önemli yer elde etti.

**Sanallaştırma olmadan**



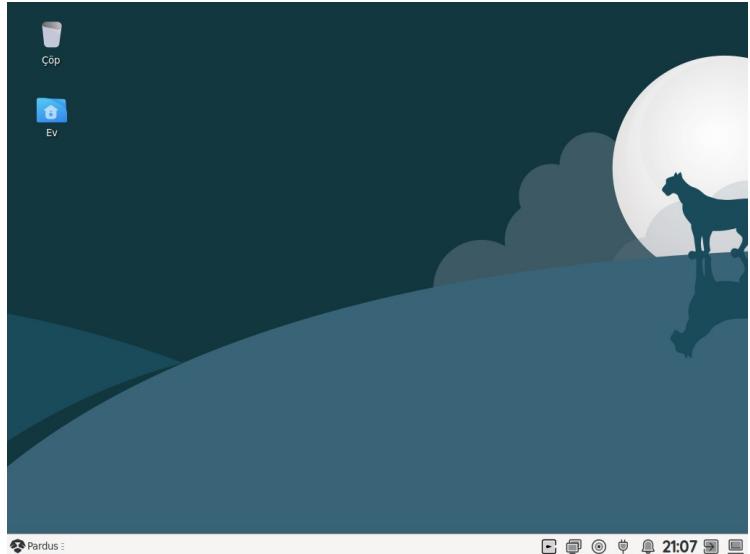
**Sanallaştırma ile**



## Masaüstü Ortamları

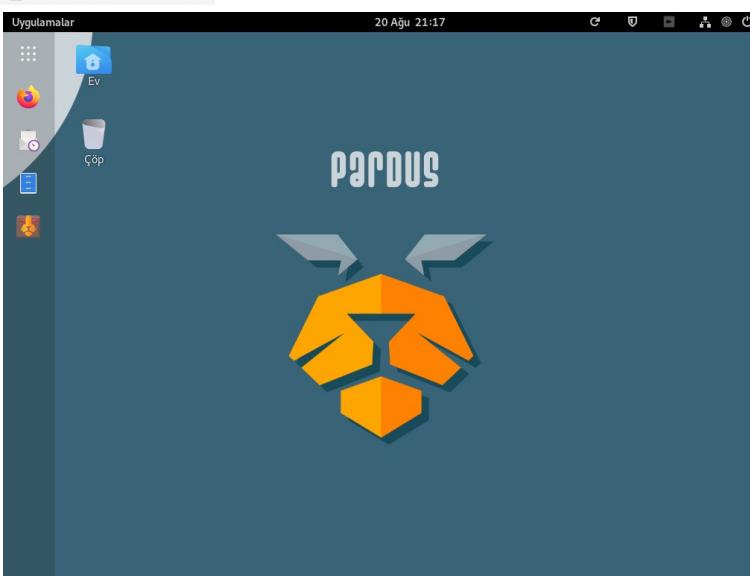
### XFCE – Pardus 21.0

Daha az bellek kullanmakta



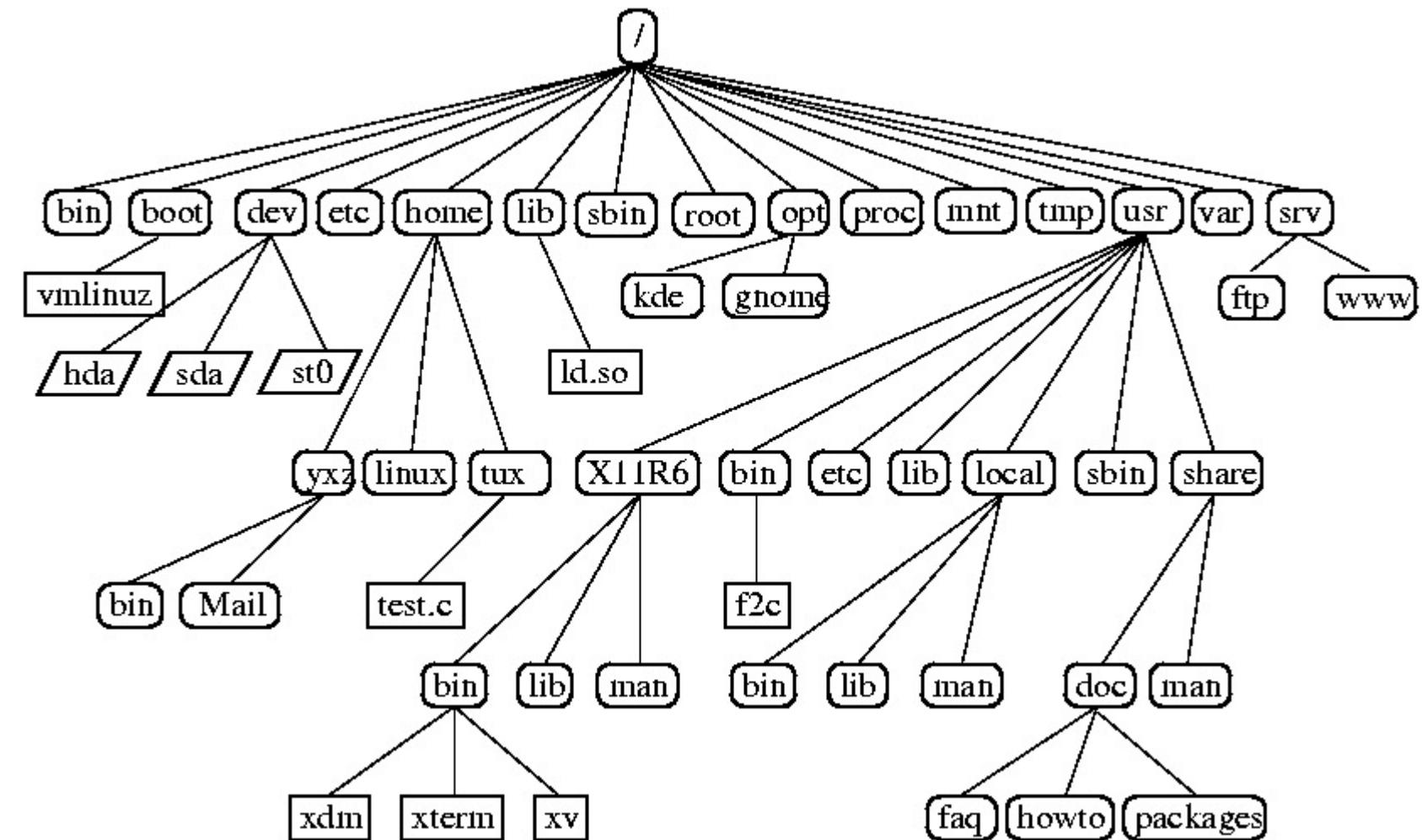
### KDE

Daha çok görsellik içermekte

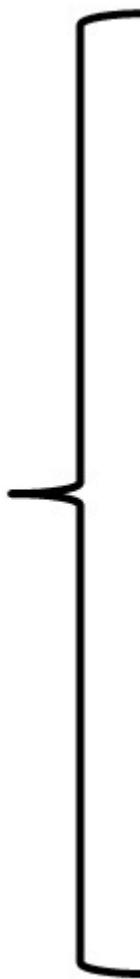


### GNOME - Pardus 21.0

## Linux üzerinde standart dizin yapısı

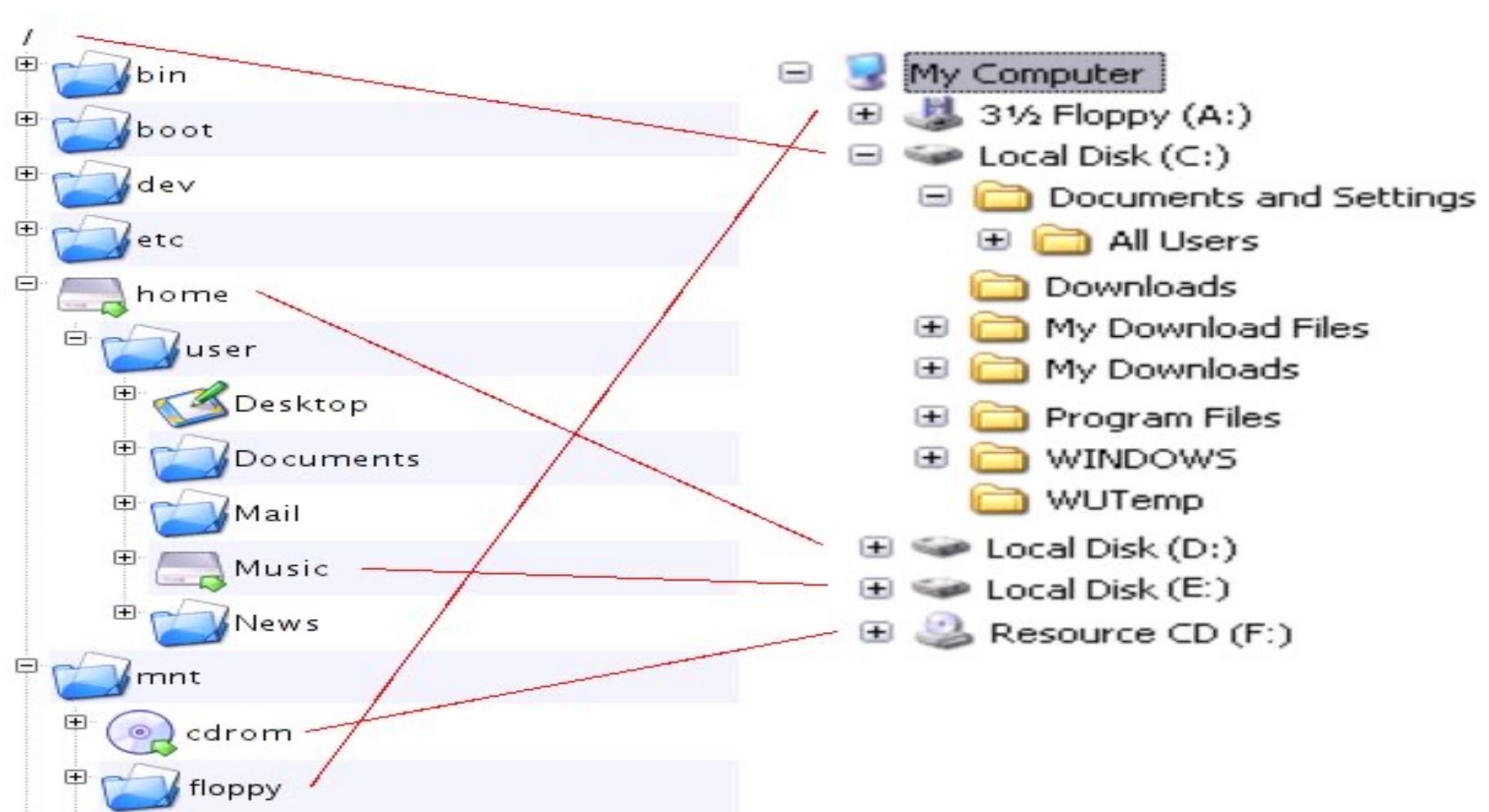


## Linux üzerinde standart dizin yapısı



/bin/	TEMEL KOMUT DOSYALARI
/boot/	ÖNYÜKLEYİCİ DOSYALARI
/dev/	AYGIT DOSYALARI
/etc/	SİSTEM AYAR DOSYALARI
/home/	KULLANICI EV DİZİNLERİ
/lib/	GEREKLİ ORTAK KÜTÜPHANELER VE KERNEL MODÜLLERİ
/media/	ÇIKARILABİLİR DEPOLAMA ORTAMLARI İÇİN BAĞLAMA NOKTASIDIR
/mnt/	SABİT DEPOLAMA ORTAMLARI İÇİN BAĞLAMA NOKTASIDIR
/opt/	ÜÇÜNCÜ PARTİ PROGRAMLAR İÇİNDİR
/sbin/	SİSTEM İLE İLGİLİ ÇALIŞTIRILABİLİR DOSYALAR
/srv/	SİSTEM TARAFINDAN SUNULAN HİZMETLERE İLİŞKİN VERİLER
/tmp/	GEÇİCİ DOSYALAR
/usr/	TÜM KULLANICILAR TARAFINDAN PAYLAŞILAN VERİLER
/var/	LOG DOSYALARI, YAZICI KUYRUKLARI GİBİ DEĞİŞKENLER
/root/	ROOT KULLANICININ ANA DİZİNİ
/proc/	KERNEL VE İŞLEM DURUMLARI HAKKINDA BİLGİLER

## Linux üzerinde standart dizin yapısı



File Systems: On the left is a typical Linux file system, and on the right is Windows' Explorer. The Linux file system consists of one "tree" with each drive attached to that tree and acting like a folder. The Windows file hierarchy consists of each physical drive having its own separate file system. The lines between the two point to where these drives appear in each graphic.

## ||||| *Peki Linux tam olarak ne?*

★ **Yol**, bir dosya veya klasörün sistemdeki konumuna verilen isim.

★ Örneğin masaüstünde bulunan Ali.txt isimli bir dosyanızın yolu aşağıdaki gibidir:

**/home/duygu/Masaüstü/Döküman.txt**  
**~/Masaüstü/Döküman.txt**

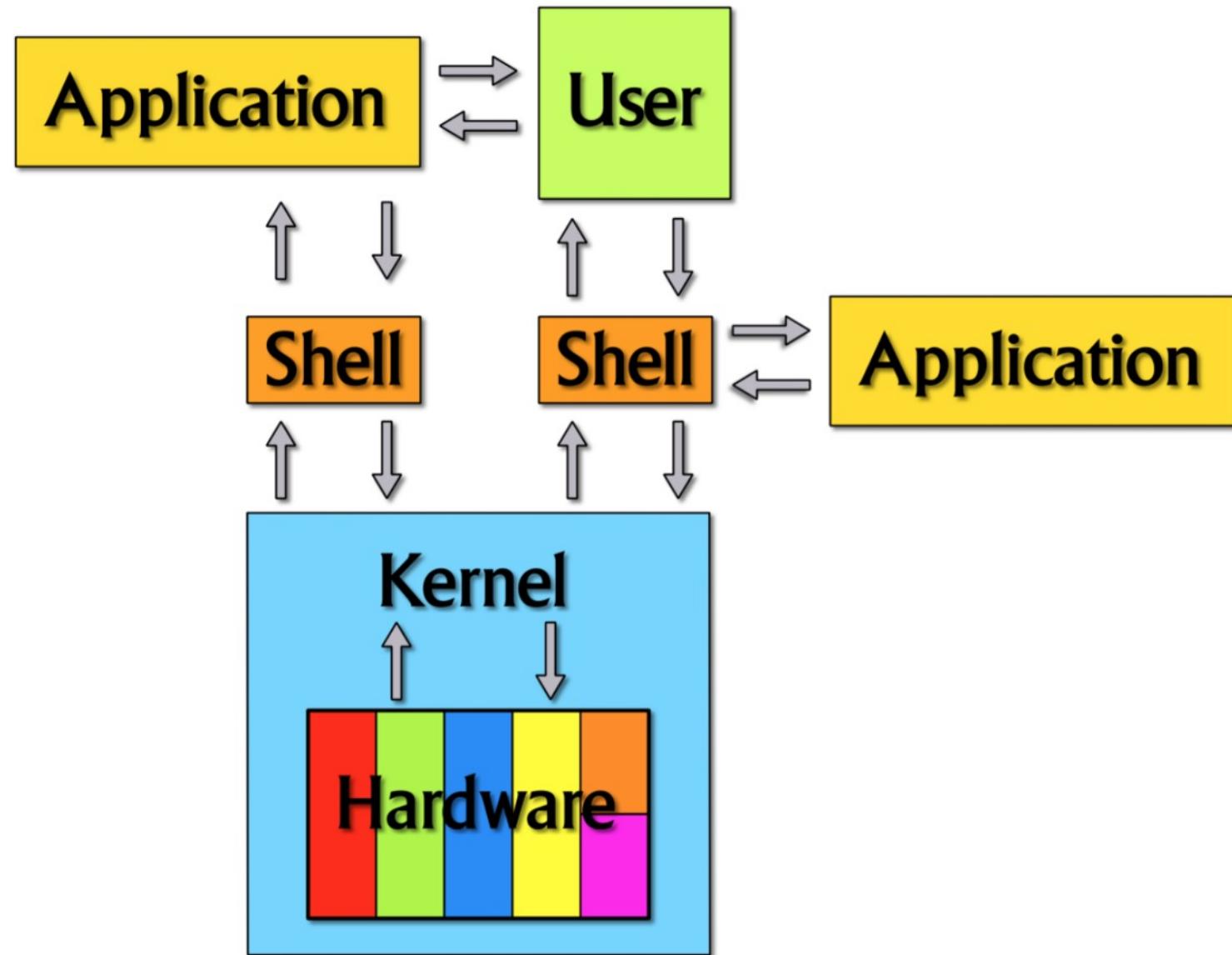
★ Herhangi bir kullanıcının masaüstü *dizin*inin yolu ise:

**/home/duygu/Masaüstü/**  
**~/Masaüstü/**

★ Sistemin yetkilendirme ile ilgili log kaydının tutulduğu dosyanın yolu:

**/var/log/auth.log**

## |||| Kabuk Kavramı



## ||||| Temel Komutlar

Dizinler		
pwd		Çalışılan dizinin tam yulunu verir
cd	<dizin1>	İstenilen dizine geçiş yapar
cd	..	Bir önceki dizine gider
ls	<dizin1>	Dosya içeriğini listeler
	-a	Gizli dahil dosyaları göster
	-R	Alt dizinleri listele
	-r	Sıralama ters çevir
	-t	Son düzenlemeye göre listele
	-S	Dosya boyutuna göre listele
	-l	Geniş listeleme formatı
	-1	Tel satır, tek dosya
	-m	Cıktı virgül ile ayrılır
	-Q	Tırnak içerisine al

## Temel Komutlar

### Dosya ve Klasörler

touch	<dosya1>	dosya1 isminde dosya oluşturur
cat	<dosya1>	dosya1'in içeriğini terminale yazdırır
echo	"Linux komutları"	İfadeyi terminal ekranına yazdırılır
head	<dosya1>	dosya1'in ilk 10 satır çıktısı
tail	<dosya1>	dosya1'in son 10 satır çıktısı
sort	<dosya1>	dosya1 isminde dosya oluşturulur
cp	<dosya1> <dosya2>	dosya1'in dosya2'ye kopyalanması
cp -r	<dizin1> <dizin2>	dizin1'in dizin2'ye kopyalanması
mv	<dosya1> <dosya2>	dosya1'in dosya2'ye taşınması
rm	<dosya1>	dosya1 silinir
rm -rf	<dizin1>	dizin1'i içindelerle birlikte silme
mkdir	<dizin1>	dizin1 adında dizin oluşturulur
rmdir	<dizin1>	İçи boş dizin1 silinir

## ||||| Temel Komutlar

### Komut Listeleri

komut1; komut2	Once komut1, sonra komut2 çalışır
komut1 && komut2	komut1 başarılırsa komut2 çalışır
komut1    komut2	komut1 başarısızsa komut2 çalışır
komut1 &	Alt kabukta komut1 çalıştır

## ||||| Temel Komutlar

### GÇ Yönlendirmeleri

komut > dosya1	dosya1 içeriği komuta girdi verir
komut1 < (komut2)	komut2 çıktısı, komut1'e girdi verir
komut > dosya1	Komut çıktısı dosyanın içeriğine yazılır
komut > /dev/null	Komut çıktısını yazdırılmaz
komut >> dosya1	Komut çıktısı dosyanın sonuna ekler
komut2 > dosya1	Komut hata çıktısını dosya1'e yazdırır
komut &> dosya1	Komut çıktılarını dosya1'e yazdırır

## ||||| Temel Komutlar

### Özel İşretler

komut1 | komut2

komut1'in çıktısını komut2'ye girdi alır

komut1 |& komut2

komut1'in hata çıktısı komut2'ye girdi

## ||||| Temel Komutlar

### Süreçler

ps	Aktif kullanıcıyla ilgili süreç listesi
ps aux	Bilgisayarda o an çalışan süreç listesi
pstree	Sistemdeki süreçleri hiyerarsık listesi
kill <pid>	Süreci sonlandırır
killall <pid>	Programla ilgili tüm süreçleri sonlandırır
killall -9 <pid>	Ölmemekte direnen süreçleri sonlandırır

## ||||| Dosya içerik arama işlemleri (grep)

- ★ Grep komutu Linux' ta dosya içerisinde arama yapmak için kullanılan bir komuttur. Bu komut ile istediğimiz herhangi bir stringin hangi dosyalar içerisinde geçtiğini bulabiliriz
  - echo "" THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
  - this line is the 1st lower case line in this file.
  - This Line Has All Its First Character Of The Word With Upper Case.
  - Two lines above this line is empty.
  - And this is the last line.""" > dosya1.txt
- ★ Aşağıdaki komut çalıştığı zaman bize içerişinde this geçen satırları verir.  
`grep "this" dosya1.txt`
- ★ İstenilen string' i birden fazla dosya içerisinde arama  
`cp dosya1.txt dosya2.txt`  
`grep "this" dosya1.txt`
- ★ Harfe duyarlılığı kaldırarak arama yapmak  
`grep -i "the" dosya1.txt`
- ★ Regular Expression (Düzenli İfade) arama  
`grep "is.*ine" dosya1.txt`
- ★ Tam kelime eşleşmesini aramak  
`grep -iw "is" dosya1.txt`

## ||||| Dosya içerik arama işlemleri

### ★ locate

Dosya ve dizin aramak için kullanılır.

Kullanımı:

# find dizin seçenekler

Dizin adı belirtilmezse bulunulan dizin ifade edilir.

Seçenekler:

-name isim: Aranılacak dosyanın ismi.

-iname isim: Aranılacak dosya ismi büyük/küçük harf duyarsız

-type tip: Aranan dosyanın tipini belirler. Tip aşağıdaki değerlerden biri olabilir.

f: Normal Dosya

d: Dizin

## ||||| Dosya içerik arama işlemleri

### ☆ find

find komutu arama yaparken her seferinde belirtilen dizinde tarama yaptığı için çok fazla dosya olan sistemlerde yavaş çalışabilir. Daha hızlı dosya aramak için locate komutu kullanılabilir. Bu komut dosya adı aramalarını direkt dosya sisteminde tarama yapmak yerine özel bir veritabanından getirmektedir. Locate komutu aramayı /var/cache/locate/locatedb veritabanı dosyasında yapmaktadır. Bu dosya locate dışında ayrı olarak çalışan **updatedb** komutu tarafından güncellenmektedir. Eğer en güncel dosyalarında aramaya dahil olmasını istiyorsanız belirli aralıklarla updatedb komutu çalıştırılmalıdır. Her gece veritabanı dosyasını güncellemek için Pardus tarafından otomatik olarak /etc/cron.daily/locate dosyasındaki cron işi çalıştırılmaktadır. Arama yapmak için dosyanın adı veya adının geçtiği kelimeleri yazmak yeterlidir.

## ||||| Dosya içerik arama işlemleri

### ☆ which

\$PATH değişkeni içerisinde tanımlı çalıştırılabilir komutların tam yolunu gösterir.  
Sadece komutlar ve shell scriptler için çalışır.

```
# which cut  
/bin/cut
```

### ☆ whereis

Parametre olarak verilen program, kaynak kodu ve yardım dosyalarının yerlerini gösterir.

```
# whereis pwd  
pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz
```

## İçerik Manipülasyon

### ★ awk

awk, güçlü bir desen arama aracıdır. Sed gibi benzerlerinden ayıran özellik ise matematik işlemler, şartlı ifadeler, değişkenler ve dosya G/C işlemleri gibi script dillerinde olan temel işlevlere sahip olmasıdır. Genelde yapılandırma dosyalarını okumak veya komut çıktılarını işlemek için kullanılır. Dosyalarla işlem yapmak için –f parametresi kullanılır.

Çağrılama şekli:

```
$ awk -f awk_script dosya  
$ awk desen {aksiyon} dosya
```

Eğer desen belirtilmezse bütün satırlar eşleşir. Sonda verilen dosya parametresi yerine metin akımı üzerinde de işlem yapabilir.

Belirtilen dosyadaki 2 ve 3 numaralı sütunu ekrana basmak için (sütunlar \$2 ve \$3 ile belirtildi):

```
$ awk < dosya '{ print $2, $3 }'
```

Bash ile biten satırları göstermek için:

```
$ awk '/bash$/' /etc/passwd
```

## İçerik Manipülasyon

### ★ sed

Sed (veya GNU versiyonu olan gsed) , oldukça yetenekli bir yazı düzenlemeye editörüdür. Tek komutla sizi büyük bir iş yükünden kurtarabilir, işte bazı kullanım örnekleri:

#### BOŞLUK KOYMA

İçinde "deneme" geçen satırların üzerinde bir boş satır oluştur:

```
$sed '/deneme/{x;p;x;}'
```

İçinde "deneme" geçen satırların altında bir boş satır oluştur:

```
$sed '/deneme/G'
```

İçinde "deneme" geçen satırların hem üzerinde hem altında bir boş satır oluştur:

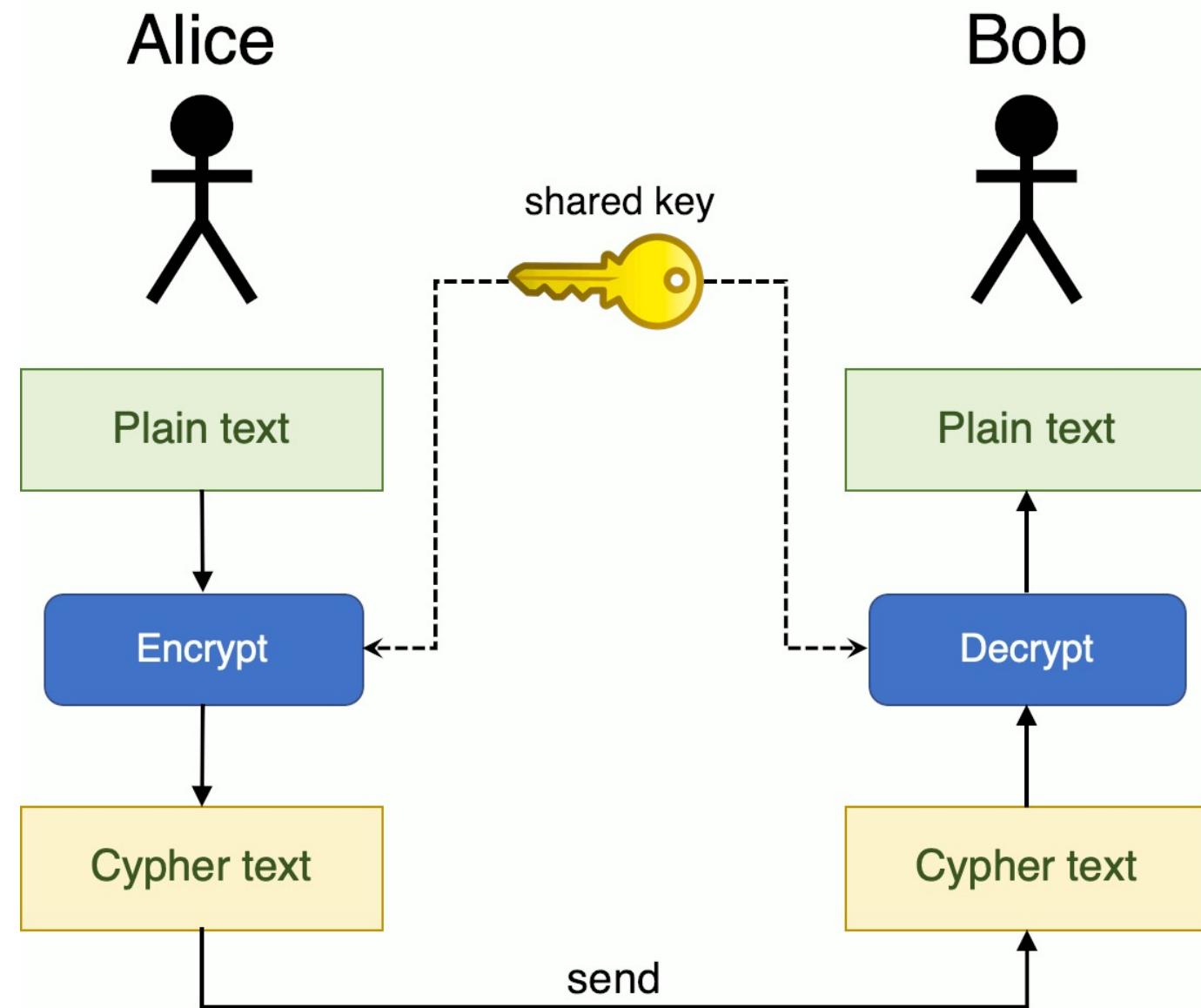
```
$sed '/deneme/{x;p;x;G;}'
```

Her 5 satırdan sonra bir boş satır ekle:

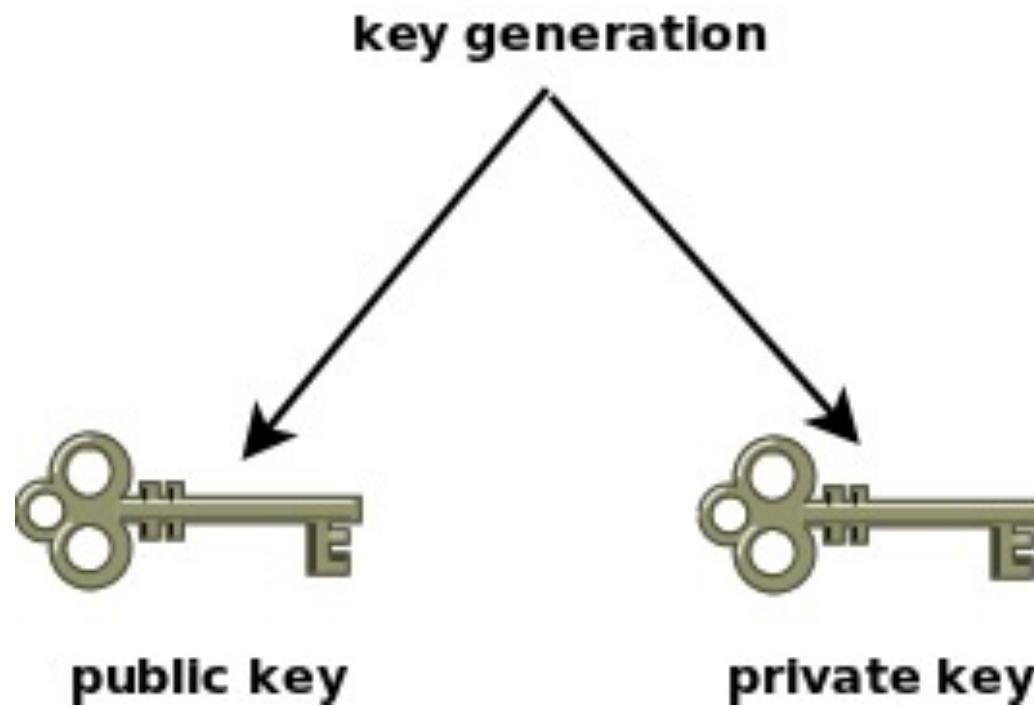
```
$gsed '0~5G' (GNU sed)
```

```
$sed 'n;n;n;n;G;' (Diğer sed sürümleri)
```

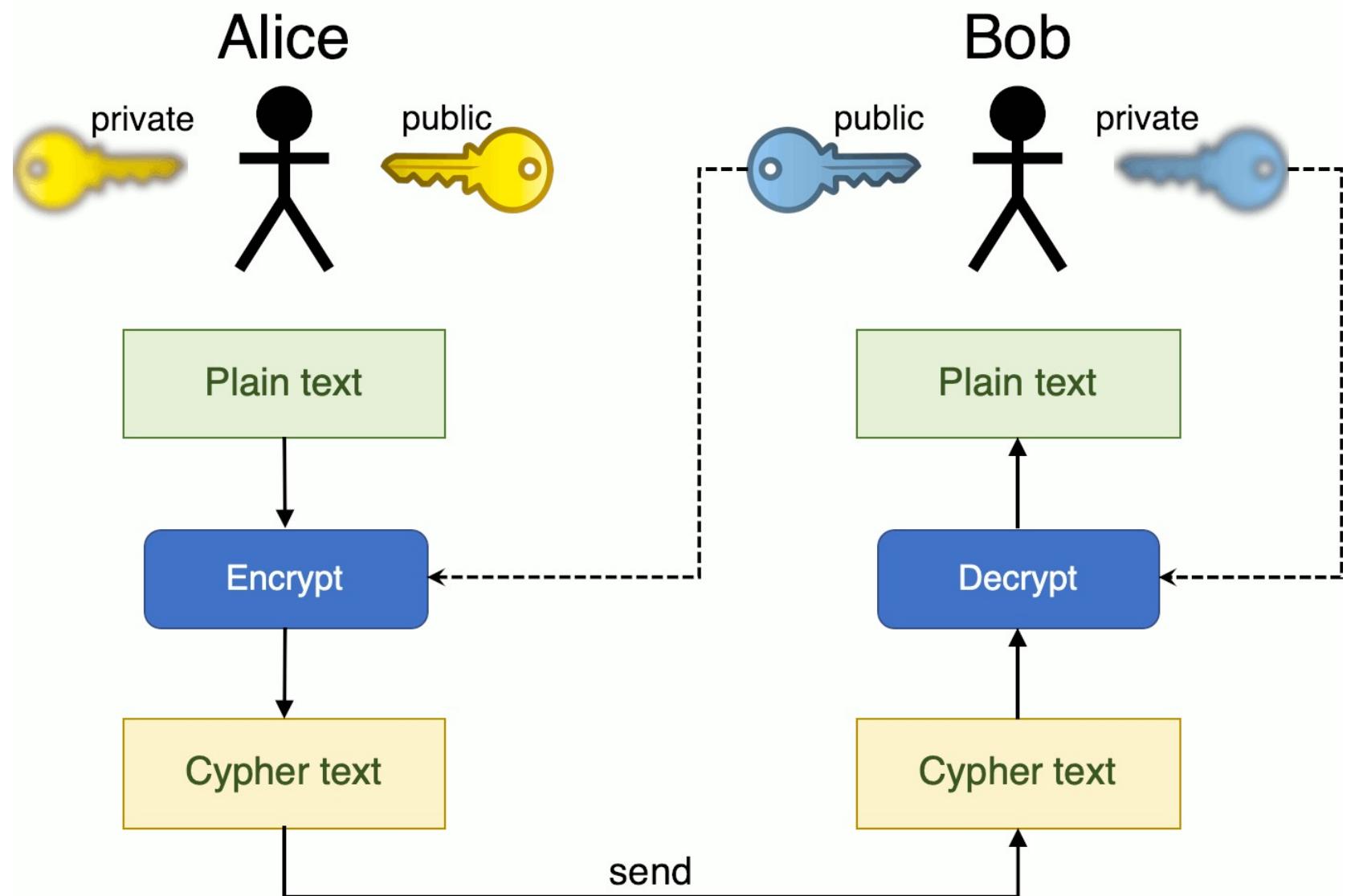
## ||||| Simetrik Şifreleme



## ||||| Asimetrik Şifreleme



## ||||| Asimetrik Şifreleme





## Özgür Yazılımın Önemi

- <http://www.linuxfocus.org/Turkce/March1998/article27.html>
- <http://ab.org.tr/ab12/sunum/202.pdf>
- <https://seminer.linux.org.tr/wp-content/uploads/ozguryazilimlisanslari.pdf>
- <http://ozgurlisanslar.org.tr/>
- <http://www.nyucel.com/2017/05/ozgur-yazlm-ile-ack-kaynagn-ne-fark-var.html>
- <http://www.nyucel.com/2017/05/ozgur-yazlm-m-ack-kaynak-m-1.html>