1. GNU/LINUX İŞLETİM SİSTEMİNİN YAPISI

**Linux Dosya Sistemi**

**/root**: kök dizin. Diğer bütün dizinler root dizinin altındadır.  
**/root**: Sistem yöneticisinin ev dizini.   
**/home**: Sistemdeki kullanıcıların ev dizinidir  
**/usr**: Tüm kullanıcılarla paylaşılan verileri içeren dizin  
**/bin**: Genel kullanıcıların kullanabileceği komutlara ait dosyaların bulunduğu dizindir. ls, mkdir, cat..  
**/sbin**: Sadece root kullanıcısının kullanabileceği komutlara ait dosyaların bulunduğu dizin.  
**/boot**: Sistem açılışında kullanılan dosyaların bulunduğu dizin  
**/dev**: Aygıtlar ve disk bölümlerine ait dosyaların bulunduğu bölüm  
**/etc**: Sisteme ait yapılandırma dosyalarının bulunduğu dizin  
**/etc/passwd** : Her kullanıcı hakkında ayrıntılı bilgilerin bulunduğu dizin.  
**/etc/group** :Gruplar hakkında ayrıntılı bilgilerin bulunduğu dizin  
**/var** : Log dosyaları, e mail ve printer kuyrukları gibi değişken sistem bilgilerini barındırır  
**/tmp** : Geçici dosyaların tutulduğu dizin  
**/proc** : Sistem süreçlerinin bilgisinin bulunduğu sanal dosya sisteminin dosyalarının bulunduğu dizin  
**/lib** :Programların ihtiyacı olan kütüphane dosyalarının bulunduğu dizin  
**/mnt** :Sistem yöneticisinin geçici olarak herhangi bir dosya sistemine ulaşmak için oluşturabileceği bağlama noktalarını içeren dizin  
**/opt** : Dağıtımdan bağımsız ekstra yüklenen paketler için kullanılan dizin  
**/media** : Kaldırılabilir aygıtların sisteme eklendiği dizin

echo $SHELL # sahip olduğumuz Shell

2. KOMUT SATIRI VE TEMEL KOMUTLAR

**man komut** : Komut hakkında bilgi verir. örnek : man ls  
**man –k disk**  #Disk ile ilgili programları öğrenmek için **ls –help** #Komut hakkında bilgi verir.  
**apropos anahtarkelime** #Dokumanlar içinde kelimeyle alakalı komutları gösterir  
**whereis** komut   
**whatis** komut

**wget indirme komutu**wget adres

**Kısayol Tuşları**

ctrl c : öldürür   
ctrl z: askıya alır  
ctrl d: bulunulan oturumu kapatır   
ctrl l: clear komutu ile aynı. Ekranı temizler. hafızadan silmez.  
ctrl e : imleç nerede olursa olsun yazılan kodun en sonuna gider.  
ctrl a : en başa gider  
ctrl w: imlecin bulunduğu yerden sola doğru ilk bosluga kadar kelimeyi siler  
ctrl k: sag tarafa doğru hepsini  
ctrl u: sol tarafa doğru hepsini siler  
ctrl r:kod tamamlama, history den daha önce yazılan kodu arama

# tab tuşu path ya da komutun devamını getiriyor, tamamlıyor.  
# tab ile devamı hatırlanamayan kodlar da gözükür

**Dosya Ve Dizin İşlemleri Komutları**

touch dosyaismi : yeni dosya oluşturur  
mkdir dizin : yeni dizin oluşturur  
mcdir -p A/{B,C,D}/{X,Y,Z}/{a,b,c}LYK  
rm dosyaismi #dosyayı siler  
rmdir dizinisimi # dizin boş iken dizini siler  
rm –rf dizinadı # dizini alt klasörleri ile beraber siler  
cd # klasör değiştirmek için kullanılır  
pwd # içinde bulunulan dizini verir  
history > historyisim # yazılan kodlar kaydedilir  
> yönlendirme  
>> birikimli yönlendirme  
| : bir komutun çıktısını diğer komutun girdisi haline getirir

**ls**   
# içinde bulunulan klasörde neler olduğunu görmek için kullanılır  
Parametreleri :   
-a gizli dosyaları da gösterir  
-l alt alta gösterir  
-d sadece directory dosyasını gösterir  
-r küçükten büyüğe sıralarak gösterir  
-R alt dizinleri de listeler   
-s boyuta göre sıralar

**cp**   
# dosya ve dizin kopyalamak için kullanılır. cp kaynak hedef   
cp abc.txt /tmp  
  
  
**mv**   
# dosya taşımak yeniden isimlendirmek için kullanılır  
Parametreleri:  
-i hedef dosya veya dizin varsa üzerine yazma için onay alır  
-f onay almadan yapar.

**Dosya Okuma Komutları**

cat : dosyanın içeriğini ekrana yazdırır (-n: satır numaraları gözükür)  
tac : dosyanın içeriğini sondanbaşa doğru gösterir  
rev : dosyanın içeriğini her satırı ters çevirerek gösterir  
more: dosyanın içeriğini sayfa sayfa gösterir  
less : ok tuşlarıyla sayfa da ilerlenebilir(space sonraki syf, b önceki sayfa, q çıkış)  
head: dosya içeriğini baştan belirtilen satır kadar gösterir. varsayılan değeri 10  
tail : dosya içeriğini sondan belirtilen satır kadar gösterir.

**tr : translate komutu**  
cat a.txt | tr A-Z a-z # tüm harfleri küçük harfe çevirmek  
cat a.txt | tr –d a | more # a.txt deki tüm “a” ları sil  
cat a.txt | tr bcd aaa # bcd leri aaa ya çevir  
cat a.txt | tr bcd aaa | tr –s a | more # birden fazla a varsa teke düşür.

**sort : sıralama komutu**sort dosyaadı # içerik sıralanırsort –r # sıralama tersine çevirilir  
sort –u # yinelenen satırları atlayarak biricikliği sağlar

**cut : istenilen kolonu gösterme komutu**cat /etc/passwd | cut –d “:” –f1   
# - d ile nasıl parçalara ayıracağımızı belirttiğimiz dizindeki ilk sütundaki kelimeler  
cat /etc/passwd | cut –d “:” –f 1 | sort   
# passwd yi yazdır, 1. sütunu göster, sort ile sırala

**wc komutu**  
Bir dosya veya komut çıktısındaki satır, karakter ve kelime sayılarını bulmak için kullanılır.  
-l : satır sayısını bulur  
-w : kelime sayısını bulur  
-c karakter sayısını bulur

**grep komutu**  
Dosyaların içerisinde arama yapar.  
grep murat deneme # deneme isimli dosyanın içinde murat kelimesi geçen satırları listeler   
grep murat /etc/passwd # bu dizinde murat kelimesi geçen satırları listeler  
grep net \*.txt # uzantısı txt olan ve adı önemli olmayan tüm txt lerde net kelimesini ara  
grep net$ a.txt satırın sonunda net olan ifadeler  
grep –i ^net a.txt # satırın başında net olan ifadeler  
grep –i net AA.txt # -i büyük harf küçük harf farketmez her ikisine de bakar  
grep –w net AA.txt # aranan ifadenin tek başına olması  
grep –l net \*.txt # içinde net geçen dosya isimleri  
grep –c net \*.txt # içinde kaç satırın uyduğunu belirtiyor  
grep net$ AA.txt # satırın sonunda net olan ifadeler  
grep –i ^net AA.txt # başında net olan satırlar  
grep ^$ AA.txt # boş satırlar  
grep –i net AA.txt | grep –i fox # içinde hem net hem de faks kelimesi geçen satırlar  
grep –i net AA.txt | grep –iv fox # net var ama faks yok  
grep –i “net|fox” AA.txt #tek başına net tek basına fox ya da her ikisni birden yazdırıyor  
grep –E –i “net | fox “ AA.txt # tek başına net tek basına fox ya da her ikisni birden   
grep –i “net.\*fox” # netten sonra sırasıyla fox gecen satırlar

grep –iw “net.\*fox # sadece net geçenler ve devamında sadece fox geçenler  
  
\([a-z]\)([a-z]\)\2\1 # birinci karakter küçük harf ikinci karakter küçük harf üçüncü karakter 2. cinin aynısını dördüncü karakter brincinin aynısı. Baştaki \önemli   
  
  
-i : büyük küçük harf fark etmez   
-w : aranan ifade tek başına olacak   
-l :uyan satırları listeler   
-c : kaç satırın uyduğunu belirtir   
$ : satırın sonu  
^ : satırın başı

**find komutu**  
find /home/vahit/ -type f –name bilgisayar  
# /home/vahit dizininde tipi dosya olan ve isminde bilgisayar olan şeyleri bul

find /etc –name \*.txt # etc içinde txt uzantılı dosyaları bul  
find /etc –type f –name “passwd”

# etc dizininde ismi passwd olan tipi dosya olan dosyaları bul  
find /var/log/ -type f - size +10K # varlog dizininde boyutu 10k dan büyük olan dosyalar

find /var/log/ -type f -size +10 –exec du sh {} + | sort –h   
# varlog daki dosya tipi dosyaların 10 kb den buyuk olanları buldu işletti ve sıraladı.

find dosyayla iligli olan aramalara bakar içeriği değil

find aranacak (/ , /home, /var/log ) -name (isme göre) -iname (büyükküçük harf önemsiz) -size (boyutlar için) -type (dosya tipleri )

Örnek : find /var/log -iname "\*.log" 2>/dev/null ==> log dosyalarını hatasız(2>/dev/null yaparak hataları çöp kutusuna atıyor) bir şekilde çıkartıyor

{} bu karakter find işleminde her bir dosya için işlem yapacağımız anlamına gelir

+ veya \ işareti {} den sonra kullanılır ve exec ten sonra kullandığımız komutu bitirmeye yarıyor. \ işareti ile birlikte ; kullanılmalı yoksa hata verir

Örnek : find /var/log -iname "\*.log" 2>/dev/null -exec file {} +

exec yapısı;

-exec komut {} \;

-exec komut {} +

find /etc/ -type f -exec grep -n root {} + 2>/dev/null ==> etc içerisindeki bütün dosyalarda root geçen satırları bize listeliyor ve hataları dev null içine atıyor

chattr +i ile girmiş olduğumuz dosyalara herhangi bir müdahalede bulunamam  
Örnek : chattr +i /etc/passwd dersek bu dosyada bi işlem yapamam  
chattr +a dosya adı ile o dosyada sadece append işlemini yaptırabiliriz

Örn; cat \*.txt > A1 diyip

cat \*.sh >> A1 dediğimizde üstüne ekleme işlemini yaptırmayabiliriz.

script akif diyip ekran görüntülerini sabahtan akşama kadar alır ve exit yada ctrl+c ile bitirebiliriz. bunu sadece more ile açabiliriz

script /home/makif/bubirdeneme.txt yazarak bunu bir txt dosyasına da kaydedebiliriz.

sort -u ile uniq arasındaki fark;

uniq olunca birden fazla a.txt olup arada b.txt varsa birden fazla olan a.txt yi tek a.txt yapıp aradaki b.txt yi b.txt olarak alır sonraki a.txt yi tekrar alır. Ama önce sort -u yapıp sonra uniq yapmak mantıklıdır çünkü sıralamada a.txt ler hep bir aradadır.

softlınkın bağlantıısnı ln -s /home/makıf/a.txt a\_soft.txt olarak kaynak dosyayı yoluyla birlikte verirsek kaynak dosyayı nereye taşırsak taşıyalım verdiğimiz yolu verecektir

**ps komutu**   
ps –aux # çalışan prosesleri listeler  
ps –aux | grep vahit # vahit kullanıcısına ait çalışan prosesleri listeler  
ps ile liste alındıktan sonra listede kill edilmek istenen prosesin id si alınır ve şu kod ile kill edilir :  
kill -9 1124

**KOD EDİTÖRÜ Vİ**

vi /tmp/vim #vi editörü açıyoruz.

**İNSERT MODU**

insert moda geçmek için i yazıyoruz. geri dönmek için ESC, çıkmak için wq

a imlecin sağından başlıyor

shift + i : satırın en başına geçiriyor  
shift + a : bulunulan satırın en sonundan insert moduna giriyor  
o : bulunduğu satırın altından başlar  
shift + o : bulunulan satırın bir üstünden insert moduna giriyor

**VİSUAL MODU**

0: bulunulan satırın başına gider  
$: bulunulan satırın sonuna gider  
d : delete # dd: satırı birinci d de seçiyor ikinci d de siliyor  
x : silme işlemi yapıyor. Tek karakter siliyor.  
R:replace, harfin üzerine gidilir r ye basılıp değiştirilmek istenen harfe basılır  
  
s:üzerinde bulunulan harfi silip insert moduna geçer  
shift + s : komple satırı silip insert moduna geçiyor  
cc: komple satırı silip insert moduna geçiyor  
ce: bir kelimeyi siler insert moduna geçer;  
shift + c : bulunulan yerden sag tarafı silip insert moduna geçer  
shift + d:bulunulan yerden sol tarafı silip insert moduna geçer

ctrl u: yapılan işlemi geri alıyor   
ctrl r : yapılan işlemi ileri alıyor

**NAVİGASYON KOMUTLARI**

h : sola, j : aşağı, k : yukarı, l: sağa  
g: iki kere basarak sayfanın en başına gidiyor  
shift + g : sayfanın en sonuna gidiyor

silme için navigasyonlar kullanılabilir d +yön şeklinde

insert modunda 1234567890 yazdık ve sonra esc ile visual moda geçtikten sonra dJ9L yi tuşladık. Bunu yazınca sadece 0 harfi kaldı

e: bir kelime gider  
de: üzerinde olunan kelimeyi siler  
15e: 15 kelime gider  
d3e : 3 kelimeyi siler

w:en yakın kelimenin baş harfine gidiyor  
. : önceki komutu tekrarlar  
b: kelimenin başına gider

gg ile en üste git d + shift + g ye bas hepsini siler u ya bas geri alır.  
: iki nokta üst üste koyunca komut kısmı açılır.  
:10 dersek onuncu satıra gider

% # [],() benzeri işaretler arasında git gel yapıyor

**KOPYALAMA YAPIŞTIRMA KESME İŞLEMLERİ**

y: kopyalama , iki defa yy yapılırsa satırı kopyalıyor.  
p: yapıştırıyor

y3ep : 3 kelime yi 3e ile belirtip y ile kopyaladık istediğimiz yere p ile yapıştırdık

y5jp: beş satırı kopyaladı ve p ile yapıştırdık

d:kesme , iki defa dd yaparsak kesmiş oluruz yani silme ile aynı

x: x tek karakter keser

f: f ye basıp bir karaktere basarsak kendinden sonra basılan ilk karaktere gider

d2ft:d silme, f silme komuyu ama ikinciye kadar sil,t de silinecek harf

**ARAMA YAPMAK**

/ aranansey : bulunduğun yerden alt tarafa doğru arama yapar yoksa yukarı da çıkar .  
? aranansey : bulunduğun yerden üst tarafa doğru arama yapar yoksa yukarı da arar  
n: aşağı dpgru aramaya devam eder  
shift + n :yukarı doğru aramaya devam eder  
/ arananşey \c : büyük küçük harf noktalama işaretleri vs ye takılmadan arar

**KAPATMA KAYDETME**

:w #kaydediyor  
:wq #kaydet çık  
:q! # kaydetmeden çıkacak  
:w dosyaismi # farklı kaydetme

:! ls –la # vi den çıkmadan terminal komutları verebiliriz.   
Mesela bir dizin oluşturmamız gerekti diyelim bu dizini vi den çıkmadan yapabiliriz.

:1,$ s/escape/haci/g # ilk sıradan başla dokumanın en sonuna git , s değiştirmek anlamına geliyor , escape değiştirilmek istenen kelime yerine koyulacak isim haci, g bir satırda değiştirmek istenen şeyler iki tane ise ikisini de değiştirir, g yazılmazsa bir tane değiştirir.

**KABUK DEĞİŞTİRME**

oh my zsh (Google )  
yum install –y zsh  
yum install –y git  
oh my zsh Google a yaz ikinci kodu, wget li olan kodu konsole yaz.  
inince vi zshrc yazarak kişiselleştirme yapabiliriz. tema vs gibi  
sonra zsh yazarak bu kabuğa geçiyoruz.çıkmak için ctrl d

**Disk Mount İşlemleri;**

Takılan bir HDD nin işletim sistemiyle birlikte çalışması için;

1.HDD nin bölümlere ayrılması

2. Bölümlerin bir dosya biçimiyle biçimlenmesi gerekmektedir.

3.İşletim sisteminin bu dosya biçimleriyle mount edilmesi gerekmektedir.

Yeni bir takılan HDD için sdb1 , sdb2 olarak bölünlenmiş şekilde adlandırır

/dev/sda1 in karşısında boot altında olan \* işareti bootloader dosyasının burda olduğunu söylüyor.

HDD yi ekledik ve fdisk -l komutuyla onun sistedem olduğunu gördük. şimdi disk bölümlemesi yapacaz.

fdsik /dev/sdb(hangi diski yapılandıracaksak partition yapacaksak onu yazıyoz) sonra yapabileceğimiz şeyleri gösteriyor burada ve biz n ye bastık yani add a new partition kısmını seçtik.sonra extend mi primary mi diye sorunca biz p yi seçiyoz.sonra kaç tane oluşturmak istediğimizi sorunca direk entera bas geç yada 1 yaz geç.hangi silindirde başlatayım diyo 1 yaz geç.tek partition oluşturcaksak hdd de default olan 652 yi seçiyoz. sonra partition tipleri çıkıyor hangi dosya sistemine uygun olması gerektiği için..biz p ye bastık sonra linux olarak bizlere gösterdi /dev/sdb olarak gösterdi.bizde m yi seçiyoz.aklımıza yattıysa w ya bas ve oluştur.sonra fdisk -l ile bakıyoz.aşağı kısımda hazır olan partition görünüyor. 1. kısım tamamlandı.

komut satırına mkfs.ext4 /dev/sdb1 yazıyoz. Çünkü oluşturduğumuz partitionı biçimlendircez. enter a basıp hallediyoz. şu an 2. aşama da tamamlandı.

bundan sonra hangi amaçla kullanılması gerektiğini bilmemiz gerekir. bu dosya sistemini bir yere bağlayıp kullanabiliriz.

biz mnt altında bi dizine bağlıcaz. dizin oluşturcaz.

mkdir /mnt/hdd

mount /dev/sdb1 /mnt/hdd

Şu an bağlama işlemi tamamlandı.

cd /mnt/hdd içine gidince lost+found diye bişe oluştu.

df -h ile gidersek mount ettiğini görebiliriz.

Oluşturup mount ettiğimiz Disk boyutunu tam göstermeme sebebi ;

disk dolduğunda login olabilmek için belli bir kısmını login olmaya (root kullanıcısı için erişim sağlamaya ) ayırıyor.

Diskin yaklaşık olarak %5 lik dilimini tutuyor.  
tune2fs -m 0 /dev/sdb1

df -h yaptığımızda belli bir kısmının verildiğini gördük.

reboot edince mount edilen disk gitti. Bunun sürekli her başlangıçta görünmesi için /etc/fstab altına bu diski ekleyip sistemi tekrardan reboot etmemiz gerekir. yani bunu yaparsak kernel gidip bu diski okuyor ve bize bu diski gösteriyor.

1. PAKET YONETİM SİSTEMİ

**YUM-RPM**

vi /etc/yum.repos.d/CentOS- # kurulan paketler, centos da herşey bu dizinin altında   
iki kere tab yaparak altında açılabilecek dosyalar gözükür birisini yazarsak içine girer.

yum # rootla çalışır.  
yum update # tüm güncellenmesi gereken paketleri günceller   
yum search zip | less #zip programlarını listeler  
yum install zip unzip # zip ve unzip programlarını yükle  
internet çalışmıyorsa : dhclient enp0s3 bu kodla çalıştırıyor.  
  
yum install lynx programını kurup bu program ile internete girebiliriz  
lynx Google.com

yum remove unzip #unzip programı silinir  
yum list installed #sistemde yuklu olan paketler  
yum list available #yüklenebilecek programlar

yum grouplist # grup olarak kurulabilecek uygulamlaar  
yum groupinstall “KDE Plasma Workspaces” # arayüz için   
yum groupinstall “GNOME desktop” # arayüz için  
yum groupinstall “Development Tools”

yum search repo # repo ları arıyoruz, paket depoları  
yum repolist  
yum install epel-release # centos un sunmadığı ekstra paketleri sunuyor  
yum info fail2ban # yazılan paket hakkında bilgi veriyor

yum provides \*/netstat # komutun ne olduğunu biliyorsunuz ama paketi bilmiyorsunuz.  
yum history list # yum da yapılan işlem geçmişi

yum reinstall # var olan paketi tekrar yüklemek için  
yum install –t # interaktif yum komutu için, yani yükleyim mi kaldırıyım mı vb sorulara direk yes olarak cevap veriyor

rpm # bu da paket yöneticisi

1. KULLANICILAR – GRUPLAR - İZİNLER

useradd : tüm kullanıcı özelliklerini ayarlayabiliyoruz.  
adduser: otomatik ayarlı kullanıcı

useradd –D kullanıcıismi #sadece ismini yazıyoruz diğerlerini kendisi belirliyor  
useradd –r kullanıcıismi #sistem kullanıcısıları  
su kullanıcıismi # kullanıcı değiştirme

useradd –m -d /home/atyuvasi -u 1905 –c “beyaz at” -g carsi –G games,Wheel -s `which sh` -e 2016-01-02 -f 103

-m ev dizini oluşturulsun  
–d ile home un altında kendi istediğimiz yeri oluşturduk  
-u ile user ailesini tanımlıyoruz. 999 ile 60001 arasında bir değer olmalı  
-c kullanıcı hakkında yorum  
-g kullanıcının özel grup hakkı   
-G dahil olduğu grupları soyluyor   
-s Shell ini değiştiriyor   
e kullanıcı bu tarihe kadar kalsın sonra askıya alınsın   
-f 103 gun sonra kullanıcının parolası sıfırlansın

**passwd**  
passwd kullanıcıadı şifre #kullanıcının şifresini değiştirme   
passwd –l besiktas #kullanıcıyı kilitledi kullanıcının parolası doğru girilse bile girmez  
passwd –u besiktas #lock kaldırıldı

chage –l besiktas #en son ne zaman parola değişmiş vb bilgileri görüntüler  
chage –M 10 besiktas #10 gün sonra parolayı değiştirmek zorunda olucak

userdel –r besiktas # home dizini vs ne varsa siler   
userdel –ı besiktas #şu kadar gün aktivite olmazsa hesabı siliyor

chsh –s `which bash` besiktas # Shell değiştirmek için kullanılır  
chsh -l #kendimizde olan shelleri goruyoruz   
chsh –s `which nologin` besiktas # bu kullanıcı oturum açamaz  
cat /etc/passwd #en sonda yazan aktif shell

usermod –s `which sh` besiktas #kullanıcı ile alakalı her şeyi buradan değiştirebiliyoruz  
usermod –G lp,dis,tty,mem,fyarci besiktas # kullanıcının gruplarını sıfırdan belirler  
usermod –aG games besiktas # -aG diğer gruplar duruyor üstüne games grubuna da ekliyor

**GRUPLAR**

cat /etc/groups #Grupları kontrol etmek için gruopadd boşluk adi

addgroup grupadı – delgroup grupadı  
adduser murat gs # murat kullanıcısını gs grubuna ekler

groupmod –g 1903 carsi #carsi grubunun idsi (1903) değişti   
groupmod –n avrat at # at grubunu avrat yaptı  
groupdel avrat #avrat grubunu siler

w: kullanıcının ne yaptığı hangi ip ile bağlandığı vs

uname –a # kernel ve kernelin ne zaman derlendiği  
cat /etc/os-release # dağıtımın hangisi olduğu surumunun ne oldugu

last #kim ne zamn girmiş ne zaman çıkmış gibi bilgiler  
lastb #sıkıntılı durumları gösteriyor başarısız girişler gibi  
  
usermod –g Wheel vahit # vahiti Wheel grubuna ekledi

**chmod**

r : okuma(4) , w: yazma(2) , x : çalıştırma(1)   
u: dosya sahibi , g : dosya grubu , o : diğer kullanıcılar  
-rwxrw-r-- # kullanıcı tüm haklara sahip, grup okuma ve yazma, diğerleri sadece okuma  
- : normal dosya   
d : dizin  
c : karakter bazında işlem yapan aygıt sürücüleri  
b : blok bazında işlem yapan aygıt sürücüleri  
l : başka bir dosyayı gösteren link dosyaları

chmod u + rwx , g-w, o-rwx deneme # deneme dosyasının sahibi tüm yetkilere sahip grubu okuma ve çalıştırma yetkisine sahip diğerleri hiçbir yetkiye sahip değil  
chmod go-rwx deneme # grup ve diğer kullanıcıların tüm yetkileri alındı  
chmod +x deneme # herkese çalıştırma izni verildi   
chmod 730 dosya ismi # user a tüm yetkiler, grupa çalıştırma ve yazma, diğerlerine yetki yok

**chown**

dosya veya dizin sahibini değiştirmek için kullanılır  
chown yenisahip dosyaismi

**su**

su kullanıcı # login olmak için kullanılıyor  
su –kullanıcı # login yapılan kullanıcının home dizine gidiyor temiz bir sayfa açıyor

su – besiktas –c “ls –lad ~” # kullanıcı değiştirmeden kullanıcı adına komut vermek

**sudo**normal bir kullanıcıya root hakları ile dosya çalıştırma izni verir

**sudo edit**

visudo #

visudo –f /etc/sudoers # kullanıcıların sudo yetkilerini düzenleyeceğimiz yer

ALL/ALL da her yetkiyi veriyor biz ALL u silerek ne yetki vermek istediğimizi kendimiz yazıyoruz

Wheel kısmına yazıyoruz bunu ve kullanıcının whell e dahil olması lazım burada Wheel bir grup

ALL=!/bin/passwd, !/sbin/\*, /sbin/visudo, !–f /etc/sudoers , !/bin/su

# diğer kullanıcılardan sudo yetkilerini almak. başında ünlem olanları yapamıyor. Diğerlerini yapıyor.

# en sondaki kullanıcının su yapma yetkisini alır.

iki yada uc cesıt sey var ya kullanının ısmını yazacaksın mesela root ALL = (ALL) ALL roota tüm yetkiler

mesela

!sbin in altındaki hiçbir şeyi kullanmasın

1. SÜREÇ YÖNETİMİ (PROSES)

ilk bilmemiz gereken proses ID lerdir. Yapılan her işleme verilen bir ID dir.   
Linuxun ilk işlemi init tir. init process IDsi 1 dir.  
Servisler felan inite tanımlanıyor. Bu açılsın şu açılsın gibi. Windows msconfig buna benzerdir.

linux da donanım bilgileri proc altındadır.

cat /proc/cpuinfo

flags işlemcimizin bize sağlamış olduğu şeylerdir. mesela İşlemcinin sanallaştırmaya izin verip vermediğine burdan bakabiliriz.

free –m #ram bilgilerini verir. Ram kullanımını verir.

Linux bazı komutları ve metadataları felan ram de cache ler. Oyüzden ram in bir kısmını felan kullanıyor. buffer/cache lik kısmı eğer çıkarırsak fiziksel oalrak kullanmış olduğumuz ram ortaya çıkmış oluyor. Yani buffer/cache ler beklemede kalıyor eğer ihtiyaç olursa bunları ekliyor. buffer/cache bizim kullanmış olduğumuz programların kullandığı ram miktarıdır.

swap olayı ise; ola ki fiziksel ram yetmedi diski ram olarak kullandırıyor.

Swap olayında diski ayrı partition olarak kullanımı var. Birde normal diskin bir kısmına yani sistemin bir kısmına swap alanı açıp o şekilde kullanım olabiliyor bu da performansı etkiliyor tabikide.

top komutu bizim işletim sistemimizin kaynak bilgilerini veriyor.windowstaki task manager gibi çalışmaktadır.

top altında wa ile gösterilen kısım I/O kısmıdır önemlidir.

top komutundayken h ye basarsak yardım komutlarını çıkarır.

top üzerindeyken k ye basarsak kill eder. kill özelliğinde ise 15 ile 9 var. bunlar durdurur. k ye basıp process id yi girdikten sonra sinyal giriyoz yani 15 yada 9 giriyoz. 15 normal durdurur. 9 ise ağzını burnunu kırarak durdurur.

bunlar kullanılmadan nasıl yapıyoz peki?

ps -sistemdeki processleri listeletiyoz. birçok parametre var. tireli ve tiresiz parametreler var.

ps ax parametreleri sistemdeki bütün prosesleri gösterir.

ps aux parametresi ise user orianted demek. Kimin uygulaması olduğu pId si nedir onları listeliyor.

ps aux |grep process\_adı yazarak çalışan processi listeliyoz. Burdanda process ID yi öğrenip kill komutuyla durdurabiliyoz.

ps faux parametresi bir processin altında çalışan processleri gösteriyor.

bu şu anlama geliyor. Normal Proc. bir de child proc. ler var. normal proc.leri sonlandırırsak child olanlar da otomatik olarak sonlanırlar.

ps -f -p 1 sadece o process id(id si 1 olanı) olanı gösterir

ps -f -p 1,2 1 ve 2 yi gösterir

ps -f -u makif parametresi makif kullanıcısının processlerini verir.

ps -f -C process\_adı o processin processlerini gösterir.

f parametresi hiyerarşik olarak gösteriyor.

kill komutu ;

kill process\_id veya kill -15 process\_id veya kill -9 Process\_id ==> bu processi durdurur.

yes "Gnu" verirsek bunu kill edene kadar Gnu bastırır ekrana.

& işareti bir programın arka planda çalışmasına yarar. (yes "Gnu" & ) bunu yazıp entera basarsak başka bir terminal açıp bunu kill etmek gerekir. yoksa durduramayız.

yes "Gnu" 2>&1 >/dev/null & yazdıktan sonra kareli parantez içerisindeki numara pre-oceslerin sırası diğeri process id yi verir.

bunları kontrol etmek (arka planda çalışanları) için jobs komutunu girebiliriz.

pgrep -l çalısan\_process\_id (eğer l yazmazsak sadece id leri verir)

fg ==> arka plandan ön plana getirme komutu

nohup yes "Gnu" ==>yes processini arka planda çalıştırıyo eğer çıktısını bi yer beirtmediysek bulundugumuz dizinde nohup.out diye biyere yazıyor.

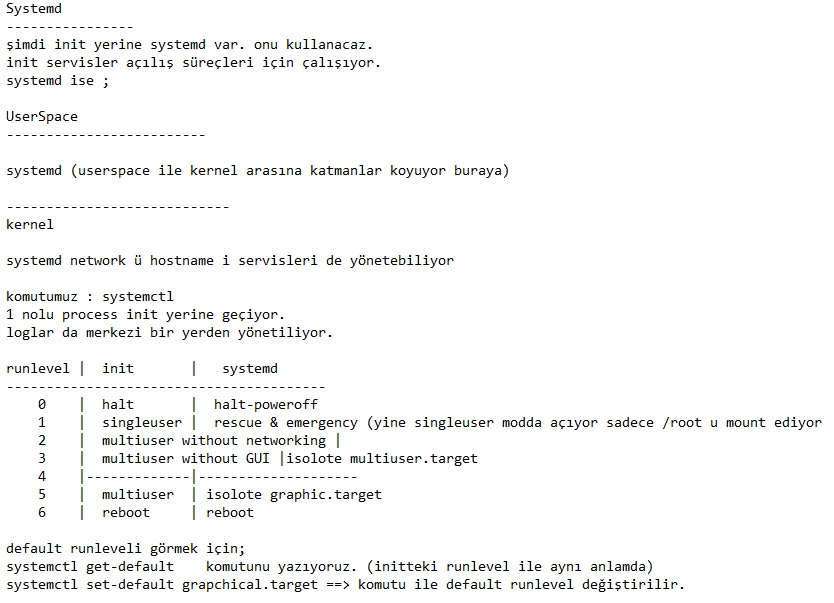
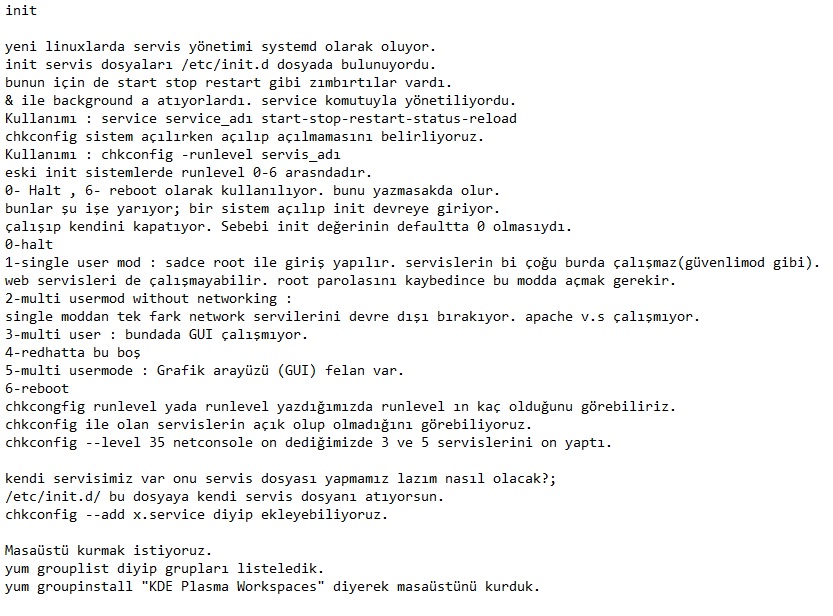
nohup yes "Gnu" > /tmp/yes.out ==> bunu da istediğimiz bi yere yönlendirdik.

watch ==> programın çıktısını izlememize yarıyor  
watch -n 1 ls -la /tmp/yes.out ==> şu an çalışan yes.out procesini 1 saniyelik aralıklarla ekranda güncelliyor.

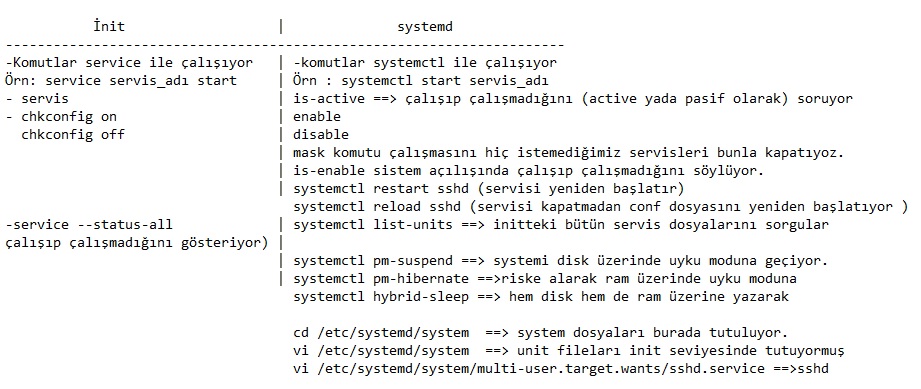
nohup yes "Gnu" > /tmp/yes.out & ==>

disown ==> arka planda çalışan yukardaki komut olsun. disown komutunu verince arkadaki çalışan uygulama child process olarak değilde normal process olarak çalışmaya devam edecektir. yani bağlı olduğu process den alıp normal process haline getirecektir.

1. SERVİS YÖNETİMİ



**init ve systemd karşılaştırması**



**journalctl :** systemd nin logları yöneten kısım.

bir işlemi yada servisi kurup çalıştırmak istedik ama çalışmadı. Bunun hatasına yada neden çalışmadığına journal -u ile bakabiliriz. Burdan da görünen hata ile birlikte sebep olan dosyaya gidip hatayı düzeltip sonra systemctl start servis\_adı dersek çalışırsa çalıştığına bakmak için systemctl status sistem\_adı olarak bakabiliriz.

biz sadece son 10 satırı görünmesini istiyoz. journalctl -n 10 yazıyoz

journalctl -n 10 -o json ==>çıktı formatı belirliyoruz. json olarak bastı.

journalctl -n 10 -o verbose

journal -fn 10 ==> (akan logları listeliyor)(stream eden datayı getiriyor) anlık log izlemeye yarıyor. Yani log düştükçe burda da görüncek

normalde /var/log/secure altından buna bakabiliriz.

/var/log/messages ==>sistem mesajları vardı.

journalctl -p warning

journalctl -p info ==> -p filtrelemeye yarıyor. -p den sonra logun cinsini giriyoz .log öncelikleri var. bunlar info error warning

journalctl -b ==>enson reboot ettikten sonraki kısmı gösterir bize.

journalctl --since="2015-06-11 23:00:00" ==> şu tarihten beri olanları listele

journalctl -ux sshd ==>(x olsada olur olmasada) açıklamalarıyla birlikte logları veriyor. yardımcı metinlerle birlikte veriyor.

vi /etc/locale.conf

cat /etc/locale.conf/

localectl status ==> local ayarların ne olduğunu gösterir.

localectl list-locales ==> sistemde yüklü olan kullanabileceğimiz lokallerin çıktısını verir.

localectl set-locale LANG=en\_US.UTF-8 ==> lokal bu şekilde set ediliyor

bunu kontrol etmek için ya /etc/locale.conf dosyasına yada localectl status ile bakabiliriz.

localectl list-keymaps ==> klavye dizilimlerine burdan bakarız.

localectl set-x11-keymap en ==> komutu ile klavye dizilimlerine en yi eklerizset-keymap kullanırsak eğer her ikisini değiştirir. x11 grafik arayüzünün klavyesini değiştirir.

**date**

date ==> saatin ve tarihin kaç olduğunu belirtir.

date +"%Y" ==> yılı gösterir

date +"%Y%M" ==>Yıl ve dakikayı birlikte gösterir

date + "%Y--%m--%d//%D" ==> tarihi ilk önce - lerle ayırır sonra birlikte yazar

date + "%Y--%m--%d // %H:%M:%S %Z" ==> time zone ile birlikte tarih ve saat verir.

date --set 2014-12-30 23:31:10 ==> bu şekilde tarih ve saat değiştirilebilir. (sakın deneme)

cal 1988 yaz doğduğun yılın takvimini, verir

timedatectl ==> buda systemd tarafında date işlemidir.

timedatectl list-timezone ==> timezoneleri verir

timedatectl set-timezone Europe/Istanbul ==> time zone yi istanbul oalrak ayarlar

timedatectl set-time ==> pek deneme sakıncalı olabilir

yum search chrony ntp ==> ntp clasic olarak yazılır. ntp sunucularına bizim saatimizi otomatik atar. chrony ise yeni geldi. bunlar beraber yazılmayacak ayrı ayrı yazılacak.

vi /etc/ntp.conf ==>ntp nin ayarları burdan yapılıyor.

ayarlar yapılırken ntp nin kapalı olması lazım  
ntpdate pool.ntp.org (ntp sunucusu girdik) sonra servisi yani ntp yi start edecez sonra systemctl status ntp ile çalışıp çalışmadığına bakacaz. sonra date diyerek tarihe bakabiliriz. saati ve tarihi buradan çekmiş oluyor.

1. SİSTEM KAYITLARI

SED (stream Editor, akgul hoca)

txt dosyalarında satır temelli işlem yapar. Şu adresleri şu satırları sil gibi işlemler yapar.

Genel Komut :   
sed option sed\_komutları dosyalar

sed komutlarının genel formatı : address {eylem}

**adres olarak**;

sayı belirleriz 1 diyebiliriz

$ diye son satırı belli ederiz.

3,89 diyebiliriz.

içinde /inet/ geçen satırlar diyebiliriz.

1, /^$/ diyebiliriz.

/ayse/,/elif/ içinde ayse geçen satırdan elif geçen satıra kadar diyebiliriz.

/elif/, $ diyebiliriz.

**Eylem olarak**;

s/regex/hedef/ g koyarsak başındaki s ifadesi sayesinde regex olan

ifadeleri hedef olarak değiştirebilirz demektir.

Ayraç olarak / değilde : \_ | ; bunları kullanabiliriz.g global anlamındadır.

d (delete) o satırı sil anlamındadır.satırı olduğu gibi siliyoruz.

p (print) defaul olarak herşeyi basıyor. Default olarak basma benim sölediklerimi bas demektir.

Örn : Sed -n 'p' Dosya\_adı ==> n parametresi default olarak basma anlamındadır.

cat dosya adı ile aynı işlemi görür.

p nin haricinde ;

r read anlamında ==> r file

i include ==> satıra dokunmuyor. bi satırın yada adresin öncesinde işlem yap

a append ==> satırın arkasında işlem yap

c change ==> satırı tümden değiştiriyor. (değişiklik yapmak için)

w içinde verilen bir kelime yada harfleri bi dosyaya yazar

Örn : /inet/w inet.txt ==> içinde inet olanları inet.txt ye yaz

**Option olarak**;

-n bas diyince default basma

-f sed dosyalarını buna basıyor burdan okuyo

-e '....', -e '...' küçük sed komutlarını birleştirmek için kullanılır.

/ /! ==> mesela a ile b arasını istiyosan burda a ile b arasının haricindekileri veriyor.

**Örn:**

sed '/^$/!w AA1.txt' AA.txt

sed '/^$/d' AA.txt > AA1.txt yukarıdakiyle bu aynı işi yapar. boş satırları silip AA1.txt dosyasına gönderir.

^$ olan kısım boş satırları gösterir. W ise kendisinden sonra gelen dosyaya gönderme yada yazma yapıyor.

W dan önceki ! işareti ise boş satırların haricindeki kısımlar al demek oluyor. d ifadesi ise boş satırları sil anlamındadır.

sed 's/^/ELIF /' AA.txt ==> Bütün satır başlarını ELIF boşluk ile değiştir anlamındadır.

sed '/elif/I s/^/AYSE /' AA.txt ==> başında elif yazanları Ayse olarak değiştirip büyük küçük harf ayrımı yapmadan yazdır anlamındadır. I harfi büyük küçük harf duyarlılığına bakıyor.

sed '$ s/$/ BITTI/' aa.txt ==> son satırın sonuna BITTI yazısını ekledik. ilk $ işareti son satırı belirtir.

2. $ işareti ise sonuna ekleyeceğimizi belirtir.

sed '10,20 s/^/ELIF /' A.txt | sed '10,20 s/$/ BITTI/' |head -20 ==>

10. ve 20. satırlar arasında satır başına elif satır sonuna bitti yazdırdık (bu 1. yöntem)

sed -e '10,20 s/^/ELIF /' -e '10,20 s/$/ BITTI/' A.txt ==> bu da 2. yöntem

cat A.txt | sed '/^inter/I s/$/Minternet/' |grep Minternet ==> satır başında inter olanların satır sonuna Minternet yazılacak.

DİPNOT : 1~2 dersek 1 den başlayıp 2şer atlayarak gider. (1..3..5..7 gibi)

sed -e '1~2 s/$/ TEK /' -e '2~2 s/^/Cift /' A.txt ==> tek numaralı satırların sonuna

TEK, çift numaralı satırların başına Çift yazdıracak.

sed '/x/s/a/ABC/' ==> x geçen satırlarda ilk bulduğu a yı ABC ile değiştircek

sed 's/a/AAAA/g' ==> bütün satırlarda he gördüğü a yı AAAA ile değistircek ve satır kescek

sed 's/ /\\\

/g komutu boşluklarda \ atıp aşağı satıra geçer

Yani \ 'ın ekran görünmesini istiyosak kendisiyle birlikte yazmamız lazım

**SED (fatih yarci)**

sed '1,10 s/x/y/g' 1 ile 10 arasındaki tüm satırlarda x yerine y yi yaz

sed ' /^at/d' tüm satırlarda at ile başlayanları sil

sed ' /^$/!w x.txt' y.txt ==> x.txt deki boş satırların haricindekileri y.txt ye aktar

s/,/\

/g ==> virgülleri satır sonuyla değiştiriyor. buradaki g nin anlamı bir satırda birden fazla x varsa 1. xsi değiştirip diğerlirini de değiştirmesi demektir. Yani sadece birini değiştirip bırakmıyor hepsni değiştiriyor global olması ordan geliyor.

sed \%ozet/%s%"\(ozet/.\*\)"%\1.html%

Bir sed de tırnaklar arasında aralık belirliyoz. ''

yukarıda özet/ olarak verilen kısım bunu anlatıyor.

s/ "\(ozet/.\*\)" burada \ lar parantezleri kaçırmak için kullanılmış. burada özet ile başlayanları bir grup haline getiriyor yani topluyor birleştiriyor.

**SED KODUNUN BÖLÜMLERİ**

Sed de ilk kısım adresi yazıyor (burada yolumuz özet geçen satırlar)

2. kısım ne yapcağımızı belirliyoruz. (ne yapacağımızı s harfinden anlıyoruz.yani değişim yapacaz anlamına geliyor.)

burada ayracımız % olarak belirlenmiş ama bu farketmez istediğimizi kullanabiliriz. ama kullandığımız bu işaret bizim aradaığımız yada yazdığımız script içerisinde olmaması gerekir.

3. kısım ise '' arasında özet/ ile başlayıp herhangi birşeyy ile devam edecek.

4. kısım ise en son da \1 ile belirttiği alan (ozet/.\*\) ile gruplanan kısımda \1 ile görünen yere html yazacak.

**AWK**

Kayıtlar üzerinde işlem yapar. Satırlar üzerinde yapar. Filtrelere ayırarak yapar.

Aritmetik işlemler de yapılabilir. While if do gibi komutları kullanılabilir. Ciddi bir programlama dilidir. Google awk tutorial

rsync -av lyk.ibu.edu.tr:pub

AWK yapısı ==> awk options awk\_command dosyalar

pattern {action}

pattern filtreler ==> $1 $2 $3 ... $20

tüm kayıta $0 ile ulaşılır.

onun parçaları $1 $2 gibi ayrılabilir.

awk ile sayıları toplayabilir ortalamasını alabiliriz. string işlemlri yapılabilir aritmetik işlemler yapılabilir. while do if dahil yapılabilir.

AWK Builtin Variables

RS (record Separator)- kayıtları nasıl ayırırız onu belirtir.

Filename - o an çalıştığımız dosyanın kayıtlarına erişebiliriz.

awk '{print $2 "\t" $1}' ==> araya tab atarak 2. ve 1. sütunları yazdırdı

awk '!/^$/ {print $2 "\t" $1}' notes.txt | awk -f AWK ==> komutları awk dosyasından oku

BEGIN{FS="\t"} ==> field leri tab ile ayırıyorum anlamındadır.

$NF ==> son field demek

$OFS ==> output field separator demek yani ortadan tab işlemiyle ayır anlamında

paste iki tane dosyayı yan yana yapıştırıyor

awk '{print $1,$4,$NF}' xdata.txt ==> burada 7 satırlık bir dosyamız vardı bu dosyada 1,4 ve en son field ı ekrana bastı

RS ==>Kayıtları birbirinden ayırmak için kullanılan separator

\n\n iki tane bundan olması demek bir tane boş satır demektir.

awk -F: '{print $1,$6}' /etc/passwd ==> -F: parametresi ile field separator ı belirtiyoruz. Burada : field separator olarak belirlenmiş oluyor.

BEGIN{OFS="\t"}

{split($0,a,":")

print a[1],a[6]}

Bunun anlamı;

$0 bir kayıt,a bir array adı , ":" bir ayrac anlamında. en son satır ise birinci ve 6. field ı bas

cat ynotes.txt|awk '{print $2 "\t" $1 }' |awk -f Awk.6 ==> 2. sütunları 60 tan büyük olanları ekrana basacak.

rsync -av akif.txt linux@lyk.ibu.edu.tr::test şifre : ozgur

dosyanın gittiğini kontrol etmek için;

rsync -av linux@lyk.ibu.edu.tr::test/akif.txt /tmp

ls /tmp diyip dosya tmp de görünüyorsa dosyamız yüklenmiş ve geri indirilmiştir.

rsync kısaltması remote senkronizasyon demektir.

touch ali.txt veli.txt elif.txt ayse.txt LYK dizininde oluşturuldular.

rsync -av /LYK /tmp/AB (LYK dizinini AB dizinine taşıdı.

touch /tmp/AB/xyz{1,2,3,4}.txt ==> xyz1 xyz2 xyz3 xyz4 şeklinde 4 tane dosya oluşturuyoruz

rsync -av /tmp/AB /LYK ==>/tmp/AB içindekileri bu dosyaya taşıyoruz.

Şu an LYK dizinindeyiz.

rm ali.txt veli.txt elif.txt ayse.txt

rsync -av --delete . /tmp/AB ==> yukarıdaki dosyaları sildi ve /tmp/AB dosyasını güncelledi.

cat /etc/rsyncd.conf ==> bu makinedeki public arşivi

rsync protokolü ile bişey yükleyebilmek için root yetkisi olmalıdır.

rsync -av akif.txt 139.179.130.100:/tmp ==> ip si verilen makinenin içine akif.txt dosyasını bu şekilde gönderebiliriz.

rsync -av akif.txt user@139.179.130.100:/tmp ==> eğer bu makine ile ip si verilen makinenin user ları eşleşmiyorsa yada o farklı bir kullanıcı ise başına bu şekilde user@ yazmalıyız.

**8. NETWORK**

(Barkın Kılıç ==>barkinkilic@gmail.com)

**OSİ KATMANLARI**

7- fiziksel katman

fiziksel katmanın içerisinde pc ye takılan kablo, o kablodaki elekt. sinyali vs. bunlar yer alır.

Network altyapısı için gerekli olan donanım kısmını temsil eder.

bizi çok fazla ilgilendiren bir kısım değil. Ağ kartı diyebiliriz(eth kartı, wifi kartı)

6- Veri Bağlantı katmanı (data link)

Fiziksel katmandan gelen verileri iletmek için gerekli olan protokolleri koşturan katmandır.

ethernet için mac protokolü kendini bulur.

mac protokolü içerisinde ilişkili olan arp protokolü veya RAPP protokolü vardır.

5- Network (AĞ) Katmanı

içinde yer alacak olan protokol ip katmanıdır. Bunlar da ipv4 ve ipv6 dır

4- Taşıma (Transport) Katmanı

TCP, UDP, ICMP

3- Oturum (Session) Katmanı

genelde bağlantı sağlandıktan sonra tcp oturumu sağlandıktan sonra üst taraftaki

uygulamalara kendi içinde bir oturum varsa onun içinde yer alan protokollerdir.

Burada Point to Point kullanılabilir.

Maksadı ise, sadce belirli erişimleri sağlamak adına veya bir kişiye sağlamak adına

çeşitli mevcut protokoller oluşturuluyor. bu na da örnek point to point veya VPN verebiliriz.

2- Sunum (presentation) Katmanı

Uygulama katmanının anlayacağı bir şekile dönüşümün sağlanması gerekir bu da bu katmanda olur.

Bunlarda text , ascii , binary

1- Uygulama (application) Katmanı

Diğer katmanlardaki bütün herşeyin hazırlanıp son kullanıcının ekrana getirilmesini sağlayan katmandır.

**Ethernet protokolü** : Ağ kartınızın elkt. sinyallerini iletirken belirli kontrolleri gerçekleştirip belirli protokollerin koşturğu yerdir.

Bakır kablo üzerinde sinyalin nasıl iletileceği, iletirken neye dikkat edileceği burada yer alır.

**ARP Protokolü** : gerçek hayattan ; pc ler haberleşirken ip ile haberleşirler ama fiziksel olrakta haberleşirler. aynı fiziksel lokasyonda bulunan pc lerin konuşacağı ip ile iletişime geçmeden bu ip adresine sahip olan kimse bana belirtebilirmi diye sorup o şekilde haberleşebilir. karşıdaki ip ise elini kaldırıp benim diyip ona bilgilerini verir. bundan sonraki cevap ve iletişim kendi aralarında gerçekleşir. Bunla beraber broadcast ve unicast kavramları gelir.

**Broadcast** : herkese bağırarak konuşma anlamında

**Unicast** : birebir konuşma anlamında

**RARP Protokolü** : fiziksel Lokasyonu yada adresini bilip onun ip adresini direk ona sormak RARP protokolüdür.

her ağ kartının kendine ait bir fiziksel kimliği olması gerekiyor. Buna da mac adresi denir. --:--:-- Üretici firmanın kendisine ait olan kısmı diğer 3 kısım ise device id olarak geçer ve o makinenin kendisine ait olan kısmıdır. yani üreticinin ürettiği aygıtın id'sidir.

Aygıttan kasıt network (ağ) kartıdır.

**ipV4 Protokolü** : adreslemeyi tanımlayan protokoldür. ip adreslerinin total uzunluğu 32 bittir.

2^8 bitten oluşan 4 parça şeklinde oluşur.

bir ağ oluşturma yada toplulukta grup oluşturma nasıl yapabiliriz.

Bunları bir gruplama modeline nasıl yerleştiririz. Ozaman subnetmask (Alt Ağ maskesi) kavramı ortaya çıkıyor.

Alt ağ maskesinin olmasının sebebi gruplar oluşturmaktır.

192.168.1.255 ip adresi vermek istiyorsam öncelikle alt ağ maskesini bilmem lazım. 255.255.0.0 verilebilir. 255.255.255.0 verirsek bunun broadcast adresi olur verilemez. O yüzden 255.255.254.0 vermemiz lazım. yani ibrenin sağ tarafının tamamının 1 olması için ibrenin yada sürgünün bir tık solda olması gerekir

5tane ağ lazım ise bize;

ibrenin sol taraf sabit kalacak. ibre solda n 1.nin sağına koyunca 0 ve 1 olmak üzere 2 tane ağ tanımlayabilirim. 2.nin sağına koyunca 4 tane , 3. sağına çekince 2^3ten 8 tane ağ oluşur bu da bize 5 tane ağ oluşturmak için yetecektir. sol tarafın hepsini sıfırladım. neetwork id si için sol 0 sağ 0 olunca 0.0.0.0/3 olur bu en çok verebileceğim network olur. makinelere ip vermek istesek sağ taraf 1er 1er artar. yani 0.0.0.1 gibi artar. bunun alt ağ maskesi ise 3. nün sağına kadar ibreyle geldiğimiz için 128+64+32 = 224 olur. Gerisi 0 olacaktır. O zaman 224.0.0.0 olacak şekilde verebiliriz.

NOT: ağ lazım ise soldan itibaren bakarak geliyoruz.

her grubun içinde max 35 tane ip lazım..

yöntem bu sefer sağ taraf host olduğu için sağdan gelmektir. sağdan 6 tane gelmemiz lazım. sol tarafın tamamı 0 olarak düşünürsek sağ tarafı da 0 olarak düşününce bir network olur. 0.0.0.0/26 olarak network tanımlayabiliriz. alt ağ maskesi olarak ise 255.255.255.192 diye tanımlayabiliriz.

NOT: gruplara ip lazım ise sağdan itibaren geliyoruz.

sınıf mantığı şudur ; 5 adet sınıf vardır. A B C D E Sınıflarıdır bunlar.

Nasıl kullanıyor?;

1-126 arası A sınıfıdır. yani network olarak 0.0.0.0/8 dir. 8 verilmesinin sebebi 8 tane subnette 1 var demektir. neden 126 ya kadar çünkü network ve broadcast ip leri verilemediği için.

128-191 arası B sınıfıdır. yani 0.0.0.0/16 dir.

192-223 arası C sınıfıdır. yani 0.0.0.0/24 dür.

D classı multicast için kullanılır yani makinelere verilmez.

E class da rezervedir. bilimsel çalışmalarda kullanmak için saklanmıştır.

192.168.2.1/32 demek bir tane hostu tanımlamak istemesidir. /32 diye tanımlamak hostu tanımlamak demektir. /32 nin haricindeki tanımlar ağı tanımlamak anlamına gelir.

0.0.0.0/0 bu da internetin tamamı demektir.

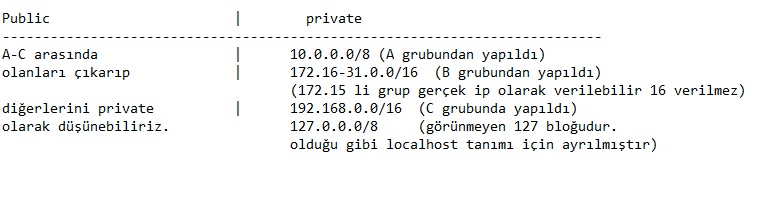
**public - private işlemleri**

classların tanımlı olduğu aralıklarda 127 gibi görünmeyenler var.

public interneti herkesin üzerinde koşturduğu grup

private özel olarak koşturulan grup

Bu işleme NAT denir. Network adress Translation



**ROUTİNG (paket yönlendirme)**

Bulunduğumuz yani erişim alanının dışında olan yere isteğimiz olduğunda posta kullanıyoz. bu tip te bir yöntem kullanılıyor. Bizim için vekillik yapması bunun maksadı. fiziksel bir iletişim yok. pc ile pc haberleşmek istediği zaman gönderim yaptıkları kurallar routing kavramıyla yapılyıor. Routing, Göndereceğimiz paketlerin ne şekilde iletileceği ile ilgili bilgileri içeriyor.

ilk etapta bir makine başka makineyle haberleşmek için network paketini hazırlayınca dikkat etmesi gereken şey şu oluyor;

10.0.0.11 böyle bir ip adresi olsun.

Makineninde kendisine ait ip adresi de 192.168.0.3/24

10.0.0.11 i aldıktan sonra yapacağı iş ilk etapta benimle aynı ağ içerisinde mi değilmi ona bakıyor. buna karar veren subnetmask tır. niye ulaştırmak için ne yapmasına karar vermek için. bunun için bi aracıya ihtiyaç var. bunu da routing denilen tablolarla sağlarız.

tabloda ;

10.0.0.11 arayüzünden interface ile 192.168.0.1 üzerinden şu paketi gönderirsin diye bi açıklama varsa interface üzerinden konuşarak kendisine bi msj ı olduğunu söylemesi lazım.

aynı ağda ise ilk olarak arp işlemini yapıyor. 192.168.0.1 ip adresi kimdeyse bana mac adresini söylermisin benim bilgilerim bu diyecek. dönüş broadcast ip adresi üzerinden olacak(192.168.0.255).

routerların gateway diye bi kimliği olabiliyor. Gatewayler int ortamında birbirleriyle bağlantı içerisinde o ağın haberleşmesinde paketlerin gönderilmesi iletilmesinden sorumludurlar. default gateway, iletişim kuracağım makineyle alakalı bilgi yer almıyorsa bu işlem yapıyor.

**TCP-UDP**

Haberleşmek için kullanılan ana protokollerdir.

**UDP** : shoot and forget(ateş et ve gerisini unut)

paketi gönderdikten sonra hiç birşeyle uğraşmıyoz. Tamamen performans amaçlıdır. Gönderilen paket hedefe ulaştımı ulaşmadımı bunları kontrol etmez.

udp nin paket başlık bilgileri;

udp için source port ve destination port önemlidir.

paketin içine bunları yerleştirip gönderiyoz.

paket karşıya ulaşmış olabilir ama paketi alan servis çalışmıyor olabilir.

**TCP** : işin içine kontrol mekanizmaları giriyor.

source port destination port bundada var.

sequence number acknowledgment number tcp de devreye giriyor.

ben senle haberleşmek istiyom hazırmısın diyo.

karşı tarafta buna karşılık veriyor oda gönderiyor sonra karşı taraf ben bunu aldım diyor. bu şekilde karşılıklı onaylaşarak alış veriş yapılıyor. Başlatma da aynı şekildedir kapatmada aynı şekildedir. Oturum açılarak yapılır. kapatma işlemi de birbirlerine haber verilerek yapılır. burdaki maksat paketler gönderildi. diğer taraftan tepki gelmedi. gönderidği paketlerle ilgili bilgi alamazsa tekrar gönderir taaki onay alana kadar. belirli bir zaman aşımı süresi vardır. maksat bir timeout süresinin dolmasıdır. bi müddet sonra askıda kalan paketlerden dolayı network meşgul olur. en sonunda onay alamazsa vazgeçer ve paketi göndermez.

**ICMP (internet control messaging Protocol) Protokolü** : görevi haberleşmenin düzgün bir şekilde ip protokolü üzerinde sağlanabilmesini yapmaktır. Tcp yapınca port kapalıysa reset paketi döner. UDP paketinde ise bu olay icmp paketiyle uyarı verir. Data göndermeye çalıştığın hat çalışma kapasitesi 5x olsun. 6x lik bi paket göndermeye çalışırsan benim msj size ım buna yeterli değil diye cevap döner.

NOT: ip paketi içerisinde TTL bilgisi vardır. eğer routing işi içinde olaki bir hata söz konusu ise bi paket sonsuz döngüye girmesin diye bu vardır(yani TTL).

herhangi bir problem varsa bunla alakalı bilgilendirme gerekiyosa bu protokol bu işe yarar.

**İki Temel Servis**

1- DNS : Adres çözümlemek için kullanılır.

2- DHCP : Network konfigürasyonlarını almak için kullanılır

**DNS ;**

Maksadımız ,isim yapısı oluşturup karşılığı bulmayı sağlamak.

Örneğin; www.lkd.org.tr alan adının çözümlemesine yardımcı olmak için dns kullanılır.

dig komutu nslookup ile aynıdır. Dns sunucularında dns sorgulaması yapıp cevap aldğmz şey.

DNS kayıtları;

cname ==>A kaydına referans olarak gösterdiğimiz kayıtlar

www.or

A ==> isimden ip ye çevirmeye yarayan kayıttır. Aynı A kaydını birden fazla takma ad ile tanımlamaya yarar. Örn;

www.linux.org.tr adresine dig komutunu verdiğimizde aslında bu adres bir cname dir asıl adresi eti.linux.org.tr dir diye bize bilgi veriyor.ALIAS (takma ad) olarak düşünebiliriz.

AAAA ==> ipV6 dan geliyor

TxT kaydı ==> bilgi amaçlı kayıt barındıran kayıtlardır.

SPF kaydı ==>TxT'nin Spam için özelleştirilmiş halidir.

PRT kaydı ==> ip yi adrese çeviren kayıtlardır.

NS kaydı ==> O domain için sorumlu olan DNS bilgilerini tutar.

MX kaydı ==> O domaine sorumlu olan mail sunucu bilgilerini tutar.

SOA kaydı ==> Alan adının sorumlularının DNS ve mail adres bilgilerini tutar.

**DHCP** : dahil olduğunuz ağ içerisinde ip adresi almak için uğraşmaktansa bir servis üzerinden bu işlemleri yapabiliriz. bu da DHCP dir. makineler ağa bağlandığında burda dhcp servisi varmı diye sorar var diyip ip adresini kiralama işlemini belli süre boyunca yaparak ordan ip alır. Kiralama olmasaydı sürekli aynı ip de kalacaktı. O da ip sıkıntısı yaşamaya sebep olacaktı.

Belli ip aralığında da ip adresi verilebiliyor.

sistemde iki tane dhcp server varsa hangisinden önce cevap alırsa ondan ip adresi alır.

arp-scan -I eth0 -l ==> bulunduğumuz ağ üzerinde kimler varsa onları çıkartıyor.

9.SSH

# Open-Ssh Kurma:

sudo apt-get install openssh-server

Uzaktaki bir sunucudaki terminale bağlanmak için kullanılıyor.

Standart portu 22.

# Bağlanma :

ssh ip -l kullaniciadi

veya ssh kullaniciadi@ip

Eğer port tanımlamışsanız :

ssh kullaniciadi@ip -p2222 [2222 : Sizin belirlediğiniz herhangi bir port olabilir(Standart 22'dir.)]

# Ssh configure dosyaları:

vim /etc/sh/sshd\_config

# Ssh port değiştirme :

Port 22 [Yazan kısımdaki sayıyı değiştirin]

# Ssh key'lerinin bulunudğu yer

~/.ssh/known\_host

#ListenAddress

Hangi ethernet kartını kullanıcağını yazabilirsiniz.

#UsePrivilegeSeparation

Servis açıldıktan sonra root yetkisi gerektiren işleri yapar sonra standart kullanıcıya döner.

#KeyRegenerationInterval

Ne kadar sıklıkta iletişim kurulacağı

#ServerKeyBits

Anahtarın uzunluğu

#LoginGraceTime

İlk bağlanma aşamasında karşı tarafın şifreyi girme süresi

#PermitRootLogin no

Bağlanırken root ile bağlanmaya izin verilsin mi? (Güvenlik için verilmesin)

# Ssh servisini başlatma/durdurma/durumunugörüntüleme

/etc/init.d/ssh [start/stop/status]

# Scp

Uzak sunucuya dosya gönderme ve almak için

# Karşı sunucudan veri alma :

scp kullaniciadi@ipadresi:~/dosyaadi.txt /home/ensar/dosyaninindirilmesiistenenyer -P 22

-P [sayi] = Port numarası

-l [sayi] = tek seferde kaç kb'lık veri göndereceği ayarlar, bant genişliğini ayarlar.

Karşı sunucuya veri gönderme :

scp /home/ensar/gonderilecekdosya kullaniciadi@ipadresi:~/gonderilecekklasor -P 22

Ssh ile karşı tarafta uygulama başlatma

ssh kullaniciadi@ipadresi -x

-x : Karşıda uygulama başlatmak için

Bağlandıktan sonra terminale uygulama adını yazın enterlayın. Program karşı tarafta çalışacak görüntü sizin ekrana gelecektir.

Ssh ile aradaki bağlantı ile web proxy

ssh kullaniciadi@ip -D8080 -P 22

Özellikler > Ağ > Ağ ayarları > Manuel proxy > SOCKS\_HOST 127.0.0.1 ayarla

Ssh ile vpn için bu ayarı girin:

socks-proxy 127.0.0.1

route 209.200.110.70 255.255.255.255 net\_gateway

# Open Ssh

Ssh 'da 2 adımla doğrulama:

http://www.syslogs.org/ssh-2-step-verification-google-authenticato/

# Kurmak için;

sudo apt-get install openssh-server

# veya ;

dpkg -L openssh-client openssh-client

# İnittab

vim /etc/inittab

Default-start :

0 Kapat

1 Restart et

2 Grafik arabirimiyle aç...

# w = ile sunucuya kimler bağlı görebirsin

ssh kullaniciadi@ipadresi [komut]

# netstat -antp |

# Sleep

sleep 100 [100 saniye uyur]

# jobs

Arkada çalışan sleepleri gösterir.

Komutu getirmek için:

fg 2

ps aux | grep sleep | grep -v grep | awk '{print $2}' | xargs kill

# Debug Level

-v, -vv, -vvv = Programın arkada ne yaptığını ekrana yazar. V sayısı arttıkça daha çok bilgi verir.

ssh -v kullaniciadi@ip

Komut satırı -X

~/.ssh/config

ssh kullaniciadi@ip -o Compressions=yes

# Stat

stat dosyaadi.txt

Dosya hakkında metadata bilgileri verir.

# Son değiştirilen dosyaları listeleme :

find . -type f -mmin 1 [son bir dakikada değiştirilen dosyalar]

find . -type f -mtime 24h [son 24 saatte değiştirilen dosyalar]

find . -type f -amin 1 [son bir dakikada erişilen dosyalar]

m = modify

a = access

c = creat

# Ssh-keygen

# Oluşturma :

ssh-keygen -t rsa - 2048 -f dosyaadi\_rsa "kullaniciadi@ip -p[portnuamrasi]"

# Key'i karşı tarafa gönderme:

ssh-copy-id -i ~/.ssh/id\_rsa.pub kullaniciadi@ip

# Sshd Configure

vim /etc/ssh/sshd\_config

Ssh isteği hangi interface'den gelsin.

ListenAddress ::

ListenAddress 0.0.0.0

ssh -D:8080 -fN kullaniciadi@ip -p 2222

-f = Arka planda calis

-N = Komut gondermeyecegim sadece tunel icin kullan

ssh -fN kullaniciadi@ip -p [portnumara] -L 111:94.73.162.50:110

# Ters Ssh (Reverse Ssh)

ssh -R [port]:localhost:22 kullaniciadi@ip

# Ngrep

ngrep -Wyline [aranankelime] port [sayi(80)] or port [baskabirport]