**İŞLETİM SİSTEMİ**

Bilgisayar,çeşitli donanım cihazlarının bir araya gelmesinden oluşan cihazdır.Bu cihazın içindeki donanımların birbirleri ile ve çevresi ile anlamlı bir bütün halinde hareket etmesini sağlayan yazılımdır. İşletim Sistemi bilgisayarın tüm donanımlarını yönterek birlikte çalışmasını sağlar.Böylece sabit bir diskten okunan verinin işlemciye gönderip işlendikten sonra ses ve görüntü olarak çıktı alınması sağlanabilir.Ayrıca işletim sistemi dış çevre iletişimi, Klavye,mouse hareketleri vs. bilgisayarlar arasındaki iletişimi de sağlar.

*İşletim Sistemleri:*

Örn:DOS,Unix,Linux,Windows,Macintos

-Solris bir UNixtir., Macintos bir Unixtir.(yani işletim sistemi ailesi olan UNix in bir dalıdır.)

-Linux bir Unix değilir.İşletim sisteminin Unix olması için POSIX standartlarını sağlaması gerekir. Linux sağlamamaktadır.

-GNU/LİNUX şuan kullandığımız işletim sisteminin asıl adıdır.Yani GNU projesi c derleyici ile derlenmiştir.(gcc ile) Kernel.org tan linux çekirdeğini görebiliriz.Andorid e de bu yapılmıştır.Google linux kernelini aldı ve derledi.Bu yüzden artık GNU/Linux olmaktan çıktı.Çünkü GNU C derleyici ile derlenmiştir.Android in de çekirdek kısmı açıktır.Linux ta sanallaştırma desteği Xen ile sağlanır.Linus Torvals aslında Xen i çekirdeğe hiç kabul etmemiştir.(Kodu çok karmaşık geldiği için)Bunu üzerine sanallaştırma KVM ile sağlanmıştır.

LİNUS TORVALS:ilk olarak, üniversitenin online sistemini sevmediği için kendi bir program yazarken linux çıkıyor ortaya ve Stallman ın başlattıgı FSF derneğinde geliştirilen UNIX için bir tamamlayıcı oluyor.

Not1:Centos, Redhat in klonudur.Redhat paralı, Centos ücretsizdir.

Not2:Açık kaynak kod: Yapılabileceklerin istenildiği gibi yapılabileceği anlamındadır.

İşletim Sistemi donanım kaynaklarınıda yönettiği için aynı sistem üzerinde farklı amaçlara hizmet eden yazılımlar kurulup kullanılabilir.

***UNIX TARİHİ***

1960 larda AT-T Bell Laboratuvarlarında vaşlayan, Multics isimli bir işletim sisteminin devamı olarak C Programlama dili ile yazılan UNIX işletim sistemi kısa zamanda üniversiteler ve araştırma kurumlarında ilgi odagı olmuştur. 1980 lerin başında ise donanım üreticileri, kendi UNIX sistemlerini geliştirmiş ve lisanslayarak para kazanmayı amaçlamışlardır ve o günlerde artık Unix bir işletim sisteminin adı olmaktan çıkıp, işletim sistemi ailesinin adını almıştır.(SysV/BSD)Bu gelişmeler sonucunda, başta UNIX in geliştiricileri olmak üzere bir çok kişi, Richard Stallman tarafından başlatılan ve amacı UNİX türevi ve ücretsiz dağıtım geliştirmek olmak GNU Projesine destek vermiştir. GNU daha sonra Stallman tarafından kurulan FSF projesi ilede desteklenmiştir. Amaç; özgürce dağıtılabilecek ve kaynak kodları açık yazılım geliştirmektir.

***LINUX TARİHİ***

Üniversitelerde bilgisayar bilimleri öğrencilerine UNIX öğretmek için kullanılan UNIX, donanım üreticileri tarafından lisansı alınıp, donanım üreticileri tarafında lisansı alınıp ve çeşitlilik arttıktan sonra, yeni bir işletim sistemine gereksinim duyulmaya başlanmıştır.Bu amaçla MINIX işletim sistemi ortaya çıkmıştır.1990 lı yılların başında Bilgisaya bilimleri öğrenci olan Lİnus Torvals,MINIX işletim sisteminden okulun sunucusuna bağlandığı programı yeterli bulmayıp,diskten çalışan kendi uygulamasını geliştirmeye başladı.Yazdığı bu uygulamanın belli bir noktadan sonra işletim sistemi gibi çalışabileceğini fark eden Torvals, comp.minix.os haber grubuna duyurdu.Ardından GNU projesi gcc(c derleyicisi ile) ve intel mimarisi üzerinde çalışan ilk Lİnux sürümünü açık kaynak kodlu olarak dağıttı.Bu yeni ÇEKİRDEK yazılımı diğer geliştiricilerinde dikkatini çekti ve GPL lisansı altında, GNU/Linux işletim sistemi olarak bugunkü adını aldı.

***LİNUX UN TEMELLERİ***

\*ÇEKİRDEK

\*LİSANS

\*DAGITIM

\*DOSYA SİSTEMİ

\*KULLANICILAR VE GRUPLAR

\*İZİNLER (YETKİLENDİRMELER)

\*PAKETLER

\*YAZILIMLAR

\*AYGITLAR

\*MASAÜSTÜ

***ÇEKİRDEK KAVRAMI***

Çekirdek sistem kaynaklarını yöneten yazılıma verilen addır.Bu anlamada bakıldığında bir işletim sistemi olarakta bahsedilebilir.Linus Torvalsın yazdığı Çekirdek yazılımıdır ve Linux ismide çekirdeğe verilen isimdir.Tüm donanımları yöneten kod çekirdektir ve donanımları yönetmede kullanılacak kod çekirdeğin içerisinde bulunur.Aygıtların kullanımını sağlayan koda AYGIT SÜRÜCÜ denir.Aygıt sürücüleri linux çekirdeğine aygıt olarak eklenebilir yada çıkarılabilir.

Not:linux katmanlı mimariye sahiptir ve bu mimariden dolayı yavaştır.Ama linux zaten hız için değil stabilite için yazılmıştır.

katmanlar içerden dışarı doğru: ÇEKİRDEK>UYGULAMAR(kabuk da burdadır)>USER SPACE

**LİSANS KAVRAMI**

Kullanıcıların yazılım üzerindeki hakları olarak özetlenebilir.Linux işletim sistemi GPL ile lisanslıdır.GPL lisansı günümüz yazılım teknolojilerinin gelişiminde önemli role sahiptir.GPL lisansını en önemli kısıtlaması:"Yazılım üzerinde değişiklik yapılması halinde yazılım tekrar kaynak kodunun açık olarak dağıtlması" zorunluluğudur.

Not:Yazılımlarla neler yapılıp, nelerin yapılamayacağına dair kısıtlamalar.Linux GPL lisanslıdır. GPL derki;

açık kaynak kodlu yazılım varsa elinde, sen bunu alıp herşeyi yapabilirsin, sadece kodunu kapatamazsın.

***DAĞITIM KAVRAMI***

Dağıtım, Linux çekirdeği üzerine farklı yazılımları da ekleyerek sunan alternatif işletim sistemidir. Farklı amaçlarla çıkmış olabilir.

Not:linux çekirdeği üzerindeki katmalarda farklılaşan yazılımlardır.

NOT:HBA Fİber kart?

NOT:Redhat güncelleştirme desteği satar.işletim sistemini satmaz.

Paket kavramı yokken kaynak koddan derleme işlemi vardı.Kütüphaneler make e yazılıp derleniyordu. Redhat güncellemeleri paket olarak göndermektedir.İstersen paketlerin kaynak koduda açıktır. derleyebilirsen para vermedende kullanabilirsin.

***DOSYA SİSTEMİ KAVRAMI***

Verilein sabit disk üzerinde saklandığı sisteme DOSYA SİSTEMİ denir. Linux dosya sistemi dizin yapısı agaç mimarisinde bir yapıdan oluşmaktadır.

*1.DOSYA SİSTEMİ HİYERARŞİSİ*

Dosya sistemi hiyerarşisi,belirli ortak özelliklere sahip dosyaların hangi dizinde tutulacağını belirler.Linux felsefesinden dolayı bu yapı için herhangi bir kısıtlayıcı etken koymamıştır fakat standartlaşma adına uyulması iyi olan ama zorunlu olmaya kurallar vardır. Mesela dağıtımlar, yapılandırma dosyaları /etc nin altına ev dizini dosyalarını /home un altına koymaları standartlaşma açısından yararlıdır.

*2.DOSYALAR VE İZİNLER*

Lİnuxta dosyalar ve bu dosyalar üzerinde çalışmak önemlidir.Çoğu kullanıcı işlerini dosyalar üzerinden yapar, yapılandırmalarını yapılandırma dosyaları ile gerçekleştirir.Bu yüzden dosyalara ait izinler hem çalışılabilirlik hemde güvenlik açısından önemlidir.

*3.UZANTILAR*

linux ta dosyaların sonundaki uzantıların windowstaki gibi özel bir anlamı yoktur.Linux ta bir dosyanın uzantısının .txt olması bu dosyanın bir metin dosyası olmasını gerektirmeyeceği gibi hiçbir uzantıya sahip olmamasıda metin dosyası olmasına engeldeğildir.Dosya tipini belirleyen dosyanın uzantısı değil dosyanın içeriğidir.Uzantılar daha çok son kullanıcıların farklı tipteki dosyaları birbirinden ayırt edebilmesini kolaylaştırması açısından önemlidir.Ayrıca GNOME,KDE gibi masaüstü ortamlarında dosyaların çalıştırılabilirliği açısından uzantılar önemlidir.

***KULLANICI VE GRUP KAVRAMI***

Linux işletim sisteminde de diğer işletim sistemlerinde olduğu gibi "yetkili kullanıcı" ve "normal kullanıcı" kavramı vardır.En yetkili kullanıcı ya "root" denir.Bu kullanıcıya diğer işletim sistemlerine nazaran daha fazla yetki verilir.

*1-NORMAL KULLANICILAR*

Sistem üzerinde değişiklik yetkisi olmayan sadece sadece sistemi kullanmak üzerine yaratılmış kullanıcı hesaplarıdır.

*2-GRUPLAR*

Belirli işlemleri yapabilecek kullanıcılar bir araya gelerek kullanıcıları oluşturur.Böylece Tek tek kullanıcıya yetki vermek yerine gruba izin verilerek aynı anda birçok kullanıcı yetkilendirilmiş olur.

***İZİN KAVRAMI***

Linux üzerinde işlerimleri çoğu dosya ve dizinler üzerinden gerçekleştirildiğinden kullanıcı ve gruplara izinler atamak mümkündür.Dosya ve Dizinlere verilen isimler aynıolsada farklı anlamar taşımaktadır.

***1.DOSYA İZİNLERİ:OKUMA(R)-YAZMA(W)-ÇALIŞTIRMA(X)***

Dosya sahibi,dosya sahibinin grubundaki kullanıcılar ve diğer kullanıcılar olarak 3 ayrı izin tanımlaması vardır.Kullanıcılar,ilgili izinlere göre dosyalar üzerinde değişiklik yapabilirler.

a-OKUMA:bir text dosyası üzerinde okuma izni bulunan bir kullanıcı bu zizin sayesinde dosyayı okuyabilir.

b-YAZMA:Dosya üzerinde yazma bir başka deyişle değişiklik yapma hakkına sahip olmaktır.

c-ÇALIŞTIRMA:Dosya üzerinde çalıştırma iznine sahip olmaktır. Genelde script gibi çalışıtırlabilir dosya tipleri üzerinde etkilidir.Eğer kullanıcı bu izizne sahip değilse dosyayı çalıştımrya çalıştığında hata mesajı alır.

***1.DİZİN İZİNLERİ:LİSTELEME(R)-DOSYA İŞLEMLERİ(W)-GEÇİŞ(X)***

a-Listeleme:Dizin içindeki dosya ve dizinleri listeleme hakkına sahip olacaktır.

b-Dosya işlemleri:Dizin içerisindeki dosyaları veya dizinleri silme,yeni dosyalar ekleme iznine sahip olacaktır.

c-Geçiş:Dizin içerisine geçiş hakkına sahip olacaktır.

***PAKET VE YAZILIM KAVRAMI***

Linux işletim sistemine yazılım yüklemenin birden fazla yolu mevcuttur.Çoğunlukla kullanılan yöntem,Kaynak kod üzerinden uygulamayı derlemektir.Bu yöntemde bağımlılıkların tespit edilmesi, bazı durumlarda işin içinden çıkılmaz bir hal almasına neden olabilir.Ayrıca bu yöntemde silme gibi bazı işlemlerin yapılması zorunludur.Bu sebeplerden ötürü çoğu dağıtımda paket kavramı kullanılmakatadır.İşletim sistemi üzerine yüklenen bütün yazılımlar paketler halinde hazırlanır.Neyin yüklendiği bilindiği için dogrulama ve silme gibi ek işlemleri yapmaktandan bizi kurtatırır.

En çok bilinen paket yöntemleri RPM ve DEP yöntemleridir.Bazen bir paket yükleneceği zaman sistemde yüklü olması gereken bazı paketler ön şart olabilir. Böyle bir durumda ön şart paket internet üzerinden yada ortam

aracılığıyla sağlanabilir. Bu işi kolaylaştırmak için RPM için YUM, DEB için apt-get yazılımları kullanılır. Bu yazılımlar sayesinde, paket yükleneceği zaman en güncel sürümü bulunur ve yüklenir ve programlar aracılığıyla paketler güncel tutulur.

***AYGIT KAVRAMI***

Linux işletim sistemi kendisine bağlı olan tüm aygıtları, /dev dizini altında bir dosya ile imgeler.Uygulamalar, aygıtlar ile iletişim bu dosya üzerinden sağlar.Dosyaların isimlendirilmesinde bazı standartlar vardır.

Aygıt sürücüleri:

-IDE Diskler:hd veya sd

-SATA Disker:sd

-USB,SCSI Diskler:sd

-Parelel Portlar:lp

-Seri Portlar:ttyS

-Uç birimler:tty

AYGIT İSİMLENDİRMESİ

aygıt no ya göre->/dev/fd0,/dev/fd1

aygıt adına göre->/dev/hda,/dev/sdb

***MASAÜSTÜ KAVRAMI***

Linux işletim sistemi, diğer UNIX benzeri işletim sistemlerinde olduğu gibi masaüstü ortamlı olarak değil,konsol erişimli olarak tasarlanmıştır. Yani LİNUX sistemlerde Masaüstü ortamı ayrı bir yazılımdır ve çekirdekten bağımsızdır.

NOT:1.Xpencere sistemi

2.Pencere yöneticisi

3.Masaüstü ortamı

1.X pencere sistemi:

tamamen istemci sunucu mimarisi ile çalışır.

A(istemci)------------>B(sunucu) (aradaki ag baglantısı) hem Xistemcisi hemde Xsunucusu aynı makinede çalıştır.xdmcp protokolunu kullanır.

not:vnc,rdp, nx bunlar uzak masaüstü protokolleridir. Bağlanan kişinin mouse klavye hareketlerini yakalayıp, görüntüyü karşıdan ister.

xdmcp; linux ince istemci mimarisinde çalışır.Tek makinede çalışıyoruz.İstemci ile sunucuyu ayırırsak ince istemcili makine olur.

Bu yüzdn ince istemcili sistemler linux üzerinde kolay yapılır.

-Xprojesinin amacı; makinenin ekran kartını yönetip renkli görüntüler oluşturmaktır.Xprojesi ekran kartını otomatik olarak algılar.

\*\*\*\*Görüntüyü yaratan X istemcisidir. Görüldüğü yer ise X Sunucusudur.Bu yüzden ekran kartını X SUNUCUSUNDAN AYARLARIZ.

A(istemci)---------->laptop,windows (arasındaki bag xming ile sağlanır.)

Yapılandırma Dosyası:

/etc/X11/xorg.conf

uygulamalar kendisi ile ilgili ayarları dosyalarda tutar.

/etc/X11/xorg.confd ile .conf uzantılı dosyalarda yapılandırmaya dahil edilir.

Xor -config /etc/X11/xorg.conf

X çalışmazsa;

Xorg -configure yaparsak: bu bütün donanımları algıayıp xorg.conf dosyası hazırlar.

2-Pencere yöneticisi:Ara katmandır.(pencereleri çizer)

3-Masaüstü ortamı: (GNOME metacity, KDE Kwin pencere yöneticisi kullanır.)

-siyah ekrandan startx ile grafik ekrana baglanbiliriz.Login ekranını istemez çünkü zaten root ile startx yaparız.

-OPENSTANDART: zorlayıcı olmayan standart, birşeyi yapmaya zorlamayan tavsiye eden standarttır. Örnegiğin dosya formatı bu standarttır. dosyaların nerelere atılacağı ile ilgili bilgiler verir. zorunluluk değildir.

-/bin dizini: binary dizininin kısaltmasıdır.

-/boot: bilgisayarın açılmasıyla ilgil bütün dosyalar burdadır. vmlinuz sistemin açılmasını sağlar.Çekirdek budur.Boyutu küçüktür ama çalışma esnasında büyüme özelliği vardır.Dinamik olarak ihtiyaç duyulan moduller çalışma zamanında eklenir.

-/dev :çalışma zamanında kullanılan uygulamalar.

-/dev/zero : sonsuza kadar hep 0 biti verir.

-/dev/null : nasıl veri gönderilirse gönderilsin yok eder.

-/dev/random

-/dev/urandom: unlimited random.

Bilgisayarda random yoktur. Çünkü bilgisayar deterministic bir alettir.(ne yapacagı belli)Bilgisayarda rastegele veri üretimi işlemci saatine göre belirlenir.Bilgisayarda deterministic olmayan tek şey kullanıcı hareketleridir.Bunu kullanarak urandom rastegelelik üretir.Sertifika oluşturmak için kullanılır.Çekirdek derlerken kullanılır.Çekirdekte çünkü random bir değer tutar.

-/etc: programların conflarının tutulduğu yerdir.Programların nasıl çalışması erektiğini söyler.

-/home: ev diziniir. linuxta root un ev dizini home da değildir.Linux ta ya rootsundur. Yada kullanıcı.

-/lib:çalışma zamanı kütüphaneleri,dinamik kitaplıklar belleğe birkez yüklenir.Kullanıcı çağırmadıkça çalışmaz. Ayrıca,çekirdeğe dinamik olarak ekleyip çıkarabileceğimiz modullerde burda bulur.

-/lib/modules çekirdeğin versiyon numarasını verir.

-.ko uzantılı dosyalar kernel objecttir.

-/media: usb baglandığında örnegin buraya bağlanır.diski taktığımız anda bağlanır ve dizin oluşturulur.Ancak bu sadece masaüstü ortamı üzerinden olur.Siyah ekranda olay farklıdır.oluşan klasore atılan dosyalara diski çıkarsak bile ulaşabiliriz.bu yüzden linux hardisk olmasa bile bir şekilde çalıştırılabilir.

-/mnt: sistem yöneticisinin geçici baglantılar yapmak için kullandığı yerdir.

-/opt:3.parti yazılımların kurulacağı yerdir.Dağıtımla birlikte gelmeyen lisanssız yazılımlar burda bulunur.

-/proc:şuana kadar bütün dizinler harddisk üzerindeki datalardı ama bu çekirdek ile iletişime geçebilen bir kanaldır.Aslında uygulamalarda bu kanaldan iletişime geçer.

-not:Ctrl+A ile komut satırının başına hareket edilir.

not:proc dizini çalışma zamanında sanal olarak üretilir.işlem sistem çagrısı aldığı zaman çıktı verir.

-free:/proc/meminfo nun çıktısını anlaşılabilir olarak verir.

-free -m ile megabyte cinsinden çıktı alınır.

-/sbin:sistem binarysi, yapılandırma ve sistem yönetimiyle ilgili dosyalar burdadır.

-/temp:geçici dizin, güvenlik gerektirmeyen dosyalaar buraya yazılır.özel bir izne sahiptir: birinin yazdığını herkes görür ama sadece yazan kişi içeriğini değiştirebilir yada silebilir.(trickybit izni)

-/usr:paylaşılabilir alandır.

-/var(variable):değişken dosyalar burdadır.mesela log dosyası.

örn:yazıcıya bir çıktı gönderdik, sistem bunu kuyruklar,değişken kuyrukları var ın altında tutar.hatta sistem kendi değişkenlerini bile burda tutar.

\*\*\*\*\*\*\*\*\*\*\*

i386:

Intel x86 uyumlu işlemciler.

ppc:

PowerPC işlemciler Macintosh ve IBM pSeries bu gruba girer.

x86\_64 amd64:

AMD-64 bit işlemciler.intelin 64 bit işlemciside bu gruba girer.

IDE Diskler

\*Primary Master :/dev/hda: Ana IDE sürücüsü

\*Primary Slave :/dev/hdb: Yardımcı IDE sürücüsü

\*Secondary Master: /dev/hdc: İkincil ana IDE sürücüsü

\*Secondary Slave: /dev/hdd: İkincil yardımcı IDE sürücüsü

SCSI,Fiber,SATA,USB Diskler

Birinci disk:/dev/sda

ikinci disk:/dev/sdb ...

dik bölümleri ->/dev/sda2,/dev/hdb7

***SWAP(TAKAS ALANI)***

Bilgisayarda çalışan her süreç RAM de bir takım bloklar kullanır.Bu bloklara sayfa adı verilir. Bellek içerisindeki,işlemci tarafından çok kısa süre içerisinde kullanılacak olan sayfalara çalışma kümesi denir. Linux belleğe yapılan bu erişimleri göz önünde tutarak, mümkün oldukta bu sayfaların ramde tutulmasını sağlar.Bilgisayarda çalışan çok sayıda süreç varsa; çekirdek RAM de tutulan bu sayfaları disk üzerinde bulunan, bu işlem için ayrıl ayrılmış alana taşıyarak, RAM in boşaltılmasını sağlar.

TAKAS ALANININ BOYUTU NE OLMALI?

Bu büyüklüğün belirlenmesinde esas olarak, sistemde çalışan süreçlerin göz önüne alınması gerekir.Sunucular masaüstü bilgisayarlara oranla daha fazla takas alanı ayrılır.(Genellikle RAM miktarının 2 katı).Genellikle RAM miktarının 2 katı olması önerilir ama Ram i çok büyük olan sistemler için bu çok da mantıklı bir oran değildir.

**MASAÜSTÜ**

**X Pencere Sistemi**

X Pencere Sistemi, UNIX tabanlı bir işletim sistemine grafiksel arayüz sağlar. 1980'lerin başında MIT tarafından başlatılmış bir projedir ve X.org isimli açık-kaynak projesi olarak devam etmektedir.

X Pencere sistemi istemci/sunucu yapısında çalışır. X sunucusu, istemcinin istekleri doğrultusunda istemcideki pencereleri, iletişim kutularını ve düğmelerin çizilmesini sağlar. İstemcinin belirli bir donanıma sahip olması gerekli değildir. Yani aynı X sunucusuna farklı donanıma sahip istemciler bağlanabilirler.

X pencere sisteminin ağ üzerinden çalışabiliyor olması aynı anda birden çok iş istasyonunun yönetimine, thin-client yapısının kurulmasına, uygulamaların uzakta çalıştırılmasına ve kurulumların standartlaştırılmasına imkan tanır.Genellikle X sunucusuna istemciler yerel olarak bağlanır.

Standart bir Fedora kurulumunda hem X sunucusu hem de X istemcisi aynı makina üstünde çalışmaktadır. Merkezi bir X sunucusuna bağlı bir sistem kurmak, sistem ve yazılım bakımlarını kolaylaştırır.

**X11**

Fedora sistemler ile gelen X sunucusunun temel yapılandırma dosyası **/etc/X11/xorg.conf**' tur. Dosya olmaması durumunda X sunucusu çalışmaya başlarken donanımları algılayarak açılmaya çalışır.

**/etc/X11/xorg.conf.d/** dizini içerisindeki .conf uzantılı dosyalar da yapılandırmaya dahil edilir.

X yapılandırmasını yapmak için **system-config-display** komutu veya menüden **System > Administration > Display** seçeneği kullanılabilir. Kullanılan çözünürlük ve renk derinliği bu arayüz kullanılarak değiştirilebilir. Kullanılan monitör ve ekran kartı **Hardware** sekmesi kullanılarak ayarlanabilir.

**Pencere Yöneticisi**

Pencere yöneticisi X sunucusunun başlamasının hemen ardından başlar.

Pencere yöneticisi ekrandaki pencerelerin denetiminden sorumludur. X istemcisindeki pencerelerin çizimi ve ekrandaki kontroller ve özelliklerin gerçeklenmesini sağlamaktır. Pencere yöneticisi olmadan ekrandaki pencereler taşınamaz, tekrar boyutlandırılamaz, küçültülemez vs…

Linux üstünde kullanılan bir çok pencere yöneticisi vardır. Bazıları; **fvwm, icewm, blackbox,windowmaker,metacity,kwin** …

**Masaüstü Ortamı**

Masaüstü ortamında bir veya daha fazla pencere yöneticisi ve ortak kütüphanelerden geliştirilmiş birçok uygulamayı içerir. Masaüstü ortamı ile uygulamaların birbirine benzemesi ve birbirleriyle rahatlıkla iletişim kurması sağlanmıştır.

Fedora sistemler ile gelen varsayılan masaüstü ortamı **GNOME**'dur. Bunun dışında **KDE** de dağıtımla birlikte gelir.

GNOME'un varsayılan pencere yöneticisi **metacity**'dir. KDE pencere yöneticisi olarak **kwin** kullanır.

Bunların dışında bir çok masaüstü ortamı vardır. Xfce, CDE, Xpde... Fedora sistemlerde kullanılacak olan masaüstü seçimi **/etc/sysconfig/desktop** dosyası içinde belirtilir. Dosyanın boş olması ise varsayılan masaüstü ortamı GNOME olduğu için kullanılan masaüstü ortamının GNOME olduğunu işaret eder.

**Switchdesk**

Kullanılan masaüstü ortamını veya pencere yöneticisini değiştirmek için **switchdesk** komutu kullanılabilir.

Masaüstü ortamını değiştirebilmek için kullanıcı giriş ekranında Sessions kısmından gerekli seçim yapılabilir.

**X Sunucu Yapılandırması**

X-sunucusu gerekli bilgilere ***xorg.conf*** dosyasını okuyarak ulaşır. Bu dosyada donanımlar hakkında bilgilerin yanında kullanılacak fontlar, renkler gibi bilgilerde bulunur.

Fedora sistemlerde X-sunucusu üzerinde yapılmak istenen bir değişiklik  **etc/X11/xorg.conf** dosyası üzerinden ya da **system-config-display** görsel yapılandırma aracı sayesinde yapılabilir X sunucusunun yapılandırma dosyası dağıtımdan dağıtıma değişebilir. Örneğin Ubuntu ile gelen X sunucusu yapılandırma dosyası **/usr/share/X11/xorg.conf.d** altında yer alır.

Fedora 10 sonrası sistemlerde ***xorg.conf*** dosyası oluşturulmaz. X yapılandırması X sunucusu başlatıldığı zaman belirlenir. Genelde sunucu üzerinde elle yapılandırmaya ihtiyaç duyulmaz, duyulduğu zamanda **system-config-display** komutu yardımıyla yapılandırma dosyası oluşturulabilir. Bunun için öncelikle **system-config-display** paketinin kurulu olması gerekir. xorg.conf dosyasının var olan sürücü yapılandırması üzerinden oluşturulması için aşağıdaki komut kullanılabilir:

[root@ila ~]# **system -config -display --noui**

**Xorg.conf Dosyası**

xorg.conf yapılandırma dosyası sistem açılışında aşağıda belirtilen dizinlerde aranır.

etc/X11/$XORGCONFIG

/tmp/Xorg-KEM/etc/X11/$XORGCONFIG

/etc/X11/xorg.conf4

/etc/X11/xorg.conf

/etc/xorg.conf

/tmp/Xorg-KEM/etc/X11/xorg.conf.<hostname>

/tmp/Xorg-KEM/etc/X11/xorg.conf-4

/tmp/Xorg-KEM/etc/X11/xorg.conf

/tmp/Xorg-KEM/lib/X11/xorg.conf.<hostname>

/tmp/Xorg-KEM/lib/X11/xorg.conf-4

/tmp/Xorg-KEM/lib/X11/xorg.conf

Xorg.conf dosyası her bir ayrı işlev için secenekleri belirten bölümlerden oluşur. Her bölüm belli bir forma sahiptir.

**Section** "bölüm\_adı"

Bölüm\_girdisi

...

**EndSection**

**---**

**Bölüm\_adı seçenekleri:**

**Files:** Dosya yolları tanımı.

**ServerFlags:** Sunucuya belirtilecek özel işaretleri tanımı.

**InputDevice:** Fare, Klavye gibi aygıtları tanımı. Eski sistemlerde ki Keyboard ve Pointer bölüm tanımları da bu tanımlama içinde yer alır.

**Module:** Dinamik modül yükleme.

**Device:** Grafik aygıtları tanımı.

**VideoAdaptor:** Xv video adaptör tanımı.

**Monitor:** Ekran tanımı.

**Modes:** Ekran ayarları tanımı.

**Screen:** Ekran yapılandırma tanımı. X pencere sisteminde klavye, fiziksel ekran gibi eleman birleşimleri ekran olarak adlandırılır.

**ServerLayout:** sunucu yerleşim tanımı.

**DRI:** DRI(Direct Rendering Interface) yapılandırma tanımı.

**Vendor:** Ürün satıcı tanımları.

**Örnek Yapılandırma**

**Section** “InputDevice”

**Identifier** “Keyboard0”

**Driver** “kbd”

**Option** “XkbModel” “pc105”

**Option** “XkbLayout” “tr”

**EndSection**

Bu bölümde kullanılan klavyenin modeli, sürücüsü, yerleşimi gibi özellikler hakkında bilgi vermektedir.

Xorg.conf yapılandırma dosyasında herhangi bir değişikliği, sistemi text modunda açtıktan sonra **startx** komutuyla X sunucusu elle çalıştırılarak test edilebilir. X sunucusu istenildiği gibi çalışıyorsa yapılandırmada sorun yok demektir. Aksi takdirde X oturumunu kapatarak yapılandırma dosyası kontrol edilmelidir.

X sunucusunun yeniden başlatmak için **Ctrl+Alt+Backspace** tuş kombinasyonunu kullanabilir ya da var olan X sunucu sürecini öldürülebilir.

**[root@ila ~]#** ps ax| grep X

**1463 tty7 Ss+ 0:08 /usr/bin/Xorg :0 -br -verbose -auth /var/run/gdm/auth-for-gdm-FGdmzm/database -nolisten tcp**

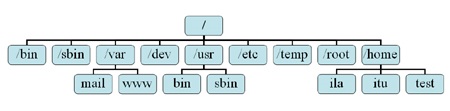
**[root@ila ~]#** kill 1463

**DOSYA SİSTEMİ**

**Dosya Sistemi**

Dosya sistemi, verilerin sistem üzerindeki düzeni ve saklanması metodudur. Dosya sistemleri veri saklama amacı ile kullanılan tüm aygıtlarda bulunmaktadır. Linux, bir çok işlemin dosyalar aracılığı ile gerçekleştirildiği bir yapıya sahiptir ve belirli bir dosya sistemi hiyerarşisi bulunmaktadır. Dosya Sistemi Hiyerarşi Standartı (FHS - File System Hierarchy Standard) olarak adlandırılan bu yapının tüm dağıtımlar tarafından kullanılması hedeflenmektedir. Ancak birçok dağıtım hala buna uyum sağlayamamıştır. Standartlaşmış bir yapı kullanmak yazılımların gerekli dosyalarının nereye kopyalayacağını belirleyebilmesi, kullanıcıların kurulu yazılımlar ile ilgili dosyaların bulunduğu yol ve dizinleri öngörebilmesini sağlar.

Red Hat türevi dağıtımlar FHS standartlarını dağıtımlarında uygulamaktadır.

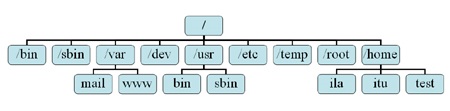


**Kök Dizini**

Linux dosya sistemi yapısında en üstte kök **dizin (/)** bulunur ve bütün dosya sistemi, farklı donanımlar da dahil olmak üzere, bu kök dizininin altında yer alır.

Sistemin açılabilmesi için kök dizininin çekirdek tarafından bilinmesi gerekir ve bütün dizinler bu dizinin altında bulunur.

**/etc, /bin , /sbin** gibi sistemi doğrudan etkileyen yapılandırmaları barındıran dizinler / dizini ile aynı bölümde olmak zorundadırlar, bunun dışında kalan diğer dizinler farklı disk bölümlerinde konumlandırılabilirler.



**Temel Dizinler**

**/bin Dizini**

Hem sistem yöneticileri hem de kullanıcıların kullanabileceği komutları içerir.

Sistemin tek kullanıcılı modda ihtiyaç duyacağı ve betikler tarafından kullanılan komutlar da bu dizinde bulunur.

**/boot Dizini**

Açılış sırasında kullanılan dosya ve dizinlerin bulunduğu dizindir.

Önyükleyici yapılandırması

Çekirdek imajı vb.

**/dev Dizini**

Aygıt dosyalarının bulunduğu dizindir.

**fd\***: Disket sürücüleri

**hd\***: IDE diskler ve diskler üzerinde bulunan bölümler

**sd\***: SCSI/SATA diskler ve diskler üzerinde bulunan bölümler

**lp\***: Paralel Port

Bu dizin altındaki bazı özel dosyalar:

**/dev/zero**

Sıfır (ASCII 0x00) karakterinin üretildiği bir dosyadır. Disk ve dosyaların üzerine yazmak için gereken veri bu dosyadan alınabilir.

**/dev/null**

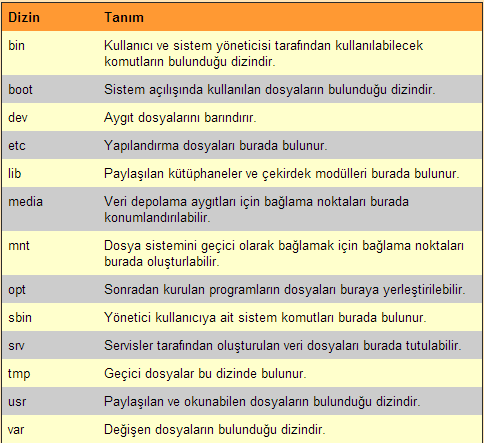
Bu dosyaya yazılan her veri sistem tarafından yok sayılır. Bu nedenle bu dosya bir kara deliğe de benzetilir. Genellikle uygulamaların istenmeyen çıktılarını devre dışı bırakmak amacı ile kullanılır.

**/dev/random**

Rastgele sayıların üretildiği bir dosyadır. Donanımların oluşturduğu etkileri kullanarak rastgele sayı üretir.

**/dev/urandom**

Rasgele sayı üretecidir, çıktı vermesi herhangi bir duruma bağlı değildir, her zaman veri alınabilir.



**/etc Dizini**

Yapılandırma dosyalarının bulunduğu dizindir. Bu dosyalar programların nasıl çalışması gerektiğini bildiren dosyalardır ve çalıştırılabilir (binary) veya dinamik değişen dosya olamazlar. Ayrıca bu dizinin altında alt dizinlerinde özel olarak servisler için yapılandırma dosyaları toplanmış olabilir.

Web servisi: **/etc/httpd/**

NTP servisi: **/etc/ntp/**

**/home Dizini**

Kullanıcıların varsayılan olarak ev dizinlerinin oluşturulacağı dizindir, kullanıcı açma işlemi sırasında ev dizini olarak bu dizinden farklı dizinler de kullanılabilir.

ila isimli kullanıcının ev dizini: /home/ila

**/lib Dizini**

Sistemin açılması ve açıldıktan sonra düzgün çalışabilmesi için gerekli kitaplıkların bulunduğu dizindir. **/bin ve /sbin** dizinlerindeki komutlar ve birçok program buradaki kitaplıkları kullanır.

Ayrıca **/lib/modules** dizininin altında çekirdek sürümüne uygun olarak modüller bulunur, böylece farklı sürümlerdeki modüllerin karışması engellenmiş olur.

**/media Dizini**

Ortam aygıtlarının bağlama noktalarının bulunduğu dizindir.

Aşağıdaki dizinleri barındırabilir.

**/media/floppy** - Disket Sürücü

**/media/cdrom** - CD-ROM Sürücü

**/media/cdrecorder** - CD Yazıcı

**/media/zip** - Zip Sürücüsü

**/mnt Dizini**

Standartlarda tanımlanan belirli bir kullanım amacı olmamakla birlikte genellikle geçici olarak herhangi bir dosya sistemine ulaşmak için oluşturulan bağlama noktalarını içerir.

Sistemin çalışması için gerekli dosyalar bu dizinin içinde bulunmamalıdır.

**/opt Dizini**

Dağıtım dışı (3. parti) program dosyalarının bulunması için tasarlanmış dizindir. Genellikle alt dizin olarak paket ismi veya paketi çıkaran firma ismi kullanılır. Sistem tarafından sağlanan program dosyaları burada bulunmaz.

**/proc Dizini**

Bu dizin aslında bir dizin veya bir bölüme karşılık gelmemektedir. Sistem , çalışan süreçler ve aygıtlarla ilgili bir çok bilgi **/proc** dizininde bulunur. **/proc** dizini sistem çalışırken var olan sanal bir dosya sistemidir.

Bazı komutlar bu dizin altından bilgi alarak çıktı üretir.

Ayrıca çekirdeğin davranışını değiştirmek için **/proc/sys** dizinin altındaki dosyalar değiştirilebilir.

Bellek bilgileri: **/proc/meminfo**

Takas alanı bilgileri: **/proc/swaps**

İşlemci bilgileri: **/proc/cpuinfo**

**/root Dizini**

root kullanıcısının ev dizini olarak kullanılan dizindir.

İstenildiği takdirde bu dizin değiştirilebilir.

**/sbin Dizini**

Sadece sistem yönetimi için sistem yöneticisi tarafından kullanılan komutlar **/sbin, /usr/sbin, /usr/local/sbin** dizinlerinde bulunur.

**/sbin** dizinindeki komutlar sistemin açılışı, kurtarılması veya onarılması için **/bin** dizinindeki komutlara ek olarak gelmiştir.

**/usr/sbin** dizinindeki komutlar sistemin düzgün açılmasının ardından kullanılabilecek komutlardır.

**/usr/local/sbin** dizinindeki komutlar sadece yerelde bulunan sistem yönetim komutlarıdır.

**/tmp Dizini**

/tmp dizini geçici olarak oluşturulacak dosyaların bulunabileceği dizindir.

Kullanıcıların ve programların bu dizine yazma hakkı vardır.

**/usr Dizini**

Paylaşılan ve okunabilen dosyaların bulunduğu dizindir. Sistemin düzgün açılmasından sonra gerekli olacak dosyalar bu dizinin altında bulunur. **/usr** dizininin yapısı kök dizinin yapısına benzerdir ve aşağıdaki dizinleri barındırmak zorundadır.

**/usr/bin**: Kullanıcıların kullandıkları komutlar

**/usr/include**: C programlarının kullandığı başlık (header) dosyalarını içerir.

**/usr/lib**: Kitaplıklar

**/usr/local**: Yerel hiyerarşi (yerel olarak kurulmuş dosyalar burada yer alır)

**/usr/sbin**: Sistem komutları

**/usr/share**: Mimariden bağımsız paylaşılan veriler

Bu dizinlerden ayrı olarak aşağıdaki dizinler de bulunabilir.

**/usr/X11R6**: X pencere sistemi (versiyon 1 release 6)

**/usr/games**: Oyunlar

**/usr/src**: Kaynak kod

**/var Dizini**

/var dizini değişen dosyaların bulunduğu dizindir. Bu dosyalar arasında kuyruk dizinleri, yönetim ve log dosyaları, geçici dosyalar bulunur.

Bu dizinin altında bulunan önemli dizinler şunlardır:

**/var/spool** Kuyrukta bekleyen dosyaların bulunduğu dizin ( yazıcı, e-posta)

**/var/run** Çalışan prosesler ile ilgili bilgilerin bulunduğu dosyalar (PID)

**/var/lock** Kilit dosyaları

**/var/log Log** dosyaları

/boot dışındaki dizinlerin neredeyse hepsi paylaşılabilir niteliktedir, fakat pratikte /lib, /bin, /sbin gibi kritik dizinlerin paylaşılmaması gerekmektedir.

**Yardım**

Komut satırında tüm komutları kullanım şekilleri ile birlikte ezberlemek mümkün değildir. Bu sebeple komut satırı üzerinde komutların kullanım şekilleri konusunda yardım gerektiğinde şu komutlara başvurulur.

**man**

**info**

**whatis**

**apropos**

**man**

man” linux’un sunduğu metin tabanlı yardım sayfalarıdır. Sistem çağrıları, komutlar ve bazı dosyalar hakkında yardım alınabilir. Tüm bu bilgiler **/usr/share/man** dizini altında bulunan dosyalarda tutulmaktadır.

[root@ila ~]# **man ls**

...

Man sayfalarında aşağıdaki tuşlarla hareket edilir:

boşluk(space) tuşu : bir sayfa ileri gitmek için

**/** : kelime arayabilmek için

**q** : çıkmak için

Linux sistemlerdeki tüm komutlar **/usr/share/man** dizini altında belirli gruplar halinde tutulmaktadır. Bu gruplar komutların kullanım alanlarına göre oluşturulmuştur.

**Man grupları**

1 Kabuk Komutları

2 Sistem Çağrıları

3 Kütüphaneler

4 Aygıt Dosyaları

5 Dosya Tipleri

6 Oyunlar

7 Diğer

8 Servis ve Sistem Yönetim Araçları

9 Çekirdek Programları

**info**

**info** yardım sayfaları da man komutu ile aynı mantıkta hazırlanmıştır fakat biraz daha detaylı bilgi içerir.

[root@ila ~]# **info man**

...

info sayfalarında hareket ederken:

**\*** : karakteri yardım sayfasında başlıkları ifade eder, bu karakterin üzerinde gelip **<ENTER>** tuşuna basıldığında ilgili başlığın içeriği görüntülenir.

**h** : info kullanım bilgileri

**q** : çıkış için

**boşluk(space) tuşu** : bir sayfa ileri gitmek için

**whatis**

Man kitaplıklarında arama gerçekleştirerek aranan komut hakkında bilgi sunar.

[root@ila ~]# **whatis man**

man (1) - an interface to the on-line reference manuals

man (7) - macros to format man pages

**apropos**

Komuta parametre olarak verilen kelimeyi whatis gibi man kitaplıklarında hem komut hem de açıklama alanında aratarak ekrana ilgili sonuçları döner.

[root@ila ~]# **apropos ila**

acpi\_available (1) - test whether ACPI subsystem is available

apm\_available (1) - test whether APM subsystem is available

mailaddr (7) - mail addressing description

pam\_mail (8) - Inform about available mail

pcimodules (8) - List kernel driver modules available for all currently plugged in PCI devices

**Dosya Tipleri**

Linux sistemlerde dosya tipini belirleyen dosyanın içeriğidir. Genellikle dosyaların uzantıları sistem yöneticisine kolaylık sağlamak açısından tanımlanmaktadır (Bazı programların kullanımlarında dosya uzantısı önem kazanabilmektedir.)

Linux işletim sistemi temelde 2 farklı dosya tipi bulunmaktadır:

**Aygıt Dosyaları** : Sistemde bulunan donanım ile haberleşmeyi gerçekleştiren dosyalardır

**Karakter Aygıt Dosyaları**: Bu tip aygıt dosyası ile erişilen donanımlara veriler birim zamanda karakter yollanarak yapılır.

Örn: Konsol arabirimi --> **/dev/console**

**Blok Aygıt Dosyaları**: Bu tip aygıt dosyası ile erişilen donanımlara veriler birim zamanda veri blokları halinde yollanarak yapılır.

Örn: Sabit diskler --> **/dev/sda**

**Sıradan Dosyalar**: Aygıt dosyaları dışında kalan dosyalardır. Çok fazla çeşitte sıradan dosya tipi bulunmaktadır.

Dosyaların tipini öğrenmek için **file** komutu kullanılır.

[root@ila ~]# **file /dev/sda**

/dev/sda: block special

Not:\*\*\*komutlar\*\*\*\*\*\*

\*man:(gruplar halinde komut dokumanlarını tutar)1.grup kitaplıkları 2.grupsistem çagrıları 3.kütüphaneler...9.çekirdek programları

SYNOPSIS:

-[] :seçimli demek.

-... :istediğin kadar tekrarlayabilirsin demek.

-man kitaplığında arama:shit+/

not siyah ekranda:ctrl+f pageup, ctrl+b pagedown demek

arama yaptık bulduğu bir sonraki kelimeye N ile gidiyoruz.Bir önceki kelimeye shift+N :shift tersine işlem yürüt demek.

\*info:man gibidir ama kitap gibi yapmışlar.

\*whois:man unixlerdede var ama bu unixlerde yoktur.man ı bir veritabanı gibi düşünürsek what is benim man kitaplığımda nerelerde olduguna bakar.what is bazı durumlarda yetmez örn, yapacagımız işlem kopyalama ama hangi komutu kullanacaığımızı bilmiyoruz o zaman;

-apropos:what is gibi çalışır ama her yere bakar.

\*linuxta 2 tip dosya vardır.

1-aygıt dosyaları

2-sıradan dosya:pdf gibi harddiskte metadatası olan dosyadır.

\*\*cd(change directory)\*\*

-dizin agacında gezinti yapmaya yarar.

\*\*pwd\*\*

-print working directory şuan bulunduğumuz dizini verir.

\*\*tam yol:/ ile başlar. /etc/X11/Xorg.confd

\*\*göreceli yol:xorg.conf.d/ gibi bulunduğun dizine göre görecelidir.

örn:cat /proc/meminfo bellek ile ilgili bilgileri verir.

-lspci:pci slotlarına takılı olan cihazları listeler.

-cd - :bulunduğum dizinden bir önce neredeysem oraya gider.

-cd : kullanıcını kendi ev dizinine döner.

-cd .. :dizin agacında bulunduğumuz yerin bir üst dizinine çıkar.

- cd . :bulunduğumuz yer anlamına gelir.

-cd ~ :bulunduğumuz kullanıcını ev dizinine döner.

touch :dosya oluşturur.

-file dosya:dosyanın tipini söyler.

-file -i:dosyanın hangi metin dosyası formatında olduğunu söyler.

not:utf8 de ilk 2 bit diger 6 bite bakılıp bakılmayacağını kodlar. Böylece ilk 2 bit sayesinde 2 üzeri 62 tane varyasyonla kodlama yapılabilir.Dosya isimlendirirken kullanılmayan tek karakter / dir.yani:

mkdir dosya1/dosya2 yapamayız.Ama mkdir -p dizin1/dizin2 olabilir. -p path verileceğini söyler.

mkdir -p -v dizin1 (verbose:detaylandırma yapar yani yaptıgı işlemi ekrana yazar.)

\*\*\*ls\*\*\*\*\*

. ile başlayan dosyalar gizlidir.Burda bahsedilen görsel olarak gizlemektir. Güvenlik ile ilili gizlemeden bahsetmiyoruz.

linux ta her kullanıcı ayrı ayrı tutulur her kullanıcının ev dizini ayrıdır.

-ls -l :long list çıkan listede;

linkler l ile ,dizinler d ile temsil edilir. bu listede root root gibi bir satır vardır. ilk root dosya sahibi ikinci root dosya grubu demektir.

- ls -l -h (h:human readable megabyte olarak çıktı. verir.)

-ls -l -h -d = ls -lhd:verdiğim dizinin kendisinin boyutuna bakmak istiyorum der.

-ls -l :alfabetik sıralar. ls -lst:boyutuna göre sıralama yapar.(s:boyuta göre sıralar.-t verirsek değiştirme tarihine göre sıralar.)

-ls -R /root :roottan başlayarak bütün dizinleri listeler.

-ls -1

-ctrl+L konsolu temizler.

\*\*\*\*\*\*Taşıma\*\*\*\*

-mv NEYİ NEREYE ÖRN:mv dosya1 dosya2 /tmp :herzaman en sağdaki hedeftir.

isim değiştrme mv dosya1 ./sertac

mv -f: parola sormaz

\*\*\*cp\*\*\*

-cp burdanal burayakopyala cp ali veli ./

-kopyalama işleminde -a paramtereside kullanılabilir.

\*\*stat\*\*

-dosyalar 2 farklı alandan oluşur.1.dosyanın verisi 2.dosyanın öz niteliklerinin tanımlı olduğu metadata,

stat time stample ilgili bilgi verir.

access alanı,modify alanı;dosya içeriğinin son değişim zamanı,change:dosya içeriğinin son değişim zamanı.Birth:oluşturulma tarihi

Not access time sürekli değiştiğinden metadata değildir.öyle oslaydı eğer change alanıyla sonsuz döngüye girerdi.

-touch aslında dosyanın zaman zaman damgasını değiştirir. change time hariç.(öyle olsa bu da sonsuz döngüye girer.)

\*\*silme\*\*

-rm

-updatedb diyerek locate komutunun kendi veri tabanını güncelleyebiliriz.

-locate aramasında önemli olan dosyanın sayısıdır.Çok fazla dosya varsa, locate her gece kendi veri tabanınını güncellediğinden, güncellemey yapmasını englleemeliyiz.

-find : gerçek zamanlı dosyalar üzerinde arama yapar.Sadece isme göre arama yapmaz. Datası hariç hemen hemen herşeye göre arama yapılabilir.örnegin içinde sertac geçen dosyaları getir diyemeyiz. işe yaramaz.meta dataya göre arama yapılmasına izin verir.

-find / : heryerde ara demektir.

- find a birşey vermeden kullanırsak dizinin altındakileri satır satır verir.

-find ile isme göre arama:

find / -name sertac

NOT: free -g : diskteki boş alanlaı gigabyte cinsinden verir.

NOT: linux çekirdeği bütün disk i o larını bellekte tutar.Aslında önbellekte arama yapar.find komutu diskte arama yapmak ister ama çekirdek onu önbellekte arama yapmaya zorlar. Bu yüzden çok hızlıdır.

-echo 1> /proc/sys/vm/drop\_caches diyerek cache i boşaltabiliriz.

-find / -user root : sahibi root olan dosyaları listeler.

-find /\! -user root : sahibi root olmaya dosyaları listeler.

-find /\! -user root type -f -name a => normalde aradığımız şeyin ismini tam vermemiz gerekir ama bilmiyoruz diyelim.

-find /\! -user root type -f -name "a\*" a ile başlayanları listeler.

-find /\! -user root type -f /\!-name "a\*"

-find / -name 'a\*' -name 'b\*' =>hem a ile hemde b ile başlayanları bul demek yani find anahtarları VE ler. veyalamak için

-find / -name 'a\*' -o -name "b\*" => a veya b ile başlayanlar.

-find / -size 0 => boyutu 0 olanları verir.

-find / -size 1M

-find /-size +1M -size -3M =>boyutu 1 mb ın üstünde olanlar ile 3 mb ın altında olanları bul.

-find / -nouser =>sahipsiz dosyları listeler.

\*\*\*\*find /\!-user root -name 'b\*' -type d -exec ls -l {}\; => -exec find ın çıktısını exec blogunda verilen komutla çalıştıracağım demektir.-exec yerine ok kullanılabilir. bu şekilde yapılan her işlem için onay bekler.

-find / -links +80 => dosya sisteminde link sayısı 80 den fazla olanları listeler.

-ls il : dosyaları inode numaralarıyla birlikte verir.

-find / -links +50 -links -60 -fls /root/cikti (fls: ls nin çıktısını dosyaya yazar.)

-find /root -maxdepth 3 -mindepth 2

NOT:which man dersek dosyanın çalıştırılabilir yolunu verir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BİTİS\*\*\*\*\*

NOT:

\*usb baglama:

-mount: komutu ile grafik ekranda usb yi taktığımızda otomatik mount edilir.nereye baglandığını örmek için mount komutunu kullanırız. Ayrıca grafik ekrandan usb yi baglamak içinde kullanılır.

-dmesg:çekirdeğin kendi log bufferınıı ekrana döker.En son satır en güncel satırdır.(Çekirdek kendi loglarını dosyaya yazmaz Ram üzerinde bir alana yazar.)

dmesg, -c anahtarıyla birlikte çekirdeğin bufferını temizleyebilir.

dmesg ile çekirdeğin usb ye hangi etiketi verdiğini görebiliriz.(sdb1 gibi bir etiket atar.)

-fdisk -l ile disklerle ilgili bilgileri görüntüleyebiliriz.

-mount NEYİ NEREYEbgalacaksın ;NEYİ kısımda usb yi temsil eden bir aygıt dosyası kullanılır.

ÖRN: mount /dev/sdb1 /mnt/ ile usb baglayabiliriz./mnt/ sistem yöneticilerin baglantı yaptıı temsili yerin adı.

\*baglantı çözme:

-umount /mnt/ yada umount /dev/sd1/; ister bagantı noktası mnt yi istersekte aygıt dosyasının oldugu yeri belirterek unmount edebiliriz.

\*Elle yapılan ütün baglantılar geçicidir. Kalıcı olmasını istiyorsak bazı dosyalara bazı bişeyler yazmamız gerekir.

\*Bagı çözemeyeceğimiz bazı durumlarda örnegin "device is busy"; bagı çözmemiz gerek, çözmeden usb yi çekersek ya dosya sistemi bozulur, yada veri kaybı olur. Çünkü usb asenkron çalışır.Grafik ekranda diyelim ki pencere kapandı(progress bar). Pencere kapandı diye hemen usb yi çekmemeliyiz.Arka planda veri bufferı atılmıştır ve unmount ettiğimizde ancak bu buffer boşaltılır.örnegin 16 gblık usb ye 8 9 gb veri atılırken unmount ettiğimizde komut satırında bekler ve biz anlarız ki halen veri aktarımı var.

\*Grafik ekran yoksa usb otomatik algılanmaz. Elle baglmamız gerekir.

\*Konsoldan cd baglama:

-cd baglarken dmesg e log düşmez.Çünkü media takıp çalıştırıyoruz."fdisk -l" de yemez.Bu yüzden;

/dev/cdrom adında bir sembolik link kullanılmıştır.

ls -l dersek;

/dev/crom->sr0 a linkliolduğunu görebiliriz.

NOT:Aygıt dosyaları "udev" klasorunun altından yönetilir.

-mount /dev/cdrom/ /mnt/ ile cdyi baglarız.Bu durumda cd nin içeriğine

ls -l /mnt ile erişilir.

NOT:CD ler için kullanılan dosya formatı ISO 9660 tır.

-cd yi bagladık fiziksel olarak çıkarmak için düğmeye bastığımzda vermez.Bagı çözmemiz gerekir.

\*cd çözme;

-umount /dev/cdrom ile bagı çözebiliriz. Diger bir yöntem eject komutudur.Sadece cd ler için geçerlidir.usb de yemez usb yi elle sen baglarsın sen çözersin.

\*Network paylaşımı:

-ls -l /var/ISO yada du -sh /var/ISO ile isoları listeleriz(s:summarize)

-mount nfs 192.168.0.200:/var/ISO /mnt ile başka bir adresten mnt ye baglama yapabilriz.(nfs:dosya paylaşım servisi)

-mount cifs 192.168.0.200:/var/ISO /mnt (Windows üzerinden dosya paylaşımı) =>mount cifs -o username=xxx,domain=ITU

//yazilim.cc.itu.edu.tr /mnt

\*cd imajını içeriğini görmek;

-mount -o loop /var/ISO/ds1-4.4.10.iso /mnt

-ls /mnt ile baglantı yapıp iso nun içeriğini görebiliriz.

\*init3 diyerek konsola düşebiliriz. init1 grafik ekrandır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BİTİS\*\*\*\*\*\*\*\*\*\*\*\*

**Dosya ve Dizin İşlemleri**

**Özel Klasör Gösterimleri**

. -> Bulunulan dizin

.. -> bir üst dizin

- -> bir önceki dizin

~ -> kullanıcının ev dizini

~**kullanıcıAdı** -> belirtilen kullanıcının ev dizini

Klasör işlemleri için kullanılan komutlar:

**pwd** : Bulunulan dizini öğrenmek için kullanılır.

**cd**: Dizin değiştirmek için kullanılır.

[root@ila ~]# pwd

/root

[root@ila ~]# **cd /var/**

[root@ila var]# **pwd**

/var

[root@ila var]# **cd** ..

[root@ila /]# **pwd**

/

[root@ila ~]# **cd**

[root@ila ~]# **pwd**

/root

[root@ila ~]# **cd** ..

/etc

**Dosya Oluşturmak**

Dosya oluşturmak için **touch** komutu kullanılır. Komut en temel hali ile kullanıldığında, çalışılmakta olan dizin içerisinde boş bir dosya yaratılır.

Linux ‘ta dosya isimerinin uzunluğu 256 karaktere kadar uzayabilir. Dosya isimleri büyük-küçük harfe duyarlıdır. Büyük harf, küçük harf, sayı ve bir çok noktalama işareti dosya adı içerisinde kullanılabilir.

/ işareti dizin yolu belirtmede kullanıldığı için dosya isimlerinde kullanılamaz. Ayrıca ? , \* gibi karakterlerin de özel anlamları olduğu için dosya isimlerinde kullanılmaları önerilmez.

. işareti ile başlayan dosyalar gizli dosyalardır.

[root@ila ~]# **touch ila**

[root@ila ~]# **file** **ila**

ila: empty

**Dizin Oluşturmak**

Dizin oluşturmak içim **mkdir** komutu kullanılır.

**mkdir** komutuna verilebilecek bazı parametreler:

-**p**: oluşturulacak dizinin üst dizinleri yoksa bunların da yaratılmasını sağlar.

-**v:** oluşturulan dizinler kullanıcıya bildirilir

[root@ila ~]# **mkdir iladizin**

[root@ila ~]# **file iladizin**

iladizin: directory

[root@ila ~]# **mkdir** **test1**/**test2**

mkdir: cannot create directory `test1/test2': No such file or directory

[root@ila ~]# **mkdir -p test1/test2**

[root@ila ~]# **file** **test1**

test1: directory

[root@ila ~]# **file** **test1**/**test2**

test1/test2: directory

[root@ila ~]# **mkdir** -**v** **test3**

mkdir: created directory `test3'

**Dizin Listelemek**

**ls** komutu kullanılarak dosya veya dizinler listelenebilir. ls komutu parametre aldığı dizinin içeriğini listeler, eğer parametre olarak bir dizin ismi verilmediyse bulunulan dizinin içeriği listelenir.

ls komutunun aldığı bazı parametreler aşağıdaki gibidir.

-**a** : Tüm dosyaları görmek için kullanılır. Gizli dosyalar da dahil.

-**d** : Bir dizine ait bilgilerin görüntülenebilmesi için kullanılır.(ls komutu dizin ismi ile birlikte çalıştırıldığında o dizinin içeriği listelenir.)

-**l** : Liste şeklinde çıktı üretir. Dosyaların izinleri, sahibi boyutu, son güncellenme tarihi ve adını ekrana yazdırır.

-**h** : Dosya boyut bilgisini kolay okunur biçimde gösterir.

-**R** : Tekrarlı olarak çalışıp listelenen dizinlerin alt dizinlerininde listelenmesini sağlar.

-**S** : Boyuta göre sıralı listelemeyi sağlar. Boyutları aynı ise isme göre sıralar.

-**t** : Son değişiklik tarihine göre sıralı listelemeyi sağlar.

[root@ila ~]# **ls -lSh /var/log/**

total 1.3M

-rw-r--r-- 1 root root 19M Feb 24 21:22 lastlog

-rw------- 1 root root 394K Feb 24 21:24 messages

-rw-rw-r-- 1 root utmp 249K Feb 24 21:22 wtmp

-rw-r--r-- 1 root root 82K Feb 22 14:34 scrollkeeper.log

-rw-r--r-- 1 root root 47K Feb 24 04:03 prelink.log

**Dosya ve Dizin Taşıma**

Dosya veya dizinleri taşımak için mv komutu kullanılabilir. Dosyalar taşınırken eğer hedef olarak gösterilen isimde bir dosya var ise bu dosyanın üstüne yazma işlemi yapılır. Dizinler taşınırken bu isimde hedef olarak gösterilen dizin varsa bu dizinin içine kopyalama işlemi yapılır.

[root@ila ~]# **ls**

dosya2 ila

[root@ila ~]# **ls ila/**

dosya1

[root@ila ~]# **mv dosya2 ila/**

[root@ila ~]# **ls**

ila

[root@ila ~]# **ls ila/**

dosya1 dosya2

**Dosya ve Dizin Kopyalama**

Dosya veya dizin kopyalamak için **cp** komutu kullanılabilir.

cp komutunun alabileceği temel parametreler aşağıdaki gibidir:

-i - Yapılacak işlem için onay alır. Red Hat sistemlerde root kullanıcısı için cp komutu cp -i komutunun aliasıdır.

-**a** - -dpR parametrelerinin kullanılması ile aynıdır.

-**d** - Bağlantıların korunmasını sağlar.

-**p** - Sahip ve tarih bilgileri saklar.

-**R** - Tekrarlı olarak işlem yapar.

-**u** - Değişmiş olan dosyaları kopyalar.

-**v** - Ekrana işlem ile ilgili bilgi bastırır.

[root@ila ~]# **ll**

total 8

-rw-r--r-- 1 root root 16 Feb 24 03:16 ila

[root@ila ~]# **cp -v ila ila\_kopya**

`ila' -> `ila\_kopya'

[root@ila ~]# **ls -l**

toplam 12

-rw-r--r-- 1 root root 0 2010-08-10 15:08 ila

-rw-r--r-- 1 root root 0 2010-08-10 15:09 ila\_kopya

**Dosya ve Dizin Bilgilerini Görüntülemek**

**stat** komutu izin, oluşturulma zamanı, boyut gibi bilgileri görüntülemek için kullanılabilir

[root@ila ~]# **stat test1**

File: `test1'

Size: 4096 Blocks: 8 IO Block: 4096 directory

Device: ca01h/51713d Inode: 622617 Links: 2

Access: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2010-08-10 10:41:14.000000000 +0300

Modify: 2010-08-10 10:41:14.000000000 +0300

Change: 2010-08-10 10:41:14.000000000 +0300

**Dosya ve Dizin Silmek**

**rmdir**: Boş bir dizini silmek için kullanılır.

**rm**: Dosya veya dizin silmek için kullanılır.

Komuta verilebilecek parametreler:

**r** : alt dizin ve dosyaların da silinmesi sağlanır.

**v** : yapılan silme işlemi ile ilgili bilgi döner.

**i** : komutun interaktif modda çalışması sağlanır, her işlem için onay istenir.

**Bağlantılar**

İki tip bağlantı çeşidi vardır: **sembolik** ve **sert (hard) bağlantı.**

Sembolik bağlantıda dosyaya kısayol oluşturulur ve dosyaya ulaşılmak istendiğinde bu bağlantıdan erişilebilir, hedef dosya silinirse bağlantı boşa çıkar.

Sert bağlantıda dosyada değişiklik olduğunda bu değişiklikler bağlantıda da gerçekleşir. Hedef dosya silinse bile bağlantıdan dosyaya ulaşılabilir.

[root@ila ~]# **ll**

total 8

-rw-r--r-- 1 root root 93 Feb 23 22:32 ila

[root@ila ~]# **ln ila hard\_link**

[root@ila ~]# **ll**

total 16

-rw-r--r-- 2 root root 93 Feb 23 22:32 hard\_link

-rw-r--r-- 2 root root 93 Feb 23 22:32 ila

[root@ila ~]# **ln -s ila sem\_link**

[root@ila ~]# **ll**

total 20

-rw-r--r-- 2 root root 93 Feb 23 22:32 hard\_link

-rw-r--r-- 2 root root 93 Feb 23 22:32 ila

lrwxrwxrwx 1 root root 3 Feb 23 22:33 sem\_link -> ila

**Dosya Okuma**

Dosya okuma için temel olarak kullanılabilecek komutlar:

cat

more

less

tac

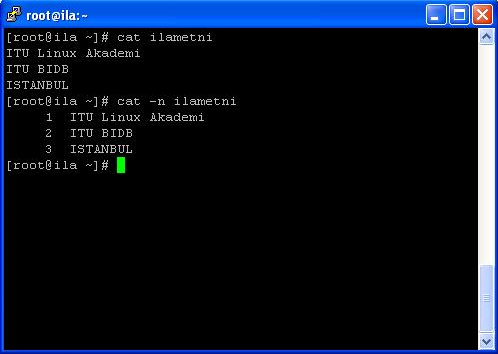
rev

head

tail

**cat**

Dosya içeriğinin ekrana yazdırılmasını sağlar

-n anahtarı kullanılırsa satır numaraları ile ekrana yazar.

**More**

Dosya içeriğini sayfa sayfa gösterir

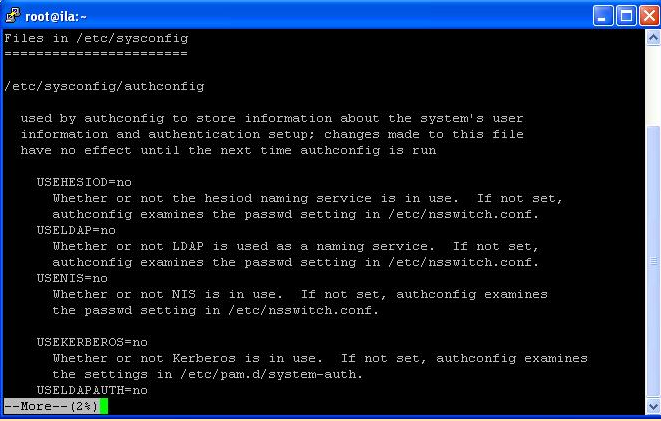
[Boşluk] veya [F] bir sayfa sonraya gider.

[Enter] bir satır sonraya gider.

[/] ile arama yapılabilir, [N] bir sonraki arama sonucuna gider.

[B] bir sayfa geriye gider.

Çıkış için [Q]



**Less**

Aşağı ve yukarı ok tuşlarıyla sayfada ilerlenebilir.

[Boşluk] veya [F] bir sayfa sonraya gider.

[Enter] bir satır sonrasına gider.

Yön tuşları ile içerik içinde hareket edilebilir.

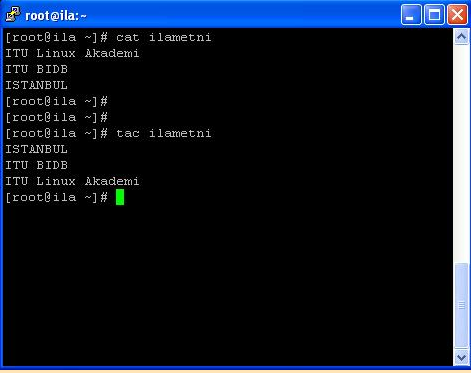
[B] bir sayfa geriye gider.

[/] ile arama yapılabilir, [N] bir sonraki arama sonucuna gider.

Çıkış için [Q]

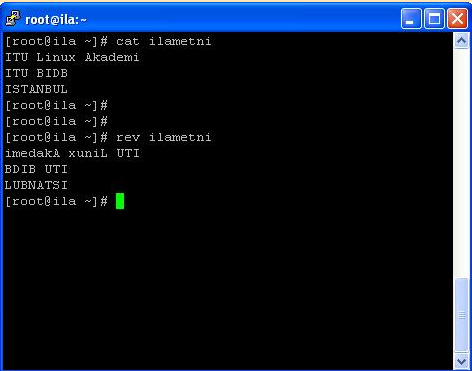
**tac**

Dosyanın içeriğini sondan başa doğru gösterir.



**rev**

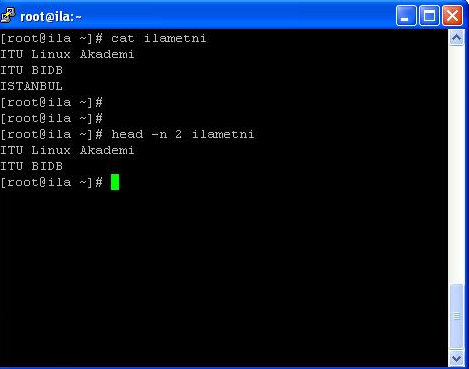
Dosyanıniçeriğini her satırı ters çevirerek gösterir.



**head**

Dosyanın başından belli bir miktar gösterir.

-n parametresi ile kaç satır göstereceği belirtilir.

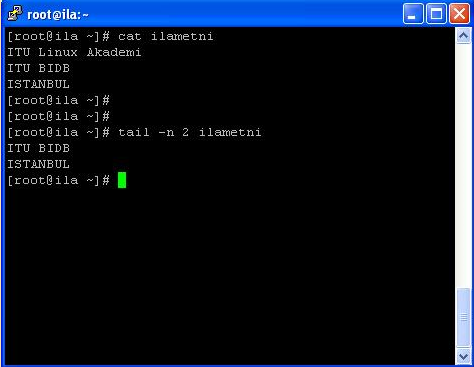


**tail**

Dosyanın sonundan belli bir miktar gösterir.

–n ile kaç satır göstereceği belirtilebilir.

–f parametresi ile değişen bir dosyanın sonuna gelen satırlar dinamik olarak görülebilir, log dosyaları incelenirken sıklıkla kullanılır.



**Dosya İşlemleri**

join

paste

split

uniq

cut

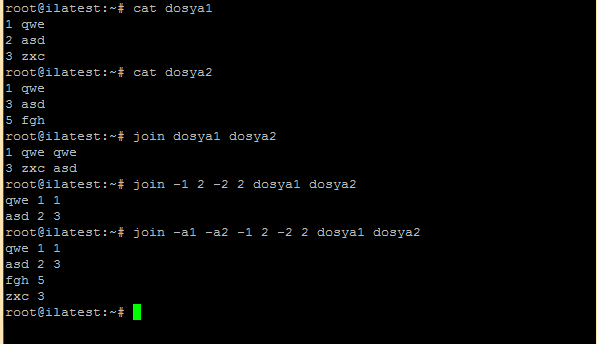
**join**

İki dosyanın içeriğini karşılaştırarak birleştirmek amacıyla kullanılır. Birleştirme işlemi için ilk sütunun ortaklığını kullanır.

Farklı parametreleri ile komutun gelişmiş özellikleri kullanılabilir.

Örnekte “-1” 1. dosyayı ve “-2” de 2. dosyayı ifade etmektedir. 1. ve 2. dosyanın 2. sütunlarındaki ortaklık kullanılarak birleştirme işlemi gerçekleştirilmiştir.

İlk durumda ortak olmayan satırlar işlem dışı kalmıştır. –a parametresi kullanılarak bu satırların da işleme dahil edilmesi sağlanabilir.

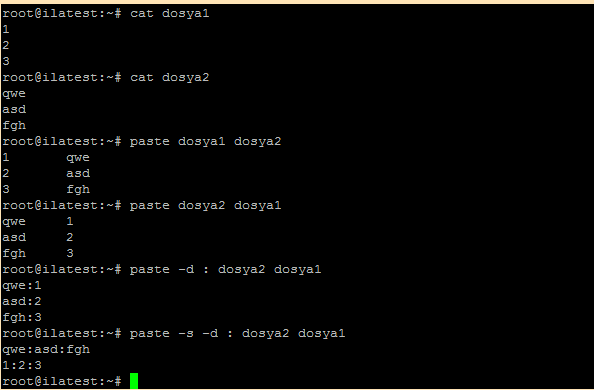


**Paste**

İki dosyayı satır satır birleştirir.

-d: birleştirme ayracı kullanmaya imkan tanır.

-s: Bir dosyanın içeriğini sıralı bir şekilde tek satırda kullanır.



**split**

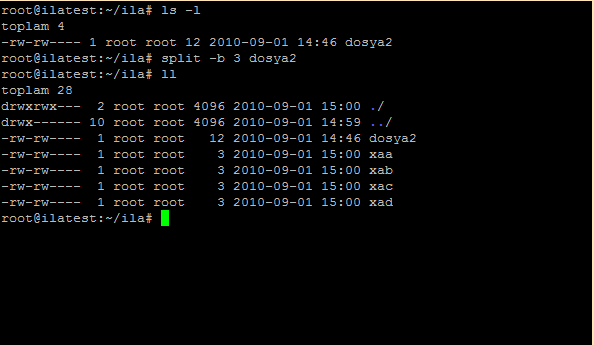
Bir dosyayı boyutuna veya satır sayısına göre parçalara ayırmak için kullanılır.

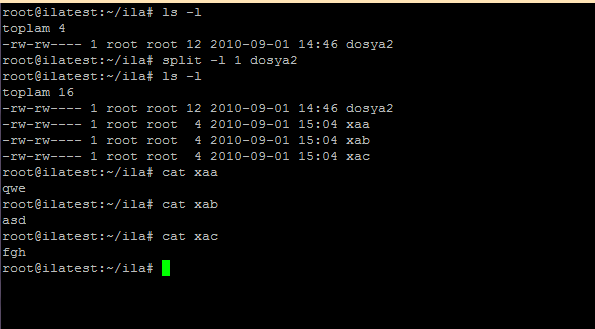
-b: her dosyaya kaç byte lık veri yazılacağı belirlenir( bu parametreden sonra K (kilobyte), M(megabyte), G(gigabyte) deyimleri kullanılabilir.)

Örnek:

root@ilatest:~# **split -b 2k dosya**

-l: dosyanın kaç satırlık bölümlere ayrılacağı belirtilir. Eğer komuta hiçbir parametre verilmemişse komut 1000 satırlık bir bölümleme yapar.

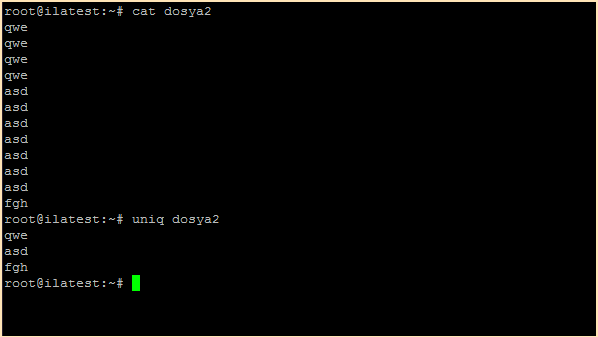




**Uniq**

Bir dosya içindeki tekrar eden satırların görüntülenmesini engeller.

-i anahtarı büyük küçük harfe bakılmaksızın fark hesaplanmasını sağlar



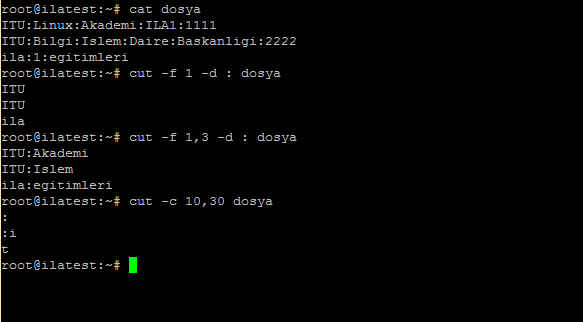
**Cut**

Bir dosyanın içeriği veya komutun çıktısını karakter boyut veya alan gibi bilgilerini kullanarak özelleştirmek ve ekrana yazdırmak için kullanılır.

-f: Belirli bir alan bilgisini alabilmek için kullanılır.

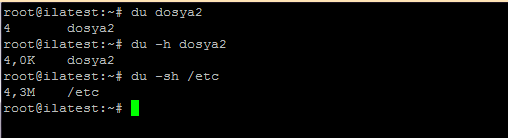
-d: Ayraç tanımlayabilmek için kullanılır.

-c: Karakter kesme işlemi yapabilmek için kullanılır.



**du**

Dosya veya dizinlerin (alt dizinleri ile birlikte) kapladığı alanı görüntülemek için kullanılır.



**Dosya ve Dizin Arama Komutları**

locate

find

which

whereis

**locate**

**locate** komutunun ardından aranan dosyanın ismi verilerek arama yapılabilir.

locate komutu **/var/lib/mlocate/mlocate.db** veri tabanında tutulan bilgilerden arama yapar. Disk üstünde değil. **updatedb** komutu ile disk üstündeki dosyalar taranarak bu veri tabanı güncellenebilir. Red Hat türevi güncel sistemlerde bu tarama günlük olarak yapılmaktadır.

[root@ila ~]# **locate httpd.conf**

/usr/share/system-config-httpd/httpd.conf.xsl

/etc/httpd/conf/httpd.conf

[root@ila ~]# **file /var/lib/mlocate/mlocate.db**

/var/lib/mlocate/mlocate.db: data

**Find**

**find** komutu kullanılarak dosya sisteminde arama yapılabilir. Bu komut çalıştırıldığı anda aranan dizinleri tarayarak sonucu ekrana getirir, bu nedenle locate komutuna göre daha yavaş ama daha güncel sonuçları ekrana getirir ve isim dışındaki kriterler ile arama yapılabilir. Kullanım şekli aşağıdaki gibidir.

**find aranacak\_dizin parametre**

find komutunun alabileceği bazı parametreler şunlardır:

-**name exp** : İsim ile arama yapmak için kullanılır. Dosyanın tam ismi verilebilceği gibi, \* ve ? gibi yer tutucu karakterlerde kullanılabilir.

-**user kullanıcı** : İstenilen kullanıcıya ait dosyların bulunması için kullanılır.

**-group grup** : Belirtilen gruba ait dosyaların bulunması için kullanılır.

**-perm izin** : Belirtilen izine sahip dosyaların bulunması için kullanılır.

-**newer dosya1** : Belirtilen dosyadan daha yeni dosyaların bulunması için kullanılır.

**-size [+|-] büyüklük** : Belirtilen büyüklüğe sahip dosyaların bulunması için kullanılır.

-**anewer dosya1** : Belirtilen dosya yapılan erişimden daha sonra erişilmiş dosyaları bulmak için kullanılır.

**-amin [+|-] süre** : Süre ile belirtilen dakika önce erişilen dosyaları bulmak için kullanılır.

**-atime [+|-] süre** : Süre ile belirtilen gün önce erişilen dosyları bulmak için kullanılır.

**-cmin [+|-] süre** : Süre ile belirtilen dakika önce durumu değiştirilen dosyaları bulmak için kullanılır.

-**ctime [+|-] süre** : Süre ile belirtilen gün önce durumu değiştirilen dosyları bulmak için kullanılır.

-**links [+|-] sayı :** Sayı ile belirtilen miktarda hard-link’e sahip dosyaları bulmak için kullanılır.

**-mmin [+|-] süre** : Süre ile belirtilen dakika önce değiştirilmiş dosyaları bulmak için kullanılır.

**-mtime [+|-] süre** : Süre ile belirtilen gün önce değiştirilmiş dosyaları bulmak için kullanılır.

**-nouser** : Hiçbir tanımlı kullanıcıya ait olmayan dosyaları bulmak için kullanılır.

**-nogrup** : Hiçbir tanımlı gruba ait olmayan dosyaları bulmak için kullanılır.

**-regex patern** : İsmi patern ile belirtilen regular expresion’a uyan dosyları bulmak için kullanılır. Dikkat edilmesi gereken nokta regular expresion dosya ismi üzerinde değil tam yol bilgisi üzerinde denenmektedir.

find programı sadece dosya bulmak için kullanılmamaktadır. Bulunan dosyalar üzerinde işlem

yapma yeteneğine de sahiptir. Bulunan dosyalar üzerinde yapılabilecek temel işlemler ve arama kriterine eklenmesi gereken parametreler aşağıdaki gibidir:

**-exec komut;** : Bulunan dosya için komut ile belirtilen komutu çalıştırır. Komut ile belirtilen argümanda bulunan {} işareti bulunan dosya ismi ile değiştirilir.

**-ok komut**; : Exec parametresinden tek farkı komut çalıştırılmadan önce kullanıcıdan onay istemesidir.

**-ls** : Dosya ile ilgli belirli özellikleri listeler(ls–dils)

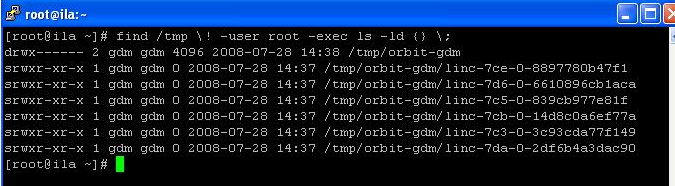
-**fls** : ls çıktısını standart çıkış yerine bir dosyaya aktarılmasını sağlar.

find komutu ile birlikte ayrıca aşağıdaki mantıksal operatörler arama kriterlerini birleştirmek için kullanılabilir:

-**a** : ve

-**o** : veya

**\!** : değil



**Which**

Komutların bulunduğu tam yolu veya alias'ını gösterir.

[root@ila ~]# **which man**

/usr/bin/man

[root@ila ~]# **which pwd**

/bin/pwd

[root@ila ~]# **which ls**

alias ls='ls--color=tty'

/bin/ls

**whereis**

**whereis** komutu aranan kelimenin çalıştırılabilir yolunu, kaynağını ve yardım sayfalarının yolunu gösterir.

Bazı parametreleri şunlardır:

-**b** : Sadece çalıştırılabilir yolu arar

-**m** : Sadece yardım sayfalarını arar

-**s** : Sadece kaynağıarar

[root@ila ~]# **whereis pwd**

pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1p/pwd.1p.gz /usr/share/man/man1/pwd.1.gz

[root@ila ~]# **whereis -b pwd**

pwd: /bin/pwd /usr/include/pwd.h

[root@ila ~]# **whereis -m pwd**

pwd: /usr/share/man/man1p/pwd.1p.gz /usr/share/man/man1/pwd.1.gz

**İzinler**

Linux'te temel dosya sistemi güvenliği kullanıcıların dosya ve dizinler üzerindeki erişim izinlerinin belirlenmesiyle sağlanır. Bir dosya veya dizinlere ait 3 grup izin vardır:

Dosya sahibinin izinleri

Dosya grubunun izinleri

Sistemdeki diğer kullanıcıların izinleri

Dosya ve dizinlerin sahip oldukları izinleri görmek için **ls-l(d)** ya da **stat** komutları kullanılabilir.

İzinler ls -l(d) çıktısının ilk kısmında belirtilir ve 4 kısımdan oluşur.

NOT:

1.İzinler dosyanın sahibine verilir.(rwx)

2.İzinler dosyanın grubuna verilir.(rwx)

3.İzinler digerlerine verilir.(rwx)

-izinler için chmod kullanılır.

r:4

w:2

x:1

NOT:root a izin işlemez okuma yazma izni olmasa bile root gider okur da yazarda.Çünkü çekirdekte root haklarıyla çalışır.Kernele sınırsız izin vardır.

-useradd ila ile ila kullanıcısı eklenebilir.

-linux ta grubu olmaya kullanıcı olmaz rubu tanımlanmayan kullanıcı için aslında kullanıcı ile aynı isimde birde grup oluşturulur.

-su(switch user) ile root accountunu istediğimiz kullanıcıya geçirebiliriz.

-whoami : o anki kullanıcının kim olduğunu verir.

-chmod u=rx,g=,o=r /home/ila (u:sahibi,g:grubu,o:other) ==>bunun adı sembolik izin atama

-chmod 504 /home/ila ==>numeric izin atama ==>bu iki satır aynıdır.

-belli bir izine ekleme yapmak istersek :chmod o+x /home/ila (other a execute vermiş olduk) Aynı şekilde ; chmod o-rwx ile otherdan bütün yetkileri almış oluruz. chmod o-x /home/ila (otherdan execute u çıkardık.)

-chown : dosyanın sahibini değiştirir.

-chgrp :dosyanın grubunu değiştirir.

NOT:chmod 707 Grup hariç herkes herşeyi yapar. Bazen ruba izin vermemiz gerekebilir.

Bakma sırası:1-kullanıcı msn?

2-grup musun?

3-sahibi misin? dir.

-chmod -x (-x burda herkes demektir.)

-chmod -R 777 dizina1 :hiyerarşik olarak herkese bütün izinleri atar.

-chmod -R o+rX dizina1/ (X sadece execute için varıdr. Dosyalara çalıştıma izni vermede dizinlere ver demektir.)

-lsattr ve chattr metadata ile ilgilidir. Bir dosyayı root bile olsan değiştiremezsin demenin yoludur.

-ls attr isim yada chattr +i isim dersek bu dosyayı root bile silemez artık, çünkü izinleri zaten root verip alır.

NOT:sembolik linklere izin vermenin hiçbir anlamı yoktur.

\*\*\*ÖZEL İZİN TİPLERİ:

-Aslında izinler 3 karakter değil 4 karakter ile tanımlanır.

1.suid biti:ilgili dosya çalıştırıldığında dosyanın sahibi kimse dosyanın hakları o haklara bürünür. /etc/shadow un izinlerinde görebilriz.

2.SGID biti:hiç bir kullanıcı grupsuz hareket edemez.(Kim çalıştırırsa çalıştırsın program) bunu grup hakkı olarak dizine verdik diyelim iç dizinlerede aktarılır.

3.Sticky Bit:temp de görülür.Kişi kendi yarattığı dosyayı silebilir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bitis\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1** : Dosya tipinin belirtildiği kısım

**-** : 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

**234** : Dosya sahibinin dosya üzerindeki izinlerinin tanımlandığı alandır.

**567** : Dosya grubunun dosya üzerindeki izinlerinin tanımlandığı alandır.

**890** : Sistemdeki diğer tüm kullanıcıların dosya üzerindeki izinlerinin tanımlandığı alandır.

[root@ila ~]# **ls -l install.log**

rw-r--r-- 1 root root 64477 Jul 8 15:44 install.log

[root@ila ~]# **ls -ld Desktop/**

drwxr-xr-x 2 root root 4096 Aug 1 18:00 Desktop/

[root@ila ~]# **stat install.log.syslog**

File: `install.log.syslog'

Size: 8325 Blocks: 32 IO Block: 4096 regular file

Device: 301h/769d Inode: 774147 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2006-08-03 17:04:06.000000000 +0300

Modify: 2006-07-08 15:44:00.000000000 +0300

Change: 2006-07-08 15:44:00.000000000 +0300

**İzin Türleri**

r - okuma izni (4)

Dosya içeriği okunabilir mi?

Dizin içindeki dosya(ların) listesi alınabilir mi?

w - yazma izni(2)

Dosya içeriği değiştirilebilir mi?

Dosya silinebilir mi?

Dizin içinde yeni dosya veya alt dizinler oluşturulabilir mi?

x - çalıştırma izni (1)

Dosya çalıştırılabilir mi?

Dizine geçilebilir mi?

**Varsayılan Dosya/Dizin İzinleri**

Yeni oluşturulacak dosya ve dizinler için izinlerin belirlenmesi

Normalde yeni oluşturulan dosya ve dizinler aşağıdaki izinlere sahip olurlar:

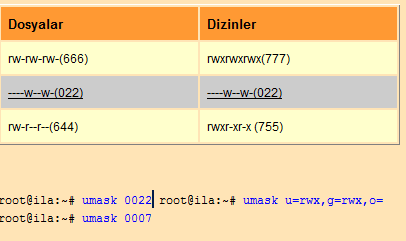
Dosya : rw-rw-rw-

Dizin : rwxrwxrwx

Dikkat edilirse bu izinlere sahip dosya ve dizinlerin sistemdeki her kullanıcı için okuma ve yazma izni

olduğu görülür. Bunu engellemek için **umask** komutu kullanılarak varsayılan izinler düzenlenir. Varsayılan olarak umask 0022 değerine sahiptir, bu sayısal ifade dosya ya da dizine verilen değil, kaldırılan izinleri göstermektedir.

umask değeri sembolik olarak da atanabilir. Bunun için **“umask u=rwx,g=rwx,o=”** şeklinde bir komut kullanılabilir. Bu ifade ile birlikte oluşturulacak olan tüm dizinler 770, dosyalar ise 660 izinleri ile birlikte oluşacaktır.



**İzin Değiştirme İşlemleri**

Dosya ve dizinlere ait izinler sahipleri tarafından chmod komutu ile değiştirilebilir.

**chmod [verilecek izinler] dosya/dizin\_adı**

chmod komutu iki farklı şekilde kullanılabilir:

İzinin sayısal değeri kullanılarak

İzinin sembolik ifadesi kullanılarak

**Sayısal değerle izin atamak**

[root@ila ~]# **ls -l install.log**

-rw-r--r--1 root root 62789 Feb 24 11:36 install.log

[root@ila ~]# **chmod 660 install.log**

[root@ila ~]# **ls -l install.log**

-rw-rw----1 root root 62789 Feb 24 11:36 install.log

**Sembollerle izin atamak**

Kullanıcı gruplarının sembolleri

Dosyanın sahibi: u (users)

Dosyanın grubu: g (group)

Diğerleri kullanıcılar: o (others)

**İzinlerin sembolleri**

Okuma izni: r (read)

Yazma izni: w (write)

Çalıştırma izni: x (execute)

Dosya/dizine ilgili izni vermek için "+", izinleri kaldırmak için "-" sembolleri kullanılır.

**Örnekler**

chmod 700 dosya (dosya sahibine tüm izinler verilir, diğer tüm izinler kaldırılır)

chmod 644 dosya (Dosya sahibine okuma ve yazma, gruba ve diğer kullanıcılara sadece okuma izni verilir)

chmod +x dosya (herkese çalıştırma izni verilir)

chmod u+rwx dosya (dosya sahibine tüm haklar verilir)

chmod g-wx dosya (dosya grubundan yazma ve çalıştırma iznini kaldırılır)

chmod u+rwx,g+rx,o+x dosya (dosya sahibine tüm izinler, gruba okuma ve çalıştırma izni, diğer kullanıcılara çalıştırma izni verilir)

chmod u=r,g=r,o=r dosya (herkes sadece okuma iznine sahip olur.)

[root@ila ~]# **ls -l install.log**

-rw-rw---- 1 root root 62789 Feb 24 11:36 install.log

[root@ila ~]# **chmod u+rw,g-w,o+r install.log**

[root@ila **~]# ls -l install.log**

-rw-r--r-- 1 root root 62789 Feb 24 11:36 install.log

**Dosya Sahibi ve Grubunun Değiştirilmesi**

Dosya sahibini değiştirmek için chown, dosya grubunu değiştirmek içinse chgrp komutu kullanılır.

chown sahip[:[grup]] dosya

chown -R sahip[:[grup]] dizin

chgrp grup\_adı dosya

[root@ila ~]# **ls -l deneme**

-rw-r--r-- 1 root root 0 Feb 24 14:14 deneme

[root@ila ~]# **chown user1 deneme**

[root@ila ~]# **ls -l deneme**

-rw-r--r-- 1 user1 root 0 Feb 24 14:14 deneme

[root@ila ~]# **chown user1:user1 deneme**

[root@ila ~]# **ls -l deneme**

-rw-r--r-- 1 user1 user1 0 Feb 24 14:14 deneme

[root@ila ~]# **ls -l test/**

total 0

-rw-r--r-- 1 root root 0 Feb 24 14:16 a

-rw-r--r-- 1 root root 0 Feb 24 14:16 b

-rw-r--r-- 1 root root 0 Feb 24 14:16 c

[root@ila ~]# **chown -R user1:root test/**

[root@ila ~]# **ls-l test/**

total 0

-rw-r--r-- 1 user1 root 0 Feb 24 14:16 a

-rw-r--r-- 1 user1 root 0 Feb 24 14:16 b

-rw-r--r-- 1 user1 root 0 Feb 24 14:16 c

[root@ila ~]# **ls -ld test/**

drwxr--r-- 2 user1 root 4096 Feb 24 14:16 test/

[root@ila ~]# **ls -ld test/**

drwxr--r-- 2 user1 root 4096 Feb 24 14:16 test/

[root@ila ~]# **chgrp user1 test/**

[root@ila ~]# **ls -ld test/**

drwxr--r-- 2 user1 user1 4096 Feb 24 14:16 test/

**Dosya Özniteliklerinin Değiştirilmesi**

Bazı dosya sistemleri izinlerin yanı sıra veri günlükleme, sıkıştırma ve değişmezlik gibi ek öznitelikleri desteklemektedir. Buna ait işlemler **chattr** komutu ile yapılmakta ve **lsattr** komutu ile listelenmektedir.

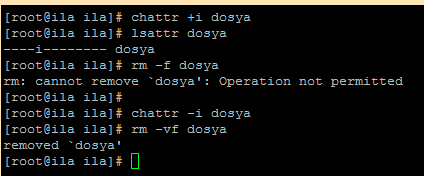
**a** : Tanımlandığında, dosya üzerine sadece yeni veri eklenebilir fakat var olan veriler değiştirilemez veya silinemez

**i** : Tanımlandığında, dosya içeriği herhangi bir şekilde değiştirilemez (silinemez, isim değişikliği yapılamaz)

**A**: Tanımlandığında, dosyaya yapılan erişimlere ait zaman bilgisi (atime) güncellenmez.

Komut Kullanımı:

chattr [+|-]<Öznitelik> Dosya ...



**Link Dosyaları ve İzinleri**

Link dosyalarının izinlerine bakıldığında herkese, tüm izinlerin olduğu görülür. Ama bu dosyalar aslında hedef gösterdikleri dosyanın izinlerini kullanırlar

[root@ila ~]# **ls -l /etc/rndc.key**

lrwxrwxrwx 1 root root 30 Feb 24 11:34 /etc/rndc.key-> /var/named/chroot/etc/rndc.key

[root@ila ~]# **ls -l /var/named/chroot/etc/rndc.key**

-rw-r--r-- 1 root named 132 Feb 24 11:23 /var/named/chroot/etc/rndc.key

**Özel İzin Tipleri**

**SUID (4) (s)**

Çalıştırılabilir dosyaların, kendi sahibinin yetkileri ile birlikte çalıştırılmasını sağlar.

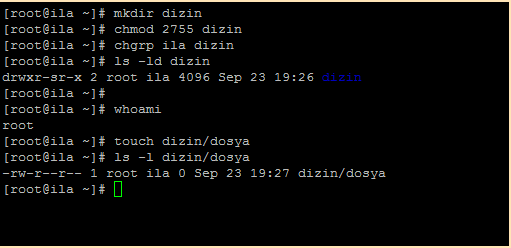
**SGID (2) (s)**

Dosya: SUID ile benzer özelliktedir, dosyanın grubu hakları ile çalıştırılmasını sağlar.

Dizin: Dizin altında yeni bir nesne oluşturulduğunda, nesne grup sahiplik izni dizin ile aynı olur.

**Sticky Bit (1) (t)**

Bu iznin atandığı dosya ya da dizini kim yaratmışsa sadece o silebilir. Örn: /tmp



**Dosya Sistemlerine Erişim**

Değiştirilebilir aygıtlardaki dosyalara erişebilmek için öncelikle bu aygıtların bağlanması gereklidir. Bu aygıtları bağlamak için **mount** komutu kullanılır. Eğer X pencere sisteminde(5. seviye) bu aygıtlara ulaşmak istenilirse bağlama işlemi otomatik olarak yapılır. Eğer X pencere sistemi çalışmıyorsa bu dizinleri elle bağlamak gereklidir. mount komutu parametre verilmeden kullanılırsa bağlanmış olan dizinleri listeler. mount komutunun temel kullanım şekli aşağıdaki gibidir:

**mount -t type bağlanacak\_aygıt bağlama\_noktası**

**type** : Bağlanmak istenen dosya sistemi tipi

Fedora ile gelen çekirdeğin desteklediği dosya sistemleri şunlardır:

adfs, affs, autofs, coda, coherent, cramfs, devpts, efs, ext, ext2, ext3, hfs, hpfs, iso9660, jfs, minix, msdos, ncpfs, nfs, nfs4, ntfs, proc, qnx4, ramfs, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, usbfs, vfat, xenix, xfs, xiafs

/**etc**/**fstab** dosyasında hangi cihazın hangi dizine bağlanması gerektiği belirtilmiştir. Burada belirtilen

bağlama işlemlerinde sadece mount dizin\_adı komutunu yürüterek bağlama yapılabilir.

[root@ila kurs]# **cat /etc/fstab**

# Thisfile is editedbyfstab-sync-see'manfstab-sync' fordetails

LABEL=/ / ext3 defaults 1 1

/dev/devpts /dev/ptsdevpt sgid=5,mode=620 0 0

/dev/shm /dev/shm tmpfs defaults 0 0

/dev/proc /proc proc defaults 0 0

/dev/sys /sys sysfs defaults 0 0

/dev/fd0 /media/floppy auto pamconsole,exec,noauto,utf8,managed 0 0

/dev/hda /media/cdrecorder auto pamconsole,exec,noauto,managed 0 0

**Disket Sürücüye Erişim**

X Pencere Sisteminde Masaüstünde Computer simgesine tıklayıp, onun içindeki Floppy Drive:floppy simgesine tıklamak disket sürücüsünün bağlanmasını sağlayacaktır.

[root@ila ~]# **mount|grep floppy**

/dev/fd0 on /media/floppy type vfat(rw,nosuid,nodev,\_netdev,utf8)

Eğer X pencere sistemi çalışmıyorsa disket sürücüyü bağlamak için

[root@ila ~]# **mount /media/floppy/**

veya

[root@ila ~]# **mount /dev/fd0**

komutları kullanılabilir. Bağlama işlemi sonucunda disket sürücüye /media/floppy dizininden ulaşılabilir

Başka bir disket takılmak istendiğinde öncelikle bu bağlantının koparılması gereklidir.

[root@ila ~]# **umount /media/floppy/**

veya

[root@ila ~]# **umount /dev/fd0**

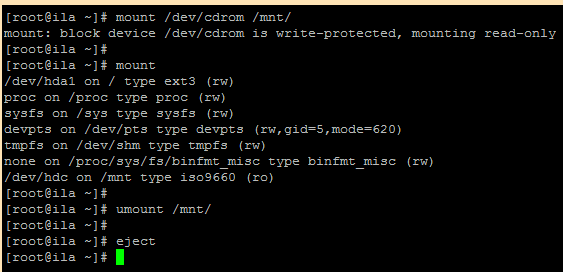
**Cdrom a Erişim**

X Pencere sisteminin çalışıyorken CD takılması durumunda CD otomatik olarak bağlanır ve masaüstünde kısayol oluşur.

Eğer X Pencere sistemi çalışmıyorsa **mount** komutu ile **cdrom**/**cdrecorder** bağlanabilir:

Başka bir CD takılmak istendiğinde öncelikle bu bağlantının koparılması gereklidir. Bağlantıyı koparmak için aşağıdaki komut kullanılabilir:

NOT: eject komutu CD/DVD aygıtının tepsisinin çıkmasını sağlar.

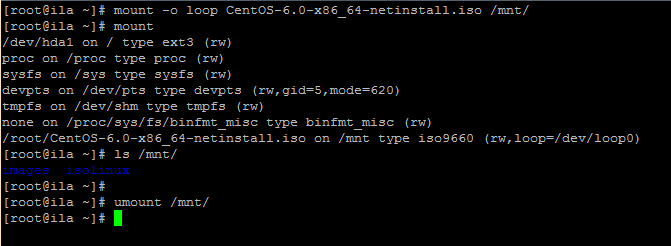


**CD/DVD İmajına (iso) Erişim**

CD imajları iso (iso9660) formatında hazırlanmaktadır. Bu formattaki bir dosya **mount** komutu ile dizin ağacına bağlanarak içeriğine salt okunur olarak ulaşılabilinir.

Bunu yapmak için **mount** komutuna **-o loop** seçeneği verilmelidir.

**mount -o loop <iso\_dosyası> <Bağlama\_dizini>**



**USB Cihaza Erişim**

USB cihazların bağlama noktaları **/etc/fstab** dosyasında bulunmaz. Genellikle USB cihazların adresi **/dev/sda** olarak adlandırılır. Ancak sistemde SCSI disk veya SATA disk varsa cihaz ismi **/dev/sdb, /dev/sdc** … şeklinde değişebilir. Bu cihazı bağlamak için kullanılabilinecek komut aşağıdaki gibidir. **/mnt/usb** dizininden cihaz erişilmek isteniyorsa,

[root@ila ~]# **mount /dev/sda /mnt/usb**

Bağlantıyı koparmak için

[root@ila ~]# **umount /dev/sda**

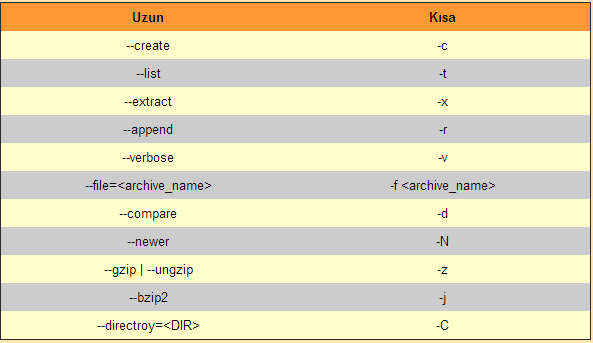
**Arşivleme**

Arşivleme ve yedekleme işlemleri birçok yolla yapılabilir. Bunun için var olan Linux komutları kullanılabileceği gibi ücretli ya da ücretsiz araçlar da kullanılabilir. Hatta eğer bir depolama cihazı kullanılıyorsa cihaz ile birlikte gelen yedekleme yetenekleri bile kullanılabilir.

Bu bölümde, arşiv ve yedekleme işlemlerinde sık kullanılan komutlar anlatılacaktır.

**Tar**

Arşivleme için kullanılabilinecek komutların başında tar komutu gelir. tar ile bir çok dosyayı tek bir dosyada birleştirmek ve hatta birleştirirken sıkıştırma işlemi yapmak mümkündür. tar komutunun sık kullanılan seçeneklerinin uzun ve kısa isimleri aşağıdaki gibidir:



[root@ila tar]# **tar -cvf dosyalar.tar [ab]**

a

b

[root@ila tar]# **tar -tf dosyalar.tar**

a

b

[root@ila tar]# **tar -rvf dosyalar.tar c**

c

[root@ila tar]# **tar -tf dosyalar.tar**

a

b

c

[root@ila tar]# **tar -xvf dosyalar.tar**

a

b

c

**cpio**

**cpio** komutu bir arşiv oluşturmayı, oluşturulmuş bir arşivi açmayı ya da bir dizinin içeriğini başka bir dizine aktarmayı sağlar.

cpio girdi olarak arşivi alınacak dosyaların veya dizinlerin listesini alır.

-**o** seçeneği arşivin oluşturulamasını sağlar.

-**i** seçeneği arşivin açılmasını sağlar.

**-t** seçeneği arşivin içeriğinin listelenmesini sağlar.

**-v** seçeneği arşivlenen dosyaları ekrana yazdırır.

**>** işareti ise komutun çıktısını bir dosyaya yönlendirir.

Arşiv açılmak istendiğinde –**idv** seçenekleri kullanabilir. Eğer var olan dosyalar arşivdekiyle aynı veya daha yeniyse bu dosyaların üstüne yazılmayacaktır.

**–u** seçeneği verildiği takdirde bu dosyaların üzerine yazılabilir.

[root@ila tar]# **find . -name "\*.txt" | cpio -ov > txtler.cpio**

./d.txt

./b.txt

70 blocks

[root@ila tar]# **rm b.txt**

rm: remove regular file `b.txt'? y

[root@ila tar]# **cpio -idv < txtler.cpio**

cpio: d.txt not created: newer or same age version exists

d.txt

b.txt

70 blocks

**gzip**

**gzip** komutu ile dosyaların sıkıştırılması sağlanabilir. Parametre olarak dosya ismi verildiğinde verilen dosyayı sıkıştırıp eski halini siler.

Sıkıştırılan dosyanın korunması istenirse aşağıdaki gibi bir komut kullanılabilir.

**gunzip** komutu ile **gzip** komutu ile sıkıştırılmış olan dosyalar açılabilir.

[root@ila ~]# **ls -l tree\***

-rw-r--r--1 root root 0 Nov26 14:31 tree.cpio

[root@ila ~]# **gzip tree.cpio**

[root@ila ~]# **ls -l tree\***

-rw-r--r--1 root root 30 Nov26 14:31 tree.cpio.gz

[root@ila ~]# **ls tree\***

tree.cpio

[root@ila ~]# **gzip -c tree.cpio > tree.gz**

[root@ila ~]# **ls tree\***

tree.cpio tree.gz

[root@ila ~]# **gunzip tree.cpio.gz**

**bzip2**

**bzip2** komutu da dosya sıkıştırmak için kullanılan bir programdır. Burrows-Wheeler blok sıralamalı metin sıkıştırma algoritmasını ve Huffman kodlamasını kullanarak dosyaları sıkıştırır. Gzip’in kullandığı algoritmaya göre daha iyi bir algoritma kullandığı için daha iyi sıkıştırır.

Kullanımı **gzip** ile aynıdır.

gzip komutunda olduğu gibi **bunzip2** komutu ile sıkıştırılmış dosya geri alınabilir.

[root@ila ~]# **ls tree\***

tree.cpio

[root@ila ~]# **bzip2 -c tree.cpio > tree.bz2**

[root@ila ~]# **ls tree\***

tree.cpio tree.bz2

**UYGULAMA 1**

1 - Bulunduğunuz dizinde Uygulama isminde bir dizin oluşturun.

2- Oluşturduğunuz Uygulama dizininin içine geçip a, b ve c isminde 3 adet dosya oluşturun.

3- Oluşturduğunuz a dosyasını işaret eden d isminde bir kısayolu (sembolik) Uygulama dizini içerisinde oluşturun.

4- Uygulama dizininin izinlerini, sahibine tüm haklar, grubuna sadece dizini listeleme hakkı ve diğer kullanıcılara ise sadece geçiş hakkı verecek şekilde sembolik yol ile atayın.

5- a, b ve c dosyalarının izinlerini, sadece sahibine okuma hakkı olup diğer tüm kullanıcıların hiçbir şekilde erişemeyeceği şekilde sayısal yol ile ayarlayın. ( Tek satır komut ile yapın

6 - Sistemde yum.conf dosyasının yerini bulun.

7 - Uygulama dizininin ismini uygulama\_2 olarak değiştirin.

8 - Sistemde /var dizini altında, n harfi ile başlayan tüm dosya ve dizinler arasından root harici kullanıcılara ait olanlarını bulun. İPUCU: find komutu size yardımcı olacaktır.

9 - 8. maddede bulduğunuz dosyaları yaratmış olduğunuz uygulama\_2 dizini altına kopyalayın. İPUCU: find komutunun -exec seçeneğinden faydalanabilirsiniz.

10 - /etc/sysconfig/ ve /etc/X11 dizinlerini uygulama\_2 dizini içerisinde yedek.tar.gz dosyasına arşivleyin.

11 - uygulama\_2 dizinini silin.

**CEVAP 1**

**1** - [root@ila ~]# pwd

/root

[root@ila ~]# mkdir Uygulama

[root@ila ~]#

**2 -** [root@ila ~]# cd Uygulama/

[root@ila Uygulama]# touch a b c

[root@ila Uygulama]# ls

a b c

[root@ila Uygulama]#

**3** - [root@ila Uygulama]# ln -s a d

[root@ila Uygulama]# ls -l

total 0

-rw-r--r-- 1 root root 0 Sep 24 21:14 a

-rw-r--r-- 1 root root 0 Sep 24 21:14 b

-rw-r--r-- 1 root root 0 Sep 24 21:14 c

lrwxrwxrwx 1 root root 1 Sep 24 21:15 d -> a

[root@ila Uygulama]#

**4** - [root@ila Uygulama]# chmod u=rwx,g=r,o=x /root/Uygulama

[root@ila Uygulama]# ls -ld .

drwxr----x 2 root root 4096 Sep 24 21:15 .

[root@ila Uygulama]#

**5 -** [root@ila Uygulama]# chmod 400 a b c

[root@ila Uygulama]# ls -l

total 0

-r-------- 1 root root 0 Sep 24 21:14 a

-r-------- 1 root root 0 Sep 24 21:14 b

-r-------- 1 root root 0 Sep 24 21:14 c

lrwxrwxrwx 1 root root 1 Sep 24 21:15 d -> a

[root@ila Uygulama]#

**6 -** [root@ila Uygulama]# locate yum.conf

/etc/yum.conf

/etc/yum.conf.rpmnew

/usr/share/logwatch/default.conf/logfiles/yum.conf

/usr/share/logwatch/default.conf/services/yum.conf

/usr/share/man/man5/yum.conf.5.gz

[root@ila Uygulama]# find / -name yum.conf

/etc/yum.conf

/usr/share/logwatch/default.conf/logfiles/yum.conf

/usr/share/logwatch/default.conf/services/yum.conf

[root@ila Uygulama]#

**7** - [root@ila Uygulama]# cd ..

[root@ila ~]# mv Uygulama uygulama\_2

[root@ila ~]#

**8 -** [root@ila ~]# find /var -name 'n\*' \! -user root

/var/named/chroot/var/named/data/named\_mem\_stats.txt

/var/named/chroot/var/named/data/named.run

/var/named/chroot/var/run/named

/var/run/named

/var/lib/ntp

[root@ila ~]#

**9 -** [root@ila ~]# find /var -name 'n\*' \! -user root -exec cp -a {} /root/uygulama\_2/ \;

[root@ila ~]#

**10** - [root@ila ~]# tar -czf uygulama\_2/yedek.tgz /etc/sysconfig/ /etc/X11/

tar: Removing leading `/' from member names

[root@ila ~]# ls -l uygulama\_2/yedek.tgz

-rw-r--r-- 1 root root 69637 Sep 25 09:23 uygulama\_2/yedek.tgz

[root@ila ~]#

**11 -** [root@ila ~]# rm -rf uygulama\_2/

[root@ila ~]#

**METİN EDİTÖRLERİ (Vim)**

**Linux Sistem Yöneticiliği ve Editörler**

Her ne kadar Linux sunucularda sistemin ve servislerin yönetimi için çeşitli yönetim arayüzleri geliştirilse de, bu grafik arayüzlerin hem sayısı az hem de yetenekleri kısıtlıdır. Aynı zamanda aktif servis veren bir sunucu üzerinde sürekli masaüstü ortamı çalışması da sistem kaynaklarının bir bölümünü tüketmektedirler. Bu nedenle Linux sistem yöneticileri çoğunlukla bu ihtiyaçlarını sistem ve servislere ilişkin yapılandırma dosyalarını düzenleme/değiştirme yoluna giderek karşılarlar. Linux sistem yöneticiliğinde, metin (text) editörlerine gereksinim işte bu noktada duyulur.

**Hangi Editörler**

Linux dağıtımlarında kullanılan bir çok editör bulunmaktadır ve dağıtımdan dağıtıma çeşitlilik gösterebilmektedir

Bu editörlerden biri olan vim, bir UNIX geleneğidir; UNIX tabanlı hemen her sistemde bulunur. Ayrıca gelişmiş özellikleri göz önünde bulundurulduğunda da pek çok editörden daha avantajlı bir seçimdir

Vim, vi'ın geliştirilmiş sürümüdür. Vi ile vim arasındaki farklar sistemdeki **/usr/share/vim/vim72/doc/vi\_diff.txt.gz** dosyasında mevcuttur. Vim editörünün bazı özellikleri:

\*Yapılan değişilikleri geri alma mekanizması

\*Çoklu pencere

\*Sentaks renklendirme

\*Komut modunda dosya ismi tamamlamasıdır.

**Vi/vim**

NOT:

Vİ->unix ten gelmiştir.Kullanması zor.Yetenegi boldur.

VİM->vi improved,linuxlarda kullanılabilir durumdadır.

Birkaç modu vardır.

1-komut modu

2-escape modu

3-visual mod

\*komut modunda :q deyip tab a bastık örnegin q nun neyin kısaltması olduğunu öğrenebiliriz.

\*anaconda\_ks.cfg;Kickstart dosyası;linux kurarken seçtiğimiz ayarlar bu dosyada tutulur. bir sonraki kurulumda bu dosyayı göstererek aynı ayarlarla bir kez daha kurulum yapabiliriz.

\*Escape moddayken;

-H J K L -> yukarı aşagı sağa sola satır satır hareket edilebilir.

-vim +30 anaconda\_ks.cfg imlec 30. satırda olarak dosya açılır.

-vim +/net anaconda\_ks.cfg net kelimesinin ilk geçtiği satırına git.

-vim -d ; şuan çalıştığımız dosya ile önceki hali arasındaki farkı verir. Sistem yöneticileri için hayat kurtarır.Yapılandırma dosyasını değiştirmeden önce her zaman ir yedeği alınır.Servise yapılandırma dosyası sonra okutulur.

-vim -d anaconda anaconda\_degisen yada vim -diff anaconda anaconda\_degisen

-vim -x anaconda ;dosyayı şifreler.

çıkarken :w osman dersek osman isminde bir dosyaya kaydetmiş oluruz.

-shift+G ile son satıra gidilir.

-G+G ile 1.satıra gidilir.

NOT:bir işlemi birden fazla kez yapmak istersek önce kaç kez yapılacağını yazıp sonra işlemi gireriz.

-z+Enter ile satır başı yaparız.

-W:saga, B:sola hareket etemke için kullanılır.

-$ yani AltGr+4 ile satır sonuna Alt+o ile satır başına gidilir.

-satır kopyalama y+y, bulunulan satır hafızaya alınır.

-p ile satır yapıştırılır.(Bulunulan satırın altına yapıştırılır.) satırın üstüne yapıştırmak için Shift+P kullanılır.

-örn:satırı 6 kez kopyalamak istiyoruz. 6 ya basıp sonra y+y yapıoruz.

\*kesme işlemi aynı zamanda silme anlamına gelir ayrıca silme işlemi yoktur.

-d+d ile kesip P ile yapıştırabiliriz.

\*kelime kopyalamak istiyoruz.y+w

\*bulunduğum satıdan itibaren satır sonuna kadar yapıştır.y+$

\*x del gibi çalışır ayrıca del yoktur.

\*Bulundugun karakteri değiştirme:R deyip + ya bastık mesela o an üzerinde bulunduğumuz karakteri + ile değiştirir.

\* her u ya bastığımızda 1 geri alınır.

\*ctrl+R ile bir sonraki haline gelir.

\*\*\*\*\*\*\*\*INSERT MOD(i)\*\*\*\*\*\*\*\*\*\*\*

\*escape moda geçmek istersek;esc veya ctrl+c kullanabiliriz.

-a kullanırsak append eder. İmleci bir saga kaydırarak işleme başlar.

-shift+a satır sonuna giderek ordan eklemeye başlar.

-o yenibir satır açar.(alt satırına)

-shift+o bu de yeni bir satır açar.(nereye açtığına bak)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Komut mod(:)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*arama:/net imlecin bulunduğu yerden itibaren arar.

-N: bir sonraki

-Shift+N:Bir önceki

NOT:Arama case-sensitive dir.

-:/net\c ile case sensitivity kaldırılır.(net kelimesini küçük büyük harf duyarlılığı olmadan arar.)

-:?net imlecin bulunduğu yerin üstünde arar.

\*bul ve değiştir; :500,1000 s/NET/SERTAC/(i) 500 ile 1000. satır arasında geçen NET leri bul SERTAC ile degistir(i; büyük harf küçük harf duyarlılığını kaldırır.s syntax demek,500,1000 yerine % kullanırsak bütün dosya üzerinde işlem yapılır.)

NOT:\*\*normalde bulduğu satırdaki ilk kelime değiştirir geçer bulduğu satırdaki tüm eşleşmeleri değiştir demek için i anahtarından sonra g kullanırız.

\*\*Yanında birde c anahtarı verirsek her değiştirme işlemi için onay alır.

ÖRN: :%g/SERTAC/d içinde SERTAC geçen satırları sil.

\*Dosyayı farklı yere kaydetmek için;

:w /root/fasulye bu dosyanın 50 ile 100. satırları arasını şuraya yaz demek istersek;

:w 50,100 /root/bolunmus

NOT: :x=:wq=Shift+z+z

- ^ satırbası

- $ satırsonu regex karakterleri

:% s/root/\U&/g (U:upper case demek,root u ROOT ile değiştirir.)

- Komut modundayken ! e basarak konsolun komut satıra düşebiliriz.Enter a basarak işleme devam edebiliriz.

-R+!+KOMUT ; komutun çıktısını dosyaya ekler.

-vim dosya1 dosya2 dersek;

dosya1 açılır. :next ile dosya2 ye :prev ile dosya 1 e tekrar geçiş yapabiliriz.

-:set all bütün vim ayarlarını ekrana döker.

-:set nu satır numarası ekler.

-:set nonu satır numarasını kaldırır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VİSUAL MOD\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-v ile geçilir. sağ sol ok tuşlarıyla karakterler seçilir.

-blok seçtik diyelim. Shift+< ve Shift+> ile içeri dışarı girinti çıkıtnı yapabiliriz topluca.

-Ctrl+ vile satırı seçmeden hareket edebiliriz örnegin ~ ya basıp bir alanı seçtik örneğin o alandaki büyük harfleri küçüge çevirebiliriz

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*

Bazı Linux dağıtımlarında vi, vim’in **alias** ı olacak şekilde ayarlanmıştır.

Vim'in escape,insert ve command olmak üzere 3 ana modu vardır. (Kullanışlı olması açısından “Visual mod” da anlatılmıştır.)

root@ila ~]# cat /etc/issue

Red Hat Enterprise Linux AS release 4 (Nahant Update 7)

Kernel \r on an \m

[root@ila ~]# alias

alias cp='cp -i'

alias l.='ls -d .\* --color=tty'

alias ll='ls -l --color=tty'

alias ls='ls --color=tty'

alias mv='mv –i

alias rm='rm -i'

alias vi='vim'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

**ESCAPE Mod**

Bir dosya, düzenleme yapmak üzere  **vim** editörü ile açıldığında, **ESCAPE** modda başlatılır.

vim < seçenekler> <dosya adı>

Ancak gerektiği durumlarda birden fazla dosya adı da parametre olarak verilerek açılabilir.

**vim**: Normal, varsayılan yapılandırma ile çalıştırmadır.<br>

**vim -**: Metin vim'e standart girişten aktarılır. İstenilen metin girildikten sonra [CTRL]-[D] tuş kombinasyonu kullanılır ve dosya kaydedilip **vim**'den çıkılır.<br>

**vim +150 dosya** : "dosya" isimli dosya açılır ve imleç 150. satıra gelir.

**vim +/fedora dosya** : "dosya" isimli dosya açılır ve imleç içinde "fedora" geçen ilk satıra konumlandırılır.

**vim -d dosya1 dosya2**: Birden fazla dosyayı açıp aralarındaki farkı gösterir. (**vimdiff** komutu gibi çalışır)

**vim -h**: vim'in komut satırı seçenekleri ile ilgili kısa yardım almak için kullanılır.

**vim -x**: vim'in sözcükleri dosyaya yazarken kullanıcıya sorarak öğrendiği bir anahtar kelimeye göre şifreleme yapacak şekilde başlatılmasını sağlar.

**Dosya Üzerinde İstenilen Yere Gitme (Navigasyon)**

**\*[H],[J],[K],[L] (Yön Tuşları)** : Karakter bazında sola, aşağı, yukarı, sağa hareket

**\*[CTRL]-[F] :** Bir sayfa ileri

**\*[CTRL]-[B]** : Bir sayfa geri

**\*[Z] (ENTER) :** İmleçin bulunduğu satır ekranın en üstüne gelecek şekilde ekranı yeniden düzenler.

**\*[SHIFT]-[G] :** Dosyadaki son satıra gider.

**\*[n SHIFT-G] :** Dosyadaki n. satıra gider (Dosya başına gitmek için 1G)

**\*[0] :** İmlecin bulunduğu satırın başına gider.

**\*[$] :** İmlecin bulunduğu satırın sonuna gider.

**\*[W] :** Bir sonraki kelimenin başına gider.

**\*[B] :** Bir önceki kelimenin başına gider.

**Karakter/Kelime/Satır/Paragraf Silme Yapıştırma**

**[X]**,ya da **[x]:** İmlecin üzerinde bulunduğu karakteri siler.

**[d]-[w]:** İmlecin üzerinde bulunduğu kelimeyi siler.

**[d]-[d]:** İmlecin üzerinde bulunduğu satırı siler.

**[y]-[y]**ya da **[Y]:** İmlecin üzerindeki bulunduğu satırı kopyalar.

**[d]-[$]**ya da **[D]:** İmlecin bulunduğu noktadan satır sonuna kadar siler.

**[d]-[0]:** İmlecin bulunduğu noktadan satır başına kadar siler.

**[d]-[}]:** Paragraf silme

**[P]**ya da **[p]**: [Y]-[Y], [D]-[D], [D]-[W] veya [X]ile en son silinen satır/kelime/karakteri yapıştırır.

**[SHIFT]-[p]**: En son silinen metni imlecin bir üstündeki satırdan başlayacak şekilde yapıştırır.

**[u]:** Yapılan işlemleri sırasıyla geri almak için (undo) kullanılır.

**[CTRL]-[r]:** Geri alınan işlemi tekrarlamak (redo) için kullanılır.

Yukarıdaki komutlardan önce bir sayı yazılırsa, komut o sayı kadar işletilir.

**[3]-[d]-[d]-** 3 satır siler.

**[2]-[P]** - En son silinen metni 2 kez yapıştırır.

Karakter Düzeltme

**[R]-[CHAR]** - İmlecin üzerinde bulunduğu karakteri [CHAR] ile belirtilen karakter ile değiştirir.

**INSERT Mode**

**INSERT** mod dosya üzerinde metin yazılabilen moddur.

**[İ] :** İmlecin bulunduğu yere ekleme yapmak için kullanılır.

**[A] :** İmlecin bulunduğu yerin sağına ekleme yapmak için kullanılır.

**[SHIFT]-[A] :** İmlecin bulunduğu satırın sonuna ekleme yapmak için kullanılır.

**[O] :** İmlecin bulunduğu yerin bir alt satırına yeni bir satır eklemek için kullanılır.

**[SHIFT]-[O] :** İmlecin bulunduğu yerin bir üst satırına yeni bir satır eklemek için kullanılır. INSERT moddan ESCAPE moda geçmek için [ESC] veya [CTRL]-[C] tuşularına basılır.

**COMMAND Mod**

Vim'de komut moduna geçmek için ESCAPE moddayken [:] tuşuna basılır.

Dosya kaydedip çıkmak, arama yapmak gibi temel özellikler bu modda kullanılır.

**:w** - Dosya üzerinde yapılan değişiklikleri kaydet.

**:wq** - Dosya üzerinde yapılan değişiklikleri kaydet ve vim'den çık (:x komutu da aynı işi görür).

**:q** - Dosyadan çık

**:q!** - Dosyadaki değişiklikleri kaydetmeden çık.

**:w!** - Dosya üzerinde yapılan değişiklikleri kaydet (salt okunur dosyalar için)

**:wq!** – Dosya üzerinde yapılan değişiklikleri kaydet ve vim'den çık (salt okunur dosyalar için/ :x! komutu da aynı işi görür)

**Dosya üzerinde arama yapmak**

\* **/<aranacak\_kelime>**- Dosya içinde <aranacak\_kelime>'yi bulur ve imleç oraya gider.

\* **?<aranacak\_kelime>** - Dosya içinde <aranacak\_kelime>'yi bulmak için ters yönde işlem yapar.

\* **[N]** - En son aranan kelime dosya içinde birden fazla yerde bulunuyorsa, arama yönünde bir sonrakini bulur.

\***[SHIFT]-[N]** - En son aranan kelime dosya içinde birden fazla yerde bulunuyorsa, arama yönünün tersi yönde bir sonrakini bulur.

\* **</ | ?>;<aranacak\_kelime>\c**  - Aramayı küçük/büyük harfe duyarsız (case-insensitive) yapar.

**Dosya Üzerinde Kayıt Bulmak ve Değiştirmek**

Bazı kısaltmalar:

\* **s** - Değiştir (Substitute)

\* **d** - Sil (Delete)

\* **g** - Satırdaki tüm eşlemeleri değiştir (global)

\* **c** - Yapılacak her değişiklik için onay iste (confirmation), case-insensitive

\* **U** - Büyük harf (uppercase)

\* **L** - Küçük harf (lowercase)

\* **%** - İstenen değişikliği dosyadaki tüm satırlarda yap

\* **^** - Satır başı

\* **$** - Satır sonu

\* **x,y** - İstenen değişikliği x ile y satırlarıarasında yap (x ve y dahil)

:5,23 s/ila/itulinuxakademi/g

:4,20 s/ali/veli/

:% s/root/admin/gc

:13,19 s/^deneme$//

:g/^$/d

:% s/root/\U&amp;/g (Burada & "root" kelimesini tutuyor)

**İleri Dosya İşlemleri**

\* **:w** dosya2 - Üzerinde çalışılan dosyayı dosya2 olarak kaydedilir

\* **:25,30w dosya2**- Üzerinde çalışılan dosyanın 25-30. satırlarını dosya2 olarak kaydedilir

\* **:!**- Dosya üzerinde işlem yapılırken komut satırı(shell) programları yürütülür.

\* **:r!<command>-** <command> isminde bir komut varsa yürütülür ve komut çıktısı imlecin bulunduğu yere yerleştirilir(bash kabuğunun komut tamamlama özelliği geçerlidir).

\* **vim dosya1 dosya2**- Aynı anda açılmak istenen dosyaların listesi verilir. İlk dosya üzerinde işlem yapılacak dosyadır ve belleğe yüklenir. İlk dosya üzerinde yapılan değişiklikler kaydedildikten sonra,

\* **:next**- sonraki dosyaya gider

\* **:prev**- komutuyla önceki dosyaya geçiş yapar.<br>

Satır numaralarını Göstermek

\* **:set nu** (satır numaralandırmayı kaldırmak için :set nonu komutu yürütülür)

Bazı vim Özellikleri

\* **:set all**- vim'in çalışmasına ilişkin tüm ayarları listeler.

\* **:set ic**- Metin aramalarında küçük/büyük harf duyarlılığını kaldır (set noic)

\* **:set ts=32**- [TAB] tuşunun basıldığında 32 karakter ilerlesin.

\* **:set history**- Geriye dönük ne kadar komut tutulduğu görüntülenir, varsayılan değeri değiştirmek için **:set history=XY** komutu kullanılır. (Vim ile komut geçmişi, aranılan kalıplar gibi bilgiler kullanıcının ev dizininindeki .viminfo dosyasında tutulur)

**VİSUAL Mod**

3 farklı şekilde başlatılabilir ve bu modlardan çıkış yine aynı şekilde gerçekleştirilebilir.

**v** : Seçim işlemi karakter bazında olur

**V** : Seçim işlemi satır bazında olur.

**Ctrl-V** : Seçim işlemi blok bazında olur.

\* Girinti yapılması ve geri alınması (<, >)

\* Satır silme işlemleri (d)

\* Büyük-küçük harf dönüşüm işlemleri (~)

Gibi çoklu satır işlemlerinde büyük kolaylık sağlar.

**Uygulama**

1. Tek komutla hatali harfleri siliniz

Bu satirrdaki hattali harfwleri sifliniz.

2. Eksik satiri tamamlayiniz.

Bu eksik satir

Bu eksik satirin tam hali.

3. Bir alt satira gecip ayni satiri tekrar yaziniz.

Bu satirindan iki tane olacak.

4. Satirdaki gereksiz kelimeler silinecek. (Her kelime tek komutla silinecek.)

Bu satirdaki linux gereksiz kelimeler dawew silinecek.

5. Tek komutla satirin tekrarlanan kismi silinecek.

Bu satirin iki defa yazilan kismi silinecek. yazilan kismi silinecek.

6. Tekrarlanan 2 satir tek komutla silinecek.

a)www.linux.org

b)fedora.redhat.com

c)www.linux.org.tr

c)www.linux.org.tr

c)www.linux.org.tr

d)www.belgeler.org

7.Satirlar baslarindaki harfe gore siraya sokulacak. Bunun icin kesme sonra yapistirma kullanilacak.

d)www.belgeler.org

a)www.linux.org

b)fedora.redhat.com

c)www.linux.org.tr

8. Hatali harfler replace edilerek alttaki satirin kopyasi elde edilecek.

Bu sarrrdaki hadalar ast satara gore ceplace ediaerek duzeldilecek.

Bu satirdaki hatalar alt satira gore replace edilerek duzeltilecek.

9. Ilk satirdaki hatali yerden sonrasi degistirilerek ikinci satirla ayni olacak.

Buradaki satir ikinci satirla ayni olacak.

Buradaki satir birinci satirla ayni olacak.

10. Asagidaki satir kopyalanip bir alt satira yapistirilarcak.

Bu satir kopyalanip bir alt satira yapistirilacak.

11. Asagidaki satirlarin basina topluca tek seferde girinti eklenecek. Ardından sadece buyukharf yazan kısmın tek seferde buyuk harfe donmesi saglanacak.

a: buyukharf

b: buyukharf

c: buyukharf

11. Once butun eski kelimeleri yeni ile sonra tum yeni kelimeleri eski ile degistirilecek.

Bu ve sonraki satirlarda gecen tum eski kelimeleri yeni kelimesi ile degistirilecek. Hic eski kelimesi kalmayacak.

Sonra butun yeni kelimeleri eski kelimesi ile degistirilecek.

Toplam 6 tane yeni kelimesi olacak.

12. Dosya kaydedilip cikilacak.

KULLANICI VE GRUP YÖNETİMİ

NOT:KULLANICI YÖNETİMİ

Kullanıcı bir dosya yarattığında defaultta bir grup bilgisi gerekmektedir.Bu bilgi birincil gruptan alınır.

-Kullanıcı ve grup isimleri 32 karakterle sınırlıdır.

-Rakamala başlayamaz.s

-Linux çok kullanıcılı bir sistemdir.BU kullanıcılar kolay bir şekilde özelleştirip hesapları yönetilebilir.

-her kullanıcıya bir uid atanır.

-her gruba bir gid tanır.

-useradd komutu ile kullanıcı eklenir.

ÖRN:useradd -c 'ila Kursiyeri' -d /home/kursiyer -g root -G ila,kursiyer1,kurisyer2 (-c:ad,soyad ; -d : ev dizini nerde olmalı; -g :kullanıcının 1.cil grubunu eklemey yarar. Defaultta kullanıcı ile aynı isimde grup açılır. -G:ek gruplar atanır.)

-Çok kullanıcılı ortamlar aslında Linux çekirdeği için çokta önemli değildir.Linux sisitemlerde kullanıcı açılında otomatik olarak bir uid atanır.(Bunu "id kullanıcıadı" diyerek ögrenebiliriz.) Aynı zamanda bir gid atanır.

-id root dersek; uid=0 değeri atanır.uid=0 is izin işlemez mantığı burdan gelmektedir.uid=0 özeldir, yetki uid nin 0 olmasından dolayı gelir.Özel yetkiler verilir.

-Redhat dagıtımları grup oluşturduğumuzda evdizini otomatik olarak home un altına atanır.Ubuntuda ise kullanıcı oluşturduğumuzda ev dizini otomatik olarak oluştutulmaz -m parametresi vererek oluşturmamız gerekmektedir.

-/etc/skel/ ;local account la çalışıyorsak ,yeni kullanıcı açtığımızda buraya koyduğumuz şeyler otomatik olarak kullanıcıların evdizinine atalır. Ancak sadece yeni kullanıcı oluşturulurken aktarılır.

-su(swtich user): bunu sadece root yapabilir yani uid si 0 olan kullanıcı yapabilir. Bu yüzden parola istemeden geçiş yapılabilir.

-whoami: o anki kullanıcı verir.

-passwd: bu komut ile kullanıcı kendi şifresini değiştirme hakkına sahiptir.Normalde kullanıcı bunu set edemez ama root edebilir.örn :passwd kursiyer2

-su root ken herrhangi bir kullanıcıya parolasız geçiş sağlatır.Ama bir kullanıcıdan diğerine parola ile geçilir.

-exit komutu yada CTRL+D ile o anki kullanıcıdan çıkış yapılabilir.(bu şekilde var olan kabuk oturumuda kapatılabilir.)

-su kursiyer1 dersek;root un ev dizininde kalınır.Ancak;

su -l kursiyer1 dersek kursiyer1 in evdizninine gidilir.Sanki kullanıcı login olmuş gibidir.Çevresel değişkenler falan ayarlanır.-l vermeden geçiş yapılan bir önceki komutun anlamı aslında root haklarından vazgeç demektir.

-usermod -g kursiyer2 kursiyer2

-groupadd :grup eklemek için kullanılır.

-finger kursiyer1 ;kullanıcı hakkında bilgi sorgulayabiliriz.Aslında bu bir unix protokoludur.

-chfn kursiyer1: kusiyer1 in fingerprinti değiştirilebilir.

-chgne -l root

-userdel fasulye ; dedikten sonra evizini kalır.ls -l /home/ ile ev dizninin kullanıcı haklarını görüntüleyebiliriz.Bu tür dosyalara orphen file denir.

-useradd -u 501 fasulye (uid si 501 olana yeni bir kullanıcı eklenir.Artık sildiğimiz fasulye kullanıcısının ev dizinin sahibi olur.)

-Bir dosyanın sahibi yoksa bunu sadece root kullanabilir.Genellikte root ta bunları kullanmadığından dolayı temizlemek gerekir.

-/etc/passwd dosyası; bütün kullanıcılara ait bilgiler tutulur.Root bunu elle editleyebilir.Şuana kadar giridğimiz bütün komutlar gelir bu dosyayı editler.BU dosyadaki hersatır bir kullanıcıya denk gelir.

-vim /etc/login.df ; dosyasında login oldunduğunda nelerin default olarak ayarlanacagı bu dosyada belirlenir.

etc/passwd dosyasında en son eklenen kullanıcı son satıra eklenir.Satır yapısı;KULLANICI:ŞİFRE şekilde devam eder.

6%4=2 -> 2 çıktısının hangi girdi verilerek alındığını bilemeyiz.Hashleme de buna benzerdirçGeri dönülmez bir şekilde girdiyi değiştirir.Eskiden bu şifreler bu dosyada tutlmakta idi ancak şuanda şifrelerin hashlari tutulmaktadır.

Şİfreler:/etc/shadow un altında tutulmaktadır.

-/etc/passwd içerisinde root adını girip sertac gibi herhangi bir şey yapabiliriz.artık uid si 0 olan kişi sertac olur root olmaz mesela.Bunun aktif olması için login olmamız gerekir.çünkü /etc/shadow un altında sertac kullanıcı yoktur.(bu dosyaya da sertac yazmamız gerekbilir.)

passwd: accounta ait bilgiler tutar.

uid=0 degerini /etc/passwd/ un altından degisitirirsek; root u çoklamış oluruz. Ama bu karışıklıga yol açabilir.

-/etc/passwd de kullaniciadi:sifre:uid:gid:.. diye devam eder ve fingerprint tutar.Evdizininide bu satırda görebiliriz.

-firewall sayesinde kullanıcı haklarıyla bir virüs gelip çalışsa dahi firewall sayesinde herhangi bir yeri bozamaz.Ama root haklarıyla gelip çalışırsa ki buna rootkit denir. Herşeyi darmadagın edebilir.

Sistem admini bir sisteme giriş yaparken sistemde keyloger olup olmadığına dikkat etmek zorundandır.

-/etc/shadow / u incelersek:

kullanıcıadi:şifre(hashlenmiş hali)... diye gider notlardan devamına bak.

hashlenmiş şifre ;login oldugumuzda oluşur. şifrenin hash ı tutturulursa sistem açılır.Bu bütün sisitemlerde aynıdır.Aynı şifreyi başka kullanıcıda aldı diyelim farklı bir hash oluşur.

Yapısı: $6...(hashlemede kullanılcak algoritmanın kodu 6 ile başlar)$...(bu alana tuz biber denir random üretilir.Burası farklı oldugundan farklı hashler türer.)$.......

Aynı hash i alıp diğer kullanıcının hashine kopyalarsak şifreyi aktarmış oluruz.

vim /etc/shadow izninden dolayı read only açılır.ls -l /etc/shadow diyerek bakabiliriz.

-/etc/group: gruplar bu dosyanın içerisindedir.Yapısı:

grupadı:grubunsifresi:grubun GID degeri(gid=0 in özel bir anlamı yok):BU gruba dahil olanların listesi

-grubun şifresi alanınıda aynı mantıkla /etc/gshadow/ un altında tutar.Şİfreyi bir grubun hakkını almak istediğimizde kullanabiliriz.Çok karşımıza çıkmaz.Unix zamanından kalmadır.Mainframe hiyerarşisinde kullanılmıştır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*bıtıs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Kullanıcı Yönetimi**

Linux, çok kullanıcılı bir sistemdir. Bu kullanıcılar kolay bir şekilde özelleştirilebilir, hesapları yönetilebilir.

Linux'ta aynı zamanda grup kavramı da mevcuttur, gruplar bir veya birden fazla kullanıcıdan oluşan ve özellikle güvenlik açısından kullanılması tercih edilen bir yapıdır.

Diğer işletim sistemlerindeki grup yapılarına benzemekle birlikte, her kullanıcı açılışında ona ait bir birincil grup oluşturulması ve bu gruba dahil edilmesi durumu ile diğer sistemlerden farklılık gösterir.

Linux sistemlerde her kullanıcı için tekil bir numara atanır ve buna **UID** (User ID) denir. Aynı zamanda açılan her grup içinde tekil bir numara atanır ve buna da **GID** (Group ID) denir.

**Kullanıcı İşlemleri**

Kullanıcı isimleri atanırken:

* 32 karaktere kadar kullanıcı adı oluşturulabilir.
* Büyük/küçük harfler kullanılabilir
* Rakam kullanılabilir fakat kullanıcı isimlerinin harfle başlamamasına özen gösterilmelidir.
* . ve \_ kullanılabilir ama önerilmez.

Linux sistemler kullanıcıları isimleri üzerinden değil aslında UID (ve GID) numaraları üzerinden takip eder.

UID değeri genellikle sistem hesaplarında 500'ün altından atanır ve normal kullanıcılara 500 ve üzerinde UID veya GID numaraları verilir.

UID numarası 0 olan kullanıcı sistemdeki en yetkin kullanıcıdır (root).

**Kullanıcı Ekleme**

Kullanıcı eklemek için **useradd** komutu kullanılabilir. Ayrıca **adduser** komutu da useradd komutuna yönlendirilmiştir.

* -m parametresi ile çalıştırıldığında kullanıcının ev dizini oluşturulup, bu dizine **/etc/skel/** dizininde bulunan dosyalar da kopyalanır. Red Hat türevi sistemlerde bu parametreyi vermeye gerek yoktur, sistem otomatik olarak bu işlemleri yapar.
* -u parametresi ile kullanıcı açılırken kullanıcıya istediğimiz bir UID'yi verebiliriz.
* –c parametresi kullanıcı hakkında ekstra (adı soyadı, telefon gibi) bilgilerin tutulmasını sağlar. Bu alana girdiğimiz bilgileri , ile ayırmamız gerekir.
* -g parametresi ile kullanıcının birincil grubu belirtilebilir. Red Hat türevi sistemlerde kullanıcı açıldığı zaman eğer bu parametre ile bir grup belirtilmezse, kullanıcı açılması ile birlikte kullanıcı adı ile aynı isimde bir grup da açılır.
* –G parametresi ile kullanıcıyı açarken dahil olacağı ek grupları belirtebiliriz.
* -d parametresi ile kullanıcının ev dizini belirtilebilir.
* -s parametresi ile kullanıcının kullanacağı kabuk belirtilebilir.
* -e parametresi ile kullanıcı hesabının disable olma tarihi belirtilir.

**Örnekler**

ila1 isminde genel bir kullanıcı ekleme

[root@ila ~]# **useradd -m –c "İTÜ Linux Akademi,1234567" ila1**

ila2 isminde egitmenler ve root gruplarına dahil (birincil grup değil) kullanıcı eklemek.

[root@ila ~]# **useradd -G egitmenler,root ila2**

31 Aralık 2011 de kullanım dışı kalacak ila3 kullanıcısı

[root@ila ~]# **useradd -e 2011-12-31 –s /bin/zsh ila3**

**Kullanıcı Bilgilerini Görüntüleme/Değiştirme**

**id**

Kullanıcıların GID ve UID’lerini bulmak için id komutu kullanılabilir. Aynı zamanda kullanıcıların SeLinux içeriklerini de gösterir.

**su**

**su** komutu ile kullanıcılar arasında geçiş yapılabilir. – veya –l parametresi ile çalıştırıldığında kullanıcının shell’e login olmasını sağlamış olur.

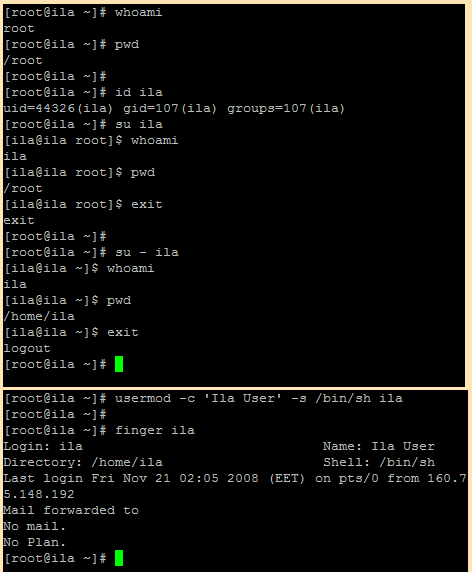
**usermod**

Kullanıcı bilgileri üstünde değişiklik yapmak için **usermod** komutu kullanılabilir. Bu komuta aşağıdaki parametreler verilerek kullanıcı ile ilgili bilgiler değiştirilebilir:

* -c <açıklama> : Açıklama alanı değiştirilebilir.
* -d <ev\_dizini> : Ev dizini değiştirlir. Eğer –m parametresi de eklenirse var olan ev dizini yenisine taşınır.
* -e <bitis\_tarihi> : Hesabın son kullanım süresi değiştirilebilir.
* -f <gun\_sayısı> : Kullanıcı şifresi expire olduktan disable olmasına kadar geçecek olan süre ayarlanabilir.
* -g <birincil\_grup> : Kullanıcının birincil grubu belirlenebilir
* -G <grup[1,2,3]> : Kullanıcının üyesi olacağı gruplar belirtilir. Her grup, diğerinden virgül ile ayrılır.Eğer kullanıcı listelenen gruplar haricinde bir grubun da üyesi ise, kullanıcı bu gruptan çıkarılır.
* -l <kullanıcı\_adı> : Kullanıcı adı değiştirilebilir.
* -p <parola> : Kullanıcının şifresi değiştirilebilir ancak burada vermek istediğimiz şifreyi encrypt edilmiş bir şekilde burada vermek gereklidir.
* -s <kabuk> : Kullanıcının kullandığı kabuk değiştirilebilir.
* -u <kullanıcı\_kimliği> : Kullanıcı UID’si değiştirilebilir. Ev dizinindeki tüm dosyaların sahipliği de yeni UID olarak değiştirilir ancak ev dizini dışındaki dosyaların izinleri el ile değiştirilmelidir.
* -L : Kullanıcı şifresini kilitler. Şifrenin önüne ! işareti koyarak parolanın geçersiz olmasını sağlar.
* -U : Kilitlenmiş bir parolayı açar. Yaptığı iş parolanın önündeki ! işaretini kaldırmaktır.

**finger**

**finger** komutu ile kullanıcı bilgilerine ulaşılır.



**chfn**

**chfn** komutu kullanıcı bilgilerini değiştirmek için kullanılabilecek başka bir komuttur. Komut parametrelerle çalıştırılabildiği gibi parametre vermeden de kullanılarak adım adım bilgilerin güncellenmesini sağlar.

Kullanımı aşağıdaki gibidir:

**chfn [ -f tam\_isim ] [ -o ofis ] [ -p ofis\_tel ] [ -h ev\_tel] [ kullanıcı\_ismi ]**

**chsh**

**chsh** komutu ile kullanıcının kullandığı kabuk değiştirilebilir. –l parametresi sistemde var olan kabukları listeler.

**passwd**

Kullanıcıların şifrelerini ayarlamak (sadece root yetkilidir) veya kullanıcıların kendi şifrelerini değiştirmek için kullandıkları komuttur.

**passwd [Kullanici\_Adi]**

**chage**

**chage** komutu ile kullanıcıların şifre yaşlandırma politikaları değiştirilebilir. Bu işlem passwd komutu ile

de yapılabilir.

Sık kullanılan parametreleri şunlardır:

* -l : Yaşlandırma hakkında bilgi verir.
* -E : Hesabın geçersiz olacağı tarihi(expire) belirtilir.
* -M : Şifrenin geçersiz olacağı gün sayısı belirtilir.
* -m : Şifre değiştirme işlemi yapılabilmesi için geçmesi gereken minimum gün sayısı.
* -I : Şifrenin geçersiz olmasından sonra hesabın kilitlenmesi için geçmesi gereken süre.

[root@ila ~]# **chfn -f "İTÜ Linux Akademi" ila**

[root@ila ~]# **chfn ila**

Changing finger information for ila.

Name [İTÜ Linux Akademi]:

Office []: 1234567

Office Phone []: 123

Home Phone []: 9876543

Finger information changed.

[root@ila ~]# **finger -m ila**

Login: ila Name: İTÜ Linux Akademi

Directory: /home/ila Shell: /bin/bash

Office: 1234567, 123 Home Phone: 987-6543

Last login Fri Dec 2 16:31 (EET) on pts/1 from ao01.ila.itu.edu.tr

No mail.

No Plan.

[root@ila ~]# **chsh -l**

/bin/sh

/bin/bash

/sbin/nologin

/bin/ksh

/bin/tcsh

/bin/csh

/bin/zsh

[root@ila ~]# **chsh -s /bin/sh ila**

Changing shell for ila.

Shell changed.

[root@ila ~]# **date**

Thu Mar 2 12:25:56 EET 2006

[root@ila ~]# **chage ila -E 2006-04-15**

[root@ila ~]# **chage ila -M 20**

[root@ila ~]# **chage ila -I 10**

[root@ila **~]# chage ila -m 5**

[root@ila ~]# **chage ila -W 25**

[root@ila ~]# **chage -l ila**

Last password change : Mar 02, 2006

Password expires : Mar 22, 2006

Password inactive : Apr 01, 2006

Account expires : Apr 15, 2006

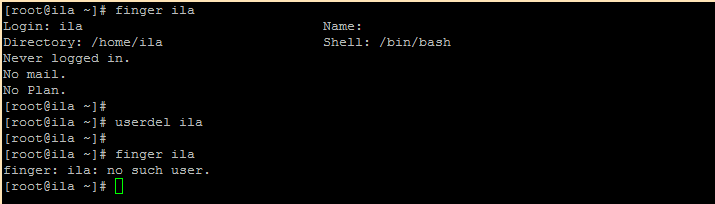
Minimum number of days between password change : 5

Maximum number of days between password change : 20

Number of days of warning before password expires : 25

**Kullanıcı Silme**

Kullanıcı silmek için **userdel** komutu kullanılabilir. Komut ile kullanıcının **/etc/passwd** ve **/etc/group** dosyalarındaki kayıtları silinir. Eğer –r parametresi kullanılırsa kullanıcının ev dizini de silinir.



**/etc/passwd Dosyasının Yapısı**

**/etc/passwd** dosyasında kullanıcı ile ilgili bilgiler bulunur. Dosyanın her satırı bir kullanıcı ile ilgili bilgileri içerir. Her satır : (iki nokta üst üste işareti) ile ayrılmış 7 alandan oluşur.

**kullanıcı\_adı:şifre:uid:gid:user\_info:ev\_dizini:kabuk**

ila:x:503:503:İTÜ Linux Akademi,1234567,123,9876543:/home/ila:/bin/bash

Bu alanlar sırasıyla aşağıdaki anlamları içerir:

* **kullanıcı\_adı** - Kullanıcının sisteme girerken kullandığı isim.
* **şifre** - Şifre /etc/shadow dosyasında tutulduğu için burada x olarak görünür.
* **uid** - Kullanıcı kimliğini belirleyen bir sayıdır.
* **gid** - Kullanıcının grubunu belirleyen bir sayıdır.
* **user\_info** - Kullanıcının gerçek ismi ve hesapla ilgili diğer açıklamaların yer aldığı alan. Buradaki bilgiler , ile birbirinden ayrılır.
* **ev\_dizini** - Kullanıcının ev dizini bu alanda belirtilir.
* **kabuk** - Kullanıcı sisteme girdiğinde açılacak varsayılan kabuk burada belirtilir.

[root@ila ~]# cat /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

halt:x:7:0:halt:/sbin:/sbin/halt

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin

operator:x:11:0:operator:/root:/sbin/nologin

games:x:12:100:games:/usr/games:/sbin/nologin

gopher:x:13:30:gopher:/var/gopher:/sbin/nologin

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

nobody:x:99:99:Nobody:/:/sbin/nologin

dbus:x:81:81:System message bus:/:/sbin/nologin

vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin

abrt:x:499:499::/etc/abrt:/sbin/nologin

haldaemon:x:68:68:HAL daemon:/:/sbin/nologin

saslauth:x:498:498:"Saslauthd user":/var/empty/saslauth:/sbin/nologin

postfix:x:89:89::/var/spool/postfix:/sbin/nologin

sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

tcpdump:x:72:72::/:/sbin/nologin

ntp:x:38:38::/etc/ntp:/sbin/nologin

test:x:500:500::/home/test:/bin/bash

ila:x:501:501::/home/ila:/bin/bash

**/etc/shadow Dosyası**

**/etc/shadow** dosyasında kullanıcının şifresi ile ilgili bilgiler bulunur. **/etc/passwd** dosyası gibi her satırı bir kullanıcı ile ilgili bilgileri tutar.

Bu dosyanın yapısı aşağıdaki gibidir:

**kullanıcı\_adı:şifre:last:may:must:warn:expire:disable:reserved**

ila:$1$lQ5v9OsdsEh9p/yAlPSm3/3B4W.:13119:0:99999:7:::

* **kullanıcı**\_**adı** : Kullanıcının sisteme girerken kullandığı isim.
* **şifre** : Kullanıcının şifresi şifrelenmiş bir şekilde burada bulunur.
* last : Şifrenin en son ne zaman değiştirildiğini tutar. Buradaki değer 1 Ocak 1970’den itibaren geçen gün sayısı olarak tutulur.
* **may** : Kullanıcının şifresini değiştirebilmesi için geçmesi gereken gün sayısı.
* **must** : Kullanıcının şifresini ne kadar daha kullanabileceğinin gösterir. 99999 yazıyorsa herhangi bir kısıtlamanın olmadığı anlamına gelir.
* **warn** : Kullanıcının, şifresinin kullanım süresi dolmadan kaç gün önce uyarılacağı belirtilir.
* **expire** : Kullanıcının, şifresinin kullanım süresi dolmasından kaç gün sonra disable olacağı belirtilir.
* **disable** : Kullanıcı hesabının disable edilme (kullanılamaz) süresi; bu değer 1 Ocak 1970’den sonraki gün sayısı olarak tutulur.
* **reserved** : Bu alan ayrılmış alandır ve herhangi bir amaçla kullanılmamaktadır.

**[root@ila ~]# cat /etc/shadow**

root:$6$wossOTJw$oWShjHce7ZwDeBXGO0QKac/kO.K.6fA1:15246:0:99999:7:::

bin:\*:14790:0:99999:7:::

daemon:\*:14790:0:99999:7:::

adm:\*:14790:0:99999:7:::

lp:\*:14790:0:99999:7:::

sync:\*:14790:0:99999:7:::

shutdown:\*:14790:0:99999:7:::

halt:\*:14790:0:99999:7:::

mail:\*:14790:0:99999:7:::

uucp:\*:14790:0:99999:7:::

operator:\*:14790:0:99999:7:::

games:\*:14790:0:99999:7:::

gopher:\*:14790:0:99999:7:::

ftp:\*:14790:0:99999:7:::

nobody:\*:14790:0:99999:7:::

dbus:!!:15040::::::

vcsa:!!:15040::::::

abrt:!!:15040::::::

haldaemon:!!:15040::::::

saslauth:!!:15040::::::

postfix:!!:15040::::::

sshd:!!:15040::::::

tcpdump:!!:15040::::::

ntp:!!:15042::::::

test:!!:15042:0:99999:7:::

ila:!!:15246:0:99999:7:::

**Grup Yönetimi**

Linux sistemlerde gruplamalar üzerinden dizin erişim yetkilerinin ayarlanması sağlanır.

Linux sistemlerde bir grup içerisine sadece kullanıcılar eklenebilir, grup içerisine grup eklenmesi durumu yoktur.

**Grup İşlemleri**

Burada gruplar üzerinde yapılabilcek işlemler ile komutlardan bahsedilecektir.

**Grup Ekleme**

Grup eklemek için groupadd komutu kullanılabilir.

Kullanımı:

**groupadd [-g gid ][-r] [-f] grup**

* -g : GID verilmesini sağlar.
* -r : Grubun sisteme ait bir grup olduğunu belirler. 499'dan küçük ilk sayı otomatik olarak atanır.
* -f : Açılacak grup sistemde varsa hata vermeden komutun çalıştırılmasını sağlar.

[root@ila ~]# **groupadd -g 550 ila-kursiyer**

[root@ila ~]# **groupadd ila-kursiyer**

groupadd: group ila-kursiyer exists

[root@ila ~]# **groupadd -f ila-kursiyer**

**Grup Bilgilerini Görüntüleme/Değiştirme**

Grup bilgilerini değiştirmek için **groupmod** komutu kullanılabilir. –g parametresi ile GID, –n parametresi ile grup ismi değiştirilebilir.

[root@ila ~]# more /etc/group | grep ila-kursiyer ila-kursiyer:x:504:ila1,ila2

[root@ila ~]# /usr/sbin/groupmod -n ila-ogrenci ila-kursiyer

[root@ila ~]# more /etc/group|grep ila-ogrenci ila-ogrenci:x:504:ila1,ila2

[root@ila ~]# /usr/sbin/groupmod -g 507 ila-ogrenci

[root@ila ~]# more /etc/group | grep ila-ogrenci

ila-ogrenci:x:507:ila1,ila2

**Grup Şifreleri**

Grup şifrelerini değiştirmek veya gruplarla ilgili yönetim işlemlerini yapmak için **gpasswd** komutu kullanılabilir. Kullanıcılar **newgrp** komutunu kullanarak üye oldukları grup dışında farklı bir grup hakkına sahip olabilirler. Kabuktan çıkış yaptıklarında bu grubun hakkı da yok olacaktır.

[ila@ila ~]$ id -Gn

ila

[ila@ila ~]$ newgrp ila-test2

Password:

[ila@ila ~]$ id -Gn

ila-test2 ila

gpasswd komutu seçenek verilmeden kullanılırsa parametre olarak verilen grubun şifresini değiştirir.

gpasswd komutununun seçenekleri aşağıdaki gibidir:

**gpasswd <seçenekler> grup\_adı**

Seçenek verilmeden kullanılırsa grubun şifresini değiştirir.

* -**a** : Gruba kullanıcı ekler.
* -**d** : Gruptan kullanıcı çıkarır.
* -**R** : newgrp komutu ile gruba erişimi kapatır.
* -**r** : Gruba ait şifreyi kaldırır.
* -**A** : Grup yöneticilerini belirler. (birden fazla kullanıcı ismini parametre olarak alabilir.)
* -**M** : Grup üyelerini belirler. (birden fazla kullanıcı ismini parametre olarak alabilir.)

**Grup Silme**

Grup silme sırasında groupdel komutu kullanılabilir.

[root@ila ~]# **groupdel grup\_adı**

**/etc/group dosyası**

**/etc/group** dosyasında gruplar ile ilgili bilgiler bulunur. Bu dosyada alanlar : işareti ile birbirinden ayrılmıştır. Bu dosyanın yapısı aşağıdaki gibidir:

**grup\_adi:şifre:gid:username1,username2,username3,...**

ila:x:500:ila1,ila2

* **grup\_adı** : Bu kısımda grubun ismi belirtilir.
* **şifre** : /etc/passwd dosyasında olduğu gibi bu alanda da x harfi bulunur. Grubun şifresi /etc/gshadow dosyasında şifrelenmiş olarak tutulur.
* **gid** : Grubun kimliğini belirleyen sayıdır.
* **username[1-3]** : Son alanda bu gruba üye olan kullanıcıların kullanıcı adları virgülle ayrılmış olarak bulunur.

**[root@ila ~]# cat /etc/group**

root:x:0:root

bin:x:1:root,bin,daemon

daemon:x:2:root,bin,daemon

sys:x:3:root,bin,adm

adm:x:4:root,adm,daemon

tty:x:5:

disk:x:6:root

lp:x:7:daemon,lp

mem:x:8:

kmem:x:9:

wheel:x:10:root

mail:x:12:mail,postfix

uucp:x:14:uucp

man:x:15:

games:x:20:

gopher:x:30:

video:x:39:

dip:x:40:

ftp:x:50:

lock:x:54:

audio:x:63:

nobody:x:99:

users:x:100:

dbus:x:81:

utmp:x:22:

utempter:x:35:

floppy:x:19:

vcsa:x:69:

abrt:x:499:

cdrom:x:11:

tape:x:33:

dialout:x:18:

haldaemon:x:68:haldaemon

saslauth:x:498:

postdrop:x:90:

postfix:x:89:

sshd:x:74:

tcpdump:x:72:

slocate:x:21:

stapdev:x:497:

stapusr:x:496:

ntp:x:38:

test:x:500:

ila:x:501:

[root@ila ~]#

**/etc/gshadow Dosyası**

/etc/shadow dosyasında kullanıcıların şifre bilgileri bulunduğu gibi grup şifreleri ve grup yönetimi ile ilgili bilgiler /etc/gshadow dosyasında bulunur.

Dosyanın yapısı aşağıdaki gibidir:

**grup\_adı:şifre:grup\_yöneticisi:grup\_üyeleri**

ila:8.MyhSj1uYfoY:ila:ila1,ila2,ila-test

* grup\_adı : Grubun ismi belirtilir.
* şifre : Grubun şifresi şifrelenmiş halde bulunur.
* grup\_yöneticisi : Grup yöneticilerini belirtir. Yöneticiler gpasswd komutunu kullanarak grup üyelerini düzenleme yetkilerine sahip olurlar.
* grup\_üyeleri : Grup üyelerini gösterir.

**[root@ila ~]# cat /etc/gshadow**

root:::root

bin:::root,bin,daemon

daemon:::root,bin,daemon

sys:::root,bin,adm

adm:::root,adm,daemon

tty:::

disk:::root

lp:::daemon,lp

mem:::

kmem:::

wheel:::root

mail:::mail,postfix

uucp:::uucp

man:::

games:::

gopher:::

video:::

dip:::

ftp:::

lock:::

audio:::

nobody:::

users:::

dbus:!::

utmp:!::

utempter:!::

floppy:!::

vcsa:!::

abrt:!::

cdrom:!::

tape:!::

dialout:!::

haldaemon:!::haldaemon

saslauth:!::

postdrop:!::

postfix:!::

sshd:!::

tcpdump:!::

slocate:!::

stapdev:!::

stapusr:!::

ntp:!::

test:!::

ila:!::

[root@ila ~]#

**/etc/login.defs Dosyası**

Kullanıcı ve grup işlemlerine ait varsayılan değerlerin bulunduğu bir dosyadır. İçinde yer alan bazı deyimler şu şekildedir:

PASS\_MAX\_DAYS 99999

PASS\_MIN\_DAYS 0

PASS\_WARN\_AGE 7

# useradd komutu ile atanacak minimum ve maksimum UID değerleri

UID\_MIN 1000

UID\_MAX 60000

# Sistem kullanıcılarına ait hesaplar

#SYS\_UID\_MIN 100

#SYS\_UID\_MAX 999

# groupadd komutu ile atanacak minimum ve maksimum UID değerleri

GID\_MIN 1000

GID\_MAX 60000

# Sistem gruplarına ait hesaplar

#SYS\_GID\_MIN 100

#SYS\_GID\_MAX 999

Bu dosyada tanımlanabilecek tüm deyimler hakkında bilgi için man sayfasına bakınız.

[root@ila ~]# man login.defs

**Uygulama 1**

1. Bilgisayaraınızda ogrenciler isminde GID değeri 666 olan bir grup açın.
2. Makinanızda ogrenci isminde bir kullanıcı açın ve bu kullanıcının birincil grubunun ogrenciler olmasını sağlayın.
3. Makinanızda <isminiz> kullanıcı adına sahip, ek olarak ogrenciler grubuna dahil olan ve UID değeri 600 olan bir hesap açın.
4. ogrenci ve <isminiz> kullanıcılarına birer şifre atayın.
5. Bilgisayarınızın 2. terminal ekranında (Ctrl + Alt + F2) açtığınız <isminiz> kullanıcısı ile oturum açın.
6. isminiz kullanıcısı altında iken ogrenci hesabının haklarını alın.
7. ogrenci haklarında iken ogrenci ev dizininde adı dosya.txt olan ve içeriğinde "Merhaba" yazan bir dosya yaratın.
8. ogrenci haklarında iken ogrenci ev dizininin grup izinlerini, gruba dahil olan kullanıcıların bu dizine geçme ve listeleme işlemlerini yapacak şekilde düzenleyin.
9. ogrenci haklarını bırakın ve isminiz kullanıcısı ile ogrenci kullanıcısının ev dizinini listeleyin.

**Çözümler 1**

**1 -** [root@ila ~]# groupadd -g 666 ogrenciler

[root@ila ~]#

**2 -** [root@ila ~]# useradd -g ogrenciler ogrenci

[root@ila ~]#

**3 -** [root@ila ~]# useradd -G ogrenciler -u 600 isminiz

[root@ila ~]#

**4** - [root@ila ~]# passwd ogrenci

Changing password for user ogrenci.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@ila ~]# passwd isminiz

Changing password for user isminiz.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@ila ~]#

**5 -** <Ctrl + Alt + F2>

**6 -** [isminiz@ila ~]$ su -l ogrenci

Password:

[ogrenci@ila ~]$

**7 -** [ogrenci@ila ~]$ vim dosya.txt

[ogrenci@ila ~]$

**8 -** [ogrenci@ila ~]$ chmod g+rx /home/ogrenci

[ogrenci@ila ~]$

**9 -** [ogrenci@ila ~]$ logout

[isminiz@ila ~]$ ls -l /home/ogrenci

total 4

-rw-r--r-- 1 ogrenci ogrenciler 8 Sep 29 19:49 dosya.txt

[isminiz@ila ~]$

**PAKET YÖNETİMİ VE PROGRAM KURULUMU**

**Paket Kavramı**

NOT:

PAKET YÖNETİMİ

Bilgisayara uygulama kurup kaldırmak için kullanılır.

-Kaynak kodu al,bilgisayarda derle, çalıştır.Mantık budur. Paket yöneticileri de genelde bunu yapar.

\*Google da Clamav diye aratalım(open source antivirüs)

-DOwnload diyelim

-tar ı açalım.(tar -zxf Clamav...)

-Dosyanın içine geç.

-less INSTALL ı oku

\*\*Configura çalıştır diyor.Configure demek; Kurulumdan önce toplanacak bilgilerin toplamasını sağlar.

\*\*make:derleme aşamasıdır. Kaynak kodu derleriz. Neyin ne şekilde nasıl ve ne sırada derleneceği makefile da yazar.Büyük projeler parçalardan oluşur.

\*\*make install: ilgili dosyaların dizin agacında ilgili yerlere gitmesi sağlanır.

-Redhat in attıgı mantıga göre: zaten işlemciler bir önceki işlemcinin komut setini de destekliyor.Dolayısıyla eski bir işlemci mimarisi seçelim.(32 bit için i386 seçilmiş.)Önden buna göre derleyelim ve arşiv haline getirelim(cpio arşivi.). Bu arşiv dosyasına bir gtup bilgiler ekliyelim ve bu dosyaların nerelerde depolanacagı bilgilerini verelim buşekilde rpm paketi oluşturulmuştur.(REdhat package manager)

Redhat türevi dağıtımlar baştan aşagı rpm lerden oluşur.(Yani derlenmiş paketler)

-Digeri ise DEBIAN dpkg; debian tabanlı dağıtımlarda bunu destekler.

\*\*RPM\*\*(dpkg ile farkları vardır.dpkg son kullanıcılar için kolaylıklar sağlarken, rpm in sistem yöneticilerine yönelik sağladıgı kolaylıklar vardır.)

-make komutu da make.rpm ile gelir.Bunad bir paket başka bir pakete bagımlılık olarak gösterilebilir.

-cat /etc/redhat-release (|ubuntu-release|\*-release) ile dagıtımın sürümünü öğrenebiliriz.

-gcc(paket adı)-4.0(sürümno)...-rpm derlenme numarası-...(hangi dagıtım için oldugunun kod numarası).rpm==>rpm paketinin okunuşu

-rpm komutu daha çok sorgulama yapmak için kullanılır.

-rpm -i gcc-4......rpm ile derleme yapabiliriz.dediğimizde bir uyarı çıkacak bu uyarı başka rpmleride istediğini söyler.Onları da gidip itünün ftpsinden çekebiliriz.itü nün ftp sinde rpm paketleri var.

-rpm -q ElectircFense ile paketin yüklenip yüklenemediğini öğrenebeiliriz.

-rpm -e ElectricFense diyerekte paketi kaldırabiliriz.(neyin nereye konduğuna dair bilgi localde bir veritabanında tutulduğundan dolayı bu şekilde paket kaldırılabilir.)

-rpm -ivh ElecricFense...rpm şekilde kurulum yapılabilir.

-rpm -Uvh ElectricFense...rpm ile paketi Upgrade edebiliriz.

-rpm -Fvh ElectricFense...rpm u ile işlevi aynıdır ama u da paket yüklü değilse yükler.F de paketin yüklü olması gerekir.

-rpm -ivh --test ElectricFense....rpm ; paketle ilgili bir problem olup olmadığını paketi yüklüyormuş gibi yaparak kontrol eder.

-rpm -q paketadı ile paket aranabilir.

-rpm -qa ile bütün yüklü rpmleri listeler.

-rpm -ql ElectricFense ; daha önceden yüklediğimiz ElectricFense paketi hangi dosyalarıda beraberinde getirdi. BUnları listeler.

-rpm -qf /root/ ;root un evdizini hangi paketle geldi diye sorabiliriz.

-rpm -qi ElectricFense ; ElectricFense Paketinin ismini gösterir.

Şuana kadar yüklü paketlerle çalışıtık ya dosya yüklenmemişse yada yoksa;

-rpm -qpl ElectricFense.....rpm veya rpm -qpi ElectricFense....rpm ; ile paketin içeriğine falan bakabiliriz.

-rpm -q --scripts sendmail(paket adıdır bu) ;preinstall: yüklemeden önce çalışan komutlar ;postinstall: yüklemeden sonra çalışan komutlar. preuninstall ve postunistall gibi kaldırma öncesi ve kaldırma sonrası içinde çalışabiliriz .Yani rpm paketleri;aynı zamanda betikte çalıştırabilir.Bunlara bakarak paketin yüklenirken zararlı birşeyler yapıp yapmadığını kontrol edebiliriz.

-rpm2cpio ElectricFense(cpio arşivi olarak açar) | cpio -it(içeriği listeler)

-rpm -v sendmail (sadece dosyları yüklemez md5 ini falanda alır.)

\*\*\*\*YUM\*\*\*\*(yellowdog; bir dağıtım adı)

-mantık insanlar paketleri internette aramasın; insanların aradığı paketleri bir yazılım gitsin depolarda arasın bulsun ve bilgisayara yüklesin.(Debian daki apt ye denk gelmektedir.)

1.yum install gcc ->1.depoların veritanı indexlerini günceller.

2.bagımlılıkları çözer.

-yum depodaki en güncel sürümle çalışır.Mümkün olan en üst sürüme çalışır. Zaten mantık olarakda olması gereken budur en günceli en iyisidir.

3.Bagımlılıkları listeler ve yükleyip isteyip istemediğini sorar.

-vim /etc/yum.repos.d/ internette bakılacak olan repoların adını tutar.repolarla ilgili net adresleri falan vardır.burada baseurl ve mirrorlist diye iki repo adresi vardır.Birine baglanamazsa diğerine bakar(mirorlistten bakmaya devam eder.)

-yum search gcc : içerisinde gcc geçen paketleri listeler.

-yum provides /etc/X11/ hangi paketlerin /etc/X11 i getirdiğini söyler.

yum provides \*http.conf\* :(http.conf un nerde olduğunu bilmiyorsak örnegin böyle hangi paketlerin bunu getirdiğini bulabiliriz)

-yum update paketadi ; yum update(tüm makinayi günceller) paket adıda verince paketi günceller.(bir sonraki minor sürüme günceller major sürüme geçiş yoktur. Buna update-rool over versioning denir.)

-yum remove paket adi (bunu kullanmak iyi değildir çünkü paketin bagımlılıkları ile birlikte kaldırır. örnegin böyle libgcc yi kaldırmaya kalksak neredeyse makinedeki herşeyi kaldırmış oluruz.)

-rpm -e paketadi (paket silmek için kullanılır.)

-rpm -e --nodeps libgcc : bagımlılıkları salla demektir.

-yum localinstall /Downloads/ElectiricFense....rpm paketi verilen adresten kur bagımlılıklar git internetten bul demektir.

\*\*\*\*Clamav a dönersek;

-./configure dedik (yum ile gcc yi kurduk)

-şuan başka bir program istiyor onuda yükleriz(yum ile) BU sefer kullanıcı eklememizi ister.

-useradd -r clamav :(BUn kullanıcıyı clamav kullanacak o yüzden ismide clmav olsun.) \*\*uid olarak 500 ün altında deger verilir.

daha sonra yine configure ettik.(--help ile bakabiliriz.)

\*\*bir paketin configure u olmamaısıda mümkündür. olmak zorunda değildir.

\*\*\*make: bildiğimiz derleme aşaması

\*\*\*make install :bilgisayarda ilgili dosyaları ilgili yerlere attık.

-bu aşamadan sonra bu şekilde kurduğumuz kaldıramayız. Çünkü nereye hangi dosyayı attığını bilmiyoruz. BU yüzden paket kullanmak iyidir.Ama paketin kötü yanı ise bazen paketi hazırlayanın sundugu seçenekler bize uymayabilir.

\*\*\*\*\*DPKG\*\*\*

.dep uzantılıdır.Rpm paketleri için birsürü depo bulabiliriz.Ama debian paketleri için bulamayız.Çünküde bian tarafından komut satırında paket eklenip kaldırılması pek istenmez.

----RPM -----DPKG

YÜKLEME -i paketadi -i dosyaadi

PAKET YÜKLÜ MÜ? -q paket adi -l paket adı örn:dpkg -l nano ;nano paketi ile ilgili paketleri listele

PAKETİN İÇERİĞİNDEKİ DOSYALARIN LİSTESİ -ql paket -L

DOSYA YOLUNUN HANGİ PAKETLE GELDİĞİ -qf dosyayolu -S dosyayolu

PAKETİ SİL -e paketadi --remove(-r) :yapılandırma dosyalarını bırakır.

dpkg -L nano:nano ile bilgisayarımıza yüklenecek paketlerin listesi --purge(-p) :yapılandırma dosyalarını siler

dpkg ile genelde paket kurmayız sorgu yaparız.

rpm getirdigi bütün dosyaları spec dosyasına bakarak siler.Eger programın conf dosyasında değişiklik yaptıysak .old uzantısı koyup bırakır.

-yum depodaki paket listesini download eder.Eger birşey yükliyeceksek tekrar indirir.bu yüzden bir süre cache te tutar.

-dpkg de farklıdır.paket bilgilerini cache te tutmaz.Herhalukarda; paketin farkına netten indirilip indirilmeyecegeine bakar.

-yum depodaki en güncel bilgiyi alır.

-apt cache deki bilgiyle hareket eder.

-dpkg için arka plan komutu oldugu man ında bile yazar.apt (advance package tool) kullanılır.Redhat tarafında da yum kullanılır.

\*\*\*\*\*yum\*\*\*\*vs\*\*\*\*\*apt\*\*\*\*\*

-----yum

-depo bilgilerini /etc/yum.repos.d de tutar.

-install paketadı

-update: tüm makineyi update eder.

-yum yüklediği herpaketi /var/cache/yum un altına zaten download eder.

yum un cache i şişer. yum cache all ile bütün cache i uçurabiliriz.

-yum search pattern

-yum info paketadi:paket bilgilerini döker.

-yum provides dosya:dosyayı getirecek paketleri listeler.

------apt

-/etc/apt/sources.list den dosya bilgilerini görüntüleyebiliriz.

-apt-get install paketadi

-apt-get update :cache paketlerin güncel sürümlerini çekebilmek için günceller.

-apt-get upgrade :upgrate etmeden önce cache update etmek gerekir.

do-release-upgrade :ubuntuyu en güncel sürüme günceller.

apt-get dist upgrade :bazen projeleri dağıtımlar ekleyip çıkarabilir.bu tür paket değişimleri olmadan güncelleme yapmak istiyorsak kullanabiliriz.

-apt-get -d install vim: sadece paketi download eder.(.deb olarak)

-apt-cache clean : cache uçurur. /var/cache/apt/archive :apt nin cache idir.

-apt-cache search vim (veya pattern)

-apt-cache showpkg vim (veya pattern)

-apt-file search file (vasrsayılanda yüklü gelmez yüklemek ve update etmek gerekir.->apt-file update ile.)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Linux işletim sistemine program yüklenmesi kaynak kod üzerinden ilgili porgamın kütüphaneleri ile birlikte derlenmesi ve ardından dosyalarının ilgili dizinlere taşınması şeklinde olur.

Program derleme işleminin standart bir yolu yoktur. Her projenin kendine ait farklı bir derlenme yolu vardır ve proje ile birlikte gelen açıklama dosyaları ile anlatılır. Yine de çoğu proje 4 temel adımı izler:

1. Kaynak kodun arşiv formatında (gz, tar.gz, tar.bz, vb …) elde edilip derleme dizini içine bu arşivin açılması.
2. Sistemdeki yüklü program ve kütüphanelerin kontrol edilmesi ve/veya kurulum seçeneklerinin belirlenmesi **(./configure.pl)**
3. Proje kaynak kodunun derlenmesi **(make)**
4. Derleme işleminden sonra elde edilen dosya ve dizinlerin sistem ağacında ilgili yerlere kopyalanması **(make install)**

Kaynak kod üzerinden derlemelerde farklı kütüphanelerin derlenmesi sistemde bu kütüphanelerin bulunmaması gibi sıkıntılar kurulum işlemini zorlaştırmaktadır. Paket kavramı işte bu zorlukları aşmak amacı ile ortaya çıkmıştır. Böylece hiçbir teknik bilgisi olmayan son kullanıcının bile kolayca Linux işletim sistemine uygulama yüklemesi mümkün kılınmaktadır.

Paket yönetim sistemleri sayesinde aşağıdaki özellikler de sağlanır:

Paket yönetim sistemlerinde her paket küçük bir bileşen olarak hazırlanır. Böylece büyük projelerin sadece istenen bileşenleri yüklenebilir.

Güçlü sorgulama yapıları sayesinde yüklenen paketler ve dosyalar hakkında çeşitli bilgiler alınabilir. Örneğin; bir paket kurulduktan sonra dosyalarının ne şekilde değiştiği veya bir dosyanın hangi paket tarafından sisteme yüklendiği öğrenilebilir.

Eğer işletim sistemi üzerine hep Paket sistemi ile kurulum yapılırsa bir üst işletim sistemine çıkmak tek komutla kolayca yapılabilmektedir.

Paket yönetim sistemi ile kurulan bir projeyi sistemden kaldırmak kaynak kodla kurulum sistemine göre son derece kolay ve kesindir.

Günümüzde tüm modern Linux dağıtımlar paket sistemi kullanmaktadır. Farklı paket yapısı projeleri olsa da en yaygın olan Red Hat tarafından geliştirilen **RPM** (**RedHat Package Manager**) ve Debian sistemlerde geliştirilen **Debian Package Management** yapılarıdır. Çoğu dağıtım bu projelerden kendine yakın olanı kısmen ya da tamamen kulanmaktadır. Örneğin; Ubuntu sistemler Debian sistemini kullanır.

**RPM (RedHat Package Manager)**

RPM, Red Hat türevi sistemlerde, Mandriva ve SuSe dağıtımlarında kullanılan paket yönetim sistemidir.

RPM paketleri aşağıdaki biçimde isimlendirilir.

**<isim>-<sürüm>-<yayım>[<dağıtım>].<mimari>.rpm**

* <**isim**> : Yazılımın ismi
* <**sürüm**> : Yazılımın sürümü
* <**yayım**> : Aynı sürüm yazılım kullanılarak üretilen kaçıncı derleme olduğu bilgisidir. Buradaki durum projenin kodunda değil RPM paketinin derleme seçenekleri veya yapılandırma dosyalarında değişiklik olduğunu gösterir.
* <**dağıtım**> : Bu bilgi RPM dosyasının hangi dağıtıma ait olduğunu gösterir. Dağıtım bağımsız paketlerde bu bilgi olmayabilir. Örneğin; fc4 (Fedora Core 4), suse100 (Suse Linux 10.0), mdk (Mandriva Linux)
* <**mimari**> : Paketin hangi mimari için hazırlandığı.i386, x86\_64, noarch

gcc-4.1.2-48.el5.i386.rpm

kernel-2.6.18-194.11.1.el5.x86\_64.rpm

asciidoc-8.4.5-4.1.el6.noarch.rpm

**RPM Veritabanı**

RPM yapısında sisteme kurulan her paket hakkındaki bilgiler /var/lib/rpm dizini altındaki dosyalara kaydedilir.

[root@localhost ~]# file /var/lib/rpm/\*

/var/lib/rpm/Basenames: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Conflictname: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/\_\_db.001: data

/var/lib/rpm/\_\_db.002: X11 SNF font data, LSB first

/var/lib/rpm/\_\_db.003: X11 SNF font data, LSB first

/var/lib/rpm/Dirnames: Berkeley DB (Btree, version 9, native byte-order)

/var/lib/rpm/Filemd5s: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Group: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Installtid: Berkeley DB (Btree, version 9, native byte-order)

/var/lib/rpm/Name: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Packages: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Providename: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Provideversion: Berkeley DB (Btree, version 9, native byte-order)

/var/lib/rpm/Pubkeys: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Requirename: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Requireversion: Berkeley DB (Btree, version 9, native byte-order)

/var/lib/rpm/Sha1header: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Sigmd5: Berkeley DB (Hash, version 8, native byte-order)

/var/lib/rpm/Triggername: Berkeley DB (Hash, version 8, native byte-order)

Buradaki \_\_db ile başlayan dosyalarda bir sorun oluşması durumunda aşağıdaki komutla veri tabanı yeniden yaratılabilir.

[root@ila ~]# rpm --rebuilddb

[root@ila ~]#

**rpm Komutu**

RPM paketleri üzerindeki tüm işlemler rpm komutu ile yapılır. Bazı örnek işlemler:

* Paket kurulumu
* Paket güncellemesi
* Paket kaldırılması
* RPM veritabanının sorgulanması
* Paket dosyasının sorgulanması
* Kaynak kodundan paket oluşturulması
* Paket onaylanması
* Paket dosyasının onaylanması

Paket kurulumu, güncellenmesi veya kaldırılması sırasında, rpm komutu sistemin uygun durumda kalması için bir kaç kontrol yapar. Bazı kontroller şunlardır:

* Paket kurulması için diskte yeterli yer var mıdır?
* Paket kurulabilmesi için gerekli kütüphane ve paketler kurulu mudur?
* Paket daha önce kurulmuş mu veya kurulum sırasında dosya çakışması olacak mı?

Normalde tüm koşullar sağlanmadığı takdirde paket kurulmaz ama gerekli durumlarda parametreler verilerek bu koşulların kontrol edilmemesi sağlanabilir.

**RPM Paket Kurma/Güncelleme/Kaldırma**

Yükleme işlemleri için komutun genel kullanım şekli.

**rpm <Yapılacak\_İşlem> [<İşlem\_Seçenekleri>] <Paket Dosyasının Yolu>**

* Paket Kurma: Paket kurmak için -i (--install) işlem seçeneği kullanılır. Kurulacak paket ismi olarak birden fazla paket dosyası gösterilebilir.
* Paket Güncelleme: Kurulu olan bir paketi güncellemek için -U (--upgrade) veya -F (--freshen) seçenkeleri kullanılır. -U seçeneği paket sistemde yüklü ise günceller değilse de yükler ama -F seçeneği sadece paket sistemde yüklü ise güncelleme yapar.

Kurulum adımlarında en sık kullanılan işlem seçenekleri:

-v : kurulum hakkında bilgi verir.

-h : Kurulum işleminin aşamalarını 50 adet “#”işareti ile gösterir.

Paket Kaldırma:Kaldırma işlemi için komutun genel kullanım şekli:

rpm <-e|--erase> [<İşlem\_Seçenekleri>] <Paket\_Adı>

* Paket kaldırılırken RPM veritabanında bağımlılık kontrolü yapılır ve ilgili pakete bağımlı olan ve sisteme yüklenmiş başka bir paket varsa kaldırma işlemi yapılmaz. Eğer İşlem\_Seçeneği olarak --nodeps kullanılırsa bağımlılık kontrolü yapılmadan paket sistemden kaldırılır.

**RPM Paket Sorgulamaları**

rpm komutunun -**q** seçeneği kurulu paketin versiyonunu ve yayım numarasını öğrenmek için kullanılabilir.

**rpm -q <paket\_adı>**

Bu komutun çıkışı paket ismini, sürümünü ve derleme numarasını verir.

[root@ila ~]# **rpm -q firefox**

firefox-1.0.7-1.2.fc4

-**qa** seçeneği sistemde kurulu olan tüm paketlerin listesini verir.

[root@ila ~]# **rpm -qa**

basesystem-8.0-5

device-mapper-1.01.02-1.0

hdparm-5.9-1

db4-4.3.27-3

fedora-release-4-2

rpm komutu kullanılarak herhangi bir dosyanın hangi pakete ait olduğu veya bir paketin hangi dosyaları içerdiği öğrenilebilir.

-qf <dosya\_adı> ile parametre olarak verilen dosyayı içeren pakete ulaşılabilinir.

[root@ila ~]# **rpm -qf /etc/passwd**

setup-2.5.44-1.1

**-ql <paket\_adı>** ile paketin içerdiği dosyalar listelenebilir.

[root@ila ~]# **rpm -ql setup**

/etc/aliases

/etc/bashrc

/etc/csh.cshrc

/etc/csh.login

rpm komutu -qi seçeneği ile çalıştırılırsa parametre olarak verilen paket hakkında bilgi alınır.

[root@ila rpm]# **rpm -qi setup**

Name : setup Relocations: (not relocatable)

Version : 2.5.44 Vendor: Red Hat,Inc.

Release : 1.1 Build Date: Tue 06 Sep 2005 07:05:10 PM EEST

Install Date: Fri 24 Feb 2006 10:03:19 PM EET Build Host: sizzler.build.redhat.com

Group : System Environment/Base Source RPM: setup-2.5.44-1.1.src.rpm

Size : 37073 License: public domain

Signature : DSA/SHA1, Wed 07 Sep 2005 02:38:59 AM EEST, Key IDb44269d04f2a6fd2

Packager : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>

Summary : A set of system configuration and setup files.

Description :The setup package contains a set of important system configuration and

setup files, such as passwd, group, and profile.

Eğer paket sistemde kurulu değilse -p seçeneği ile rpm dosyası gösterilerek bilgilerin rpm dosyasından alınması sağlanabilir.

**rpm -qip <paket\_dosya\_adı>**

rpm komutu -**qp --scripts** parametreleri ile çalıştırılırsa paket kurulumu sırasında çalıştırılan betiklere ulaşılabilir.

[root@ila rpm]# **rpm -qp --scripts sendmail-8.13.4-2.i386.rpm**

preinstall scriptlet (using /bin/sh):

/usr/sbin/useradd -u 47 -d /var/spool/mqueue -r -s /sbin/nologin mailnull >/dev/null 2>&1

/usr/sbin/useradd -u 51 -d /var/spool/mqueue -r -s /sbin/nologin smmsp >/dev/null 2>&1

exit 0

postinstall scriptlet (using /bin/sh):

#

# Convert old format to new

#

RPM yapısında, sistemde kurulan her bir paketin her dosyası için veri tabanında kayıt tutar. Bu kayıt dosya büyüklüğünü, sahibini, izinleri gibi bilgileri barındırır. Bu kayıtlar kullanılarak hangi dosya kurulumdan sonra ne şekilde değişmiştir belirlenebilir. –V seçeneği ile paketin içerdiği dosyalarda olan değişikliklere ulaşılabilinir.

**rpm –V <paket\_adı>**

Komutun çıktısı her bir değişen dosya için bir satır içerir. Bu satır üç alandan oluşur:

Değişiklik tipi

* . Değişiklik yok.
* 5 MD5 checksum değişmiştir.
* D Aygıt major ya da minor numarası değişmiştir.
* G Sahip grup değişmiştir.
* L Link yolu değişmiştir.
* M Dosya biçimi değişmiştir.
* S Dosya büyüklüğüdeğişmiştir.
* T Değiştirilme zamanıdeğişmiştir.
* U Sahip kullanıcı değişmiştir.

Dosya Tipi

* c Yapılandırma dosyası.
* d Dökuman dosyası.
* g Paket içeriğinde bulunmayan dosya.
* l Lisans dosyası.
* r Beni Oku (readme) dosyası

Dosyanın tam yolu

[root@ila ~]# **rpm -V sendmail**

..?...... c /etc/mail/domaintable

..?...... c /etc/mail/mailertable

.......T. c /etc/mail/sendmail.cf

..?...... c /etc/mail/virtusertable

S.5....T. c /var/log/mail/statistics

**RPM Paketlerinin Açılması**

RPM paketleri özel cpio formatında arşiv dosyalarıdır. Arşivin başında ikili formatta RPM paketine ait bilgiler bulunur ve beraberindeki dosyalar cpio formatında arşivlenmiştir. Rpm dosyaları **rpm2cpio** komutu ile ikili veri kısmında ayrılıp cpio arşiv formatında standart çıktsına yazar. Ardından da cpio komutu ile arşiv açılarak dosya sistemine veriler yazılabilir.

[root@localhost rpm]# rpm2cpio vim-enhanced-7.0.109-6.el5.x86\_64.rpm | cpio -i --make-directories

5518 blocks

**Kaynak RPM Dosyasından Kurulum**

Kaynak RPM paketlerinde kaynak kod dosyaları bulunur. Bu dosyalar derlenerek RPM dosyaları oluşturulur.

Kaynak RPM dosyalarının ismleri src.rpm ile biter.

Öncelikle src.rpm paketi normal paket kurulurmuş gibi kurulur ve böylece paketin ilgili dosyaları kullanıcının ev dizininin altına rpmbuild altına kopyalar. Bu dizinlerin içeriği şunlardır:

\* BUILD Kaynak RPM'ler bu dizinde açılır.

\* RPMS Kaynak RPM derlendiğinde yeni RPM paketi burada saklanır. Bu dizin her mimari için ayrı bir altdizin içeriyor.

\* SOURCE Kaynak kod, genelde TAR dosyası olmaktadır, ve yamalar bu dizinde bulunur.

\* SPECS Bu dizin SPEC dosyaları içerir, SPEC dosyaları paket kaynaklar listelerini, yamaları, derlenme-zaman seçenekleri, kurulum sonrası adımları ve paket bilgilerini içerir.

\* SRPMS Yeni oluşturulan kaynak RPM dosyaları bu dizinde saklanır.

Kaynak RPM’den RPM oluşturmak için SPEC dizinine geçilir ve aşağıdaki komut çalıştırılır:

rpmbuild -bb <spec\_dosya>

Bu işlemin sonucunda RPM dizininin altında ilgili RPM'ler oluşur.

[root@ila ~]# rpm -ivh sudo-1.6.8p8-2.4.src.rpm

1:sudo ########################################### [100%]

[root@ila ~]# cd rpmbuild/SPECS/

[root@ila SPECS]# rpmbuild -bb sudo.spec

......

[root@ila SPECS]# ls rpmbuild/RPMS/i386/

sudo-1.6.8p8-2.4.i386.rpm sudo-debuginfo-1.6.8p8-2.4.i386.rpm

[root@ila SPECS]# rpm -ivh rpmbuilds/RPMS/i386/sudo-1.6.8p8-2.4.i386.rpm

Preparing... ########################################### [100%]

1:sudo ########################################### [100%]

**YUM (Yellowdog Updater Modified)**

RPM paketleri her ne kadar kurulum ortamları ile birlikte gelse de güncel paketler her zaman internet üzerinden temin edilir. Bir paketi internet üzerinden bulunacağı zaman bağımlılığı olan diğer paketleri de bulup yüklemek gerekir. YUM bu işlemi kolaylaştırmak ve internet bağlantısı olduğu sürece RPM yapısındaki işlemleri kolayca yapmak için tasarlanmıştır.

YUM ile tek bir komutla bağımlılıkları ile birlikte paketler kurulabilir ve tek komutla sistem güncellenebilir.

**YUM Yapılandırma Dosyaları**

YUM’un genel yapılandırma dosyası **/etc/yum.conf'** dur. Bu dosyaya ek olarak güncellemeleri hangi sunuculardan çekeceği bilgilerini aldığı **/etc/yum.respos.d/** dizini altındaki .**repo** uzantılı dosyalara bakar. Güncellemelerin çekildiği sunuculara **depo** ismi verilir.

Yapılandırma dosyası **[]** parantezler ile bölümlere ayrılır**. [main]** bölümü ilk bölümdür ve tüm depolar için aksi belirtilmedikçe genel ayarları tanımlar. **[depo\_ismi**] şeklinde her depoya ait kendi yapılandırması yapılabilir. Normalde dağıtımlar kurulduklarında kendi resmi depolarına ait yapılandırmalar hazır bir şekilde kurulurlar.

Genel olarak bir depo tanımı en basit haliyle şu şekilde yapılır.

[Depo\_İsmi]

name=Depo\_İsmi

baseurl=…

enabled=<1|0>

Buradaki önemli alanlar **baseurl** ve **enabled** alanlarıdır. Enabled değeri **1** ise depo kullnılır, **0** ise depo tanımlı olsa bile kullanılmaz. Baseurl alanına FTP sunucusu (**ftp://),** web sunucusu (**http://)** veya yerelde bir dizin ise (**file://)** protokolleri ile adres girilebilir. Daha detaylı bilgiye man **yum**.**conf** ile bakılabilir.

**yum Komutu**

yum komutuna verilen ikincil komutlar ile işlemler yapılır. Bunların bazıları:

* yum install paket1 paket2 [...] : Parametre verilen paketlerin son sürümlerinin kurulması için kullanılır.
* yum update paket1 paket2 [...] : Parametre olarak hiçbir paket adı verilmezse sistemde kurulu tüm paketleri günceller.
* yum check-update : Sistmde kurulu paketlere ilişkin güncelleme olup olmadığını görmek için kullanılır.
* yum upgrade paket1 paket2 [...] ,i Çalışan Linux dağıtımına ilişkin güncelleme yapmak için kullanılır, yum update –obsoletes komutuna eşdeğerdir.
* yum remove/erase paket1 paket2 [...] ,i Parametre verilen paketleri, bağımlılık oluşturduğu paketler ile birlikte sistemden kaldırır.
* yum list [...] : Sistemde kurulu/güncellenebilecek paketlerin listesini verir.
* yum info [...] : Paketler hakkında açıklama ve kısa bilgi verir.
* yum [provides/whatprovides] : Belirli bir paket ile gelen dosyaları yada bir dosyanın hangi paket ile geleceğini aramak için kullanılır.
* yum clean [packages/headers/all] : Yerel sistemde tutulan paket veya paketlere ilişkin başlık bilgilerinin silinmesi için kullanılır.
* yum groupinstall grup1 grup2 [...] : Belirtilen grubun paketlerini kurmak için kullanılır.
* yum search : Paket aramak için kullanılır.
* yum localinstall <paket> - Yereldeki RPM paketlerini kurmak için kullanılır.
* yum localupdate : Sistemdeki RPM paketleri ile sistemi güncellemek için kullanılır.
* yum deplist paket1 paket2 [...] : Belirtilen pakete ilişkin bağımlılık bilgilerini listeler.

**DPKG (Debian Package Management)**

DPKG isminde olduğu gibi Debian sistemler için geliştirilmiş bir paket yönetim sistemidir. Günümüzde Ubuntu gibi birkaç farklı dağıtımda bu yapıyı kullanmaktadır.

Debian paket isimlendirmesi RPM yapısına çok benzer.

**<isim>\_<sürüm>-<yayım>[<dağıtım>]\_<mimari>.deb**

* <isim> : Yazılımın ismi
* <sürüm> :
* <yayım> : Aynı versiyon yazılım kullanılarak üretilen kaçıncı derleme olduğu bilgisidir. Buradaki durum projenin kodunda değil paketin derleme seçenekleri veya yapılandırma dosyalarında değişiklik olduğunu gösterir.
* <dağıtım> : Bu bilgi DEB dosyasının hangi dağıtıma ait olduğunu gösterir. Dağıtım bağımsız paketlerde bu bilgi olmayabilir. Örneğin ubuntu
* <mimari> : Paketin hangi mimari için hazırlandığı.i386, amd64, all

linux-generic\_2.6.32.24.25\_i386.deb

gcc-4.4-base\_4.4.3-4ubuntu5\_amd64.deb

terminator\_0.8.1-1\_all.deb

**dpkg Komutu**

Debian sistemi RPM sisteminden komut yapısı olarak daha farklıdır ama bir paket yönetim sistemi olarak aynı görevleri (örneğin bağımlılık kontrolü, disk alanı kontrolü gibi) yapar.

Paket yönetimi için **dpkg** temel komuttur. Komutun genel kullanım şekli:

**dpkg [seçenekler][işlem] [paket dosyası]**

dpkg <[-i|--install]> <Paket\_Dosyası> : İlgili paketin yüklenmesini sağlar.

dpkg <[-l|--list]> [Paket\_İsmi]: Paket ismi veirlmezse tüm paketleri verilirse ilgili isme uyan tüm paketleri listeler.

dpkg <[-L|--listfiles]> <Paket\_İsmi>: Verilen paket ile sisteme yüklenen dosya ve dizinlerin listesini döker

dpkg <[-S|--search]> <Dosya\_İsmi>: Verilen dosya ismini sisteme yükleyen paketleri listeler.

root@ubuntu:~# dpkg -S /etc/dpkg/dpkg.cfg

dpkg: /etc/dpkg/dpkg.cfg

dpkg <[-s|--status]> <Paket\_İsmi> : Verilen paket ismine ait paket detaylarını ekrana yazar.

dpkg <[-r|--remove]|[-P|--purge]> <Paket\_İsmi>: İsmi verilen paketin silinmesini sağlar. Purge seçeneği kullanılır ise yapılandırma dosyaları da silinir.

dpkg <-x|--extract> <Paket\_Yolu> <Açilacak\_Dizin\_Yolu>: Paket\_Yolu ile verilen paketin içeriğinin Açilacak\_Dizin\_Yolu dizini altına açılması sağlanır.

root@ubuntu:~# dpkg -x gcc-4.4-base\_4.4.3-4ubuntu5\_amd64.deb /tmp/acma/

Daha fazla bilgi için man dpkg.

**root@ubuntu:~# dpkg -l linux-\***

Desired=Unknown/Install/Remove/Purge/Hold

| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend

|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)

||/ Name Version Description

+++-====================================-

un linux-doc-2.6.32 <none> (no description available)

ii linux-firmware 1.34.1 Firmware for Linux kernel drivers

ii linux-generic-pae 2.6.32.24.25 Complete Generic Linux kernel

un linux-headers <none> (no description available)

un linux-headers-2.6 <none> (no description available)

ii linux-headers-2.6.32-24 2.6.32-24.43 Header files related to Linux kernel version 2.6.32

ii linux-headers-2.6.32-24-generic-pae 2.6.32-24.43 Linux kernel headers for version 2.6.32 on x86

ii linux-headers-generic-pae 2.6.32.24.25 Generic Linux kernel headers

un linux-image <none> (no description available)

un linux-image-2.6 <none> (no description available)

ii linux-image-2.6.32-24-generic-pae 2.6.32-24.43 Linux kernel image for version 2.6.32 on x86

ii linux-image-generic-pae 2.6.32.24.25 Generic Linux kernel image

un linux-initramfs-tool <none> (no description available)

un linux-kernel-log-daemon <none> (no description available)

un linux-restricted-common <none> (no description available)

un linux-source-2.6.32 <none> (no description available)

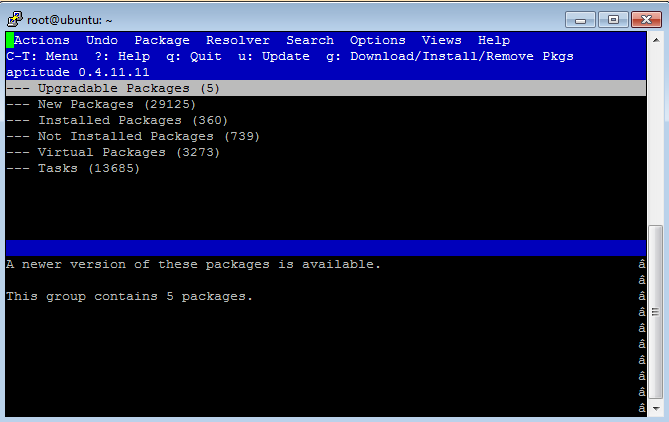
un linux-tools <none> (no description available)

**aptitude Komutu**

dpkg komutu ile yapılan işlemleri konsol ekranında metin bazlı arayüz üzerinden kolayca yapmaya yarar.

Daha fazla bilgi için man aptitude

Grafik arayüzden çıkmak için **q** tuşuna basılır.



**APT (Advanced Packaging Tool)**

APT yapısı RPM mimarisinde YUM aracının rolünü üstlenmektedir. İnternet üzerinden paketlerin kurulması veya güncellenmesi işlemlerinin kolayca yapılmasını sağlar. YUM yapısından farklı olarak belirli görevlere ait farklı komutları bulunmaktadır.

APT yapısının yapılandırma dosyaları **/etc/apt/** dizini altında bulunur. Paketleri çekeceği depo bilgilerini ise **/etc/apt/sources.list** dosyasından okur.

**apt-get Komutu**

Paket yükleme üzerine işlemler apt-get komutu ile yapılır.

Komutun genel kullanım şekli:

**apt-get [<Seçenekler>] <İşlem> [<Paket\_İsmi>]**

**apt-get update** : /etc/apt/sources.list dosyasında tanımlı depolardaki bilgilere bakarak APT paket önbelleğini günceller. Kurulum yapılmadan önce bu işlemin yapılarak güncel bilgilerle işlem yapılmasında fayda vardır.

**apt-get upgrade** : Tüm sistemin güncellenmesi sağlanır.

**apt-get dist-upgrade** : Tüm sistemde güncelleme yapara fakat eğer bir paket sistemde çakışma yaratacaksa ilgili paketi güncellemez.

**apt-get install Paket1 …** : İsmi verilen paketleri sisteme yükler.

**apt-get remove Paket\_İsmi** : İsmi verilen paketi sistemden kaldırır fakat yapılandırma dosyalarını silmez.

**apt-get purge Paket\_İsmi :** İsmi verilen paketi yapılandırma dosyaları ile birlikte sistemden kaldırır.

**apt-get <[clean|autoclean]>** : Paket yükleme önbelleğindeki gereksiz dosyaları silerek disk alanında yer kazanılmasını sağlar. Autoclean kullanılırsa sadece tekrar yüklenme ihtimali olmayan paketlere ait verile silinir.

apt-get komutuna seçenek olarak **-d (--download)** verilirse paketi yüklemek yerine sadece depodan yerel sisteme indirir.

apt-get install komutuna **--no-upgrade** seçeneği verilirse bağımlılıkla gelen yeni bir paket varsa yüklememesini sağlar.

Daha fazla bilgi için **man apt-get**

**apt-cache Komutu**

**apt-cache** arama ve raporlama gibi işlemlerde kullanılır. Veri çıkarırken sistemdeki ve depodaki verilere bakar.

**apt-cache showpkg Paket\_İsmi** : Verilen isimdeki pakete ait bilgileri ekrana döker.

**apt-cache stats :** Sisteme yüklenen paketler hakkında istatistiki bilgi sunar.

**root@ubuntu:~# apt-cache stats**

Total package names: 39457 (1,578k)

Normal packages: 29933

Pure virtual packages: 526

Single virtual packages: 2747

Mixed virtual packages: 296

Missing: 5955

Total distinct versions: 32528 (1,822k)

Total distinct descriptions: 32528 (781k)

Total dependencies: 211488 (5,922k)

Total ver/file relations: 33425 (535k)

Total Desc/File relations: 32528 (520k)

Total Provides mappings: 5833 (117k)

Total globbed strings: 149 (1,936)

Total dependency version space: 1,057k

Total slack space: 104k

Total space accounted for: 10.1M

**apt-cache search Arama\_Metni** : Verilen Arama\_Metnini veri tabanı ve depo önebelleklerindeki tüm paket ismi ve açıklamalarında arar.

**apt-cache depends Paket\_İsmi** : Verilen paketin tüm bağımlılıklarını ekrana yazar.

**apt-cache rdepends Paket\_İsmi** : Verilen pakete bağımlı olan tüm paketlerin listesini verir.

Daha fazla bilgi için **man apt-cache**

**apt-file Komutu**

Paketlerdeki dosyalara ait listeleme işlemlerinde kullanılır.

**apt-file <[search|find]>** **Arama\_Paterni** : Verilen arama paternine uyan dosya isimlerini getiren paketlerin isimlerini listeler.

**apt-file <[list|show]> Arama\_Paterni** : Arama paternine uyan paket isimlerinin içeriğindeki dosyaları ekrana listeler.

Daha fazla bilgi için **man apt-file**

**UYGULAMA 1**

1. Yapmak istediğiniz DHCP yapılandırmasına ilişkin internette yaptığınız aramada dhcpd.conf dosyasını düzenlemeniz gerektiğini okudunuz, fakat sisteme hangi paketi kurmanız gerektiğini bilmiyorsunuz. yum komutunu kullanarak hangi paketi kurmanız gerektiği bilgisini sorgulayınız.
2. yum komutunu kullanarak, sistemde, kurulu olmayan ve kurulu RPM paketi bulunmayan xpdf paketi hakkında kurmadan bilgi alınız.
3. Yaşadığınız disk alanı doluluk sorununu - kısa süreli - gidermek için yerel sistemde yum’un barındırdığı paketleri yum komutu ile siliniz(Bu paketler /var/cache/yum dizini altında tutulmaktadır).
4. Sistemde kurulu paketlere ilişkin güncelleme olup olmadığını kontrol ediniz.
5. Sistemde kurulu olan perl paketini güncelleyiniz.
6. Sistemde yüklü olmayan switchdesk RPM paketini yükleyiniz.

**UYGULAMA CEVAP 1**

**1 -** [root@ila ~]# yum provides '\*/dhcpd.conf'

Loaded plugins: presto, refresh-packagekit

12:dhcp-4.1.1-15.fc13.x86\_64 : Dynamic host configuration protocol software

Repo : fedora

Matched from:

Filename : /etc/dhcp/dhcpd.conf

ltsp-server-5.1.95-1.fc13.x86\_64 : LTSP server

Repo : fedora

Matched from:

Filename : /etc/ltsp/dhcpd.conf

sblim-cmpi-dhcp-test-0.5.5-1.fc13.x86\_64 : SBLIM WBEM-SMT DHCP - Testcase Files

Repo : fedora

Matched from:

Filename : /usr/share/sblim-testsuite/dhcpd.conf

logwatch-7.3.6-50.fc13.noarch : A log file analysis program

Repo : fedora

Matched from:

Filename : /usr/share/logwatch/default.conf/services/dhcpd.conf

12:dhcp-4.1.2-4.ESV.R2.fc13.x86\_64 : Dynamic host configuration protocol software

Repo : updates

Matched from:

Filename : /etc/dhcp/dhcpd.conf

logwatch-7.3.6-55.fc13.noarch : A log file analysis program

Repo : updates

Matched from:

Filename : /usr/share/logwatch/default.conf/services/dhcpd.conf

**2 -** [root@ila ~]#

[root@ila ~]# yum info xpdf

Loaded plugins: presto, refresh-packagekit

Available Packages

Name : xpdf

Arch : x86\_64

Epoch : 1

Version : 3.02

Release : 16.fc13

Size : 801 k

Repo : updates

Summary : A PDF file viewer for the X Window System

URL : http://www.foolabs.com/xpdf/

License : GPLv2

Description: Xpdf is an X Window System based viewer for Portable Document Format

: (PDF) files. Xpdf is a small and efficient program which uses

: standard X fonts.

**3 -** [root@ila ~]#

[root@ila ~]# yum clean all

Loaded plugins: presto, refresh-packagekit

Cleaning up Everything

0 delta-package files removed, by presto

[root@ila ~]#

**4** [root@ila ~]# yum check-update

...

[root@ila ~]#

**5** [root@ila ~]# yum update perl

...

[root@ila ~]#

**6** [root@ila ~]# yum install switchdesk

...

[root@ila ~]#

**UYGULAMA 2**

1. Yapmak istediğiniz DHCP yapılandırmasına ilişkin internette yaptığınız aramada dhcpd.conf isimli dosyayı düzenlemeniz gerektiğini buldunuz, fakat sisteme hangi paketi kurmanız gerektiğini bilmiyorsunuz. apt-file komutunu kullanarak hangi paketi kurmanız gerektiği bilgisini sorgulayınız.
2. apt-cache komutunu kullanarak, sistemde, kurulu olmayan ve kurulu paketi bulunmayan xpdf paketi hakkında kurmadan bilgi alınız.
3. Sistemde yüklü olmayan vi improved metin editörünü kullanabilmek için vim paketini yükleyin.
4. Sisteminizdeki tüm paketleri güncelleyin.

**CEVAP 2**

**1-**

root@ila:~# apt-get install apt-file

...

root@ila:~# apt-file update

...

root@ila:~# apt-file search dhcpd.conf

bcfg2-server: /usr/share/bcfg2/Hostbase/repo/dhcpd.conf.head

debian-installer: /usr/share/doc/debian-installer/talks/fosdem07/fosdem1/etc/dhcp3/dhcpd.conf

dhcp3-server: /etc/dhcp3/dhcpd.conf

dhcp3-server: /usr/share/doc/dhcp3-server/examples/dhcpd.conf

dhcp3-server: /usr/share/man/man5/dhcpd.conf.5.gz

di-netboot-assistant: /usr/share/doc/di-netboot-assistant/examples/dhcpd.conf.multiarch.gz

di-netboot-assistant: /usr/share/doc/di-netboot-assistant/examples/dhcpd.conf.simple.gz

di-netboot-assistant: /usr/share/doc/di-netboot-assistant/examples/dhcpd.conf.upstream

ebox-dhcp: /usr/share/ebox/stubs/dhcp/dhcpd.conf.mas

elilo: /usr/share/doc/elilo/examples/netboot/dhcpd.conf

etherboot: /usr/share/doc/etherboot/contrib/dhcpdconfeg/dhcpd.conf

etherboot: /usr/share/doc/etherboot/contrib/initrd/dhcpd.conf.etherboot.include.gz

logwatch: /usr/share/logwatch/default.conf/services/dhcpd.conf

ltsp-server: /usr/share/doc/ltsp-server/examples/dhcpd.conf

ltsp-server-standalone: /etc/ltsp/dhcpd.conf

manpages-ja: /usr/share/man/ja/man5/dhcpd.conf.5.gz

mythbuntu-diskless-server-standalone: /etc/ltsp/dhcpd.conf

picalib: /etc/picalib/DHCP/dhcpd.conf

udhcpd: /usr/share/doc/udhcpd/examples/udhcpd.conf

udhcpd: /usr/share/man/man5/udhcpd.conf.5.gz

root@ila:~#

**2-**

root@ila:~# apt-cache show xpdf

Package: xpdf

Priority: optional

Section: universe/text

Installed-Size: 36

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Michael Gilbert <michael.s.gilbert@gmail.com>

Architecture: all

Version: 3.02-2ubuntu1.1

Replaces: xpdf-i (<= 0.90-8)

Depends: xpdf-reader, xpdf-utils, xpdf-common

Conflicts: xpdf-i (<= 0.90-8)

Filename: pool/universe/x/xpdf/xpdf\_3.02-2ubuntu1.1\_all.deb

Size: 1338

MD5sum: 4267de6b45d3e8c1d2ca83a92cc8f03f

SHA1: ea0f75d5ff1bbd31ed74a1e24ca79fd18688bb87

SHA256: 606a430f38caa535251dda6c18fa3a0a0cbcd1845780e0d14027bfce018be814

Description: Portable Document Format (PDF) suite

xpdf is a suite of tools for Portable Document Format (PDF) files. (These are

sometimes called 'Acrobat' files after the name of Adobe's PDF software.)

.

The tools include xpdf, a PDF viewer (in the package xpdf-reader),

and PDF converters (including to/from PostScript) (in the package

xpdf-utils).

.

This package is intended for compatibility with previous versions of

this package only. You can safely remove it from your system.

Homepage: http://www.foolabs.com/xpdf/

Bugs: https://bugs.launchpad.net/ubuntu/+filebug

Origin: Ubuntu

Package: xpdf

Priority: optional

Section: universe/text

Installed-Size: 36

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Michael Gilbert <michael.s.gilbert@gmail.com>

Architecture: all

Version: 3.02-2ubuntu1

Replaces: xpdf-i (<= 0.90-8)

Depends: xpdf-reader, xpdf-utils, xpdf-common

Conflicts: xpdf-i (<= 0.90-8)

Filename: pool/universe/x/xpdf/xpdf\_3.02-2ubuntu1\_all.deb

Size: 1340

MD5sum: 8b8e59a44d97c712dfb325cbe864700a

SHA1: ba92a246352e406851b94f28839132bba9d2bab4

SHA256: 94c10cd1a9cabedea46dae36568e9c3f3304306fd4e84cc6931787cc57b1ecb0

Description: Portable Document Format (PDF) suite

xpdf is a suite of tools for Portable Document Format (PDF) files. (These are

sometimes called 'Acrobat' files after the name of Adobe's PDF software.)

.

The tools include xpdf, a PDF viewer (in the package xpdf-reader),

and PDF converters (including to/from PostScript) (in the package

xpdf-utils).

.

This package is intended for compatibility with previous versions of

this package only. You can safely remove it from your system.

Homepage: http://www.foolabs.com/xpdf/

Bugs: https://bugs.launchpad.net/ubuntu/+filebug

Origin: Ubuntu

root@ila:~#

**3-**

root@ila:~# apt-get install vim

...

root@ila:~#

**4-**

root@ila:~# apt-get upgrade

...

root@ila:~#

**KABUK**

**Kabuk (Shell) Nedir ?**

NOT:

İŞletim sistemi ile kullanıcı arasında sağlanan arayüz.

Komut satırıda arayüzdür.Sadece kullanıcılar için değil, uygulamalar içinde arayüz sağlar.Lİnus torvals ın kullandığı ilk uygulamadır.

Çekirdeğin üzerinde kabuk koy dağıtım olsun.Torvals shell üzerinden donanımla etkileşime geçmiştir.

Bash(Bourne again shell): farklı kabukların faklı yetenekleri vardır.Farklı amaçlara odaklanmışlardır.

-kabukta bir programdır.

-sh yaz çalıştır.Kabuk çalışır.[root@ila100~] şekilde kısıma prompt denir.

-exit yazıp çıkılabilir.

cat /etc/shells ;makinemizde yülkü olan kabukları listeler.

dash deyip farklı bir prompt a geçilebilir.

/etc/passwd den kullanıcıya kabuk atanabilir.

-çevresel değişkenler tamamen kabukla alakalıdır.O anki oturumumuza dair bilgileri tutar.windows tada vardır. %windows% örnegin windods un yüklü olduğu yeri gösterir.

-echo ne yazarsak bize gerir verir.Lİnuxta herşey küçük programlardan oluşur.

-degişken tanımlama: degisken=deger ne sagında ne solunda boşluk olmamalıdır.çünkü boşluk linuxta paramtre ayracıdır. Bir değişkenin değerini ekrana yazdırmak istiyoruz.

echo $degisken ; echonun degiskenden haberi bile yok. KOmutu çalıştırmadan önce kabuk devreye girer ve degeri getirir.Doları görünce kabuk hemen kendi değişken bellegine bakar ve degeri getirir.

-degisken=Merhaba diyerek değişkene yeni bir değer atayabiliriz

DEgişkenler 2 ye ayrılır:

1.Çevresel değişken: Kabuk açılışında otomatik set edilir.set komutu ile bunları listeyelebiliriz.

2.o anki değişken: Çevresel değişkenler büyük harfle başlar.DISPLAY bizde çalışacak X istemcisinin üretiiği verinin hangi IP ye gönderileceğini belirler.echo $DISPLAY diyerek bakabiliriz.

$HOME ;uygulmayı çalıştıracak kişinin ev dizinini verir.

$USER ;uygulamayı çalıştıran kim

hostname= echo $HOSTNAME

--kabugun kendi ayarları:

.historyfile echo $HISTFİLE

echo $HISTSIZE kaç komut geri dogru tutulacak.

echo $SHELL şuan kllanılan kabuk hangisi

echo $PWD pwd komutu gidip bu değişkeni okur.

Lİnux ta program çalıştımanın yolu tam yolunu yazmaktır. /bin/bash dersek çalışır.bash yazarsakta bunun sebebi kabuktur.

echo $PATH değişkenidir. KAbuk burda yazılan dizinlerin altındaki execute izni olan dosyaları arar. Bulur çalıştırır. Binary dosyamızın yolu path de değilse istedğimiz kadar kısa yol yzlaım çalışmaz.

-Bir değişkene değer atanırken başıdnaki $ işareti kalkar.

PATH=eskicikti:yeni yol diyerek ekleyebiliriz bu yöntem dogru değil.

PATH=$PATH:/tmp/bin şeklinde path genişletilebilir.

1.Local değişken 2.Global değişken

Bir kabuktayken başka bir kabuk daha açtık.

$PATH global bir değişkendir.

echo $isim=hakan dedik başak bir kabuk açtığımzda echo $isim yazarsak hakan gelmez çünkü localdir.

export komutu kullanarak bir değişken global seviyeye çıkarılabilir.

export isim dememiz gerekir.

bir değişken hem set edilip hemde globale çıkarılabilir.

export isi=hakanveli

export PATH=$PATH:/tmp/bin

Gnome terminali kapatıp yeniden açtığımızda ayarlarımız gider kalıcı olmasını istiyorsak .bashrc ye yazmamız gerekir.BU sadece kendi profilimizle alakalıdır.

/etc/bashrc ise bütün kullanıcılar için ayarları set eder

echo $PS1 ;prompt screen in patternini verir.

PS1=SERTAC yazarsak prompt SERTAC# eşklinde çıkar.

-linuxta komuttan dönen değerler 0-255 arasında değer alır.

-echo $? ;programa atanan değerleri verir. 0 dönerse program başarılı sonlanmıştır.

Bir önceki komuttan dönen sayısal değeri ? ile öğreniriz.

bir değişkeni başka bir değişkene atama

deg=$PATH

echo $deg ;kabuk değişkenin degeri açıp echo ya öyle gönderir.

KISAYOLLAR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tab:komutu tamamlar,Dosya dizin agacında gezerken kullanılır.

-ctrl+alt+f1.. f2

-ctrl+r(reverse index search) :komutsatırında arama, ctrl+l ile önceki komutlara geriye doğru gideriz.

-ctrl+c 2 numaralı kesme sinyali gönderir.

-ctrl+d girdi bekleyen programa artık sana girdi vermiyecem der.

örn :vim - ile komut satırından dosyaya girdi yapabiliriz.ctrl+d ile de girdiyi sonlandırabilirz.

-siyah ekranda ctrl+s ;ekranı dondurur.Klavyeden girdi yapmamıza ragmen ekrana basmaz..ctrl+q diyerek ekranı açabiliriz.Grafik ekrandaki terminalde yemez.

-ctrl+a imleci satır başına alır.

-ctrl+e imleci satır sonuna alır.

-ctrl+k imlecin oldugu yerden satır sonuna kadar siler.

-ctrl+u imlecin oldugu yerden satır başına kadar siler.

Kabuk için özel karakterler\*\*\*\*\*\*\*\*\*\*\*

$ :degişkenler için kullanılır.

boşluk parametre ayracaıdır.

-ls -l /dev/sd\* sd ile başlayan dosyaları listeler.

\* 0 veya daha fazla karakter demek.

? 1 karakter demek.

ör:ls -l /dev/sd? kabuk devreye girip /dev/sda ile oluşabilecekleri yanyana alıp ls -l ye parametre olarak veriyor.ls -l nin ? den haberi bile yoktur.

ls -l /dev/sd? :1 karakter

ls -l /dev/sd?? :2 karakter

ls -l /dev/sd[adkg]1 : a,d,k,g karakterlerinden herhangi birini barındıran ve 1 ile biten.

ls -l /dev/sd[a-ckg]1 :a dan c ye ve k,g

ls -l /dev/sd?[a-zA-Z1-9] :alfanumerik serisi

echo Hakan

echo hakan{veli,egitmen} :hakanveli,hakanegitmen yazar.

örn:yapılandırma dosyalrından birini değiştirip yedeğini alacaz:(yedekleme aynı dizine yapılır.)

echo /etc/yum.repos.d/fedora.repo{,.ydk}

KOpyalamak için:

cp /etc/yum.repos.d/fedora.repo{,.ydk}

peki fedora.repo{,.ydk} adında dosya oluşturmak istersek;

touch fedora.repo\{,.ydk\} ; bunu tırnak işareti kullanarakata yapabiliriz.

-çift tırnakla tek tırnak ın farkı tek tırnak her türlü özel anlamı öldürür affetmez. echo "$HOSTNAME" ile echo '$HOSTNAME' farkına bak.

-çift tırnak hala değişken isimlendirmesi yapmaya devam eder. echo '$HOST'"'"NAME' tektırnak blogun da " hariç hiçbirşey yemez .

örn: ls -l /dev/sd?[!0-5] :0 ile 5 arasında olmaya demek.

Kabugun ! için olumsuzluk anlamı vardır find ve exec ile birlikte \! şekilden kullandık .bu demekki ! find paramtre olarak gidiyor. Kabuk bu senin ! in degil deriz.; 2 komutu birbirinden ayırır.

-alias bash kabugu ile alakalıdır.

alias komutu makinadaki aliasları listeler.

alias zaman=date dersek zaman dediğimizde date komutunu çalıştır demektir.

alias aslında binary değildir.Kabuk sadece binaryleri çalıştırırdı.BU kabuk içinde programlanmıştır.örn:echo linuxta en çok kullanılan komuttur.her seferrinde gidip diskten getirmek yerine kabuga gömülmüştür.

-man bashbuiltins le kabuga gömülü komutları görebiliriz.

-unalias zaman dersek alias kalkar.

alias ı bashrc ye yazarsak kalıcaı hale gelir.

----------Bilgiyar program çalıştırıdığında 3 işaretçi atanır.

1-stdin=0 (0:işaret no) ->terminal

2-stdout=1 ->terminal

3.stderr=2 ->terminal (programın standart olmayan çıktısı)

-w komutu ile o anki kullanıcıları ve nereler de login oldukalrını görebiliriz.

ll /dev/pts/0 den bakabiliriz.

-ls anaconda-ks.cfg :burda komut paramtre almışdır.stdin i sallamaz.

cat<anaconda-ks.cfg : sağdaki komutu kabuk bellegine aldı.cat çalıştırıldığında std i besledi. her komut standat girişinde gelen veriyi sallamaz örn ls komutu.

cat<<son :son yazan bir satır gelene kadar gelen veriyi bufferlar.bunu kesmenin yolu ctrl+ d ye basmak ya da son yazmaktır.

-----------output-------

date komutunu çıktısı normalde çalıştırdığımızda stdout a yazar.derdimiz bunu dosyaya yazmak.

prog>dosya

date>cikti

date>cikti : hep üstüne yazar.

eger;

prog>>cikti

date>>cikti

date>>cikti :appendable yazar hep sona ekler.

bu aslında bir dosyanın içini boşlatma teknigidir.örn :1gb lık bir dosyanın içini boşaltmak istiyoruz.(silip yeniden oluşturmak INODE unu değiştirir.Bunuda istemiyoruz.)

echo>cikti dersek dosyanın içini boşaltır. tek ok bu yğzden tehlikelidir.

ls

anaconda\_ks.cfg>cikti

-cat cikti.

-----------------strderr(2)-----

-prog 2>dosya

-prog 2>>dosya

ls anaconda\_ks.cfg olmayandosya 2>cikti

-ls in çıktısını yok etemek istiyoruz./dev/null :kara delige yollarız.

ls>/dev/null

-ls anakonda\_ks.cfg olamayandosya 2>ckti>cikti : hem hatayı hemde çıktıyı dosyaya yazmak istiyoruz. bu şekilde bozuk yazar.

-ls anaconda\_ks.cfg olmayandosya 2>>cikti>>cikti .BU işi yapar yada merge(&) kullanabilirz

ls anaconda\_ks.cfg olmayandosy &>cikti

---------------------bir programın statndart çıktısını alıp başka bir programın standart girdisine şutlamak----------------------------------

program1\_stdout | program2\_stdin

- ls |cat

- ls |vim -

- ls |wc(word count)

örn:cat anaconda\_ks.cfg |wc :çıktısı 129(satır) 251(kelime) 2666(karakter sayisi)

örn:rpm -qa |wc -l :makinemizde yüklü olan rpm paktelerinin sayısını verir.

-rpm -qa kullanarak bütün paketlerin çıktısını hem ekrana hemde dosyaya eklemek istiyoruz. BUrda tee programı devreye girer.

rpm -qa |tee rpmler :rpmleri hem ekrana hem dosyaya yazar.tekrar çalıştırırsak dosyanın üzerine yazar. ekleme yapmasını istiyorsak -a parametresini veririz.

örn: ls anakonda\_ks.cfg olmayan

no such....

anaconda\_ks.cfg şekliden çıktı verir.

la anakoda\_ks.cfg |wc -l dersek 1 verir ama 2 vermesini istiyoruz.

ls anaconda\_ks.cfg olmayan 2&1 |wc -l

not:echo stdin i dinlemez

komut | echo yaz : komuttan geleni dinlemez ekrana yaz yazar.

passwd nin izinlerine bakmak istiyoruz.

-which passwd

ls -l /usr/bin/passwd

yada;

ls -l `which passwd` (ters tırnak bu;bunun içinde yazanı kabuk gidip çalıştırır.stdout unu gelen programa paratme olarak gönderir.)

bu durumda uygun olmaya komut çıtısı olursa çakılırız. mesela which ls için çakılırız çünkü çıktı tek satır değildir.

\*\*uname -r :çalışan çekirdek sürümünü listeler.Bir değişkene atamak istersek.

cekirdek=$(uname -r)

echo cekirdek şeklinde kullanırız.

-liste=`rpm -qa`;cat liste;echo $liste>ciktis : bu şekilde tek bir satır halinde bitişik olarak çıktı verir. satır satır istiyorsak;

-liste=`rpm -qa` ;echo "$liste" :""lar multiline girdiyi multiline çıktı şeklinde verir.

-ls -l `find /root/` : liste büyürse arguman too long hatası alınır.(ls -l `find \` dersek bu hatayı alabiliriz.)

-find \ | xargs ls -l :grup grup yanındaki komuta paramtre olarak verilir.standart girdiden aldıgı veriyi yandaki komuta grup grup gönderir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FİLTRE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

grep çıktı süzme komutu: dogrudan dosyalara çalışabilir. STandart girdi ile bir komutun çıktısı üzerinde de çalışabilir.

-grep net anakonda\_ks.cfg : dosya üzerinde

-grep net<.....:standart girdi ile

-grep --color anahtarı ile çalışır çünkü bir aliastır.

-grep aramalaraı şöyle yapar.(büyük küçük harf duyarlıdır.

-grep NETWORK /usr/..../|wc -l NETWORK ile network yazmamız arasında fark var.

-grep -i ile büyük küçük harf duyarlılıgı kalkar.

-grep network$ /usru/share/doc/initscript/syslog.txt :(satır sonu network olanları arar.)

-find ile arama yaptıgımızda dataya bulaşmıyordu ama ama grrep bulaşır.

-grep -r NETWORK /usr/share/docs/initscripts-9.30.1/ :içerisinde NETWORK satırı geçenleri getir.

-grep -lri NETWORK /usr/..../

-rpm -qa |grep -i^s|wc -l : makinemizdeki s ile başlayan rpm paketlerinin sayısı.

-grep -E = egrep aynı komuttur.

-egrep 'error/warning/critic' /var/log/messages/ :içerisinde tırnak içerisinde geçen ifade olanları loglar.

-grep -F network$ /usr/share/.../sysconfig.txt :-F ile ne yazdıysam onu ara demektir regular ifadeyi değil network$ ı aramış oluruz.

-grep -Ev '#$|^#'v/etc/yum.conf yum.conf daki boş satırları kaldırdık.

-cat /etc/passwd| cut -d : -f 1,3 1çve 3. sutunları getirir. peki 3. ve 1. sutunların gelmesini istersek burda cut biter.awk kullanılır.

-cat /etc/passwd |awk -F:'{print $1 $3}' biraz düzenlersek.

-cat /etc/passwd|awk -F:'{print $1"-->"$3}'

-cat /etc/passwd |awk -F : '$3>500{print $1"-->"$3}' (şeklinde şart ifadeside verebiliriz.)

-ls -l |awk '{print $1" "$3}' (boşluk, tab ,field seperator olunca fark edemez. hertürlü düzenler.)

-grep net anaconda\_ks.cfg dosyasındaki net kelimelerini değiştirmek istiyoruz.

-sed s/net/SERTAC anakonda\_ks.cfg |grep SERTAC s:subsution bul degiştir. bu şekilde düzenler gösterir ama değiştirmez değiştirmek istersek;

-sed -i s/net/SERTAC anakonda\_ks.cfg | grep SERTAC

-çıktıyı alfabetik sıralamak istersek:

cat /etc/passwd |awk -F : '$3>500 {print $1"-->"$3}' |sort (alfabetik olarak sıralar.sayısal sıralamak için sort -n kullanılır.)

\*\*\*\*KABUK PROGRAMLAMA\*\*\*\*\*\*\*\*\*\*\*\*

vim prog1.sh sh uzantısı biz anlayalım diye linux için bir anlamı yoktur.

echo merhaba

date

-deyip kaydettik.

-çalıştırmak için 2 yol vardır.

1.kabuk ismi dosya (bash prog1.sh)

2../prog1.sh aslında burdada kabuga çalıştırmasını söylüyoruz 1. yolla aynıdır. Ama burda permission denied alabilriz.

-kabuk programı demek: komutların alt alta bir dosyaya yazılması demektir.

sh prog1.sh sh komutu ile çalıştırmak.

#:açıklama için kullanılır. ilk satıra özel anlamı olan bir satır eklenir.

#!/bin/bash ilk satır budur. hangi programla çalışacagını söyleriz.burada dosyayı bash ile çalıştır denir.

--#!/usr/bin/bash

#ACİKLAMA

echo merhaba

date

exit 5 ---> programı sonlardırmak için verilen değer buna kabuktan echo $? ile bakabiliriz.

-source prog1.sh diyerekte çalıştırabiliriz.

-. ile bulundugumuz kabukta çalışıtır deriz.bash falan diyerek aslında başka bir kabugu çalıştırmış oluruz.yani programda exit dediğimizde çagırdığımız kabuk kapanır. Eger bulunduğumuz kabugun özelliklerini değiştiren bir program yazarsak. source yada . ile çalıştırmak zorundayız.

. prog1.sh yada

source prog1.sh şeklinde.

-vim prog2.sh

#!/bin/bash

#aciklama

#girilen dosya yolunda dosya olup olmadıgını kontrol etsin

echo merhaba

tarih=`date`

echo suan saat $tarih

echo lutfen dosya yolu girin

read dosyayolu

echo $dosyayolu

ls $dosyayolu 2>/dev/null //komut basarili ise 0 döner hata olunca ise saçma sapan şeyler basmasını istemezsek ls $dosyayolu &>/dev/null şeklinde kullanılır. Bütün standart çıktıları çöpe atar.

if [$? -eq 0]

then

echo dosya var

else

echo dosya yok

fi

--------------------------

not: echo -n ile alt satıra inmeden girdi alır. read i ise -s ile kullanırsak yanına aldığı girdiyi göstermez.

-------------------------

ŞİFRE BETİĞİ

#!/usr/bin/bash

echo merhaba

date

echo -n lutfen sifre girin

read -s sifre

#echo girilen sifre $sifre

-----------------

if["$sifre" =='123'] //metin esitlikleri == ile karşılatırılır.

then

echo sifre dogru

else

echo sifre yanlis

fi

--------yada

if["$sifre"=='123']

then

echo 'sifre dogru'

elif[$USER=='root'] //veya elif[$(id -u)=='0'] uid si 0 olan kullanıcıya bakar bu yöntem daha iyidir.

echo 'root a tum yollar acik'

else

echo sifre yanlis

-----------------

not:[ in adı aslında test tir man test diyerek bakabiliriz linux ta herşey komuttur.[ da bir komuttur.

DONGULER\*\*\*\*\*

program4.sh

#!/usr/bin/bash

#/tmp/dur dosyası oluşana kadar durma.

#bir dongu içerisinde tarih ve saat yaz.

while[! -e "tmp/dur"]

do

date

sleep 1 //1 saniye bekle

done

echo `ls -l /tmp/dur`

---------------------

seq 1 1 10 (ilk 1 baslangic ikinci 1 artım 10 sonlama degeri) yada seq 10 -1 0 gibi.

for x in `seq 1 1 10`;do echo $x done

for ((x=1;x<=10;x++)) ;do echo $x;done //(()) ile aritmetik işlemleri C stilinde yazabiliriz.

-----bc hesapmakinesi konsola bc yazarak açabiliriz.

-for y in `locate /`;do echo $y;done; ==> ls -l `locate /` --arguman listem uzunhatası verebilir.

-ls -1 ile alt alta listeler bu şekilde for a girdi olabailecek şekilde alabiliriz.

&& :mantıksal ve

|| mantıksal veya ; kabuk herzaman true ya varmaya çalışır ekrana basmak için

------------------------

-vim program5.sh (programa çalıştırırken paratme verilerek çalıştırılacak.)

echo $1

echo $2

echo tum parametreler $@

echo parametresayisi $#

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Kabuk (Shell) Nedir?

Kabuk, kendi üzerinden program çalıştırılmasına imkan veren yazılımdır. Sistem veya kullanıcılar, programları kabuk üzerinden çalıştırır. Böylece kabuk, program çalıştırmada kullanıcılara bir arayüz görevini görür.

Kabuk üzerinden program çalıştırılabileceği gibi kabuk üzerine gömülü komutlar da olabilir. Bunlar ayrı birer süreç olarak çağrılmadan kabuk tarafından çalıştırılan komutlardır. (Bkz: man builtins)

Kabuk üzerinden sürekli yapılan işlemleri kısaltmak veya kolaylaştırmak için betikler (scriptler) yazılabilir.

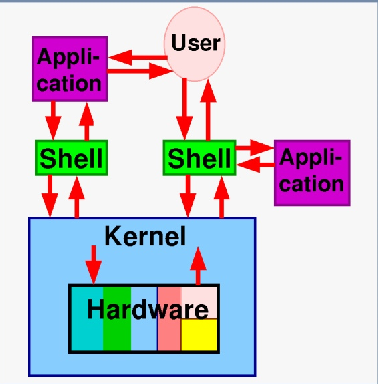
Farklı kullanım tarzlarına veya işletim sistemlerine uygun farklı kabuk yazılımları bulunmaktadır.

Bundan dolayı her kabuğun kendisine gömülü komutları veya betik yazma tarzı da farklıdır. Günümüzde en çok kullanılan kabuk **sh (Bourne Shell)** kabuğunun gelişmiş sürümü olan bash **(Bourne Again Shell**) kabuğudur.

Bazı bilinen kabuk türleri:

* bash (Bourne Again Shell): Linux sistemlerin çoğunda varsayılan kabuk olarak gelir. Öncüsü sh kabuğu ile uyumludur, komut geçmişi tutması, dosya adı ve komut isimlerini tamamlaması en önemli özelliklerindendir.
* sh (BourneShell): bash'in atası olan kabuktur.
* csh (C Shell): Berkeley'de geliştirilmiş, komut yapısı itibariyle C'ye benzer bir yapı kullanan kabuktur.
* ksh (KornShell): UNIX tabanlı sistemlerde popüler bir kabuktur.
* tcsh (Tenex C Shell): csh kabuğunun gelişmiş halidir. Dosya adı ve komut tamamlama özelliği vardır.
* zsh (Z Shell): ksh kabuğuna benzer bir kabuktur.

Linux sistemlerde yüklü olan kabukların listesi /etc/shells dosyasında tutulur. Ayrıca chsh komutuna -l anahtarı verildiğinde bu dosya içeriğini gösterir.



**[root@ila ~]# cat /etc/shells**

/bin/sh

/bin/bash

/sbin/nologin

/bin/tcsh

/bin/csh

/bin/ksh

**[root@ila ~]# chsh -l**

/bin/sh

/bin/bash

/sbin/nologin

/bin/tcsh

/bin/csh

/bin/ksh

**Kullanılan Kabuğun Değiştirilmesi**

Herhangi bir kabuk ile oturum açtıktan sonra farklı bir kabuğa geçilmek istendiğinde tek yapılması gereken diğer kabuğun normal bir program çalıştırır gibi çağrılmasıdır. Bu şekilde çalışma anında değiştirilen kabuk **exit** komutu veya **[CTRL]-[D]** tuş kombinasyonu ile sonlandırıldığında bir önceki kabuğa dönülür.

**[root@ila ~]# sh**

sh-3.2# ksh

# bash

**[root@ila ~]# exit**

exit

# exit

**sh-3.2# exit**

exit

Oturum açılırken kullanılan kabuk **chsh** komutu ile değiştirilir. Bu komuta -s anahtarı verilerek oturum açılırken kullanılacak olan kabuk belirlenebilir. Root yetkisindeki kullanıcı diğer tüm kullanıcıların oturum açılış kabuklarını değiştirebilir.

Chsh komutu /etc/passwd dosyasındaki ilgili kullanıcıya ait oturum açılırken kullanılacak kabuk alanını değiştirir. Dolayısı ile chsh ile yapılan değişiklik bir sonraki oturum açılışında etkin olacaktır.

**[root@ila ~]# chsh** -l

/bin/sh

/bin/bash

/sbin/nologin

/bin/ksh

/bin/tcsh

/bin/csh

/bin/zsh

**[root@ila ~]# chsh -s /bin/csh**

Changing shell for root.

Shell changed.

**Çevresel Değişkenler**

Bash kabuğu üzerinde bir özel çevresel değişkeni ayarlamak ya da yaratmak aşağıdaki şekildedir.

**DEGİSKEN\_ADI=DEGİSKEN\_DEGERI**

[root@ila ~]# DENEME='Bu bir deneme metni'

[root@ila ~]# echo $DENEME

Bu bir deneme metni

Görüldüğü gibi bir değişkene değer atanırken başına $ işareti koyulmaz.

Yukarıda yaratılan DENEME değişkeni bir genel çevresel değişken değil özel çevresel değişkendir. Genel çevresel değişken olması için **export** komutu kullanılmalıdır.

[root@ila ~]# export DENEME

Aynı işlem aşağıdaki şekilde de yapılabilir.

[root@ila ~]# export DENEME='Bu cevresel degisken.'

[root@localhost ~]# echo $DENEME

Bu cevresel degisken.

Bazı önemli çevresel değişkenler ve anlamları:

**DISPLAY**: X görüntü modunda görüntü çıktısının ağ üzerinden nereye yollanacağını belirler. Genellikler değer :0.0 veya <makine\_ismi>:0.0 şeklindedir. Bu görüntünün kendi üzerinde kaldığını belli eder.

**HOME**: Oturum açan kullanıcının ev dizini yolunu belirtir.

**HOSTNAME**: Linux sistemin ağ ismini tutar.

**HISTFILE**: Oturum açan kullanıcının geriye dönük komutlarını hangi dosyada saklayacağını belirler.

**HISTSIZE**: Oturum açan kullanıcının geriye dönük kaç komutunun saklanacağı bilgisini tutar.

**LANG**: Linux sistemin o anki çalışma dilinin ne olduğu bilgisini tutar.

**LD**\_**LIBRARY**\_**PATH**: Bazı programlar sistemde yüklü kütüphanelerin yolunu bu değişkenden öğrenir.

**MAIL**: Oturum açan kullanıcının maillerinin nerede saklanacağı bilgisini tutar.

**SHELL**: Oturum açılan kabuğun tam yol bilgisini tutar.

**PWD**: O anda altında bulunulan dizin yolunu tutar. Böylece bazı programlara tam yol verilmediğinde bulunulan dizini varsayılan dizin olarak kullanmaları sağlanır.

**PATH**: Bu en önemli çevresel değişkendir. Kullanıcının çalıştıracağı komutların sistemde hangi dizinler altında aranacağı belirtilir. Burada tanımlanmamış bir dizin altındaki program tam yolu yazılmadığı sürece çalıştırılamaz. Değişkenin değeri her bir dizin arasına : konarak tanımlanmış bir değerdir. Örn: [root@ila ~]# echo $PATH

/usr/lib/qt-3.3/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/libexec/sdcc:/root/bin

**PS1**: Bash kabuğunun komut beklerkenki görüntüsünü (prompt görüntüsünü) belirler. Burada özel anlamları olan kesme karakterleri ile çok değişik şeyler yapılabilir.

**USER** ya da **USERNAME**: Linux sisteme oturum açan kullanıcı adı bilgisini tutar.

?: En sık kullanılan değişkenlerden biridir. Son çalışan komutun durumunu gösterir. Eğer bu değişken değeri 0 ise program düzgün çalışmış demektir. 0 dışındaki değerler düzgün çalışmadığını gösterir. 0 dışındaki değerlerin anlamları her program için özeldir.

[root@ila ~]# ls anaconda-ks.cfg

anaconda-ks.cfg

[root@ila ~]# echo $?

0

[root@ila ~]# ls olmayandosya

ls: olmayandosya: No such file or directory

[root@ila ~]# echo $?

2

[root@ila ~]# echo $?

0

**Kabuk Kısayolları**

Çevresel değişkenler sistem üzerinde çalışan programların ortak veya özel olarak ihtiyaç duydukları bilgilerin tutulduğu yerlerdir. Programlar belirli çevresel değişkenler üzerinden çalıştıkları sisteme ait bilgileri alabilirler. Örneğin HOSTNAME değişkeni üzerinden sistemde çalışan bir program sistemin adının ne olduğunu bu değişkenden öğrenebilir.

Çevresel değişkenlerin listesine **env** komutu ile ulaşılabilir.

Programların kendilerine ait özel çevresel değişkenleri olabilir. Bu tür değişkenlerde veriler program çalıştığı sürece tutulur ve sadece ilgili program ulaşır. Sistemde çalışmakta olan kabuk da bir program olduğu için çalışma sürecinde kendisine ait çevresel değişkenler tutar. Hem özel hem de genel çevresel değişkenlerin listesine **set** komutu ile ulaşılır. Aynı zamanda set komutu kabuk özelliklerinin de ayarlanmasında da kullanılır lakin bu konumuz dışıdır.

Özel olarak belirli bir çevresel değişkenin değeri okunmak istendiğinde echo komutu kullanılır.

[root@localhost ~]# **echo** $**HOSTNAME**

localhost.localdomain

Burada görüldüğü gibi bir çevresel değişkenin değeri kullanılacağı zaman çevresel değişkenin isminin başına **$** işareti konulur.

Komut satırı üzerinden komut yazmak yorucu veya hatalı yazmaya olanaklı bir yapıdadır. Kabuğun bu zafiyetini gidermek ve hatta bazı durumlarda avantaja bile çeviren birçok kısayol tuşu bulunmaktadır.

[TAB] : Komut veya dosya adı tamamlaması yapar. Komut tamamlaması yaparken PATH çevresel değişkenini dikkate alır. [Ctrl]-[Alt]-[Fonksiyon Tuşları]: Sistemde tanımlı olarak açık bulun an konsollar arasında geçiş yapılmasını dağlar. Red Hat türevi sistemler 7 konsol ile açılırlar ve 7. konsol grafik arayüz için kullanılır.

[Yukarı], [Aşağı], [PageUp], [PageDown]: Komut geçmişinde hareket ederek önceki çalıştırılan komutların geri çağrılmasını sağlar.

**[Ctrl]-[r]:** Komut geçmişi içinde arama yapılmasını sağlar. Ctrl-r basıldıktan sonra aranacak komuta ilişkin harfler yazılıp aranan bulunana kadar Ctrl-r basılmaya devam edilir. Aramayı sonlandırmak için Ctrl-g tuşlarına basılır.

**[Ctrl]-[c**]: Çalışan programı sonlandırır.

**[Ctrl]-[d]:** Girdi girişi bekleyen programa girdi girişinin bittiğini belirtir.

**[Ctrl]-[s]:** Konsolu dondurur Dondurulmuş konsol Ctrl-q tuş kombinasyonu ile çözülür.

**[Ctrl]-[z]:** Çalışmakta olan programı durdurup arka plana gönderir.

**[Ctrl]-[Insert]:** Mouse uygulaması (gpm) çalışıyorsa seçilen veriyi kopyalar.

**[Shift]-[Insert]:** Kopyalanan veriyi yapıştırır.

**[Ctrl]-[a]:** İmleci komut satırının başına getirir.

**[Ctrl]-[e]:** İmleci komut satırının sonuna getirir.

**[Ctrl]-[u]:** İmlecin bulunduğu yerden satır başına kadar siler.

**[Ctrl]-[k]:** İmlecin bulunduğu yerden satır sonuna kadar siler.

**[Ctrl]-[l]:** Ekranı temizler.

**[Ctrl]-[Alt]-[Del]:** Sistemin yeniden başlatılmasını sağlar. Bu durum değiştirilebilir.

**Kabuk Açısından Özel Karakterler (Wildcards)**

Kabuk üzerinde işlem yaparken birden fazla dosya ismini kısa yoldan tanımlama ya da isminin belirli bir kısmı bilinen dosyalar üzerinde işlem yapmak için dosya isimlerinde özel karakterler kullanılabilir.

**\* Karakteri**

0, 1 veya daha fazla karakter yerine geçmektedir.

Örn: \*.conf tabiri sonu .conf ile biten dosyalar, sys\* sys ile başlayan dosyalar anlamına gelmektedir.

[root@ila ~]# ls -l /etc/l\*p.conf

-rw-r--r--1 root root 6647 Mar 9 2005 /etc/ldap.conf

-rw-r--r--1 root root 2954 Dec9 2003 /etc/lftp.conf

**? Karakteri**

1 karakter yerine geçer.

Örn: ???.conf üç harfli .conf uzantılı bir dosyayı temsil eder.

[root@ila ~]# ls

xaz xvz xyz xyzt

[root@ila ~]# ls x?z

xaz xvz xyz

**[ ] Köşeli Parantezler**

Köşeli parantez içine yazılan harfler aranan dosya ismi eşleştirmesinde kullanılır. Örneğin ila[168].txt şeklindeki bir tabir ila1.txt, ila6.txt ve ila8.txt dosya isimlerine eş düşecektir.

Köşeli parantezler içerisinde – karakteri ile harf veya rakam aralığı verilebilir. Örneğin ila[137-9].txt tabiri ila1.txt, ila3.txt, ila7.txt, ila8.txt ve ila9.txt dosya isimleri ile eşleşecektir. Başka bir örnek olarak il[a-c].txt tabiri de ila.txt, ilb.txt ve ilc.txt dosya isimlerine eşleşecektir.

[root@ila ~]# ls /dev/hda[0-5]

/dev/hda1 /dev/hda2 /dev/hda3 /dev/hda4 /dev/hda5

Köşeli parantezler içindeki ilk karakter ^ ya da ! ise değil işlemi uygulanır. Örneğin ila[!3-9].txt tabiri ila1.txt, ila2.txt ve ila3.txt dosyaları ile eşleşir.

[root@ila ~]# ls /dev/hda[^0-5]

/dev/hda6 /dev/hda7 /dev/hda8 /dev/hda9

[root@ila ~]# ls /dev/hda[!0-5]

/dev/hda6 /dev/hda7 /dev/hda8 /dev/hda9

**{ } Küme Parantezleri**

Küme parantezi ile tek bir karakter yerine karakter kümeleri kullanılabilir. Örneğin ila.{txt,conf,pdf} tabiri ila.txt, ila.conf ve ila.pdf tabirlerine dönüşecektir.

[root@ila ~]# ls -l /etc/{sys,krb}\*.conf

-rw-r--r--1 root root 640 Jun 30 2005 /etc/krb5.conf

-rw-r--r--1 root root 2281 Mar 9 2005 /etc/krb.conf

-rw-r--r--1 root root 526 Aug 5 2005 /etc/sysctl.conf

-rw-r--r--1 root root 817 Nov 26 22:35 /etc/syslog.conf

[root@ila ~]# ls /dev/{hd,sd}[a-b]

/dev/hda /dev/hdb /dev/sda /dev/sdb

**\ Ters Bölü veya ‘’ Tek Tırnaklar**

Ters bölü işareti özel karakter olarak kesme işareti tabiri ile anılır. Bu karakter kabuk tarafından özel anlamı olan karakterlerin önüne getirildiğini sağındaki karakterin özel anlamını elinden alır. Aynı işlem eğer özel karakterler tek tırnaklar arasına yazılırsa da gerçekleşir.

[root@ila ~]# ls /dev/hd[a-b]

/dev/hda /dev/hdb

[root@ila ~]# ls /dev/hd\[a-b\]

ls: /dev/hd[a-b]: No such file or directory

[root@ila ~]# ls '/dev/hd[a-b]'

ls: /dev/hd[a-b]: No such file or directory

[root@ila ~]# ls \'/dev/hd[a-b]\'

ls: '/dev/hd[a-b]': No such file or directory

**Örnekler:**

• [dm]\* - Adı d veya m karakteri ile başlayan tüm dosya ve dizinler.

• [a-z]\*[0-9] - Adı a dan z'ye kadar küçük harflerle başlayan ve 0 ile 9 arasında rakamlarla sonlanan tüm dosya ve dizinler.

• \*.conf.{rpmnew,rpmsave} - Adı .conf.rpmnew veya .conf.rpmsave ile sonlanan tüm dosya ve dizinler.

**Komutların Kişiselleştirilmesi (alias)**

Kabuk içine gömülü alias komutu ile bir komut için takma isimler ya da özelletirilmiş komutlar oluşturulabilir. Özelleştirilen komut seçenek ve parametrelerle genişletilebilir, ancak orjinal komutlar kesinlikle silinmez.

[root@ila ~]# uptime

22:15:18 up 59 days, 5:39, 2 users, load average: 0.00, 0.02, 0.07

[root@ila ~]# alias up="uptime"

[root@ila ~]# up

22:15:28 up 59 days, 5:40, 2 users, load average: 0.22, 0.06, 0.09

Ancak komutun adı ile alias tanımlanması durumunda, gerçek komut maskelenmiş olur.

[root@ila ~]# alias rm='rm -i'

[root@ila ~]# alias mv='mv -i'

[root@ila ~]# alias cp='cp -i'

Öntanımlı ya da sonradan tanımlanan takma adlar unalias komutu ile kaldırılabilir.

[root@ila ~]# up

22:21:19 up 59 days, 5:46, 2 users, load average: 0.14, 0.12, 0.09

[root@ila ~]# unalias up

[root@ila ~]# up

-bash: up: command not found

**Program Girdi ve Çıktılarının Yönlendirilmesi**

Linux işletim sisteminde çalıştırılan her komut 3 dosya işaretçisine sahiptir. Bunlar:

* **Standart Giriş (stdin)** - Çalışan programın, dosya gibi ek kaynaklar açmadan veri girdisi almak için kullanacağı kaynağı belirtir. Çoğu durumda programlar veri girdilerini klavye üzerinden alırlar.
* **Standart Çıkış (stdout)** - Metin bazlı çalışan programın çıktılarını göndereceği kaynağı belirtir. Çoğu durumda bu, kaynak programın çalıştığı terminal penceresidir. Grafik arayüz çıktıları Standart Çıkışa gelmez ama grafik arayüze sahip programların varsa metinsel çıktıları Standart Çıkışa gelir.
* **Standart Hata (stderr)** - Çalışan programın hata gibi önem düzeyi yüksek çıktılarını göndereceği diğer bir çıkış işaretçisidir. Çoğu durumda çalışılan terminale hata çıktıları yazılır.

Program girdi ve çıktı işaretçileri dosyalara veya farklı programlara yönlendirilebilir. Böylece komutları birlikte iç içe kullanarak komut satırı etkin ve güçlü bir şekilde kullanılabilir.

Dosya işaretçileri, kullanılırlarken bir numara ile belirtilirler. Standart giriş için 0, standart çıkış için 1, standart hata için de 2 değerleri kullanılır.

**Standart Girişin Yönlendirilmesi**

Programların klavye üzerinden girdi almaları yerine bir metin dosyasından girdi almaları standart girişin yönlendirilmesi ile sağlanır. Diğer bir deyişle bir dosyanın içeriği bir programa yönlendirilmiş olur.

Standart girişin yönlendirilmesi için < operatörü kullanılır. Bu operatörün sol tarafında komut (parametreleri ile birlikte) sağ tarafında ise içeriği yönlendirilecek olan dosya yolu olmalıdır.

**<Standart Girdi Bekleyen Komut> < <Dosya Yolu>**

**wc** komutu standart girişinden aldığı metin içerisindeki veya kendisine parametre olarak verilen dosya içerisindeki satır sayısı, kelime sayısı veya karakter sayısı gibi bilgileri verir.

wc komutunun önemli parametreleri

* -l : Sadece satır sayısını gösterir
* -w : Sadece kelime sayısını gösterir
* -m: Sadece karakter sayısını gösterir.

[root@localhost ~]# cat ila.txt

Merhaba

ILA wc komutu

Egitim metni

[root@localhost ~]# wc < ila.txt

3 6 35

[root@localhost ~]# wc -l < ila.txt

3

Bir dosyanın komuta yönlendirilmesi gibi bir programa sınırılı klavye girdisi de mümkündür. Bunun için << operatörü kullanılır. Örneğin bir\_program << BITIS komutu çalıştırılırsa bir\_program programı standart girdisinden sadece BITIS yazan bir satır görene kadar metni okuyacak ve bu girdi geldiğinde standart girdisinden veri almayı bırakacaktır.

[root@localhost ~]# cat <<BITIS

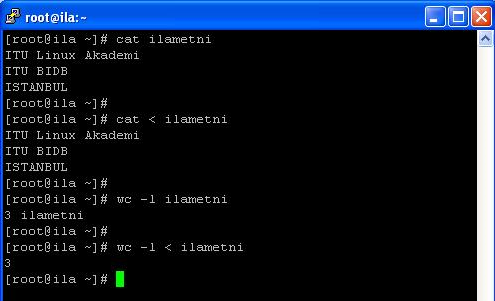
> Merhaba

> Bu metin BITIS yazana kadar okunur

> BITIS

Merhaba

Bu metin BITIS yazana kadar okunur



**Standart Çıkışın Yönlendirilmesi**

Bir programın metin bazlı çıktısı normalde ekrana yazılırken bunun ekran yerine bir dosyaya yazılması > veya >> karakterleri ile yapılır.

<Standart Çıktı Üreten Komut> [>|>>] <Standart Çıktının Yazılacağı Dosya>

> : Programın standart çıktısı sağ tarafında verilen dosya varsa içindeki veriler silinip yeni veri yazılır, dosya yoksa yaratılır.

>> : Programın standart çıktısı sağ tarafta verilen dosya varsa sondan (alttan) yeni veri eklenir, dosya yoksa yaratılır.

[root@ila ~]# ls cikti.txt

ls: cikti.txt: No such file or directory

[root@ila ~]# env > cikti.txt

[root@ila ~]# ls cikti.txt

cikti.txt

[root@ila ~]# cat cikti.txt

HOSTNAME=localhost.localdomain

TERM=xterm

SHELL=/bin/bash

HISTSIZE=1000

...

[root@ila ~]# ls -l cikti.txt > cikti.txt

[root@ila ~]# cat cikti.txt

-rw-r--r-- 1 root root 0 Sep 14 19:19 cikti.txt

[root@ila ~]# env >> cikti.txt

[root@ila ~]# cat cikti.txt

-rw-r--r-- 1 root root 0 Sep 14 19:19 cikti.txt

HOSTNAME=localhost.localdomain

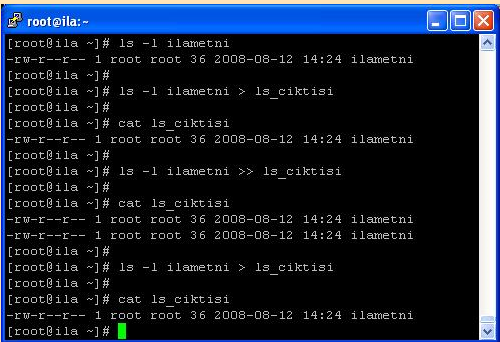
TERM=xterm

SHELL=/bin/bash

HISTSIZE=1000

...

Yukarıda ilk olarak hiçbir şekilde çalıştığımız dizinde cikti.txt dosyası yokken yönlendirme sonucunda bir tane oluşuyor ve içeriğinde env komutunu çıktısı yazılıyor. Ardından ls komutunun çıktısı > işareti ile cikti.txt dosyasın yönlendirildiğinde bir önceki env komutuna ait veriler siliniyor. Ama >> işareti ile yönlendirme yapıldığında ise operatörün sağ tarafındaki dosyaya yeni çıktı sondan ekleniyor.



**Standart Hatanın Yönlendirilmesi**

Programların yüksek önem derecesindeki (örneğin hata) mesajları normalde ekrana yazılır ama bu içerik 2> veya 2>> operatörleri ile ekran yerine bir dosyaya yazılabilir.

**<Standart Hatası Yönlendirilecek Komut> [2>|2>>] <Standart Hatanın Yazılacağı Dosya>**

[root@ila ~]# ls -l cikti.txt olmayandosya

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 976 Sep 14 19:20 cikti.txt

[root@ila ~]# ls -l cikti.txt olmayandosya 2> cikti.txt

-rw-r--r-- 1 root root 0 Sep 14 19:28 cikti.txt

[root@ila ~]# cat cikti.txt

ls: olmayandosya: No such file or directory

[root@ila ~]# ls -l cikti.txt olmayandosya 2>> cikti.txt

-rw-r--r-- 1 root root 44 Sep 14 19:28 cikti.txt

[root@ila ~]# cat cikti.txt

ls: olmayandosya: No such file or directory

ls: olmayandosya: No such file or directory

Eğer hem standart hata hem de standart çıktının aynı anda ekran yerine bir dosyaya yönlendirilmesi istenirse aşağıdaki yöntemlerden biri kullanılabilir.

• Komut &> dosya

• Komut >& dosya

• Komut >dosya 2>&1

• Komut 2> dosya >&2

[root@localhost ~]# ls -l cikti.txt olmayandosya &> cikti.txt

[root@localhost ~]# cat cikti.txt

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 0 Sep 14 19:29 cikti.txt

Bazen programların standart hata çıktıları ve ya standart çıktılarının ekrana gelmesi yani hiç gözükmesi istenmez. Bu gibi durumlarda Linux sistemlerdeki /dev/null aygıt dosyayı kullanılabilir. Bu özel bir aygıt dosyasıdır ve içeriğine ne gönderilirse yok olur. Eğer içeriği okunmak istenirse hiçbir içerik dönmeden dosya sonu bilgisi döner.

[root@localhost ~]# ls -l cikti.txt olmayandosya

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 92 Sep 14 19:29 cikti.txt

[root@localhost ~]# ls -l cikti.txt olmayandosya > /dev/null

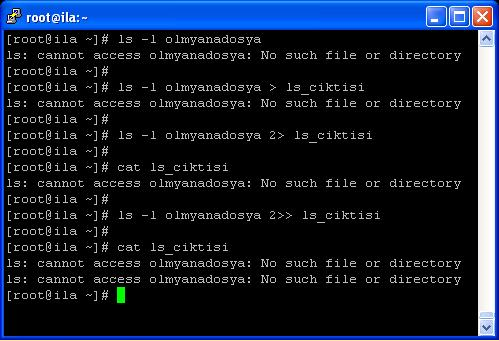
ls: olmayandosya: No such file or directory

[root@localhost ~]# ls -l cikti.txt olmayandosya 2> /dev/null

-rw-r--r-- 1 root root 92 Sep 14 19:29 cikti.txt

[root@localhost ~]# cat /dev/null

[root@localhost ~]#



**Bir Komutun Çıktısının Başka Bir Komuta Girdi Olması**

Komutların çıktıları dosyalar yönlendirilebileceği gibi standart girdi bekleyen başka bir komuta da yönlendirilebilir. Bu işlem | operatörü ile yapılır.

**<Standart Çıktı Üreten Komut> | <Standart Girdisinden Veri Alan Komut> [| ...]**

[root@ila ~]# cat cikti.txt | wc

2 16 92

| operatörü üzerinden genellikle kabuk üzerinden alınan verilerin düzenlenmesi veya filtrelenmesi sağlanır. Bunlara ilişkin komutlar ilerleyen konularda incelenecektir.

Bazı durumlarda bir programın çıktısının hem ekrana yazılması ve aynı zamanda da bir dosyaya yazılması gerekebilir. Bu durumda tee komutu kullanılır. Bu komut standart girdisinden aldığı veriyi hem ekrana hem de parametre olarak aldığı dosyaya yazar. Dosyada daha önceden var olan verileri normalde siler ama -a anahtarı kullanılırsa >> operatörü gib davranıp dosyaya sondan eklemeli olarak ekler.

tee [seçenekler] <Standart Çıktının Yazılacağı Dosya Yolu>

[root@ila ~]# cat cikti.txt | tee tee\_ciktisi.txt

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 0 Sep 14 19:29 cikti.txt

[root@ila ~]# cat tee\_ciktisi.txt

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 0 Sep 14 19:29 cikti.txt

[root@ila ~]# id root | tee -a tee\_ciktisi.txt

uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

[root@ila ~]# cat tee\_ciktisi.txt

ls: olmayandosya: No such file or directory

-rw-r--r-- 1 root root 0 Sep 14 19:29 cikti.txt

uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

**Komut Çıktılarının Parametre Olarak Kullanılması**

Bash kabuğu, komut çıktılarının bir değişkene atanarak, bunların diğer komutlara parametre olarak verilmesi imkanını sağlamaktadır.

Bir komutun çıktısını daha sonra kullanmak üzere bir değişkene atamak için iki yöntem vardır.

• degisken=`komut` ("`" Ters tırnak karakteridir.)

• degisken=$(komut)

[root@ila root]# uname -r

2.4.21-37.ELsmp

[root@ila ~]# kernel=`uname -r`

[root@ila ~]# echo $kernel

2.4.21-37.ELsmp

ya da

[root@ila ~]# kernel=$(uname -r)

[root@ila ~]# echo $kernel

2.4.21-37.ELsmp

Komut çıktıları aynı zamanda bir değişkene atanmadan komut satırından başka komutlara doğrudan parametre olarak verilebilirler. Böylece bir komutun çıktısı, kopyalanmadan veya bir değişkene atanma gibi işlemler yapılmadan doğrudan başka bir komuta parametre olabilir. Yalnız bu yöntemde komut çıktılarının diğer komutlar için anlamlı olması gerekir.

(which komutu, komutların çalıştırılabilir dosyalarının, dosya sistemi içindeki yerini verir.)

[root@ila ~]# which passwd

/usr/bin/passwd

[root@ila ~]# du -sH `which passwd`

21kB /usr/bin/passwd

Örn: Sisteme bağlı sabit disklere ait örnek bazı raporlamalar.

[root@ila ~]# ls -l /dev/{hd,sd}[a-z][0-9]\*

ls: /dev/sd[a-z][0-9]\*: No such file or directory

brw-r----- 1 root disk 3, 1 Sep 2 17:59 /dev/hda1

brw-r----- 1 root disk 3, 2 Sep 2 17:58 /dev/hda2

[root@localhost ~]# ls -l /dev/{hd,sd}[a-z][0-9]\* 2> /dev/null | wc -l

2

[root@localhost ~]# echo Sistemde `ls -l /dev/{hd,sd}[a-z][0-9]\* 2> /dev/null | wc -l` adet HDD aygıt dosyası var.

Sistemde 2 adet HDD aygıt dosyası var.

**Xargs ile Komut Çıktılarının Parametre Olarak Kullanılması**

Komut çıktıları xargs programı sayesinde diğer programlarda parametre olarak kullanılabilir.

**<Çıktı Üreten Komut> | xargs [seçenekler] [Parametre Alacak Komut]**

Burada çıktı üretene komutun çıktıları xargs programının standart girdisinden okunur ve sağ tarafında verilen programı aldığı standart girdiyi parametre olarak vererek çalıştırır. Eğer sağ tarafa bir program yazılmazsa varsayılan olarak echo programını çalıştırır ve böylece aldığı girdiyi sadece ekrana yazar.

Xargs programı genellikle parametre girilmeden kullanılsa da aldığı içeriği farklı yorumlaması üzerine parametreleri vardır. Bunlar hakkında detaylı bilgi için komutun man sayfalarına bakılabilir.

Xargs programı genellikle bir komuta çok fazla parametre verileceği durumlarda işe yarar. Komutların parametre alma sınırı aşıldığında komutlar hata vereceklerdir. Bu tür durumlarda xargs komutu kullanılırsa komutların hata vermeden çok sayıda parametre ile çalışması sağlanabilir.

Eğer sistemdeki tüm .conf uzantılı dosyaların boyutlarına ilişkin bilgi toplamak istersek bunu aşağıdaki yollar ile gerçekleştirebiliriz.

[root@localhost ~]# ls -l `locate conf`

-rw-r--r-- 1 root root 69598 Jul 27 12:57 /boot/config-2.6.18-194.11.1.el5

-rw-r--r-- 1 root root 69271 Jul 27 13:34 /boot/config-2.6.18-194.11.1.el5xen

-rw-r--r-- 1 root root 69597 May 2 11:37 /boot/config-2.6.18-194.3.1.el5

...

[root@localhost ~]# locate .conf | xargs ls -l

-rw------- 1 root root 1767 Aug 16 14:56 /boot/grub/grub.conf

-rw-r--r-- 1 root root 233 Dec 17 2009 /etc/acpi/events/power.conf

-rw-r--r-- 1 root root 236 Dec 17 2009 /etc/acpi/events/video.conf

Örneğin sistemdeki sahipsiz dosyalar ve dizinler silinmek istenirse aşağıdaki yollar kullanılabilir.

# find / -nouser -exec rm –rf {} \;

# rm -rf `find / -nouser`

# find / -nouser | xargs rm –rf

**Temel Çıktı Filtreleme Komutları**

Çıktı filtreleme komutları kullanılarak komut satırı üzerinden anlamlı raporların üretilmesi, sistem analizlerinin hazırlanması veya programlar arasında standart girdi-çıktı ilişkisinin düzgünce kurulması sağlanabilir.

Çoğu çıktı filtreleme programı veriyi standart girdisinden okuyabildiği gibi kendisine dosya ismi parametre olarak da verildiğinde işlem yapabilmektedir.

**8.7.7.1 Çıktı Süzme (grep) Komutu**

En sık kullanılan çıktı süzme komutudur. Bir dosya içerisinde veya standart girdisindeki aranan karakter katarının geçtiği satırları süzer.

**grep [seçenekler …] <istenen patern> [Aranacak dosyalar …]**

Örneğin sistemdeki /bin/bash kabuğunu kullanan kullanıcıları bulmak için aşağıdaki komut bir yol olabilir:

**[root@ila ~]# grep '/bin/bash' /etc/passwd**

root:x:0:0:root:/root:/bin/bash

sil:x:0:500::/home/sil:/bin/bash

mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash

ila:x:44326:107::/home/ila:/bin/bash

veya

**[root@ila ~]# cat /etc/passwd | grep '/bin/bash'**

root:x:0:0:root:/root:/bin/bash

sil:x:0:500::/home/sil:/bin/bash

mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash

ila:x:44326:107::/home/ila:/bin/bash

veya

**[root@ila ~]# grep '/bin/bash' < /etc/passwd**

root:x:0:0:root:/root:/bin/bash

sil:x:0:500::/home/sil:/bin/bash

mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash

ila:x:44326:107::/home/ila:/bin/bash

Grep programı ile birlikte en sık kullanılan seçenekler:

-i : Aranan paterni büyük küçük harf ayrımı gözetmeksizin eşleştirir.

-v : Aranan paternin geçtiği satırları göstermek yerine ters mantıkla geçmediği satırları gösterir.

-A <satır sayısı> : Aranan paternin geçtiği satırla birlikte verilen satır sayısı kadar sonrasını da gösterir.

-B <satır sayısı> : Aranan paternin geçtiği satırla birlikte verilen satır sayısı kadar öncesini de gösterir.

-c : Aranan paternin bulunduğu satırlar yerine aranan metinde toplamda kaç kere geçtiğini gösterir.

-r : Tekrarlı arama yapılmasını sağlar. Bu paramere ile aranacak dosya ismi yerine bir dizin ismi veirlerek aranacak paternin dizin altındaki tüm dosyalarda aranması sağlanabilir.

-l : Aranacak paterni göstermek yerine paternin geçtiği dosya isimlerini gösterir. Genellikle -r seçeneği ile birlikte yapılan tekrarlı aramalarda aranan paternin geçtiği dosya isimlerini tespit etmede kullanılır.

Grep programı patern bilgisi olarak düzenli ifadeler de (regular expressions) alabilir. Normal grep komutu basit düzenli ifadeler alırken egrep (grep -E) gelişmiş düzenli ifadeleri alır. Bununla birlikte basit metin girilecekse fgrep (grep -f) de kullanılabilir.

**[root@ila ~]# cat ila.txt**

Merhaba,

Bu grep programi icin ornek dosyadir.

Bu satirde | (Linux Borusu) var

[root@ila ~]# grep -F '|' ila.txt

Bu satirde | (Linux Borusu) var

**[root@ila ~]# grep -E 'ornek|satir' ila.txt**

Bu grep programi icin ornek dosyadir.

Bu satirde | (Linux Borusu) var

**[root@ila ~]# grep '^Bu' ila.txt**

Bu grep programi icin ornek dosyadir.

Bu satirda | (Linux Borusu) var

**8.7.7.2 Çıktı Düzenleme (awk, cut) Komutları**

Grep programı ile çıktılar üzerinden satır anlamında bir filtreleme yapılabilir ama eğer verilen standart girdi içerisinde belirli kelime sütunlarının gösterilmesi istenirse bu durumda cut veya awk programları kullanılabilir.

**cut [-d <Kelime\_Ayıracı>] -f <Gösterilecek\_Alanlar> [İncelenece\_dosyalar ...]**

Kelime ayıracı girilmemesi durumunda TAB karakteri ayıraç olarak alınır. Detaylı bilgi için man cut.

Örneğin sistemde /bin/bash kabuğu ile oturum açan kullanıcıların kullanıcı adlarını ve UID bilgilerini almak için:

**[root@ila ~]# grep /bin/bash /etc/passwd | cut -d : -f 1,3**

root:0

mysql:27

ila:44326

Cut programı daha basit işlemlerde faydalı iken awk programı ile çıktının yeniden düzenlenmesi veya belirli koşullar yazılması mümkündür.

**awk [-F <Kelime\_Ayıracı>] '[Arama\_Koşulu]{<Yazılış\_Formatı>}'**

Kelime ayıracı -F girilmeze bu durumda awk boşluk veya tab karakterini ayıraç olarak kabul eder.

Örneğin /bin/bash kabuğu ile oturum açan kullanıcılardan UID değeri 500’den büyük olan kullanıcıların UID --> kullanıcı adı --> ev\_dizini gibi formatlı bir çıktısı talep edilirse bu işlem cut komutu ile yapılamaz. Ama awk komutu ile aşağıdaki gibi yapılabilir;

**[root@ila ~]# grep /bin/bash /etc/passwd | awk -F : '$3>500{print $3"-->"$1"-->"$6}'**

**44326-->ila-->/home/ila**

Burada görüldüğü gibi awk komutunda $n sıralı halde Kelime Ayıracına göre elde edilen kelimeye karşılık bir değişken rolünü üstleniyor.

**8.7.7.3 Çıktı Değiştirme (sed) Komutu**

sed komutu standart girdisi üzerinden aldığı veri veya parametre olarak verilen dosya içerisinde kendisine verilen işlem tanımına uygun olarak değişiklik yapar.

**sed [seçenekler] [işlem tanımı] [dosyalar...]**

sed komutundaki bazı sık kullanılan işlem tanımları:

= : Satır numaralarının gözükmesini sağlar.

a\<Metin> : Her bir satırın sonuna alt satır olarak <Metin> metninin girilmesini sağlar.

i\<Metin>: Her bir satırdan önce üst satır olarak <Metin> metninin girilmesini sağlar.

s/<Düzenli\_İfade>/<Yeni\_Metin>/[İşlem\_Seçenekleri] : En sık kullanılan seçeneği budur. Girilen içerikte Düzenli\_İfade ile eşleşen metinleri Yeni\_Metin ile değiştirir. Sed komutu değişiklik işlemini normalde her satırda ilk bulduğu eşleşmeye yapar fakat aynı satırdaki diğer eşleşmeler içinde değişiklik isteniyorsa İşlem\_Seçenekleri kısmın g harfi yazılmalıdır. Aynı şekilde büyük küçük harf ayrımı olmaksızın eşleştirme yapması içinde İşlem\_Seçenekleri alanına i harfi yazılabilir.

sed komutu normalde dosya içeriklerini değiştirmez işlem gören çıktıyı standart çıktısına yazar. Standart çıktıyı > veya >> ile bir dosyaya yönlendirerek yeni bir dosyaya kayıt yapılabilir. Eğer dosya içerisinde doğrudan işlem yapılmak istenirse bu durumda -i Dosya\_Yolu seçeneği kullanılmalıdır. Bu durumda değişikliği direk dosyada yapacaktır. Bu yol kullanılırken dikkatli olunmalıdır zira yapılan bir hatanın geri dönüşü olmayabilir.

Örneğin /etc/passwd dosyasını ekrana listelemek isteyelim ama normalde : ile ayrılan satırların TAB karakteri işle ayrılmasını istersek aşağıdaki yolla bu işlemi yapabiliriz:

[root@ila ~]# sed 's/:/\t/g' /etc/passwd | head -n 3

root x 0 0 root /root /bin/bash

bin x 1 1 bin /bin /sbin/nologin

daemon x 2 2 daemon /sbin /sbin/nologin

[root@ila ~]# cat /etc/passwd | sed 's/:/\t/g' | head -n 3

root x 0 0 root /root /bin/bash

bin x 1 1 bin /bin /sbin/nologin

daemon x 2 2 daemon /sbin /sbin/nologin

Örneğin sistemdeki /bin/sh kabuğu ile oturum açanların artık /bin/bash kabuğunu kullanmasını istersek bu durumda /etc/passwd dosyasında /bin/sh kabuğu kullananların bilgilerini /bin/bash olarak değiştirilmesi gerekir. Bu işlem sed ile aşağıdaki şekilde yapılabilir:

[root@ila ~]# sed 's/\/bin\/sh/\bin\/bash/' -i /etc/passwd

**8.7.7.4 Diğer Çıktı Düzenleme Komutları**

sort komutu standart girdisinden aldığı metni veya parametre olarak verilen dosya içerisindeki metni verilen kriterlere göre sıralar.

**sort [seçenekler] [dosyalar...]**

En sık kullanılan seçenekleri:

-n : Sort komutu normalde alfabetik olarak sıralama yapar ama sayısal olarak sıralama yapması istendiğinde bu anahtar kullanılır.

[root@ila ~]# sort sort.txt

10

22

3

5

56

[root@ila ~]# sort -n sort.txt

3

5

10

22

56

-r : Sort komutunun soralamasını tersine çevirmek için kullanılır.

-u : Benzer satırları tekrarlamadan göstererek sıralama yapmasını sağlar.

uniq komutu standart girdisinden veya parametre olarak verilen dosya içerisindeki benzer satırları tekrarlamadan ekrana yazar.

[root@ila ~]# cat uniq.txt

ali

veli

10

10

veli

[root@ila ~]# uniq uniq.txt

ali

veli

10

veli

[root@ila ~]# cat uniq.txt | uniq

ali

veli

10

veli

pr komutu standart girdisinden veya parametre olarak kendisine verilen dosyadaki metni yazıcıya gönderilecek formatta ekrana yazar.

nl komutu standart girdisinden veya parametre olarak kendisine verilen dosyadaki metni ekrana satırlarını numaralandırarak yazar.

[root@ila ~]# head -n 3 /etc/passwd | nl

1 root:x:0:0:root:/root:/bin/bash

2 bin:x:1:1:bin:/bin:/sbin/nologin

3 daemon:x:2:2:daemon:/sbin:/sbin/nologin

**8.8 Kabuk Programlama**

Kabuk programlama veya kabuk betikleri yazarak komut satırı üzerinden tekrarlı yapılan işlemleri daha kolay halletmek veya belirli düzen dahilindeki işlemleri betikler ile daha kolay çözümlemek amaçlanır.

Sıklıkla kabuk programlama aşağıdaki işler gerçekleştirilir:

\* Log işleme

\* Raporlama (e-posta geri bildirimi)

\* Otomasyon

\* Günlük kısa işler (tekrarlı işler)

Her kabuğun kendine ait farklı bir programlama yazımı vardır. Buradaki anlatımlar en sık kullanılan bash kabuğu üzerinden yapılmıştır.

Basit anlamda bir kabuk programı bir metin dosyası içerisine alt alta yazılan komutlar bütünüdür.

**[root@ila ~]# cat kabuk.sh**

#!/bin/bash

#Ilk program

echo Merhaba

date

**[root@ila ~]# bash kabuk.sh**

Merhaba

Fri Sep 17 20:55:37 EEST 2010

Kabuk programlama yaparken # ile başlayan satırlar açıklama satırı olarak ele alınır

Bir kabuk programı dosyasına çalıştırma (eXecute) izni verilirse ilgili program doğrudan çalıştırılabilir.

**[root@ila ~]# chmod +x kabuk.sh**

**[root@ila ~]# ls -l kabuk.sh**

-rwxr-xr-x 1 root root 43 Sep 17 20:55 kabuk.sh

**[root@ila ~]# ./kabuk.sh**

Merhaba

Fri Sep 17 20:58:28 EEST 2010

Doğrudan çalıştırma işleminde kabuk programı içerisinde ilk satır olarak yazılan #! İle başlayan satır önemlidir. #! tabirinden sonra tam yolu yazılan program ile geri kalan betik çalıştırılır. Eğer betik içerisinde bu satır yoksa betiğin çalıştırılacağı programın elle yazılması gerekir. O anda üzerinde bulunulan kabuk ile betiğin yorumlanması istenirse source komutu veya . komutu kullanılabilir.

**[root@localhost ~]# source kabuk.sh**

Merhaba

Fri Sep 17 21:04:05 EEST 2010

**[root@localhost ~]# . kabuk.sh**

Merhaba

Fri Sep 17 21:04:11 EEST 2010

**8.8.1 Değişken Mantığı**

Diğer programlama dillerinde olduğu gibi kabuk programlama yapılırken değişkenler kullanılabilir. Kullanımı çevresel değişkenlere benzer; bir değişkene değer atamak için ilgili değişkenin değerinin yanına, boşluk bile bırakmadan, = koyarak değer ataması yapılır. Bir değişkenin değeri okunacağı zaman da ilgili değişkenin isminin başına $ işareti koyulur.

**[root@ila ~]# cat kabuk.sh**

#!/bin/bash

#Ilk program

echo Merhaba

var=`date`

echo "Simdi saat $var"

**[root@ila ~]# . kabuk.sh**

Merhaba

Simdi saat Fri Sep 17 21:16:11 EEST 2010

Bir kabuk program kendisine verilen parametreleri de özel $N (N=1,2,3,…) değişkenleri olarak alır. Örneğin bir kabuk programa ilk verilen parametre $1 değişkeninde, ikinci verilen parametre $2 değişkeninde, … tutulur.

$0 değişkeni ise çalıştırılan programın kendi ismini döndürür.

$@ değişkeni programa verilen tüm parametreleri döndürür.

$# değişkeni parametre sayısını döndürür.

$$ değişkeni de çalışan kabuk programının süreç kimlik numarasını (PID) döndürür.

**[root@ila ~]# cat kabuk.sh**

#!/bin/bash

echo "Ilk parametre $1"

echo "Ikinci parametre $2"

echo "Hepsi: $@"

**[root@ila ~]# . kabuk.sh Merhaba Dunya**

Ilk parametre Merhaba

Ikinci parametre Dunya

Hepsi: Merhaba Dunya

**[root@ila ~]# . kabuk.sh "Merhaba Dunya"**

Ilk parametre Merhaba Dunya

Ikinci parametre

Hepsi: Merhaba Dunya

Betiği sonlandırmak ve geriye değer döndürmek için exit komutu kullanılır. Bu komutun yanına verilen numara geri dönen değer olacaktır ve $? değişkenine atanır. (0 dışındaki değerle sorunlu çalışmaya işarettir)

**8.8.2 Mantıksal Kontroller**

Kabuk programın farklı koşullarda farklı şekilde çalışması istenebilir. Bunu sağlamak için if komutu veya switch – case yapısı kullanılır.

if <koşul1> [<koşul2>...]

then

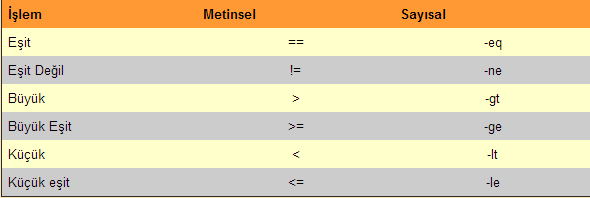
<Koşul sağlanması durumu>

else

<Koşul sağlanmaması durumu>

fi

Kabuk programlamada koşul tanımı yapılırken yazılan mantıksal operatörler sayısal veya metinsel veriler için farklıdır.



[root@ila ~]# cat kabuk.sh

#!/bin/bash

if [ $1 == 'OK' ]

then

echo OK

else

echo NOT OK

fi

[root@ila ~]# . kabuk.sh OK

OK

[root@localhost ~]# cat kabuk.sh

#!/bin/bash

if [ $# -lt 2 ]

then

echo OK

else

echo $# Parametre yeterli degil.

fi

[root@ila ~]# . kabuk.sh Merhaba Dunya

2 Parametre yeterli degil.

Koşul olarak bir dosyanın belirli halleri baz alınacaksa aşağıdaki tanımlar kullanılabilir.

-e - Dosya var

-f - Dosya var ve düzenli bir dosya

-d - Dosya klasör

-h - Dosya link

-s - Dosya boyu 0 değil

-r -w -x - Dosyanın ilgili izni var

-nt - Birinci dosya ikinciden yeni

-ot - Birinci dosya ikinciden eski

**[root@ila ~]# cat kabuk.sh**

#!/bin/bash

if [ -d '/tmp' ]

then

echo /tmp dizini var

else

echo /tmp dizini yok

fi

[root@ila ~]# . kabuk.sh

/tmp dizini var

Case-esac yapısı ile de farklı koşullar için farklı tepkiler sağlanabilir.

case $<incelenecek\_değişken> in

değer1) işlem1 ;;

değer2) işlem2 ;;

...

\*) varsayılan işlem ;;

esac

[root@ila ~]# cat kabuk.sh

#!/bin/bash

case $# in

1) echo bir pametre ;;

2) echo iki parametre ;;

\*) echo "sifir veya 2'den fazla parametre" ;;

esac

[root@ila ~]# . kabuk.sh Merhaba Dunya

iki parametre

**8.8.3 Döngüler**

Döngüler belirli durumlar sağlanana kadar tekrar etmesi gereken işlemlerde kullanılır. Kabuk programlamada for veya while ile döngü yapıları kurulabilir.

**while <koşul>**

**do**

**…**

**done**

**for <koşul>**

**do**

**...**

**done**

**for <dizi>**

**do**

**...**

**done**

**[root@ila ~]# cat kabuk.sh**

#!/bin/bash

for var in `ls -1 /`

do

du -sh $var

done

**[root@ila ~]# ./kabuk.sh**

4.6M bin

8.0K dev

4.0K etc

996K lib

4.0K proc

34M sbin

4.0K sys

12K tmp

**8.8.4 Örnek Kabuk Programları**

#!/bin/bash

for zipfile in `ls -1 /root/ziplenecekler/`

do

gzip $zipfile

ls -l $zipfile.gz && echo "$zipfile ziplendi."

done

#!/bin/bash

for x in \*

do

cp -avuf $x /yedek

done

#!/bin/bash

while [ ! -f "/tmp/test" ];

do

sleep 3

done

echo -n "dosya olusturuldu: `ls -l /tmp/test` "

**UYGULAMA 1**

1. Sisteminizde var olan kabukları listeleyin ve o anda kullanmakta olduğunuz kabuğun ismini bulun.
2. Kullanmakta olduğunu kabuk içerisindeyken ILA isminde ve içeriğinde Kur1 metnini tutan bir genel çevresel değişken tanımlayın. Bulunduğunuz kabuk içerisindeyken sh (Bourne Shell) kabuğuna geçerek ILA değişkeninin içeriğini ekrana basın. Ardından sh kabuğundan ayrılın.
3. Komut satırından "ara" diye bir komut çağrıldığında locate komutunun çağrılmasını sağlayın.
4. Tek satır bir komutla locate komutunun çalışıtırılabilir dosyasının izinlerini listeleyin.
5. Sistem üstünde yüklü olan tüm SATA, SCSI ve USB diskleri listeleyin.
6. Bir metin editörü veya cp komutunu kullanmadan /etc/group dosyasını okutup içeriğini grup\_list isimli bir dosyaya yazdırın.
7. Bir metin editörü kullanmadan grup\_list dosyasındaki GID'si 50'den büyük olan grup isimlerini listesini hem ekrana bastırın hem de bir dosyaya yazdırın.
8. Makinanızda kurulu olan rpm paketlerinin sayısını herhangi bir metin editörü kullanmadan bulun.
9. Bilgisayarınıza yum komutunu kullanarak kurduğunuz her paket için yum, /var/log/yum.log dosyasında çalışmasına ait bilgileri tutar. Bu bilgi ışığında makinanıza ‘yum install paket\_adı’ diyerek kurduğunuz paketlerin sayısını bir metin editörü kullanmadan elde ediniz.

**CEVAPLAR 1**

1---

[root@ila ~]# cat /etc/shells

/bin/sh

/bin/bash

/sbin/nologin

/bin/tcsh

/bin/csh

/bin/ksh

[root@ila ~]# echo $SHELL

/bin/bash

[root@ila ~]#

2-------------

[root@ila ~]# ILA='Kur1'

[root@ila ~]# export ILA

[root@ila ~]# /bin/sh

sh-3.2# echo $ILA

Kur1

sh-3.2# exit

exit

[root@ila ~]#

3-----------

[root@ila ~]# alias ara=locate

[root@ila ~]#

4-----------------

[root@ila ~]# ls -l $(which locate)

-rwx--s--x 1 root slocate 23856 Mar 17 2009 /usr/bin/locate

[root@ila ~]#

5-----------

[root@ila ~]# ls -l /dev/sd?

brw-r----- 1 root disk 8, 0 Sep 26 12:13 /dev/sda

brw-r----- 1 root disk 8, 16 Sep 26 12:13 /dev/sdb

[root@ila ~]#

6---------------

[root@ila ~]# cat /etc/group > grup\_list

[root@ila ~]#

7--------------

[root@ila ~]# awk -F : '$3>50{print $1}' grup\_list | tee bidosya

...

[root@ila ~]#

8---------

[root@ila ~]# rpm -qa | wc -l

687

[root@ila ~]#

9------------

[root@ila ~]# grep 'Installed: ' /var/log/yum.log | wc -l

44

[root@ila ~]#

10--------------

[root@ila ~]# find /etc/ -name '\*.conf' -type f -exec rpm -qf {} \; | grep -F 'is not owned by any package' | cut -d ' ' -f 2 > rpmsiz\_dosyalar

[root@ila ~]#

11--------------

[root@ila ~]# cat idbul.sh

#!/bin/bash

if [ "$1" == '' ]

then

echo "Lutfen bir ID degeri girin"

exit 100

else

kullanici=`cut -d ':' -f 1,3 /etc/passwd | grep :$1 | sed s/:$1$//`

echo $kullanici

fi

[root@ila ~]#

12----------------

[root@ila ~]# cat fazla\_izin\_bul.sh

#!/bin/bash

if [ "$1" == '' ]

then

echo Lutfen kullanici adi bilgisi giriniz

exit 200

else

kullaniciadi=$1

evdizini=`finger -m "$kullaniciadi" | grep '^Directory: ' | awk '{print $2}'`

if [ "$evdizini" == '' ]

then

echo "$kullaniciadi sistemde kayitli degil."

exit 201

else

for x in `find /home/ -perm /o=r`

do

echo $kullaniciadi:$x

done

fi

fi

[root@ila ~]#

Sisteminizde /etc dizini altında bulunan .conf uzantılı dosyalardan herhangi bir rpm paketi ile sisteme kurulmuş olmayanların listesini rpmsiz\_dosyalar dosyasına yazdırınız.

UID numarası verilen bir kullanıcının kullanıcı adını ekrana yazan idbul.sh kabuk programını yazın.

Kullanıcı adını paratmetre alan bir program yazın. Aldığı kullanıcı adı ile ilgili kullanıcının ev dizininde kendisi ve birincil grubu dışındaki kullanıcılara okuma izni verilen dosyaların listesini kullanıcı\_adı:dosya\_yolu şeklinde ekrana döken fazla\_izin\_bul.sh kabuk betiğini yazın ve çalıştırın.

**İŞ SÜREÇ YÖNETİMİ**

**9.1 Süreç (Process) Nedir?**

NOT:

\*\*\*\*\*\*\*\*\*\*\*\*SÜREÇ YÖNETİMİ\*\*\*\*\*\*\*\*\*\*\*\*\*

süreç makinede çalışan her uygulama bir süreçtir.belirli bir bellek alanı vardır.disk kullanımı için alanı vardır.Kısaca kaynakları(recources) vardır ve bunları çekirdek yönetir.

PID: her process ın vardır.

(parent PID) PPID: süreçler agaç yapısındadır.ana-ogul ilişkisi vardır.her sürecin bir ana süreci olmak zorundadır.BU agacın kökünede init denir. ve PID=1 dir.

-ps -aux (ppid ler listenmez.)

user(hangi kullanıcı haklarıyla çalıştıgı) pid %cpu %mem şeklinde bir çıktısı vardır. [] ile gösterilenler çekirdeğin kendi iç süreçlerine konulur.bunlarla ilgili pek bişe yapmayız.

-Run time library(linux)=dll(windows) ==> 20 tane process te çagırılırsa bellege 1 kez yüklenir.20 process te o alanı kullanır.(örn virtual bellek kullanımı)

-pts :pseudo terminal ;farazi aygıt.

-ps -ef (parent pid yi de verir.)

-ps auxw :w ile ekran bufferı genişletilir.ne kadar çok w koyarsak buffer o kadar genişler.

-ps auxf: ana-ogulları dikkate alacak şekilde çıktı verir.(terminolojide dogrudan iletişim ikilidir.)

-pstree süreçleri agac yapısında listeler.

pstree -Gahp (G telnet baglantısında görüntü bozulmasını önler.)

-pidof bash

-pgref -l -t pts/3 zor yol:ps aux|grep pts/3|awk '{print $2}'

-top : 3 snyede bir kendini yenilen dinamik bir liste çıkartır.

.ilk satır uptime ile elde edilen çıktıdır.:load avarage

.zombie: sadece bir pid tutar.Beklenmedik durumlarla ilgili durum tutar.bunları öldüremeyizBelirli aralıklarla çekirdek zombie leri toparlar.20-30 tane ye çıkarsa sayısı ters bir durum var demektir.1 2 tane ise herşey normaldir.

.Cpu alanında 1 tane işlemci varmış gibi gösterir.(1) e basarsak işlemci başına düşen yükü gösterir.(4 çekirdek varsa 4 tane çıktı olur örn.)

.shift+m bellek tüketimine göre sıralar.

.shift+p işlemci kullanımına göre sıralar.

.d ye basarak yenilenme zamanının 3 sn yeden istediğimiz sıklıga çekebilirz.

.q çıkış için kullanılır.

\*\*\* makinenin kasılıp kasılmadığına ne ile bakarız.?

w komutu ile kullanıcıların o anda neler yaptıgı izlenebilir.Bir linux sistemin ne kadar kasıldıgını burdan anlayabiliriz.(degeri 0 ile 10 arasında ise normaldir.10 u geçerse ya sistemin tasarımını değiştirmek yada donamımı iyileştirmek gerekir.)Donanım bagımlı bir fonskiyon tarafından üretilir çünkü bu değer.

--Sunucularda masaüstü arabirimi çalışmaz.

-kill 6740 (pid degeridir.)

-kill aslında process öldürmek için değildir. kill aslında sinyalleşmek için gereklidir.kill -l diyerek bu sinyallerin ne olduklarını listeleyebiliriz.

-süreç herzaman ölmeyebilir.(32 33 unixte kullanılan özel sinyallerdir. Sİnyallerin numaraalrı örtüşsün diye Lİnuxtada 32 33 pass geçilmiştir.)

-kill -9 pid zorbalıktır:) Kpanma prosedurunu falan boş ver pat diye kapat demektir. Bir süreç ölmüyorsa demekki işi var demektir.önce -15 ile öldürmeyi dene.niye ölmediğine bak mantıklı açıklaması yoksa -9 ile öldür.

-ctrl+c de aslında 2 numaralı interupt sinyali gönderir.

-kill `pidof bash` şeklinde kullanılabilir.

-killall bash : bütün bashleri öldürür.

---süreç önceliğini değiştirme:

-nice -20...0..19 a kadar deger verilebilir.(saga dogru öncelik düşer sola dogru artar.) NŞA da süreç 0 önceliği ile başlar.Root dışındaki kullanıcılar sadece 0 ile 19 arasında değer atayabilir.

-makinede boş cpu cycle oluşturmak için: cat /dev/zero kullanılır.Aygıt dosyasından sonsuz kere 0 okur. /dev/zero ve /dev/null çekirdeğin kendi içerisinde halletiği operasyonlardır.

-çalışan sürecin pid sini değiştirmek için:

-renice +19 5671

not :önceliği -5 den yukarıya alma çünkü çekirdeğin süreçlerininde önüne geçmiş olabiliriz.

-nice sleep 5 :öncelik vermezsek direk +10 veriri.(önceliği geriye almış oluruz.)

Ana süreç ogul süreç çalıştırıldığında kendi kaynakklarını ogluna teslim eder.Komutu arka plana itmek için:

-sleep 20& diyerek komutu arka plana itebiliriz.

-jobs dersek arka planda çalışan komutları görebiliriz.

-jobs -l ile PID si ile birlikte listeler.

-fg indexno :ile arka planda çalışan komutu geri alabilriz.Tekrar geri plana atarkan ctrl+z kullanılabilir. Ama bu şekilde süreç durur.Arka planda duran süreci çalıştırmak için

-bg index no kullanılır.

-pgrep -t pts/1 çalıştığımız kabuğun pidsi

-ls -l /proc :PID lerin bulundugu klasorler vardır örn:

ls -l /proc/2773/fd gibi. 0->/dev/pts/1 (0:file discrpirtor dur. 0 ı kullanırsak kabuk birkez daha açılır.)

-kabukta bir işlem yapıldıgında (kabuk bir ogul süreç çalıştırdığında ) örn: /dev/pts/1 olan işaretçiyi programa verir.Çalışan bir programın işaretçisi değiştirlemez.Arka plana atılan sürecin durmasının sebebi budur.

-sleep 200& çalıştırdık: kabugu kapattığımzda ne olur.bu da ölür.ana süreç ölürse ogul süreçte ölür. Bunu istemiyorsak;

nohup sleep 200 kullanırız.Normalde kabuk kendi kullandığı işaretçiyi ogula veriyordu.BUnu yapmıyor. Ekran çıktısı ignore edilir.Artık ekrana çıktı verilmez nohup isminde bir dosyay yazılır. (kill - deki)1 numaralı sinyale bagışıklık kazanıp init in altına yerleşir.

-genelde nohup la bişeler çalışacaksa arka planda çalışır.

nohup find / &

-nohup dosyası(nohup.out) o an nerde çalışıyorsak oraya yazar.

-nohup find / & > /dev/null istediğimiz başka bir yere yazdırabiliriz.

--network üzerinden işlem yaparken işlem uzun sürecekse hayat kurtarır zaten ekran kopmuştur. nohup yazıp arka planda çalıştırırsak sıkıntı olmaz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Süreç (process), Linux işletim sistemi (kernel) tarafından bir komutun çalıştırılmasıdır. Daha doğrusu bir komutun yürütülmesi ile bir süreç başlatılmış olur. Sistem ile kullanıcı arasında bir arayüz olan kabuk da bir programdır ve oturum açıldığı andan itibaren bir süreç oluşturur. Oluşan bu süreç ilgili oturum kapatılıncaya dek yaşar.

Süreç Yapısı

Süreç yapısı, dosya sistemi hiyerarşine benzer bir ağaç yapısına sahiptir. Bu yapıda kök (root), ana (parent) ve oğul (child) süreçler vardır. Her ana süreç yeni bir oğul süreç yaratabilir ve bu yeni süreçler de başka süreçleri yaratabilirler. İşletim sisteminde yeni süreç yaratan sistem çağrısına fork denir.

Linux işletim sistemi açıldığında, init adında (PID=1) bir süreç başlar ve bu süreç sistemin ve kullanıcıların çalıştırdığı tüm süreçlerin anası olan, kök süreçtir. Bu sürecin süreç ağacındaki yeri dosya sistemi ağacındaki kök dizini (/) gibidir.

Süreç Kimlikleri (PID, PPID)

Linux'ta başlatılan her sürece çekirdek tarafından ayrı bir kimlik PID (process identification) numarası atanır ve süreç yaşadığı sürece aynı PID numarasına sahip olur. Bir süreç öldüğünde ilgili PID numarası da boşa çıkar ve daha sonra başka bir sürece çekirdek tarafından atanabilir.

init süreci dışında sistemdeki tüm süreçler arasında bir ana-oğul ilişkisi vardır, bu nedenle her süreç PID numarasının yanı sıra, kendini yaratan sürecin PID bilgisini tutan PPID (parent process identification) numarasına da sahiptir. PPID numarası sayesinde ağaçtaki her sürecin yeri belli olur. init sürecinin ana süreci olmadığı için PPID değeri 0 olarak gözükür.

init,1

├─sshd,2141

│ ├─sshd,29114

│ │ └─bash,29118

│ │ ├─grep,29549 -B 2 -A 1 bash

│ │ └─pstree,29548 -Gap

**Süreç Yönetimi**

Linux üzerinde çalışan süreçler üzerinde listeleme, süreçlere sinyaller gönderme veya önceliklerini değiştirme gibi işlemler yapılabilir. Böylece süreçlerin ne şekilde veya nasıl çalışacağı ya da çalıştığı izlenebilir.

Linux işletim sistemindeki çoğu hizmet bir süreç olduğu için, süreçler üzerinde işlem yaparken dikkatli olmak gerekir. Yanlış veya hatalı yapılan bir işlem sistemin stabil çalışmasında sorunlara sebep olabilir.

**Ps**

ps komutu çalışan süreçler hakkında detaylı bilgilerin alınmasında kullanılan komuttur. Tüm UNIX türevi işletim sistemlerinde bu komut bulunur fakat farklı işletim sistemleri üzerinde komutun aldığı parametreleri farklılık göstermektedir. Detaylı bilgi için man sayfasına bakılmalıdır.

ps komutu ile kullanılabilecek genel seçenekler:

* a : Tüm terminallerde çalışan süreçleri göster
* x : tty’lerde çalışanlar hariç tüm süreçleri göster
* c : Süreçlere ilişkin komutların sadece isimlerini gösterir.
* e : Komutları parametreleri ile gösterir.
* l : Çıktı ayrıntılı şekilde verilir.
* r : Sadece o anda çalışan süreçleri gösterir.
* u : Süreçlerin ilişkin kullanıcıyı ve sürecin başlangıç zamanını verir.
* w : Çalışan komutların parametreleri hakkında ayrıntılı bilgi verir.
* ww : w seçeneğinden daha ayrıntılı bilgi verir.
* f : Süreçleri ağaç yapısında gösterir.

ps komutu en çok, sistemdeki tüm süreçleri ve süreçlerin başlangıç zamanını gösterilmesini sağlayan aux ve -ef seçenekleri ile birlikte kullanılır.

[root@ila ~]# ps -ef

UID PID PPID C STIME TTY TIME CMD

root 1 0 0 14:00 ? 00:00:00 init [5]

root 2 1 0 14:00 ? 00:00:00 [migration/0]

root 3 1 0 14:00 ? 00:00:00 [ksoftirqd/0]

root 4 1 0 14:00 ? 00:00:00 [migration/1]

root 5 1 0 14:00 ? 00:00:00 [ksoftirqd/1]

root 6 1 0 14:00 ? 00:00:00 [events/0]

**Diğer Süreç Listeleme Komutları**

Ps komutuna ek olarak çoğu Linux işletim sistemlerinde çalışan farklı özelleşmiş komutlar da bulunmaktadır.

**pstree Komutu**

Süreçlerin bir ağaç yapısında listelenmesini sağlar.

pstree komutu ile kullanılabilecek temel seçenekler:

* -a : Komutları seçenek ve parametreleri ile gösterir.
* -h : Şu anda yürütülen süreç ve bu sürecin ana süreçlerini göster.
* -l : Komutların seçenek ve parametrelerini daha ayrıntılı göster.
* -u : Süreçlerin sahibi olan kullanıcıları göster.
* -p : Süreçlere ait PID numaralarını göster.

**top Komutu**

Sistemde çalışan süreçleri ve bellek kullanımını gerçek zamanlı izlemek için top komutu kullanılır. Varsayılan olarak top komutu en çok işlemci kullanımına göre süreçleri listelemektedir. Bu sıralamayı bellek kullanımına göre yeniden düzenlemek için komut çalıştırıldıktan sonra [M] tuşu kullanılmalıdır, varsayılan davranışa geri dönmek içinse [P] tuşu kullanılmalıdır.

Normalde top komutu kendini 3 saniyede bir yeniler ama her boşluk tuşuna basıldığında komut kendini tekrar yeniler. Aynı şekilde d tuşuna basılarak kendini yenileme süresi değiştirilebilir.

Top komutu listeyi hazırlarken de sistemi yorduğu için yük altındaki sistemlerde uzun süreli çalıştırılmamalıdır.

**pidof Komutu**

Kendisine parametre olarak verilen süreç ismine ait PID değerlerini döndürür.

**pgrep Komutu**

Sistemde çalışan süreçler arasında arama yapmaya yarar.

* -l : Süreç isimlerini ekrana yazar.
* -u : Belirtilern kullanıcıya ait süreçleri gösterir.
* -t : Belirtilen terminalde çalışan süreçleri gösterir.

**uptime Komutu**

Sistemin ne kadar süredir ayakta olduğunu ve 1, 5 ve 15 dakikalık sistem yükünü gösterir.

**w Komutu**

uptime komutunun verdiği sistem bilgilerinin yanısıra o anda oturum açmış olan kullanıcıları, nereden

bağlandıkları ve hangi programı ne kadar süredir çalıştırdıkları gibi ek bilgiler sunar.

**[root@ila ~]# pstree -Gahp | head -n 10**

init,1

├─auditd,1999

│ ├─audispd,2001

│ │ └─{audispd},2002

│ └─{auditd},2000

├─avahi-daemon,2287

│ └─avahi-daemon,2288

├─cma,2259

│ └─cma,2260

│ └─cma,2261

**[root@ila ~]# pidof httpd**

27123 14555 14545 3934 3933 3932 3931 3930 3929 3928 3927 9176

**[root@ila ~]# pgrep -l -t pts/0**

29118 bash

[root@ila ~]# uptime

22:33:09 up 62 days, 5:41, 1 user, load average: 0.04, 0.13, 0.10

**[root@ila ~]# w**

22:31:10 up 3:47, 3 users, load average: 0.00, 0.02, 0.30

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root :0 - 18:50 ?xdm? 1:40 0.35s /usr/bin/gnome-session

root pts/2 ao124.cc.itu.edu 20:36 0.00s 0.03s 0.00s w

**Sinyaller ve Süreçler**

Süreçler çalışırken onlara sinyal gönderilebilir. Bu sayede kendi halinde çalışan bir sürecin dışarıdan etkileşime tabi olması mümkündür. Komut satırından bir sürece sinyal gönderilmek istendiğinde **kill** veya **killall** komutları kullanılır. Kill komutuna **-l** anahtarı verilerek sistemde kayıtlı olan sinyal listesine ulaşılabilir.

Çekirdek de süreçler ile etkileşim kurabilmek için sinyalleri kullanır. Örneğin kullanıcı oturumunu kapattığında ilgili oturuma ait tüm süreçlere 1 numaralı **SIGHUP** sinyalini göndererek süreçleri öldürmeye çalışır.

Kullanıcı tarafından en sık yollanan sinyaller **15** (**SIGTERM**) ve **9**(**SIGKILL**) sinyalleridir. 15 bir sürecin normal olarak ölmesi için gönderilir. Böylece süreç kendi içindeki kapanma prosedürlerini başlatır. 9 sinyali ise sürecin hemen ölmesi için gönderilir.

kill -s <Sinyal\_Numarası ya da Sinyal\_İsmi> <PID …>

kill - <Sinyal\_Numarası> <PID …>

killall -s <Sinyal\_Numarası ya da Sinyal\_İsmi> <PID …>

killall - <Sinyal\_Numarası> <PID …>

Eğer Sinyal\_Numarası ya da Sinyal\_İsmi verilmezse bu durumda varsayılan sinyal 15 (SIGTERM)’dir.

Bazı durumlarda süreçler çekirdekte çıkan bazı sorunlarda tıkanabilirler. Bu tür süreçler bellek işgal etmez ve bir iş yapmazlar ama ps gibi listeleme komutlarında listede gözükürler ve öldürülemezler. Bu tür süreçlere **zombi süreç** denir. Zombi süreçlerin sayısı çok artmadığı sürece sorun teşkil etmezler.

**[root@localhost ~]# pgrep mingetty**

1344

1356

1357

1358

1359

1360

**[root@localhost ~]# pgrep mingetty | xargs kill**

**[root@localhost ~]# pgrep mingetty**

1430

1435

1439

1442

1443

1444

**[root@localhost ~]# killall mingetty**

**[root@localhost ~]# pgrep mingetty**

1522

1525

1528

1531

1534

1537

**Süreç Öncelikleri**

Sistemde bazı süreçlerin görevleri kritiktir ve diğer süreçlerden daha önce tamamlanması gerekir. Aynı şekilde bazı süreçlerin aciliyeti ise hiç yoktur ve diğer süreçlerden çok geç bitebilir. Bu tür durumlar süreç öncelikleri ile ayarlanır.

Sistemdeki en yetkili kullanıcı(root) çalışan bir sürecin önceliğini, **renice** komutunu kullanarak aşağı ya da yukarı çekebilir. Bu sayede çok yüklü çalışan bir sistemde kritik süreçlerin daha fazla CPU zamanı kullanmaları sağlanabilir.

Linux sistemlerde süreçlerin önceliklerine nice değeri denir. Düşük nice değerine sahip süreçlerin

öncelikleri yüksek ve yüksek nice değerine sahip süreçlerin öncelikleri ise düşüktür. Bu değer -20 (en yüksek öncelik) ile 19 (en düşük öncelik) arasında değişir. Oğul süreçler oluşurken, ana süreçlerinin nice değerini alırlar. Genelde sistemde yaratılan bir sürecin nice değeri 0 olur.

Kısıtlı sistem kullanıcıları sahip oldukları süreçlerin önceliklerini sadece daha düşük değerlere getirebilirler, yani daha fazla öncelik kazandıramazlar.

Bir sürecin öncelik değerini değiştirmek için ilgili sürecin PID numarasına ihtiyaç vardır.

**renice <priority> <pid>**

**[root@ila ~]# renice -5 2239**

2239: old priority 0, new priority -5

[root@ila ~]# renice 19 2239

2239: old priority -5, new priority 19

**renice** komutunun etkileri **ps** komutunun çıktısındaki **STAT** sutünunda görülebilir ([N]- düşürülmüş öncelik, [<] - arttırılmış öncelik).

[root@localhost ~]# ps aux | grep -E 'N|\<' | head -n 5

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 2 0.0 0.0 0 0 ? S< Sep02 0:00 [migration/0]

root 3 0.0 0.0 0 0 ? SN Sep02 0:00 [ksoftirqd/0]

root 4 0.0 0.0 0 0 ? S< Sep02 0:00 [watchdog/0]

Çalışan bir sürecin nice değeri değiştirilebildiği gibi süreç daha yaratılırken de nice değeri atanabilir. Bu işlem için nice komutu kullanılır.

nice [-n Nice\_Değeri] <Komut>

nice komutuna eğer -n ile değer verilmezse varsayılan olarak 10 değerini seçer.

**[root@localhost ~]# nice sleep 20 &**

[1] 2246

[root@localhost ~]# ps auxf | grep sleep

root 2246 0.0 0.1 3792 476 pts/0 SN 01:38 0:00 | \\_ sleep 20

root 2256 0.0 0.2 3996 684 pts/0 S+ 01:38 0:00 | \\_ grep sleep

**Arka Planda Komut Çalıştırma**

Linux'te kabukta bir program/komut çalıştırıldığında, kabuk fork işlemi gerçekleştirerek bir oğul süreç başlatır. Bu oğul süreç çalışmasını tamamlayıncaya kadar ana süreç arka planda uykuda bekler. Oğul süreç, çalışması sonlandığında ana sürece başarılı olup olmadığını bildirir ve ölür. Bundan sonra ana süreç (kabuğu çalıştıran süreç) uyanır ve ön plana gelerek yeni komutlar bekler.

Kabuk üzerinde bir komut arka planda çalıştırıldığında, kabuk yine bir oğul süreç yaratır fakat uyumaz ve yeni komutların girilmesi için bekler. Bu süreç ana süreçten bağımsız olarak çalışmasını sürdürür ve tamamlar.

Sistemin kendi süreçleri dışındaki süreçler genellikle ön planda çalışırlar. Süreçler ön planda çalıştırıldıklarında, sürecin çalıştırıldığı terminale süreç sonlanıncaya dek başka bir komut gönderilemez.

Komutlar arka planda çalıştırılırsa ya da ön plandaki süreç arka plana atılırsa, terminale yeniden komut gönderilebilir.

Bir komutu arka planda çalıştırmak için:

İlgili kabukta komut yazıldıktan sonra bir boşluk bırakılarak & (ampersant) işareti kullanılır.

Ön planda çalışan bir komutu arka plana atmak için:

Komut çalışırken [CTRL ]-[Z] tuş kombinasyonu ile çalışma kesilir ve ardından bg komutu kullanılır.

Arka planda çalıştırılan ya da arka plana atılan süreçleri ön plana çıkarmak için fg komutu kullanılır.

Sistemde arka planda çalışan süreçlerin listesini almak için jobs komutu kullanılır.

jobs komutunun verdiği süreç listesindeki numaralar işlerin hangi sıra ile arka plana atıldığını göstermektedir. Buradaki numaralar bu süreçleri ön plana çıkarmak için fg komutuna parametre olarak verilir. Bu rakamların yanındaki +/- işaretleri de fg ve bg komutlarına parametre olarak verilebilir. Jobs komutuna -l anahtarı verilirse arka plandaki sürecin PID numarasını da ekrana yazar.

**[root@ila ~]# sleep 1200 &**

[1] 4402

**[root@ila ~]# jobs**

[1]+ Running sleep 1200 &

**[root@ila ~]# sleep 1000**

[1]+ Stopped sleep 1000

**[root@ila ~]# bg**

[1]+ sleep 1000 &

**[root@ila ~]# jobs**

[1]+ Running sleep 1000 &

**[root@ila ~]# jobs**

[1]+ Stopped top

[2]- Running sleep 1500 &

**[root@ila ~]# jobs**

[1]+ Stopped top

[2]- Running sleep 1500 &

**[root@ila ~]# fg 2**

sleep 1500

**Oturum Kapatılsa Bile Komutun Çalışması**

Linux terminali üzerinde oturum açıldığında çalıştırılan süreçler oturum kapatıldığında normalde kapanırlar. Bazı durumlarda oturum kapatılsa bile çalıştırılan süreçlerin işlerini bitirene kadar devam etmesi istenebilir. Bu tür durumlarda nohup komutu ile komut çalıştırılır.

**nohup <Çalıştırılacak\_Komut>**

Nohup ile bir komut çalıştırıldığında komutun çıktıları ekrana değil çalışılan dizindeki nohup.out dosyasına yazılır. Nohup ile bir komut çalıştırılırken genellikler arka planda başlatılır. Böylece kabuk üzerinden diğer işlemler yapılıp ilgili oturum sonlandırılabilir ve program arka planda işlemlerini oturum süresince de ve oturum kapandıktan sonra da devam ettirebilir.

**[root@localhost ~]# nohup find -name '\*.conf' &**

[1] 2905

**[root@localhost ~]# nohup: appending output to `nohup.out'**

**[root@localhost ~]# logout**

…

…

login as: root

root@160.75.5.174's password:

Last login: Sun Sep 19 02:30:38 2010 from 88.235.34.200

**[root@localhost ~]# cat nohup.out**

./freeradius-1.1.3/raddb/sql.conf

./freeradius-1.1.3/raddb/proxy.conf

./freeradius-1.1.3/raddb/snmp.conf

./freeradius-1.1.3/raddb/pgsql-voip.conf

./freeradius-1.1.3/raddb/oraclesql.conf

./freeradius-1.1.3/raddb/eap.conf

./freeradius-1.1.3/raddb/mssql.conf

./freeradius-1.1.3/raddb/otp.conf

./freeradius-1.1.3/raddb/postgresql.conf

./freeradius-1.1.3/raddb/clients.conf

./freeradius-1.1.3/raddb/sqlippool.conf

./freeradius-1.1.3/raddb/experimental.conf

./freeradius-1.1.3/suse/admin-httpd.conf

./freeradius-1.1.3/dialup\_admin/conf/captions.conf

./freeradius-1.1.3/dialup\_admin/conf/admin.conf

./freeradius-1.1.3/dialup\_admin/conf/naslist.conf

./.cpan/build/perl-ldap-0.39/data/slapd.oc.conf

./.cpan/build/perl-ldap-0.39/data/slapd.at.conf

./.cpan/build/perl-ldap-0.37/data/slapd.oc.conf

./.cpan/build/perl-ldap-0.37/data/slapd.at.conf

**UYGULAMA 1**

1. Sisteminizde çalışan süreçlerin PID ve PPID değerlerini listeleyin.
2. ps komutundan destek alarak root dışı haklarla çalışan süreçleri listeleyiniz.
3. Sisteminizde root dışı kullanıcılara ait dosyaları bulun (find). Bu işlemi yaparken find komutunun sistemi fazla zorlamaması için komutu önceliği düşük olarak başlatın.
4. Sisteminize o anda bağlı olan kullanıcıları ve sistem yükünü gösterin.
5. "sleep 3600" komutunu arka planda ve kesilmeksizin (nohup) başlatın. Ardından tekrar ön plana alın. Sonra tekrar arka plana gönderip çalıştırın. Bulunduğunuz kabuktan ayrılın veya terminal ekranında iseniz oturumunuzu kapatın. Son olarak çalıştırdığınız sleep sürecini bulup öldürün.

**CEVAPLAR 1**

**[root@ila ~]# ps -ef**

UID PID PPID C STIME TTY TIME CMD

root 1 0 0 Sep26 ? 00:00:00 init [3]

root 2 1 0 Sep26 ? 00:00:00 [migration/0]

...

**[root@ila ~]#**

**[root@ila ~]# ps aux | grep -v '^root'**

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

dbus 2030 0.0 0.5 6144 1320 ? Ss Sep26 0:00 dbus-daemon --system

ntp 2093 0.0 3.1 8104 8100 ? SLs Sep26 0:00 ntpd -4 -u ntp:ntp -p /var/run/ntpd.pid -g

smmsp 2123 0.0 0.5 8264 1492 ? Ss Sep26 0:00 sendmail: Queue runner@01:00:00 for /var/spool/clientmqueue

xfs 2173 0.0 0.7 7324 1996 ? Ss Sep26 0:00 xfs -droppriv -daemon

avahi 2206 0.0 0.7 6500 1976 ? Ss Sep26 0:00 avahi-daemon: running [localhost-2.local]

avahi 2207 0.0 0.2 6368 732 ? Ss Sep26 0:00 avahi-daemon: chroot helper

**[root@ila ~]#**

**[root@ila ~]# nice find / \! -user root**

...

**[root@ila ~]#**

**[root@ila ~]# w**

11:24:30 up 10 days, 18:29, 1 user, load average: 0.03, 0.01, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root pts/0 ila96.example.org 0.00s 0.04s 0.00s w

**[root@ila ~]#**

**[root@ila ~]# nohup sleep 3600 &**

[1] 10663

**[root@ila ~]# nohup: appending output to `nohup.out'**

**[root@ila ~]# fg**

nohup sleep 3600

<Ctrl+z basılır>

[1]+ Stopped nohup sleep 3600

**[root@ila ~]# bg 1**

[1]+ nohup sleep 3600 &

**[root@ila ~]# exit**

logout

<Yeniden terminal açılılır>

**[root@ila ~]# pgrep -l sleep**

10663 sleep

**[root@ila ~]# kill 10663**

[root@ila ~]#

**DİSK YÖNETİMİ**

**Dosya Sistemi**

NOT:

-Harddisk i aldık.bilgisayara taktık naparız?Biçimlendirme.(windostaki gibi)Biçimlendirerek aslında disk üzerinde dosya sistemi oluşturuyoruz.Harddisk 0 ve 1 lerden oluşan bir aygıttır.Bunalrın anlamlı olması gerekir.BU anlamalandırmada kullanılan protokele dosya sistemi denir.işletim sistemi bir dosyanın disk üzerinde nasıl tutulacagına bu protokele bakarak karar verir.Windows NTFS den anlar.Wİndows Nt zamanında çıkmıştır.Revizyonlar yapılıp yapılmadığını private bir format olduğunda bilmiyoruz.Saklandığı için Windows dışında kimse kullanamaz.(linux da)ntfs için linux tarafında reverse engineering yapılmışıt.(ürüne bakarak bunun aynısı bizde yaparız abi denilerek çıkmıştır yola)Redhat, Centos ntfs dosya sistemini okumaz bile.

Lİnux ext kullanır.

FAT32 yide windows çıkarmıştır ve herkes desteklemiştir. Çünkü publis olmuştur.

fat32 usbler için standartlaşmıştır ve fat32 de 2 gb dan büyük data tutulamaz .2 gb dan büyük veri tutabilmek için extendenFAT(extFAT)kullanılır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*EXT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*(extended file system)

ext2,ext3,ext4 gibi veriyonlar oluşmuştur.(sınırlar genişlemiştir.)

ext2 ile ext3 arasında formatlamadan geçiş yapılabilir.(journaling)

fat32 journaling dosya sistemi değildir.

ext4 ten ext3 e yani tersine bir geçiş yoktur.

(Disklerle ilgili yapılan denemeler sanal makinede yapılmalıdır.YANlış bir harekette makine çöker.)

-fdisk -l : makinemizdeki disklerle ilgili alanları listeler.(sanal disk alanı /dev/vda dır.)

-windows ta formatlamadan önce bölümlüyoruz.Bu komutun çıktısından da bu şekilde devam ederiz.

MBR(MASTER boot record)-> fdisk geçerli(ilk 512 byte bilgisayarı boot etmek için kullanır.)

GPT(GUID partition)->yenidir.gdisk

örn 4tb lık harddisk alıp,böldük,kullandık(tek bölüm olarak kullanmamanın avantajı,bakım süresi azalır.(tarama gibi))

-fdisk /dev/sda ile hangi diskle oynicaksak bunun partition table ına ulaşabiliriz.

p ye basarak görebiliriz.

n;yeni bir partition oluşturur.

bundan sonra primary mi extension mu diye sorar.

not:bilgisayar açılınca gidip,mbr nin ilk 466 byte nı alıp ram e kopyalar.bir kodun ram e alınması demek aslında çalışması demektir.

466 byte

2 byte offset

44 byte : partition table : bu bölümde en fazla 4 alana izin verilir yani bu sistemde en fazla 4 partition a izin verilir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*GPT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[mbr512 byte][....][primary part][| | | |(ıso 9660 standardına göreparçalanır.)] --> aslında halen 4 e bölebiliriz.

4 ten fazla bölümlendirmek için Extended Partition kullanılır.Bir extended partition yaratılıp, bu bölümü kendi içinde parçalara ayırabiliyoruz. Bu bölümlerden herbirine lojical partion denir.

EFI-BIOS: makineyi boot edecek kod , yine bios a yazılır.(diske yazılmaz.)GPT yi de destekler.

\*\* MBR 4 e bölümlenebilir.eskidir ama halen kullanımdadır.

.primary 1 den 4 e kadar numara alır.

.extended 4 ten 8 e kadar numara alır.seçince; formatlamay nerden başlayacaksın diye sorar first sector.

-genelde diskteki veriler dış kısımdan bozulur.orta kısımda ise hızlı okunacak veriler vardır.

fdisk in güzel yanı kaydet demediğimiz sürece gidip diske birşey yazmaz.

not: betik yazarak diski parçalıycaksan parted komutunu kullan.

şimdiye kadar bölümlendirdik ;

Artık formatlıyoruz:

-mkfs -t ext4 /dev/sd1 :mkfs.ext4 şeklinde kullanılıda vardır. (fat32 nin karşılıgı burda vfattır.)

-Mount ediyoruz.:(kalıcı mount) Bilgisayarı kapatıp açtığımızda elle yapılan mountlar uçar kalıcı olması için /etc/fstab a yazarız.

-blkid

/dev/sda5 /home ext4 defaults 0 2 (0 : dump ile backup alınıp alınmama ,2 :taranma sırarsı , genelde 1 tane 1 çokça 2 olur. amakine her açıldığında bu dosya taranır bu yüzden kalıcı olarak bağlanır.)

gdiskin kullanımı fdiskle aynıdır.

-gdisk /dev/sda1

-gpt de extended diye birşey yoktur.onun yerine GPT formatında tekrar dizer(2 tane partition alanı tutar)ilk 34 sektor basşta 34 sektorde sonda bulunur.(Sondaki baştakinin yedeğidir.)

gdiskte fdisk gibi ramde çalışır.

\*\*\*\*BAGLAMA SEÇENEKLERİ\*\*\*\*\*

defaults (fstab a yazmıştık)

nodev

exec

synk

auto ... gibi bu tabloadki herşey mount komutuna parametre olarak verilebilir.

\*\*\*\*\*\*\*\*\*\*TAKAS ALANINA NASIL MUDAHALE EDERİZ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.disk bölümünü yarat

2.formatla(mkswp ile formatlanır.)mkswp /dev/sda5

-free komutu boşalanları gösteriyordu.

-swapon /dev/sda5 ile alanı büyüyebilir.(takas alanı olarak kullanıma açabiliriz.)

-swapoff /dev/sda5 takas alanını kapat.

-cat /proc/swaps tan takas alanlarını görüntüleyebiliriz.

-takas alanı olarak bir dosyada kullanabiliriz.

512 mb büyütecaz mesela 512 mblık bir dosya oluşturacagız naparız.

-cat /dev/zero > /dev/sda harddiske 0 yazdık. Bu komutu 27 kez çalıştırırsak veriyi yok edebiliriz.mikroskobik olarak bile görünemeyecek şekilde, yok edilir.yada çok çok yüksek sıcaklıga atmamız gerekir.

\*\*\*TEMPEST:bütün elektronik cihazlar, anlamlı parazit yayını yapar.Casuslar bu yayını yakalar.türk bakanlar mal gibi bu şekilde ülke ile ilgili bilgileri ifşa ediyorlar.

-dd if=/dev/zero of=dosya ismi (if:input file of:outputfile(seçimli)); nerden girdi okuyacağı vermezen statndart girdiden okur.

(veri kopyalama tooludur aynı zamanda.)

-dd if=/dev/vda of=/takas -->/dev/vda imajının alınması.

-dd if /dev/zero of=/takas bs=1m count=512 (1m lık bloklar dan 512 kere okur) -->yani 512 mb lık dosya oluşmuş olur.

-dd bit bit okur.bs ile boyutunu vermezsek bu yüzden çok uzun sürer.

-mkswap /takas içindeki veriyi swap alanı olabilecek şekilde getirir.

-swapon /takas swap alanı yapar.

-cat /proc/swaps ile bakabiliriz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LVM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fiziksel disklerle çalışırken, boyutu aşmamak için kullanılan yazılım.Fiziksel disklerle, işletim sistemi sistemi arasında, mantıksal katman oluşturmak için, Fiziksel bir diskin snapshot ını mesela LVM olmadan alamayız.

-partprobe: Çekirdeğin partition tablolarını yeniden okumasını sağlar.(Kendi içindede; çekirdek partition tablolarını tutar.Partition tablosu uçsa bile çekirdektede disk tablosu bulunduğu için sistem bir süre daha ayakta kalır.)

-fiziksel diskler bir havuza atılır.(birden fazla da olabilir.ama bir tane olması tercih edilir.Sonra havuzdan küçük hacimler alınır.)

1.pvcreate /dev/sda2 :lvm katmanı olabilecek şekilde formatlar.fdisk ile belirli bölümler oluşturmuştuk.şimdi mantıksal bölümler ekliyoruz.

2.vgcreate :volume grubu oluşturur. vgcreate havuz /dev/sda2/ /dev/sda3 (havuzu besliyecek diskler)

-vgdisplay havuzun durumunu gösterir.

Total PE alanı:766 (pe:fiziksel eklenti) ->mantıksal alanlar oluşturulurken buna dikkat edilir.4mb lık extendlere bölünmüş durumda.

--mantıksal bölüm oluşturma: lvcreate -L 1G -n BOLUM1 havuz (-n:isim ,L:alan)

-lvdisplay

\*shutdown -f ile kapatırsak; bir sonraki açılışta, sistem taranacak şekilde kapanır.

-lvdisplay

-mksf ile oluşturduğumuz diski formatlayabiliriz.

-mount /etc/fstab

df -h ile ;baglanan disklerin dolulugunu ne kadarının dolu ne kadarının boş olduğunu bunla görebilriz.

dh -ih-->inode u ile birlikte verir.

-dosya sistemin formatlarken blok size seçilebilir.default 4 kb bloksize ı küçülttüp atıl alanıa azaltabiliriz.Ama performans düşer , veriye ulaşma zamanımız artar. Blok en küçük verinin tutulacağı alandır.(film falan atarken blok boyutunu büyük tutarsak daha hızlı atar.).Dosya sistemine veriler yazılırken blok blok yazılır. aynı blogu birden fazla dosya kullanamz. Ama solaris in xfs dosya sistemi bunu sağlar.(atıl alanı azltmak için)

-sizeon disk ve gerçek dosya boyutu diye 2 farklı kavram vardır.Dizinler 1 blok kadar yer tutar dizinlerin içini doldurduk,yer kalmadı.1 blok daha alınır.

-dizinler birer dosyadır peki dizin agacı nasıl oluşur.

\*\*\*\*her eleman ben kimi ve kime baglıyım bilgisi tutarsa dizin agacı oluşur.Dosya sistemini dzin agacı haline getiren

.

.. Dosyalarıdır./ u çagırdığımızda aslında . dosyası kullanılır.

Bu yüzden hardlinklerin agac oluşturabilmesi için mecburidir.

not:dizinler içinde bulunan dosyalar hakkında bilgi veren dosyalardır.

not:adam wordde belge yazdı gönderdi dosyayı açıp okumayı bilen kişi nerden neyi kopyaladığı,nerden yolladığı falan bir sürü veri toplatayabilir.Bu yüzden dosyayı ananymus yapmak gerekir.

-lvextend -L +100M /dev/havuz/ b1 :varsa extra alan ekler.

-resize2fs /dev/Havuz/ BOLUM1 :DİSKİ EXTEND ETTİK BÜYÜTMEYİ YAPMASI İÇİN DİSKİ DÜRTÜYORUZ.herhangi bir kesinti kapanma olmadan online yapılır.

-dosya sisteminde hata çıktı,tarama yapmak için fsck kullanılır.

-fsck -t ext4 -cvFdf

not: mountunun çözemediğimiz tek disk / tür.umount / olmaz.

-fsck:her dosyanın altında lost+found adında bir klasor oluşur.Kaybolmuş dosyaları ne yapılacagı bilinmediğinden böyle bir klasorun altına atar.Bu dosyayı silip fsck çalışıtırırsak örnegin dosya sistemi bozulur.

-fsck -f ext4 -cVFdfa /dev/HAVUZ/ bolum1 (a:assume YES hertürlü düzeltme operasyonunda onay beklerken yesle devam eder.büyük dosya sisteminde çalışıyorsa kullanmakta fayda vardır.küçük sistemlerde hatanın ne olduğunu görebilmek adına kullanmayabiliriz.)

-fdisk -l :disklerle ilgili bilgi veriyordu.

-Genişleme ihtimali varsa LVM kullanmak en mantıklısıdır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dosya sistemi, bir bilgisayarda verilerin ne şekilde tutulması gerektiğini ayarlayan bir sistemdir. Farklı işletim sistemleri farklı dosya sistemleri ile çalışabilirler (Örn. Windows XP NTFS kullanırken, Fedora 11’de ext4 kullanılabilir). Bir dosya sisteminin Linux’ta çalışması Linux çekirdeğinin desteklemesi gerekmektedir.

**Neden ext4?**

2.6.28 çekirdeğinden itibaren çoğu Linux dağıtımında varsayılan olarak ext4 dosya sistemi kullanılmaktadır. ext4 dosya sistemi, ext2 ve ext3 dosya sistemlerini temel alarak geliştirilmiştir.

**Journalling (Günlükleme)**

İşletim sisteminin düzgün kapanamadığı durumlarda, sistemin ilk açılışında e2fsck aracı dosya sistemi tutarlılığı için tarama yapar. Ancak bu işlem uzun bir zaman aldığından sistemin uzun bir süre hizmet dışı kalmasına yol açar. Journalling özelliği ile birlikte dosya sistemi üzerinde yapılan her işlem kayıt altına alınır ve bu kayıtlar Journal (günlük)’a yazılır. Bu gibi durumlarda journal’ın kontrol edilmesi çok daha kısa süre alır ve yeterlidir. Bu özellik ext3 ile gelmiş ve ext4 ile geliştirilmiştir.

**Veri Bütünlüğü ve Hız**

Journaling özelliği sayesinde disk okuma ve yazma işlemleri daha hızlı gerçekleşir ve sistemin düzgün kapanmadığı durumlarda yüksek veri güvenliği sağlanır. ext4'te gelen journal checksum özelliği sayesinde journal daha hızlı kontrol edilir.

**Kolay Geçiş**

ext3 dosya sisteminden ext4'e biçimlendirme yapmadan kolayca geçiş yapılabilir. Ayrıca ext3 dosya sistemleri ext4 olarak bağlanarak da çalıştırılabilir.

**Dosya-Dizin Sınırları**

Ext3 dosya sisteminde en yüksek dosya boyutu 2 TB, en yüksek dosya sistemi boyutu 16TB iken; ext4 ile en yüksek dosya boyutu 16 TB, en yüksek dosya sistemi boyutu 1024 PB (1 EB)'tır. Ayrıca ext3 dosya sisteminde bir dizin altında en fazla 32000 altdizin oluşturulabilirken, ext4 ile bu sınır 64000'e çıkarılmıştır.

**fdisk**

fdisk aracı bölümleme yönetme komutlarının başında gelir. Bu komutu kullanarak dosya sistemi bölümlendirilebilir, boyutları ayarlanabilir ve var olan bölümlendirmeler listelenebilir ve silinebilir.

**fdisk [-l] aygıt\_adı**

Dosya sistemi üzerindeki bölümlendirmeleri listelemek için fdisk komutuna “l” anahtarı verilir.

**[root@ila ~]# fdisk -l**

Disk /dev/hdc: 41.1 GB, 41110142976 bytes

255 heads, 63 sectors/track, 4998 cylinders

Units= cylindersof 16065 \* 512 = 8225280 bytes

Device BootStart EndBlocksIdSystem

/dev/hdc1 \* 1 2550 20482843+ 83 Linux

/dev/hdc2 2551 2681 1052257+ 82 Linux swap / Solaris

/dev/hdc3 2682 2806 1004062+ c W95 FAT32 (LBA)

/dev/hdc4 2807 4998 17607240 5 Extended

/dev/hdc5 2807 2819 104391 86 NTFS volumeset

fdisk komutuna parametre olarak bölümleme yapmak istenen disk ismi verilerek komut ile etkileşimli moda geçilir.

**[root@ila ~]# fdisk /dev/sdb**

The number of cylinders for this disk is set to 6666.

There is nothing wrong with that, but this is larger than 1024,-Harddisk i aldık.bilgisayara taktık naparız?Biçimlendirme.(windostaki gibi)Biçimlendirerek aslında disk üzerinde dosya sistemi oluşturuyoruz.Harddisk 0 ve 1 lerden oluşan bir aygıttır.Bunalrın anlamlı olması gerekir.BU anlamalandırmada kullanılan protokele dosya sistemi denir.işletim sistemi bir dosyanın disk üzerinde nasıl tutulacagına bu protokele bakarak karar verir.Windows NTFS den anlar.Wİndows Nt zamanında çıkmıştır.Revizyonlar yapılıp yapılmadığını private bir format olduğunda bilmiyoruz.Saklandığı için Windows dışında kimse kullanamaz.(linux da)ntfs için linux tarafında reverse engineering yapılmışıt.(ürüne bakarak bunun aynısı bizde yaparız abi denilerek çıkmıştır yola)Redhat, Centos ntfs dosya sistemini okumaz bile.

Lİnux ext kullanır.

FAT32 yide windows çıkarmıştır ve herkes desteklemiştir. Çünkü publis olmuştur.

fat32 usbler için standartlaşmıştır ve fat32 de 2 gb dan büyük data tutulamaz .2 gb dan büyük veri tutabilmek için extendenFAT(extFAT)kullanılır.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*EXT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*(extended file system)

ext2,ext3,ext4 gibi veriyonlar oluşmuştur.(sınırlar genişlemiştir.)

ext2 ile ext3 arasında formatlamadan geçiş yapılabilir.(journaling)

fat32 journaling dosya sistemi değildir.

ext4 ten ext3 e yani tersine bir geçiş yoktur.

(Disklerle ilgili yapılan denemeler sanal makinede yapılmalıdır.YANlış bir harekette makine çöker.)

-fdisk -l : makinemizdeki disklerle ilgili alanları listeler.(sanal disk alanı /dev/vda dır.)

-windows ta formatlamadan önce bölümlüyoruz.Bu komutun çıktısından da bu şekilde devam ederiz.

MBR(MASTER boot record)-> fdisk geçerli(ilk 512 byte bilgisayarı boot etmek için kullanır.)

GPT(GUID partition)->yenidir.gdisk

örn 4tb lık harddisk alıp,böldük,kullandık(tek bölüm olarak kullanmamanın avantajı,bakım süresi azalır.(tarama gibi))

-fdisk /dev/sda ile hangi diskle oynicaksak bunun partition table ına ulaşabiliriz.

p ye basarak görebiliriz.

n;yeni bir partition oluşturur.

bundan sonra primary mi extension mu diye sorar.

not:bilgisayar açılınca gidip,mbr nin ilk 466 byte nı alıp ram e kopyalar.bir kodun ram e alınması demek aslında çalışması demektir.

466 byte

2 byte offset

44 byte : partition table : bu bölümde en fazla 4 alana izin verilir yani bu sistemde en fazla 4 partition a izin verilir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*GPT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[mbr512 byte][....][primary part][| | | |(ıso 9660 standardına göreparçalanır.)] --> aslında halen 4 e bölebiliriz.

4 ten fazla bölümlendirmek için Extended Partition kullanılır.Bir extended partition yaratılıp, bu bölümü kendi içinde parçalara ayırabiliyoruz. Bu bölümlerden herbirine lojical partion denir.

EFI-BIOS: makineyi boot edecek kod , yine bios a yazılır.(diske yazılmaz.)GPT yi de destekler.

\*\* MBR 4 e bölümlenebilir.eskidir ama halen kullanımdadır.

.primary 1 den 4 e kadar numara alır.

.extended 4 ten 8 e kadar numara alır.seçince; formatlamay nerden başlayacaksın diye sorar first sector.

-genelde diskteki veriler dış kısımdan bozulur.orta kısımda ise hızlı okunacak veriler vardır.

fdisk in güzel yanı kaydet demediğimiz sürece gidip diske birşey yazmaz.

not: betik yazarak diski parçalıycaksan parted komutunu kullan.

şimdiye kadar bölümlendirdik ;

Artık formatlıyoruz:

-mkfs -t ext4 /dev/sd1 :mkfs.ext4 şeklinde kullanılıda vardır. (fat32 nin karşılıgı burda vfattır.)

-Mount ediyoruz.:(kalıcı mount) Bilgisayarı kapatıp açtığımızda elle yapılan mountlar uçar kalıcı olması için /etc/fstab a yazarız.

-blkid

/dev/sda5 /home ext4 defaults 0 2 (0 : dump ile backup alınıp alınmama ,2 :taranma sırarsı , genelde 1 tane 1 çokça 2 olur. amakine her açıldığında bu dosya taranır bu yüzden kalıcı olarak bağlanır.)

gdiskin kullanımı fdiskle aynıdır.

-gdisk /dev/sda1

-gpt de extended diye birşey yoktur.onun yerine GPT formatında tekrar dizer(2 tane partition alanı tutar)ilk 34 sektor basşta 34 sektorde sonda bulunur.(Sondaki baştakinin yedeğidir.)

gdiskte fdisk gibi ramde çalışır.

\*\*\*\*BAGLAMA SEÇENEKLERİ\*\*\*\*\*

defaults (fstab a yazmıştık)

nodev

exec

synk

auto ... gibi bu tabloadki herşey mount komutuna parametre olarak verilebilir.

\*\*\*\*\*\*\*\*\*\*TAKAS ALANINA NASIL MUDAHALE EDERİZ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.disk bölümünü yarat

2.formatla(mkswp ile formatlanır.)mkswp /dev/sda5

-free komutu boşalanları gösteriyordu.

-swapon /dev/sda5 ile alanı büyüyebilir.(takas alanı olarak kullanıma açabiliriz.)

-swapoff /dev/sda5 takas alanını kapat.

-cat /proc/swaps tan takas alanlarını görüntüleyebiliriz.

-takas alanı olarak bir dosyada kullanabiliriz.

512 mb büyütecaz mesela 512 mblık bir dosya oluşturacagız naparız.

-cat /dev/zero > /dev/sda harddiske 0 yazdık. Bu komutu 27 kez çalıştırırsak veriyi yok edebiliriz.mikroskobik olarak bile görünemeyecek şekilde, yok edilir.yada çok çok yüksek sıcaklıga atmamız gerekir.

\*\*\*TEMPEST:bütün elektronik cihazlar, anlamlı parazit yayını yapar.Casuslar bu yayını yakalar.türk bakanlar mal gibi bu şekilde ülke ile ilgili bilgileri ifşa ediyorlar.

-dd if=/dev/zero of=dosya ismi (if:input file of:outputfile(seçimli)); nerden girdi okuyacağı vermezen statndart girdiden okur.

(veri kopyalama tooludur aynı zamanda.)

-dd if=/dev/vda of=/takas -->/dev/vda imajının alınması.

-dd if /dev/zero of=/takas bs=1m count=512 (1m lık bloklar dan 512 kere okur) -->yani 512 mb lık dosya oluşmuş olur.

-dd bit bit okur.bs ile boyutunu vermezsek bu yüzden çok uzun sürer.

-mkswap /takas içindeki veriyi swap alanı olabilecek şekilde getirir.

-swapon /takas swap alanı yapar.

-cat /proc/swaps ile bakabiliriz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LVM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fiziksel disklerle çalışırken, boyutu aşmamak için kullanılan yazılım.Fiziksel disklerle, işletim sistemi sistemi arasında, mantıksal katman oluşturmak için, Fiziksel bir diskin snapshot ını mesela LVM olmadan alamayız.

-partprobe: Çekirdeğin partition tablolarını yeniden okumasını sağlar.(Kendi içindede; çekirdek partition tablolarını tutar.Partition tablosu uçsa bile çekirdektede disk tablosu bulunduğu için sistem bir süre daha ayakta kalır.)

-fiziksel diskler bir havuza atılır.(birden fazla da olabilir.ama bir tane olması tercih edilir.Sonra havuzdan küçük hacimler alınır.)

1.pvcreate /dev/sda2 :lvm katmanı olabilecek şekilde formatlar.fdisk ile belirli bölümler oluşturmuştuk.şimdi mantıksal bölümler ekliyoruz.

2.vgcreate :volume grubu oluşturur. vgcreate havuz /dev/sda2/ /dev/sda3 (havuzu besliyecek diskler)

-vgdisplay havuzun durumunu gösterir.

Total PE alanı:766 (pe:fiziksel eklenti) ->mantıksal alanlar oluşturulurken buna dikkat edilir.4mb lık extendlere bölünmüş durumda.

--mantıksal bölüm oluşturma: lvcreate -L 1G -n BOLUM1 havuz (-n:isim ,L:alan)

-lvdisplay

\*shutdown -f ile kapatırsak; bir sonraki açılışta, sistem taranacak şekilde kapanır.

-lvdisplay

-mksf ile oluşturduğumuz diski formatlayabiliriz.

-mount /etc/fstab

df -h ile ;baglanan disklerin dolulugunu ne kadarının dolu ne kadarının boş olduğunu bunla görebilriz.

dh -ih-->inode u ile birlikte verir.

-dosya sistemin formatlarken blok size seçilebilir.default 4 kb bloksize ı küçülttüp atıl alanıa azaltabiliriz.Ama performans düşer , veriye ulaşma zamanımız artar. Blok en küçük verinin tutulacağı alandır.(film falan atarken blok boyutunu büyük tutarsak daha hızlı atar.).Dosya sistemine veriler yazılırken blok blok yazılır. aynı blogu birden fazla dosya kullanamz. Ama solaris in xfs dosya sistemi bunu sağlar.(atıl alanı azltmak için)

-sizeon disk ve gerçek dosya boyutu diye 2 farklı kavram vardır.Dizinler 1 blok kadar yer tutar dizinlerin içini doldurduk,yer kalmadı.1 blok daha alınır.

-dizinler birer dosyadır peki dizin agacı nasıl oluşur.

\*\*\*\*her eleman ben kimi ve kime baglıyım bilgisi tutarsa dizin agacı oluşur.Dosya sistemini dzin agacı haline getiren

.

.. Dosyalarıdır./ u çagırdığımızda aslında . dosyası kullanılır.

Bu yüzden hardlinklerin agac oluşturabilmesi için mecburidir.

not:dizinler içinde bulunan dosyalar hakkında bilgi veren dosyalardır.

not:adam wordde belge yazdı gönderdi dosyayı açıp okumayı bilen kişi nerden neyi kopyaladığı,nerden yolladığı falan bir sürü veri toplatayabilir.Bu yüzden dosyayı ananymus yapmak gerekir.

-lvextend -L +100M /dev/havuz/ b1 :varsa extra alan ekler.

-resize2fs /dev/Havuz/ BOLUM1 :DİSKİ EXTEND ETTİK BÜYÜTMEYİ YAPMASI İÇİN DİSKİ DÜRTÜYORUZ.herhangi bir kesinti kapanma olmadan online yapılır.

-dosya sisteminde hata çıktı,tarama yapmak için fsck kullanılır.

-fsck -t ext4 -cvFdf

not: mountunun çözemediğimiz tek disk / tür.umount / olmaz.

-fsck:her dosyanın altında lost+found adında bir klasor oluşur.Kaybolmuş dosyaları ne yapılacagı bilinmediğinden böyle bir klasorun altına atar.Bu dosyayı silip fsck çalışıtırırsak örnegin dosya sistemi bozulur.

-fsck -f ext4 -cVFdfa /dev/HAVUZ/ bolum1 (a:assume YES hertürlü düzeltme operasyonunda onay beklerken yesle devam eder.büyük dosya sisteminde çalışıyorsa kullanmakta fayda vardır.küçük sistemlerde hatanın ne olduğunu görebilmek adına kullanmayabiliriz.)

-fdisk -l :disklerle ilgili bilgi veriyordu.

-Genişleme ihtimali varsa LVM kullanmak en mantıklısıdır.

and could in certain setups cause problems with:

1) software that runs at boot time (e.g., old versions of LILO)

2) booting and partitioning software from other OSs

(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):

Etkileşimli modda yapılabilicek işlemlerin listesini görmek için m komutu kullanılabilir.

Etkileşimli modda yapılan işlemlerin hiçbiri w komutu kullanılmadığı sürece diske yazılmaz. Böylece disk üstüne yapılacak tüm işlemler diske aktarılmadan kontrol edilebilir ve hata olması durumunda sistemi bozmadan değiştirilebilir.

Sık kullanılan fdisk komutları ve kullanım amaçları aşağıdaki gibidir:

• d - Bölümleme silme

• n - Yeni bir bölümleme oluşturma.

• p - Bölümleme tablosunu ekrana yazdırma.

• q - Değişiklikleri kaydetmeden çıkma.

• t - Bölümleme tipini değiştirme.

• w - Değişiklikleri kaydedip çıkma.

p ile ekrana bölümleme tablosu yazdırılır. Eğer bölümleme tablosunda bir değişiklik yapılmadıysa

kullanılan tabloyu gösterir. Eğer fdisk komutunu çalıştırdıktan sonra tabloda bir değişiklik yapıldıysa

bu değişklikleri de gösterir. Yapılan değişiklikler w komutu ile çıkıldığı takdirde diske yazılır. q ile çıkılırsa bu değişiklikler diske yazılmadan fdisk komutundan çıkılmış olunur.

**Yeni Bölümlendirme Oluşturma**

Yeni bölümlendirme oluşturmak için etkileşimli modda “n” seçeneği kullanılır.

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 2

First cylinder (14-652, default 14):

Using default value 14

Last cylinder or +size or +sizeM or +sizeK(14-77, default 77): +50M

Command (m for help):

İlk önce dosya sisteminin birincil mi yoksa genişletilmiş bir dosya sistemi mi olduğunu sorar. Intel mimarisindeki (ev bilgisayarları ve çoğu sunucu sistemi) bir disk üstünde en fazla 4 tane birincil (primary) bölümleme oluşturulabilecek şekilde tasarlanmıştır. 4'den fazla bölüm oluşturmak için bir adet genişletilmiş (extended) bölüm oluşturmak gerekir. Genişletilmiş bölüm içinde de istenildiği kadar mantıksal (logical) bölümlemeler oluşturulabilir.

Eğer birincil veya genişletilmiş dosya sistemi oluşturulacaksa, sonraki aşamada bölümleme numarası girilir, kullanılmayan bir numara girilmelidir. Bu numara bölümleme adlandırmasında diskten sonra gelen rakamdır. Örn /dev/sda2

Sonraki aşamada başlangıç silindir numarası girilir. [ENTER] tuşuna basılarak geçilirse varsayılan değer seçilmiş olur.

Sonraki aşamada bölümlemenin sonlanacağı son silindir numarası girilir. Burada da [ENTER] tuşu ile varsayılan değer seçilebilir. Ayrıca + işaretini kullanarak gigabyte, megabyte veya kilobyte biriminde alanın boyutu belirtilebilir. Örn. +1G, +50G veya +1500K

**Bölümlendirme Tipini Değiştirme**

Bölümlendirme tipini değiştirmek için fdisk’in etkileşimli modunda “t” seçeneği kullanılır.

Command (m for help): p

Disk /dev/hdc: 41.1 GB, 41110142976 bytes

255 heads, 63 sectors/track, 4998 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/hdc1 \* 1 2550 20482843+ 83 Linux

/dev/hdc2 2551 2681 1052257+ 82 Linux swap / Solaris

/dev/hdc3 2682 2806 1004062+ 83 Linux

/dev/hdc4 2807 4998 17607240 5 Extended

/dev/hdc5 2807 2819 104391 86 NTFS volume set

Command (m for help): t

Partition number (1-5): 3

Hex code (type L to list codes): 83

Command (m for help): p

Disk /dev/hdc: 41.1 GB, 41110142976 bytes

255 heads, 63 sectors/track, 4998 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/hdc1 \* 1 2550 20482843+ 83 Linux

/dev/hdc2 2551 2681 1052257+ 82 Linux swap / Solaris

/dev/hdc3 2682 2806 1004062+ 83 Linux

/dev/hdc4 2807 4998 17607240 5 Extended

/dev/hdc5 2807 2819 104391 86 NTFS volume set

**Diskten Bölümleme Silme**

Bölümlendirme silmek için fdisk’in etkileşimli modunda “d” seçeneği kullanılır.

Command (m for help): d

Partition number (1-5): 3

Command (m for help): p

Disk /dev/hdc: 41.1 GB, 41110142976 bytes

255 heads, 63 sectors/track, 4998 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/hdc1 \* 1 2550 20482843+ 83 Linux

/dev/hdc2 2551 2681 1052257+ 82 Linux swap / Solaris

/dev/hdc4 2807 4998 17607240 5 Extended

/dev/hdc5 2807 2819 104391 86 NTFS volume set

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table.

The new table will be used at the next reboot.

Syncing disks.

Eğer işletim sisteminin kurulu olduğu bölümlemede değişiklik yapılırsa çıkışta uyarı alınır.

**Gdisk**

GPT, UEFI BIOS yapısıyla beraber gelen bölüm tablosu (partition table) yapısıdır. MBR'daki yapının eksiklerini gidermek için geliştirilmiştir. MBR 4'e kadar bölüm yaratmaya izin verirken, GPT 128'e kadar yaratabilir. Bir MBR tablosunda yaratılan bölüm maksimum 2 terabayt olabilirken, bir GPT bölümü 8 zetabayt olabilir. GPT'nin çalışması için bilgisayarın BIOS'unun UEFI olmasına gerek yoktur.

GPT tablosu, diskin başında ve sonunda olmak üzere 2 yerde bulunur. Diskin başındaki ana tablo iken, sonundaki yedeklilik içindir. GPT tablosunun ilk birimi MBR tablosudur ve bu birim koruyucu MBR (protective MBR) olarak isimlendirilir. MBR yapısına geriye dönük uyumluluk sağlar.

Gdisk veya diğer adıyla GPT fdisk yazılımı, GPT diskleri üzerinde işlem yapmaya yarar. Yüklü değilse aşağıdaki şekilde yüklenebilir.

**[root@ila ~]# yum install gdisk**

Gdisk yazılımıyla bölüm yapısı kaydedildiğinde MBR bölüm yapısını otomatik olarak GPT'ye dönüştürür. Bu işlemi yapabilmesi için diskin başında ve sonunda en az 17KB (34 sektör) kadar boş alan bulunması gerekir. Aşağıdaki örnekte yaklaşık 20 milyon boş sektör vardır.

**[root@ila ~]# fdisk –l**

Disk /dev/vda: 16.1 GB, 16106127360 bytes

16 heads, 63 sectors/track, 31207 cylinders, total 31457280 sectors

Units = sectors of 1 \* 512 = 512 bytes

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

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

Disk identifier: 0x000ba1b0

Device Boot Start End Blocks Id System

/dev/vda1 \* 2048 1026047 512000 83 Linux

/dev/vda2 1026048 9218047 4096000 83 Linux

/dev/vda3 9218048 11319295 1050624 82 Linux swap / Solaris

Gdisk yazılımı açıldığında bölüm tablosunun ne tipte olduğunu gösterecektir. Gdisk yazılımının komut yapısı fdisk yazılımına benzerdir.

Aşağıdaki örnekte /dev/vda diski MBR yapısına sahiptir.

**[root@ila ~]# gdisk /dev/vda**

GPT fdisk (gdisk) version 0.8.4

Partition table scan:

MBR: MBR only

BSD: not present

APM: not present

GPT: not present

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Found invalid GPT and valid MBR; converting MBR to GPT format.

THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by typing 'q' if

you don't want to convert your MBR partitions to GPT format!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Command (? for help):

**Yeni Bölüm Oluşturma**

Bölüm yaratmak için için gdisk’in etkileşimli modunda “n” seçeneği kullanılır.

Ardından partition number (bölüm numarası) belirtilir.

Bölüm numarasından sonra yeni yaratılacak bölümün başlayacağı ilk sektör belirtilir. [ENTER] tuşuna basılarak geçilirse varsayılan değer seçilmiş olur.

Ardından bölümün sonlanacağı sektör belirtilir. Bunun yerine başında + ile boyut bilgisi belirtilebilir. Örnek: +10G. Kilobayt için K, Megabayt için M, Gigabayt için G karakteri kullanılmalıdır.

Sonrasında bölüm için hex kodu veya GUID bilgisi girilir. Bölüm ne için kullanılacaksa ona ait hex kodu (veya GUID) girilmelidir. Örneğin NTFS olarak formatlanacaksa 0700 girilmelidir. GUID girilmek isteniyorsa, GUID'lerin listesi buradan incelenebilir.

Command (? for help): n

Partition number (4-128, default 4): 4

First sector (34-31457246, default = 11319296) or {+-}size{KMGTP}:

Last sector (11319296-31457246, default = 31457246) or {+-}size{KMGTP}: +500M

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, Enter = 8300): L

0700 Microsoft basic data 0c01 Microsoft reserved 2700 Windows RE

4200 Windows LDM data 4201 Windows LDM metadata 7501 IBM GPFS

7f00 ChromeOS kernel 7f01 ChromeOS root 7f02 ChromeOS reserved

8200 Linux swap 8300 Linux filesystem 8301 Linux reserved

8e00 Linux LVM a500 FreeBSD disklabel a501 FreeBSD boot

a502 FreeBSD swap a503 FreeBSD UFS a504 FreeBSD ZFS

a505 FreeBSD Vinum/RAID a800 Apple UFS a901 NetBSD swap

a902 NetBSD FFS a903 NetBSD LFS a904 NetBSD concatenated

a905 NetBSD encrypted a906 NetBSD RAID ab00 Apple boot

af00 Apple HFS/HFS+ af01 Apple RAID af02 Apple RAID offline

af03 Apple label af04 AppleTV recovery af05 Apple Core Storage

be00 Solaris boot bf00 Solaris root bf01 Solaris /usr & Mac Z

bf02 Solaris swap bf03 Solaris backup bf04 Solaris /var

bf05 Solaris /home bf06 Solaris alternate se bf07 Solaris Reserved 1

bf08 Solaris Reserved 2 bf09 Solaris Reserved 3 bf0a Solaris Reserved 4

bf0b Solaris Reserved 5 c001 HP-UX data c002 HP-UX service

ef00 EFI System ef01 MBR partition scheme ef02 BIOS boot partition

fd00 Linux RAID

Hex code or GUID (L to show codes, Enter = 8300): 8e00

Changed type of partition to 'Linux LVM'

Command (? for help): p

Disk /dev/vda: 31457280 sectors, 15.0 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): 2AF9D6E8-A229-45E5-B710-27DCE9D51FC7

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 31457246

Partitions will be aligned on 2048-sector boundaries

Total free space is 19115965 sectors (9.1 GiB)

Number Start (sector) End (sector) Size Code Name

1 2048 1026047 500.0 MiB 8300 Linux filesystem

2 1026048 9218047 3.9 GiB 8300 Linux filesystem

3 9218048 11319295 1.0 GiB 8200 Linux swap

4 11319296 12343295 500.0 MiB 8E00 Linux LVM

l be used at the next reboot.

Syncing disks.

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING

PARTITIONS!!

Do you want to proceed? (Y/N): y

OK; writing new GUID partition table (GPT) to /dev/vda.

Warning: The kernel is still using the old partition table.

The new table will be used at the next reboot.

The operation has completed successfully.

**Bölüm Tipini Değiştirme**

Bölüm tipini için gdisk’in etkileşimli modunda “t” seçeneği kullanılır.

Ardından işlem yapılacak bölüm numarası (partition number) seçilir.

Sonrasında bölüm için hex kodu veya GUID bilgisi girilir. Bölüm ne için kullanılacaksa ona ait hex kodu (veya GUID) girilmelidir. Örneğin NTFS olarak formatlanacaksa 0700 girilmelidir. GUID girilmek isteniyorsa, GUID'lerin listesi buradan incelenebilir.

Command (? for help): t

Partition number (1-4): 4

Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, Enter = 8300): L

0700 Microsoft basic data 0c01 Microsoft reserved 2700 Windows RE

4200 Windows LDM data 4201 Windows LDM metadata 7501 IBM GPFS

7f00 ChromeOS kernel 7f01 ChromeOS root 7f02 ChromeOS reserved

8200 Linux swap 8300 Linux filesystem 8301 Linux reserved

8e00 Linux LVM a500 FreeBSD disklabel a501 FreeBSD boot

a502 FreeBSD swap a503 FreeBSD UFS a504 FreeBSD ZFS

a505 FreeBSD Vinum/RAID a800 Apple UFS a901 NetBSD swap

a902 NetBSD FFS a903 NetBSD LFS a904 NetBSD concatenated

a905 NetBSD encrypted a906 NetBSD RAID ab00 Apple boot

af00 Apple HFS/HFS+ af01 Apple RAID af02 Apple RAID offline

af03 Apple label af04 AppleTV recovery af05 Apple Core Storage

be00 Solaris boot bf00 Solaris root bf01 Solaris /usr & Mac Z

bf02 Solaris swap bf03 Solaris backup bf04 Solaris /var

bf05 Solaris /home bf06 Solaris alternate se bf07 Solaris Reserved 1

bf08 Solaris Reserved 2 bf09 Solaris Reserved 3 bf0a Solaris Reserved 4

bf0b Solaris Reserved 5 c001 HP-UX data c002 HP-UX service

ef00 EFI System ef01 MBR partition scheme ef02 BIOS boot partition

fd00 Linux RAID

Hex code or GUID (L to show codes, Enter = 8300): 0700

Changed type of partition to 'Microsoft basic data'

Command (? for help): p

Disk /dev/vda: 31457280 sectors, 15.0 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): 2AF9D6E8-A229-45E5-B710-27DCE9D51FC7

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 31457246

Partitions will be aligned on 2048-sector boundaries

Total free space is 19115965 sectors (9.1 GiB)

Number Start (sector) End (sector) Size Code Name

1 2048 1026047 500.0 MiB 8300 Linux filesystem

2 1026048 9218047 3.9 GiB 8300 Linux filesystem

3 9218048 11319295 1.0 GiB 8200 Linux swap

4 11319296 12343295 500.0 MiB 0700 Microsoft basic data

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING

PARTITIONS!!

Do you want to proceed? (Y/N): y

OK; writing new GUID partition table (GPT) to /dev/vda.

Warning: The kernel is still using the old partition table.

The new table will be used at the next reboot.

The operation has completed successfully.

**Diskten Bölüm Silme**

Bölüm tipini için gdisk’in etkileşimli modunda “d” seçeneği kullanılır.

Ardından silinecek bölüm numarası (partition number) seçilir ve bölüm silinir.

Command (? for help): d

Partition number (1-4): 4

Command (? for help): p

Disk /dev/vda: 31457280 sectors, 15.0 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): 2AF9D6E8-A229-45E5-B710-27DCE9D51FC7

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 31457246

Partitions will be aligned on 2048-sector boundaries

Total free space is 20139965 sectors (9.6 GiB)

Number Start (sector) End (sector) Size Code Name

1 2048 1026047 500.0 MiB 8300 Linux filesystem

2 1026048 9218047 3.9 GiB 8300 Linux filesystem

3 9218048 11319295 1.0 GiB 8200 Linux swap

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING

PARTITIONS!!

Do you want to proceed? (Y/N): y

OK; writing new GUID partition table (GPT) to /dev/vda.

Warning: The kernel is still using the old partition table.

The new table will be used at the next reboot.

The operation has completed successfully.

**mkfs**

fdisk ile bölümlendirme tablosunda yapılan değişikliklerin kullanılabilmesi için dosya sisteminin de oluşturulması gerekir. Bu işlem için mkfs komutu kullanılabilir.

mkfs komutu dosya sisteminin biçimlendirilmesi için kullanılabilir.

mkfs [-V] [-t fstype] [fs-options] device [blocks]

• -V - Dosya sistemi oluşturulurken ekstra bilgi verir.

• -t <fstype> - <fstype> kısmında oluşturulacak dosya sisteminin türü belirtilir. (ext4,vfat)

• device - Dosya sistemini oluşturacağımız aygıt belirtilir (/dev/sda2 veya /dev/hda3). Bu kısımda tamamen diskin ismi verilmez (/dev/sda veya /dev/hda gibi).

• blocks - Dosya sisteminin blok büyüklüğünün ne kadar olacağıdır. mkfs komutu otomatik olarak bu kısmı belirleyebilir.

**[root@ila ~]# mkfs -t ext4 /dev/hdc3**

mke2fs 1.41.12 (17-May-2010)

.

.

.

Writing inode tables: done

Creating journal(4096 blocks): done

Writing super blocks and file system accounting information: done

Thisfilesystemwillbe automatically checked every 23 mounts or

180 days, whichever comes first. Use tune2fs -c or-i to override.

mkfs komutu arkada çalışan bir çok komuta aracılık yapar. Aldığı parametrelere göre gerekli komutu çalıştırır. Yerine göre mke2fs yada mkfs.ext4 gibi diğer komutların kullanılmasını sağlar. Bu komutlar mkfs komutu çalıştırmadan direk çağrılarak da kullanılabilr. Bu şekilde kullanılan komutlar aşağıdaki gibidir.

mke2fs - ext2/ext3/ext4 dosya sistemi oluşturulabilir.

mkfs.ext2 - ext2 dosya sistemi oluşturulabilir.

mkfs.ext3 - ext3 dosya sistemi oluşturulabilir.

mkfs.ext4 - ext4 dosya sistemi oluşturulabilir.

mkfs.vfat/mkfs.msdos - msdos dosya sistemi oluşturulabilir.

mkfs.cramfs - cramfs dosya sistemi oluşturulabilir.

mkfs.ntfs - ntfs dosya sistemi olusturulabilir.

**Dosya Sistemlerinin Bağlanması**

Dosya sistemleri 2 yöntemle Linux dosya hiyerarşisine bağlanabilir.

Bu yöntemlerden biri “/etc/fstab” dosyasını kullanmaktır. Bu dosyada bulunan dosya sistemleri varsayılanda sistemin her açılışında otomatik olarak ilgili bağlama noktalarına bağlanacaktır.

Diğer yöntem ise “mount” komutunu kullanarak elle dosya sistemini bağlamaktır. mount komutu ile bağlanılan dosya sistemleri sistemin yeniden başlatılması durumunda otomatik olarak bağlanmazlar, elle bağlama işleminin yine yapılması gerekir.

**/etc/fstab**

/etc/fstab dosyasının yapısında her satır 6 bloktan oluşmaktadır. Örnek bir dosyanın yapısı ve her

alanın anlamı aşağıdaki gibidir:

**[root@ila ~]# cat /etc/fstab**

LABEL=/ / ext4 defaults 1 1

/dev/devpts /dev/ptsdev pts gid=5,mode=620 0 0

/dev/shm /dev/shm tmpfs defaults 0 0

/dev/proc /proc proc defaults 0 0

/dev/sys /sys sysfs defaults 0 0

/dev/fd0 /media/floppy auto pamconsole,exec,noauto,utf8,managed 0 0

/dev/hda /media/cdrecorder auto pamconsole,exec,noauto,managed 0 0

Aygıt

Bağlanılacak olan diskin aygıt dosyasının yeri, UUID değeri ya da etiketi yazılır.

Bağlama Noktası

Bağlama işleminin yapılacağı dizini belirtir. swap bölümü için bu alanda swap tanımı yer alır.

Dosya Sistemi Tipi

Bağlanılacak aygıtın dosya sistemi yer alır. auto seçeneği ile dosya sisteminin otomatik olarak belirlenmesi sağlanabilir.

Dump Seçeneği

dump komutu ile yedek alınıyorsa bu alanda 0 olması bu bölümün yedeğinin alınmasının gerekmediğini, 1 olması yedeğinin alınması gerektiğini bildirir.

Kontrol Sırası

fsck komutu ile dosya sistemi kontrol ediliyorsa hangi sırada bu sistemin kontrolünün yapılacağını belirtir. Eğer 0 değeri verildiyse bu bölüm kontrol edilmez. / dizini genelde ilk sırada kontrol edildiği için 1 değeri verilir. Geri kalan bölümler için 2 değeri verilebilir.

**Bağlama Seçenekleri**

Bağlama sırasında kullanılacak olan seçenekleri yer alır.

* defaults - Varsayılan seçeneklerdir: rw, suid, dev, exec, auto, nouser, ve async
* sync - Tüm giriş/çıkış işlemleri senkron olarak yapılır. (Yapılan her işlem anında disk üzerinde de yapılır.)
* async - Tüm giriş-çıkış işlemleri asenkron olarak yapılır.
* auto - mount komutuna –a parametresi verilerek diskin bağlanmasına izin verir.
* noauto - Sistemin mount –a komutu ile bağlanmasını engeller.
* dev - Dosya sisteminde aygıt dosyalarının kullanımına izin verir.
* nodev - Dosya sisteminde bulunan aygıt dosyalarının kullanımını engeller.
* exec - Dosya sisteminde bulunan çalıştırılabilir dosyaların çalıştırılmasına izin verir.
* noexec - Dosya sisteminde bulunan çalıştırılabilir dosyaların çalıştırılmasını engeller.
* suid - Kullanıcı veya grup bitleri set edilmiş programların, belirtilen kullanıcı veya grup izinleri ile çalışmasına izin verir.
* nosuid - Kullanıcı veya grup bitleri set edilmiş programlarda bu bitlerin işlevlerini geçersiz kılar.
* user - Normal kullanıcıların da dosya sistemini bağlamasına izin verir.
* nouser - root kullanıcısı dışındaki kullanıcıların dosya sistemini bağlamasını engeller.
* remount - Daha önceden bağlanmış bir dosya sisteminin tekrar bağlanmasını sağlar. Kullanım sebebi bağlanma seçeneklerini değiştirmektir. En çok ro olarak bağlanmış bir sistemin, rw izinleri ile bağlanmasını sağlamakta kullanılır.
* ro - Dosya sistemi salt-okunur olarak bağlanır.
* rw - Dosya sistemi okuma/yazma modunda bağlanır.

**/etc/mtab - /proc/mounts**

Sistemin bağladığı dosya sistemleri mount komutunu seçenek vermeden veya –l seçeneği ile kullanarak görülebilir. Ayrıca bağlanan dosya sistemleri /etc/mtab ve /proc/mounts dosyalarında da bulunur. /proc/mounts dosyasında disklerin dışında bağlanmış tüm aygıtlar görülebilir.

[root@ila ~]# cat /etc/mtab

/dev/hda6 / ext4 rw 0 0

/dev/proc/ proc proc rw 0 0

/dev/sys /sys sysfs rw 0 0

/dev/devpts /dev/pts devpts rw,gid=5,mode=620 0 0

/dev/shm /dev/shm tmpfs rw 0 0

none /proc/sys/fs/binfmt\_misc binfmt\_misc rw 0 0

sunrpc /var/lib/nfs/rpc\_pipefs rpc\_pipefs rw 0 0

**mount**

Disk bölümlerinin içeriğine ulaşmak için bu bölümlerin Linux dosya hiyerarşisinde bir dizine bağlanması gereklidir. Bağlama işlemini yapmak için mount komutu kullanılır.

mount kullanımı ve önemli parametrelerinin anlamları aşağıdaki gibidir:

mount <seçenekler> <bağlanacak\_dosya\_sisteminin\_yeri> <bağlanılacak\_dizin>

-a - /etc/fstab dosyasında auto seçeneği ile tanımlanan tüm dosya sistemlerinin bağlanılmasını sağlayan parametredir.

-l - Bağlanmış olan dosya sistemlerini listeler.

-F - -a parametresi kullanıldığı takdirde kullanılabilir. /etc/fstab’ta tanımlanmış her dosya sistemi için ayrı mount programının çalışmasını sağlar. Böylece bütün dosya sistemleri sırayla bağlanmak yerine paralel olarak bağlanır.

-n - Bu parametre ile dosya sisteminin bağlanma işlemi yapılır fakat /etc/mtab dosyası güncellenmez.

-f - Bu parametre ile dosya sisteminin gerçekten bağlanması dışındaki işlemler gerçekleştirilir. Bu sayede –n parametresi kullanılarak bağlanmış bir dosya sisteminin /etc/mtab’a kaydı yapılabilir.

-t <dosya\_sistem\_tipi> - Bu parametre ile bağlanacak dosya sisteminin tipi belirtilmektedir. Bu parametre verilmediği takdirde mount dosya sistemini kendisi belirlemeye çalışacaktır.

-L <label> - Bu parametre sayesinde belirtilen etikete sahip dosya sistemi bağlanır. Bu parametrenin kullanılabilmesi için /proc/partitions sanal dosyasında gerekli bilgilerin bulunması gerekmektedir.

-o <seçenek> - /etc/fstab dosyasında seçenekler alanında verilebilen değerler bu seçenekte verilebilir.

[root@ila ~]# mount -t ext4 /dev/hdc3 /home/

[root@ila ~]# mount -t vfat /dev/hdc5 /mnt/windows

[root@ila ~]# mount -t smbfs -o 'username=kullanıcı' //192.168.1.15/paylasim /mnt/samba/

[root@ila ~]# mount –t nfs sunucu:dizin /mnt/nfs

[root@ila ~]# mount /dev/cdrom

**umount**

Sisteme bağlanmış bir dosya sisteminin bağlantısını çözmek için umount komutu kullanılır.

umount komutunun kullanımı ve önemli parametrelerinin anlamları aşağıdaki gibidir:

umount <seçenekler> <aygıt | dizin>

• -a - /etc/mtab dosyasındaki bağlantılar koparlır.

• -n - /etc/mtab dosyasını güncellemeden işlemi yapar.

• -r - umount komutunun devreden çıkarma işlemi başarılı olmadığı durumda, dosya sistemini salt-okunur olarak yeniden bağlamayı dener.

• -f - Ulaşılamayan bir NFS dosya sistemi için devreden çıkarma işleminin gerçekleştirilmesini sağlar.

• -t <dosya\_sistem\_tipi> - Devreden çıkarılacak dosya sisteminin tipini belirtmek için kullanılır.

[root@ila ~]# umount /dev/cdrom

[root@ila ~]# umount –a

**df**

df komutu ile sisteme bağlanmış dosya sistemlerinin ne kadar doluluk oranında olduğu öğrenilebilir.

df komutunun sık kullanılan parametreleri ve kullanımı aşağıdaki gibidir:

df <seçenekler> …<dosya>

• -a - Tüm dosya sistemleri gösterilir.

• -h - Boyutlar okunabilir formatta ekrana yazdırılır. 1K, 104M, 16G

• -l - Sadece yerel diskler listelenir.

• -t - Sadece belirtilen dosya sistemleri listelenebilir.

• -T - Dosya sistemi tipi de yazdırılır.

• -i - Diskteki inode doluluk bilgilerini gösterir.

**[root@ila~]# df**

Filesystem1K-blocks Used Available Use% Mounted on

/dev/hdc1 19840892 5177848 13638904 28% /

/dev/shm127808 0 127808 0% /dev/shm

/dev/hdc5 104388 2508 101880 3% /mnt/ntfs

/dev/hdc3 988244 483180 454864 52% /mnt/yedek

[root@ila~]# df -hTt ext4

FilesystemType Size Used Avail Use% Mounted on

/dev/hdc1 ext4 19G 5.0G 14G 28% /

/dev/hdc3 ext4 966M 472M 445M 52% /mnt/yedek

**fsck**

fsck dosya sistemini taramak ve varsa hataları onarmak için kullanılan bir komuttur.

fsck kullanımı ve önemli parametrelerinin anlamları aşağıdaki gibidir.

fsck [-sAVRTNP] [ -C [ fd] ] [ -t fstype] [filesys... ] [--] [ fs-specific-options]

• -A - /etc/fstab dosyasında belirtilen tüm dosya sistemlerinin taranmasını sağlar.

• -C - İşlemin hangi aşamada olduğunu gösteren bir durum çubuğunun çıkmasını sağlar.

• -V - Komutun ayrıntılı çıktısını ekrana yazdırır.

• -N - Yapılması gereken işlemi yapmadan ekrana yazdırır.

• -s - Birden çok dosya sistemi kontrol edilecek ise bu işlemin sırayla yapılmasını sağlar.

• -t <fstype> - Dosya sistemi belirtilir. Normal durumlarda fsck otomatik olarak dosya sistemini anlar ama anlayamadığı durumlarda bu parametre kullanılabilir.

• <filesystems> - Bu alanda kontrolün yapılacağı bölümleme yazılır.

• -a - Bulunan hataların kullanıcıdan onay alınmadan giderilmesini sağlar.

• -r - Tüm işlemler için kullanıcıdan onay alınmasını sağlar.

• -n - Bulunan hataların kullanıcıya sorulmadan yok sayılmasını sağlar.

fsck komutu bağlanmış(mounted) bölümlemelerde kullanılmamalıdır. Bağlanmış dosya sistemlerinde bu komutun kullanılması dosya sistemine ciddi zararlar verebilir ya da veri kaybına neden olabilir.

[root@ila ~]# umount /dev/hdc3

[root@ila ~]# fsck -t ext3 -VC /dev/hdc3

fsck1.38 (30-Jun-2005)

[/sbin/fsck.ext3 (1) --/dev/hdc3] fsck.ext3 -f -C0 /dev/hdc3

e2fsck 1.38 (30-Jun-2005)

Pass1: Checking inodes, blocks, and sizes

Pass2: Checking directory structure

Pass3: Checking directory connectivity

Pass4: Checking reference counts

Pass5: Checking group summary information

/dev/hdc3: 7997/125696 files(0.5% non-contiguous), 124749/251015 blocks

[root@ila ~]# mount/dev/hdc3

**SWAP (Takas Alanı)**

İşletim sisteminin gerek duyduğu bellek miktarının artması durumunda takas alanını ( swap ) büyütmek için yeni bir takas alanı bölümlemesi oluşturulabilir. Eğer diskte bu işlemi yapacak kadar bölümlenmemiş alan yoksa takas alanı dosyası oluşturularak kullanılan takas alanı arttırılabilir.

Bunun için:

[root@ila ~]# dd if=/dev/zero of=/swap\_dosyasi bs=1024 count=262144

262144+0 records in

262144+0 records out

[root@ila ~]# mkswap /swap\_dosyasi

Setting up swap space version 1, size = 268431 kB

no label, UUID=9fb6bcf6-792d-4078-91da-ae7b8f84ef11

Swap dosyasının hemen geçerli olması için:

[root@ila ~]# swapon /swap\_dosyasi

Swap dosyasının açılış zamanından itibaren geçerli olması için fstab’a aşağıdaki satır eklenmelidir.

/swapfile swap swap defaults 0 0

[root@ila~]# more /proc/swaps

Filename Type Size Used Priority

/swap\_dosyasifile 262136 0 -1

**Kota Yönetimi**

Birden fazla kullanıcının kullandığı sistemlerde kullanıcıların yüksek oranda disk alanı kullanımını engellemek için kota sistemi uygulanabilir. İki tipte kota uygulanabilir; kullanılan dosya boyutuna veya kullanılan dosya sayısına kota konulabilir. Bu iki tip kota hem kullanıcı bazlı hem de grup bazlı uygulanabilir. Yani bir kullanıcının kullanabileceği maksimum dosya boyutu ve dosya sayısına kota konabildiği gibi bir gruba ait kullanıcıların toplam kullanabileceği dosya boyutu ve dosya sayısına da kota konulabilir. Kota yönetimini kullanabilmek için sistemde quota paketinin kurulu olması gereklidir.

Kota kullanımı 4 aşamalı işlemler sonunda uygulanabilir.

• Dosya sistemi bazında /etc/fstab dosyasında kota kullanımı açılır.

• Dosya sistemleri tekrar bağlanır.

• Kota veritabanı ve disk kullanım tabloları oluşturulur.

• Kota politikaları belirlenir.

**Kota Kullanımının Açılması**

Kota kullanımını açmak için kota uygulanmak istenen dosya sisteminin bağlanma seçenekleri arasına usrquota ve grpquota seçeneklerinin eklenmesi gereklidir. usrquota seçeneği kullanıcılara, grpquota seçeneği ise gruplara kota uygulayabilmek için gereklidir.

Aşağıdaki örnekte /home dizinine hem kullanıcı hem de grup kotası kullanımı açılmıştır.

[root@ila ~]# cat /etc/fstab

# Thisfile is editedbyfstab-sync-see'manfstab-sync' fordetails

LABEL=/ / ext4 defaults 1 1

/dev/devpts /dev/pts devpts gid=5,mode=620 0 0

/dev/shm /dev/shm tmpfs defaults 0 0

/dev/proc /proc proc defaults 0 0

/dev/sys /sys sysfs defaults 0 0

/dev/hdc3 /home ext4 defaults,usrquota,grpquota 0 0

/dev/fd0 /media/floppy auto pamconsole, exec, noauto, utf8, managed 0 0

/dev/hda /media/cdrecorder auto pamconsole, exec, noauto, managed 0 0

**Dosya Sisteminin Yeniden Bağlanması**

/etc/fstab dosyasında kota uygulanmak istenen dosya sistemlerinde usrquota ve grpquota seçeneklerin açılmasından sonra seçeneklerin geçerli olması için bu dosya sistemlerinin tekrar bağlanması gereklidir.

Dosya sistemini tekrar bağlamak için aşağıdaki yöntemlerden biri kullanılabilir:

• umount ve mount komutları kullanarak

• mount –o remount dosya\_sistemi komutu kullanarak.

• İşletim sistemini tekrar başlatarak.

[root@ila ~]# mount -o remount /home

**Kota Veritabanı ve Disk Kullanım Tabloları**

Kota veritabanını oluşturmak için kota uygulanmak istenen dizinde quotacheck işlemi yapılmalıdır.

[root@ila ~]# quotacheck -cug /home

• –c seçeneği ile çalıştırılması ile /home dizininde kullanıcı kotaları tutmak için aquota.user, grup kotalarını tutmak için aquota.group dosyaları oluşturulur.

• –u seçeneği kullanıcı –g seçeneği grup için gerekli dosyanın oluşturulacağını bildirir. Eğer hem –u hem –g seçeneği verilmezse sadece kullanıcı kotaları için gerekli olan aquota.user dosyası oluşturulur.

Gerekli dosyaların oluşturulmasından sonra var olan dosya kullanım tablolarının oluşturulması için aşağıdaki komut kullanılır:

[root@ila ~]# quotacheck -avug

quotacheck: Scanning /dev/hdc3 [/home] done

quotacheck: Checked 441 directories and 7552 files

• -a - Kota kullanımı açık tüm yerel sistemleri tarar.

• -v - İşlem sırasında ekran durum bilgisini yazdırır.

• -u - Kullanıcı disk kullanımını hesaplar.

• -g - Grup disk kullanımını hesaplar.

**Kota Atama**

edquota veya setquota komutları kullanılarak kota atanabilir.

[root@ila ~]# edquota ila

Disk quotas for user ila (uid504):Filesystem blocks soft hard inodes soft hard

/dev/hdc3 51324 0 0 0 7964 0 0

• Blocks alanı kullanıcının kullanmakta olduğu dosya boyutunu gösterir. Sonraki iki alan kullanıcının disk kotasını bildirir.

• Soft alanı kullanıcının kotasını belirtir.

• Kullanıcılar grace time adı verilen süre boyunca kotalarını aşabilir. Maksimum çıkabilecekleri boyut hard alanınında belirtilen değer kadardır.

• Inodes alanı kullanıcının kullandığı dosya sayısını belirtir.

• Son iki alandan soft dosya kotasını, hard alanı ise çıkabileceği maksimum dosya boyutunu bildirir.

edquota komutu ile gruba kota atamak için –g, Grace Time değerini değiştirmek için –t seçeneği kullanılabilir.

[root@ila ~]# edquota -t

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem Block grace period Inode grace period

/dev/hdc3 7days 7days

setquota komutunun kullanımı ve önemli parametrelerinin anlamları aşağıdaki gibidir:

setquota <seçenekler> <kullanıcı> <disk-kota> <disk-kotalimit> <inode-kota> <inode-kotalimit> <dosya\_sistemi>

• -u - Kullanıcı için kota set eder (varsayılan seçenektir).

• -g - Grup için kota set eder.

• -b - Kota ile ilgili verileri standart girdiden okur(disk-kota disk-kota limit inode-kota inode-kota limit).

• -t - Kullanıcı veya grup için gracetime atar (saniye cinsinden).

[root@ila ~]# setquota ila 600000 700000 10000 110000 /home

[root@ila ~]# quota ila

Disk quotas for user ila(uid504):

Filesystem blocks quota limit grace files quota limit grace

/dev/hdc3 512928 600000 700000 7963 10000 110000

[root@ila ~]# setquota -t 3600 7200 /home/

[root@ila ~]# edquota -t

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem Block grace period Inode grace period

/dev/hdc3 1hours 2hours

**quotaon, quotaoff, repquota**

Dosya sisteminde kota kullanımını kapatmak için quotaoff, kota kullanımını açmak için de quotaon komutu kullanılabilir.

repquota komutu ile sistemdeki kota durumu raporlanabilir.

Raporda kullanıcı adından sonra gelen 2 adet “+” veya “–“ işareti kotanın aşılıp aşılmadığını gösterir. İlk işaret disk kotası ikinci işaret dosya sayısı kotasını gösterir. “+” işareti limitin aşıldığını, “-“ ise aşılmadığını bildirir.

**[root@ila ~]# quotaoff -vug /home/**

quotaoff: quotactlon /dev/hdc3 [/home]: No such process

quotaoff: quotactlon /dev/hdc3 [/home]: No such process

**[root@ila~]# quotaon -vug /home/**

/dev/hdc3 [/home]: group quotas turned on

/dev/hdc3 [/home]: user quotas turned on

**[root@ila ~]# repquota /home**

\*\*\* Report for user quotas on device /dev/hdc3

Block grace time: 01:00; Inodegrace time: 02:00

Block limits File limits

User used soft hard grace used soft hard grace

---------------------------------------------------------

root --17784 0 0 40 0 0

ila -+ 512928 600000 700000 7963 4000 5000 02:00

**Dosya Sisteminin Yedeklenmesi**

Dosya sistemlerinde olası bir felaket senaryosunda veri kaybı olması durumuna karşı önceden planlanmış otomatize veya manuel veri yedeklemeleri yapılmalıdır. Açık kaynak kodlu yazılımlardan Amanda, Bacula gibi veri yedekleme yazılımlarının yanısıra dd, rsync komutları da dosya sistemi yedeklemesi için en çok kullanılan komutlar arasındadır.

dd

dd komutu dosya sistemi üzerinde yedekleme yapmak, rastgele veri üretmek gibi işlemler yapılmasını sağlar.

dd if=kaynak of=hedef secenekler

dd if=/dev/sda2 of=/backup/yedek.img

• Disk’i disk’e yedeklemek

dd if=/dev/hda of=/dev/hdb conv=noerror,sync

• MBR’ı yedeklemek için

dd if=/dev/hdb of=/backup/mbr.backup bs=512 count=1

• MBR’ı geri yüklemek için

dd if=/backup/mbr.backup of=/dev/hdb bs=512 count=1

**rsync**

rsync UNIX/Linux sistemlerde uzak sistemler arasında yedekleme ya da veri transferi yapma ya da veri senkronizasyonunu sağlama noktasında en gelişmiş araçlardan biridir. Uzak sistem ile, TCP-873 portunda çalışan rsyncd servisi üzerinden veri kopyalama yapabileceği gibi; uzak sisteme rsh veya ssh protokolleri üstünden de erişip kopyalama işlemlerini yapabilir.

Komutun her duruma ilişkin fazlaca seçeneği bulunmkatadır.

Genel kullanım formatı:

rsync SEÇENEKLER KAYNAK HEDEF

SEÇENEKLER:

man sayfasına bakıldığında görülebileceği üzere bir sürü seçeneği vardır. Fakat en sık kullanılan seçeneği -a (archive) ve -v (verbose) seçenekleridir. -e ssh seçeneği ile uzak bir sisteme ssh protokolü üstünden bağlanması sağlanabilir.

KAYNAK | HEDEF:

Yerelde bir dizin olabileceği gibi uzak bir sistemde belirtilebilir.

Örneğin dosyalar.ila.lab sunucusundaki /home/ dizininin bir kopyasını almak için aşağıdakine benzer bir komut kullanılabilir.

rsync -avHe ssh [root@dosyalar.ila.lab:/home /yedekler/](mailto:root@dosyalar.ila.lab:/home /yedekler/)

**LVM**

**LVM Nedir?**

LVM (Logical Volume Management, Mantıksal Hacim Yönetimi) linux'un mantıksal bölüm yöneticisidir. Bu yapı sistem yöneticilerine disk ve bölümleri yönetirken çok daha esnek bir yapı sunar. Örneğin, birden fazla harddiski birleştirip tek bir bölümmüş gibi sunabilir. LVM, harddisklerle mantıksal bölümler arasında ince bir katman gibidir.

**LVM ne işe yarar**?

- Büyük depolama sistemlerinde yeni disk ekleme, disk çıkarma, disk değiştirme ve herhangi bir kayıp olmadan bir diskten başka diske veri kopyalama yapabilir.

- Yazılımsal olarak RAID0, RAID1, RAID5, RAID6, vb. yapabilir.

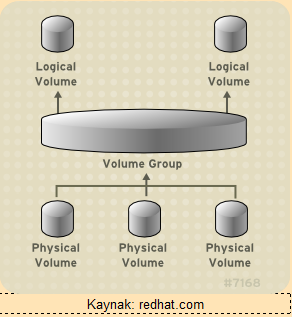
**LVM Kavramları**

Hacim Grubu (Volume Group, VG): Fiziksel hacimlerin içine atıldığı havuzdur. Mantıksal hacimler bu havuzdaki alanları kullanırlar. Bir bilgisayarda birden fazla hacim grubu olabilir.

Fiziksel Hacim (Physical Volume, PV): Fiziksel hacimler harddisk veya harddiskin bir bölümü (partition) olabilirler. LVM yapısında kullanabilmek için mutlaka bir hacim grubuna dahil olmak zorundadırlar.

Mantıksal Hacim (Logical Volume, LV): Hacim grubundaki havuz alanından istenilen miktarlarda alan alınarak oluşturulan bölümlerdir. Oluşturulduktan sonra, mantıksal hacimler istenilen dosya sistemiyle formatlanıp kullanılırlar.

Extent: LVM hacimleri yönetirken extent denen küçük parçalara böler. LVM için varsayılan extent boyutu 4MB'tır. Extent kavramı dosya sistemlerindeki blok kavramına eşdeğerdir.



**Fiziksel Hacim (PV) Operasyonları**

Fiziksel hacim (physical volume) tanımlamak için pvcreate komutu kullanılır. Bu komut belirtilen fiziksel aygıtı LVM yapısında kullanılacak şekilde ayarlar. Fiziksel aygıt diskin tamamı olabileceği gibi, bir bölümü de (partition) olabilir.

pvcreate fiziksel\_aygıt [fiziksel\_aygıt]

Mevcut fiziksel hacimler pvdisplay komutuyla görüntülenebilir.

Fiziksel hacimler pvremove komutuyla silinebilir. Bu komut ilgili aygıtta LVM'le ilgili metadata bilgilerini temizler.

Örnek:

Fiziksel disk olarak /dev/vdb aygıtının bölümleri kullanılacaktır.

[root@ila ~]# fdisk -l /dev/vdb

Disk /dev/vdb: 8589 MB, 8589934592 bytes

16 heads, 63 sectors/track, 16644 cylinders

Units = cylinders of 1008 \* 512 = 516096 bytes

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

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

Disk identifier: 0x1ef4d557

Device Boot Start End Blocks Id System

/dev/vdb1 1 8322 4194256+ 83 Linux

/dev/vdb2 8323 16644 4194288 83 Linux

Aşağıda fiziksel hacim olarak /dev/vdb aygıtının bölümleri olan /dev/vdb1 ve /dev/vdb2 kullanılmıştır. İstenildiği takdirde /dev/vdb aygıtının kendisi de kullanılabilir.

[root@ila ~]# pvcreate /dev/vdb1

Physical volume "/dev/vdb1" successfully created

[root@ila ~]# pvcreate /dev/vdb2

Physical volume "/dev/vdb2" successfully created

pvdisplay komutu ile /dev/vdb1 ve /dev/vdb2 aygıtlarının fiziksel hacim olarak yaratıldığı görülebilir. Fiziksel hacim bir hacim grubuna eklendiğinde "PE" ile ilgili kısımların bilgileri aktif hale gelecektir. PE, physical extent (fiziksel extent) demektir ve fiziksel hacime ait extent olduğunu belirtir.

[root@ila ~]# pvdisplay

"/dev/vdb1" is a new physical volume of "4.00 GiB"

--- NEW Physical volume ---

PV Name /dev/vdb1

VG Name

PV Size 4.00 GiB

Allocatable NO

PE Size 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID ojJJEG-trpp-wh1V-d317-GwM0-kQjI-U6PCuH

"/dev/vdb2" is a new physical volume of "4.00 GiB"

--- NEW Physical volume ---

PV Name /dev/vdb2

VG Name

PV Size 4.00 GiB

Allocatable NO

PE Size 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID dWWPXm-nubu-ue0U-wJ4u-VI19-u42A-d2Hz4g

[root@ila ~]# pvremove /dev/vdb1

Labels on physical volume "/dev/vdb1" successfully wiped

[root@ila ~]# pvremove /dev/vdb2

Labels on physical volume "/dev/vdb2" successfully wiped

**Hacim Grubu (VG) Operasyonları**

Hacim grubu (volume group) yaratmak için vgcreate komutu kullanılır. Komut kullanılırken, hacim grubu ismi ve en az bir fiziksel hacim belirtilmek zorundadır.

vgcreate grup\_ismi fiziksel\_aygıt [fiziksel\_aygıt]

Hacim grubuyla ilgili bilgiler vgdisplay komutuyla görüntülenebilir.

vgdisplay

Mevcut hacim grubunu silmek için vgremove komutu kullanılır.

vgremove grup\_ismi

Hacim grubuna yeni bir fiziksel hacim eklenmek istediğinde vgextend komutu kullanılır. Bu sayede hacim grubunun kullandığı havuz alanı genişletilmiş olur.

vgextend grup\_ismi fiziksel\_aygıt [fiziksel\_aygıt]

Hacim grubundan bir fiziksel hacim çıkarılmak istediğinde vgreduce komutu kullanılır. Bu komutla hacim grubunun kullandığı havuz alanı küçülür.

vgreduce grup\_ismi fiziksel\_aygıt [fiziksel\_aygıt]

Örnek:

[root@ila ~]# vgcreate VG\_TEST /dev/vdb1

Volume group "VG\_TEST" successfully created

[root@ila ~]# vgextend VG\_TEST /dev/vdb2

Volume group "VG\_TEST" successfully extended

Fiziksel hacimler, hacim grubuna eklendiklerine fiziksel extent (PE) bilgileri görünür hale gelir. Extent boyut bilgisi hacim grubuna bağlı bir niteliktir. Hacim grubu, fiziksel hacimleri extentlere böler ve mantıksal hacimlerin kullanımına sunar.

Aşağıda ilgili fiziksel hacimlere ait toplam extent sayısı, boş extent sayısı ve kullanılmış extent sayısı bilgisi görünmektedir. Mantıksal hacimler yaratıldıkça Allocated PE (ayırılmış extentler) sayısı çoğalacaktır.

[root@ila ~]# pvdisplay

--- Physical volume ---

PV Name /dev/vdb1

VG Name VG\_TEST

PV Size 4.00 GiB / not usable 3.95 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 1023

Free PE 1023

Allocated PE 0

PV UUID ojJJEG-trpp-wh1V-d317-GwM0-kQjI-U6PCuH

--- Physical volume ---

PV Name /dev/vdb2

VG Name VG\_TEST

PV Size 4.00 GiB / not usable 3.98 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 1023

Free PE 1023

Allocated PE 0

PV UUID dWWPXm-nubu-ue0U-wJ4u-VI19-u42A-d2Hz4g

vgdisplay komutunun çıktısında tüm hacim gruplarına ait bilgiler görünmektedir. Mevcut örnekte 1 tane hacim grubu vardır. Hacim grubunda kaç tane fiziksel hacim, kaç tane mantıksal hacim, toplam ne kadar alan kullanıldığı, ne kadarının daha kullanılabileceğine dair bilgiler yer almaktadır.

[root@ila ~]# vgdisplay

--- Volume group ---

VG Name VG\_TEST

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 2

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 2

Act PV 2

VG Size 7.99 GiB

PE Size 4.00 MiB

Total PE 2046

Alloc PE / Size 0 / 0

Free PE / Size 2046 / 7.99 GiB

VG UUID b1HTMB-i3UT-Xe1X-3nbf-rGVG-pxl1-4BuceB

vgreduce komutuyla fiziksel hacimler hacim grubundan çıkartılabilir. Bu işlemin yapılabilmesi için fiziksel hacimin tüm extent'leri boş (free) olmak zorundadır.

[root@ila ~]# vgreduce VG\_TEST /dev/vdb1

Removed "/dev/vdb1" from volume group "VG\_TEST"

Son fiziksel hacim, vgreduce ile hacim grubundan silinemez. Silinmeye çalışıldığında hata verir. Bunun yerine hacim grubu tamamen silinmelidir.

[root@ila ~]# vgreduce VG\_TEST /dev/vdb2

Can't remove final physical volume "/dev/vdb2" from volume group "VG\_TEST"

[root@ila ~]# vgremove VG\_TEST

Volume group "VG\_TEST" successfully removed

[root@ila ~]# vgdisplay

No volume groups found

**Mantıksal Hacim (LV) Operasyonları**

Mantıksal hacim (logical volume) yaratmak için lvcreate komutu kullanılır. Bu komut, içinde bulunduğu hacim grubundaki alandan istenilen miktarda alan yaratır. Yaratılan mantıksal hacim aygıtı /dev/[HACİM\_GRUBU\_İSMİ]/[MANTIKSAL\_HACİM\_İSMİ] yolunda bulunur.

lvcreate [ -l extent\_sayisi | -L boyut ] -n hacim\_ismi grup\_ismi

-l: Hacim yaratılırken kullanılacak extent sayısını belirtir. Örnek olarak, -varsayılan extent boyutu 4MB iken- 100 verilirse 400MB'lık bir hacim yaratılmış olur. Bu parametre yerine yüzdeli değer de girilebilir. Ör: 100%FREE. Bu 'hacim grubundaki boş alanın tümünü kullan' demektir. -L parametresiyle beraber kullanılmaz.

-L: Hacim yaratılırken kullanılacak alanı belirtir. Kilobayt için Ki Megabayt için M, Gigabayt için G, Terabayt için T harfi kullanılır. Ör: 5G. -l parametresiyle beraber kullanılmaz.

-n: Hacimin adını belirtir.

lvdisplay komutu kullanılarak mantıksal hacimlere ait bilgiler görüntülenir.

lvremove komutuyla yaratılmış mantıksal hacim silinir. 3 farklı kullanım yöntemi vardır. Parametre olarak aygıt yolu belirtilebilir, "hacim grubu adı/mantıksal hacim adı" belirtilebilir. Bu iki parametre belirtilen mantıksal hacimi siler. Son seçenek hacim grubu adı belirtilebilir. Bu, hacim grubunda bulunan bütün mantıksal hacimleri siler.

lvremove aygıt\_yolu

lvremove {hacim\_grubu\_ismi}/{mantıksal\_hacim\_ismi}

lvremove hacim\_grubu\_ismi

Örnek:

İlk başta hacim grubundaki bütün alan boştur.

[root@ila ~]# vgdisplay | grep 'PE'

PE Size 4.00 MiB

Total PE 2046

Alloc PE / Size 0 / 0

Free PE / Size 2046 / 7.99 GiB

Hacim grubunda 2 tane fiziksel hacim mevcuttur ve ikisinin de extent sayısı 1023'tür.

[root@ila ~]# pvdisplay | grep -E 'PV | PE'

PV Name /dev/vdb2

PV Size 4.00 GiB / not usable 3.98 MiB

PE Size 4.00 MiB

Total PE 1023

Free PE 1023

Allocated PE 0

PV UUID dWWPXm-nubu-ue0U-wJ4u-VI19-u42A-d2Hz4g

PV Name /dev/vdb1

PV Size 4.00 GiB / not usable 3.95 MiB

PE Size 4.00 MiB

Total PE 1023

Free PE 1023

Allocated PE 0

PV UUID ojJJEG-trpp-wh1V-d317-GwM0-kQjI-U6PCuH

VG\_TEST hacim grubundan, lvcreate komutuyla 1200 extent'lik bir hacim yaratılır.

[root@ila ~]# lvcreate -l 1200 -n mantiksal1 VG\_TEST

Logical volume "mantiksal1" created

Ardından hacim grubunda kullanılan extent sayısı ve ne kadar alan kapladığı, ne kadar boş extent olduğu ve ne kadar alanın boş olduğu görülebilir.

[root@ila ~]# vgdisplay | grep 'PE'

PE Size 4.00 MiB

Total PE 2046

Alloc PE / Size 1200 / 4.69 GiB

Free PE / Size 846 / 3.30 GiB

lvdisplay komutuyla mevcut tüm mantıksal hacimlere ait bilgiler görüntülenebilir. Mantıksal hacimin ismi, aygıt yolu, hangi hacim grubuna bağlı olduğu, kaç extent kullandığı ve haliyle boyutu görüntülenebilir.

[root@ila ~]# lvdisplay

--- Logical volume ---

LV Path /dev/VG\_TEST/mantiksal1

LV Name mantiksal1

VG Name VG\_TEST

LV UUID BEiklv-yfeg-l5hl-sAvs-9cNm-1yux-7zmZ24

LV Write Access read/write

LV Creation host, time ila.itu.edu.tr, 2013-09-06 15:18:07 +0300

LV Status available

# open 0

LV Size 4.69 GiB

Current LE 1200

Segments 2

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 253:0

Mantıksal hacim yaratıldıktan sonra standart disk bölümü olarak kullanılabilir. Burada ext4 olarak formatlanmıştır.

[root@ila ~]# mkfs.ext4 /dev/VG\_TEST/mantiksal1

lvcreate komutu -L parametresiyle boyut belirtilerek de kullanılabilir. Bu parametreyle kullanıldığında lvm extent boyutunu otomatik olarak hesaplar.

[root@ila ~]# lvcreate -L 100M -n mantiksal2 VG\_TEST

Logical volume "mantiksal2" created

Yeni yaratılan mantıksal hacim, aygıt yolu belirtilerek aşağıdaki şekilde görüntülenebilir.

[root@ila ~]# lvdisplay /dev/VG\_TEST/mantiksal2

--- Logical volume ---

LV Path /dev/VG\_TEST/mantiksal2

LV Name mantiksal2

VG Name VG\_TEST

LV UUID dtUXx2-z7HI-frwF-0Knv-2QjL-dLlv-fX9GQc

LV Write Access read/write

LV Creation host, time ila.itu.edu.tr, 2013-09-06 16:07:12 +0300

LV Status available

# open 0

LV Size 100.00 MiB

Current LE 25

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 253:1

Bir başka mantıksal hacim yaratma yöntemi oran belirtmektir. Aşağıdaki komut bütün boş alanı yeni yaratılan mantıksal hacime ayırır.

[root@ila ~]# lvcreate -l 100%FREE -n mantiksal3 VG\_TEST

Logical volume "mantiksal3" created

Ardından hacim grubunda tüm alanların kullanımda olduğu görülür.

[root@ila ~]# vgdisplay | grep 'PE'

PE Size 4.00 MiB

Total PE 2046

Alloc PE / Size 2046 / 7.99 GiB

Free PE / Size 0 / 0

Yeni yaratılan mantıksal hacime ait boyut bilgileri görülebilir.

[root@ila ~]# lvdisplay /dev/VG\_TEST/mantiksal3

--- Logical volume ---

LV Path /dev/VG\_TEST/mantiksal3

LV Name mantiksal3

VG Name VG\_TEST

LV UUID KK8rQP-j09e-yk81-2amN-Unx1-pmJX-U38JwR

LV Write Access read/write

LV Creation host, time ila.itu.edu.tr, 2013-09-06 16:45:14 +0300

LV Status available

# open 0

LV Size 3.21 GiB

Current LE 821

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 253:2

lvremove komutuyla yolu belirtilen mantıksal hacim silinir.

[root@ila ~]# lvremove /dev/VG\_TEST/mantiksal2

Do you really want to remove active logical volume mantiksal2? [y/n]: y

Logical volume "mantiksal2" successfully removed

**LVM bölümünü yeniden boyutlandırma**

Mantıksal hacimler boyutları itibariyle genişletilebilirler veya küçültülebilirler.

Bu operasyon biraz karışıktır. Çünkü mantıksal hacim bir dosya sistemiyle formatlanmış olabilir. Bu durumda boyutun değişimi dosya sistemine yansıtılmak zorundadır. Aynı zamanda dosya sistemi bir dizine bağlı ve aktif olarak kullanımda olabilir. Bu durumda yalnızca genişletme operasyonu yapılabilir. Küçültme yapılamaz. Küçültme yapılabilmesi için dosya sisteminin umount edilmesi lazımdır.

Burada genişletme operasyonuna ait adımlar belirtilecektir. Örnekte dosya sistemi bağlıdır. Bağlı değilken de durum farklı değildir. Küçültme operasyonu için de durum benzerdir. lvextend yerine lvreduce komutu kullanılır.

1. Eğer hacim grubunda boş alan yoksa hacim grubuna yeni fiziksel hacim eklenmelidir. Bu sayede hacim grubu genişletilmiş olur.

2. Genişletilmek istenen hacim grubu lvextend komutuyla aracılığıyla genişletilir.

[root@ila ~]# lvextend -L +100M /dev/VG\_TEST/mantiksal1

Extending logical volume mantiksal1 to 4.79 GiB

Logical volume mantiksal1 successfully resized

[root@ila ~]# df -h | grep -A 1 mantiksal1

/dev/mapper/VG\_TEST-mantiksal1

4.7G 138M 4.3G 4% /mnt

3. resize2fs ile dosya sistemi katmanına diskin genişlediği bilgisi yansıtılır. Böylece genişleme dosya sistemi düzeyinde görülür.

[root@ila ~]# resize2fs /dev/VG\_TEST/mantiksal1

resize2fs 1.41.12 (17-May-2010)

Filesystem at /dev/VG\_TEST/mantiksal1 is mounted on /mnt; on-line resizing required

old desc\_blocks = 1, new\_desc\_blocks = 1

Performing an on-line resize of /dev/VG\_TEST/mantiksal1 to 1254400 (4k) blocks.

The filesystem on /dev/VG\_TEST/mantiksal1 is now 1254400 blocks long.

[root@ila ~]# df -h | grep -A 1 mantiksal1

/dev/mapper/VG\_TEST-mantiksal1

4.8G 138M 4.4G 3% /mnt

4. Dosya sistemi bağlı iken (online) genişletme yapılmıyorsa dosya sistemi kontrolü (fsck, file system check) yapılmasında yarar vardır. Yukarıdaki örnekte, operasyon dosya sistemi bağlı iken yapılmıştır. Bu nedenle umount edilmeden dosya sistemi kontrolü yapılamaz.

lvextend ve lvreduce komutları dosya sistemini genişletmek ve küçültmek için kullanılır.

lvextend [ -l extent\_sayisi | -L +genişletilecek\_boyut ] mantıksal\_hacim

-l: Hacim genişletilirken kullanılacak extent sayısını belirtir. Örnek olarak, -varsayılan extent boyutu 4MB iken- 100 verilirse 400MB'lık bir genişletme yapılır. Bu parametre yerine yüzdeli değer de girilebilir. Ör: 100%FREE. Bu 'hacim grubundaki boş alanın tümünü kullan' demektir. -L parametresiyle beraber kullanılmaz.

-L: Hacim genişletilirken kullanılacak alanı belirtir. Kilobayt için K, Megabayt için M, Gigabayt için G, Terabayt için T harfi kullanılır. Ör: +5G. -l parametresiyle beraber kullanılmaz.

lvreduce [ -l extent\_sayisi | -L -genişletilecek\_boyut ] mantıksal\_hacim

-l: Hacim küçültülürken kullanılacak extent sayısını belirtir. Örnek olarak, -varsayılan extent boyutu 4MB iken- 100 verilirse 400MB'lık bir genişletme yapılır. -L parametresiyle beraber kullanılmaz.

-L: Hacim küçültülürken boşaltılacak alanı belirtir. Kilobayt için K, Megabayt için M, Gigabayt için G, Terabayt için T harfi kullanılır. Ör: -5G. -l parametresiyle beraber kullanılmaz.

**UYGULAMA 1**

NOT:

Uygulamayı sanal makineleriniz üstünde gerçekleştirin.

Uygulama için istenen disk işlemlerini sanal makineye takılı olan ve üzerinde hiçbir bölümleme olmayan disk üstünde yapın.

Makinanızda bulunan boştaki disk üstünde 1 Gb kapasitesinde takas (swap) alanı oluşturup bunu sisteminize ekleyin.

Bilgisayarınızın diski üstünde 1GB kapasitesinde 2 adet bölüm oluşturun.

Oluşturduğunuz iki adet 1GB'lık bölümü LVM yapısını kullanarak 2GB'lık tek bir bölüm olarak oluşturun.

Oluşturduğunuz 2GB'lık LVM bölümünü ext4 ile biçimlendirin. Daha sonra /etc/fstab dosyasını ayarlayarak bu bölümün /home dizinine bağlanmasını sağlayın.

**ÇÖZÜMLER**

1-----------------------------------------

[root@localhost ~]# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

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

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

Disk identifier: 0x00000000

Disk /dev/vda: 10.7 GB, 10737418240 bytes

16 heads, 63 sectors/track, 20805 cylinders

Units = cylinders of 1008 \* 512 = 516096 bytes

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

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

Disk identifier: 0x000c502b

Device Boot Start End Blocks Id System

/dev/vda1 3 2083 1048576 82 Linux swap / Solaris

Partition 1 does not end on cylinder boundary.

/dev/vda2 \* 2083 20806 9436160 83 Linux

Partition 2 does not end on cylinder boundary.

[root@localhost ~]#

[root@localhost ~]# fdisk /dev/sda

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x5f53c400.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

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

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

Disk identifier: 0x5f53c400

Device Boot Start End Blocks Id System

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-522, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-522, default 522): +1G

Command (m for help): t

Selected partition 1

Hex code (type L to list codes): 82

Changed system type of partition 1 to 82 (Linux swap / Solaris)

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

[root@localhost ~]# mkswap /dev/sda1

Setting up swapspace version 1, size = 1060252 KiB

no label, UUID=48d48c11-2c7e-4e11-b5aa-43fe598b0eae

[root@localhost ~]# swapon /dev/sda1

[root@localhost ~]#

2----------------------------------------------------------

[root@localhost ~]# swapoff /dev/sda1

[root@localhost ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 2

First cylinder (133-522, default 133):

Using default value 133

Last cylinder, +cylinders or +size{K,M,G} (133-522, default 522): +1G

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 3

First cylinder (265-522, default 265):

Using default value 265

Last cylinder, +cylinders or +size{K,M,G} (265-522, default 522): +1G

Command (m for help): p

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

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

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

Disk identifier: 0x5f53c400

Device Boot Start End Blocks Id System

/dev/sda1 1 132 1060258+ 82 Linux swap / Solaris

/dev/sda2 133 264 1060290 83 Linux

/dev/sda3 265 396 1060290 83 Linux

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

[root@localhost ~]#

3-------------------------------------------------------------

[root@localhost ~]# pvcreate /dev/sda2

Writing physical volume data to disk "/dev/sda2"

Physical volume "/dev/sda2" successfully created

[root@localhost ~]# pvcreate /dev/sda3

Writing physical volume data to disk "/dev/sda3"

Physical volume "/dev/sda3" successfully created

[root@localhost ~]#

[root@localhost ~]# pvdisplay

"/dev/sda2" is a new physical volume of "1.01 GiB"

--- NEW Physical volume ---

PV Name /dev/sda2

VG Name

PV Size 1.01 GiB

Allocatable NO

PE Size 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID t1ndxn-6zRO-1kKG-XX7m-JS9z-zWxk-4c3vwE

"/dev/sda3" is a new physical volume of "1.01 GiB"

--- NEW Physical volume ---

PV Name /dev/sda3

VG Name

PV Size 1.01 GiB

Allocatable NO

PE Size 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID iCflay-gOH7-JJ3M-CRdL-4FUs-NjRz-0DiCje

[root@localhost ~]#

[root@localhost ~]# vgcreate DISK\_GRUBU /dev/sda2 /dev/sda3

Volume group "DISK\_GRUBU" successfully created

[root@localhost ~]# vgdisplay

--- Volume group ---

VG Name DISK\_GRUBU

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 1

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 2

Act PV 2

VG Size 2.02 GiB

PE Size 4.00 MiB

Total PE 516

Alloc PE / Size 0 / 0

Free PE / Size 516 / 2.02 GiB

VG UUID frbdsW-DByj-9m1Y-NwZ8-1UZm-bCeM-viGoy5

[root@localhost ~]#

[root@localhost ~]# lvcreate -L 2G -n 2likBolum DISK\_GRUBU

Logical volume "2likBolum" created

[root@localhost ~]# lvdisplay

--- Logical volume ---

LV Path /dev/DISK\_GRUBU/2likBolum

LV Name 2likBolum

VG Name DISK\_GRUBU

LV UUID pnMcCQ-rkji-iHZL-g49C-G0li-zdHB-TUa5jp

LV Write Access read/write

LV Creation host, time localhost.localdomain, 2013-10-26 14:43:10 +0300

LV Status available

# open 0

LV Size 2.00 GiB

Current LE 512

Segments 2

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 253:0

[root@localhost ~]#

4------------------------------------------------------------------------------

[root@localhost ~]# mkfs.ext4 /dev/DISK\_GRUBU/2likBolum

mke2fs 1.41.12 (17-May-2010)

Filesystem label=

OS type: Linux

Block size=4096 (log=2)

Fragment size=4096 (log=2)

Stride=0 blocks, Stripe width=0 blocks

131072 inodes, 524288 blocks

26214 blocks (5.00%) reserved for the super user

First data block=0

Maximum filesystem blocks=536870912

16 block groups

32768 blocks per group, 32768 fragments per group

8192 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912

Writing inode tables: done

Creating journal (16384 blocks): done

Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 38 mounts or

180 days, whichever comes first. Use tune2fs -c or -i to override.

[root@localhost ~]# vim /etc/fstab

[root@localhost ~]# grep home /etc/fstab

/dev/DISK\_GRUBU/2likBolum /home ext4 defaults 0 0

[root@localhost ~]#

**AĞ AYARLARI**

**Linux'da Ağ Ayarları**

NOT:

Gnome Kde masaüstü ortamları,Linux üzerine oturmuş apayrı birer işletim sistemi gibi çalışır.Mesela Gnome,Kde .. falan donarsa o anki firefox falan gider. Artık linux bizim için siyah ekrandır.

-ifconfig : ag yapılandırmamızı gösterir.(interfaceconfig;internete çıkan ip arayüzü,ag kartı herşey)

-internete çıkan herşeye eth ile başlayan bir kelime verilir.Bide em var.Yeni sürüm çekirdeklerle beraber;

Eger donanım onboard ethernet kartı varsa(em) bunun için kullanılır.

ilk aldıgı deger: eth0,1,2... iken em1,2,3... diye gider.

-lo:loopbak(interface adi,eth,em gibi)--<localhost makinanın kendi kendini gsöteren adres.

-virbr,vnet3 -->sanal makineleri barındıran fiziksel makineler olasmından ötürü bazı zahiri ethernet kartlarıda varmış gibi görünür.

-ifconfig in çıktısında;

inet addr :IP Adress

-ip address show : ifconfig in yeni hali diyebilriz.(yeni bir komut.)

-ifconfig em1 :em1 interface mizi görüntüleyebilriz.

-ifconfig em1 down :kapatır.(interfacei interneti)Şuan,ifconfig dersek em1 i göremeyiz.s

-ifconfig -a ile bütün interfaceleri görebilriz.

-ifconfig em1 up :interneti açar.

ama ping www.google.com aheln yapamayız.

ping 127.0.0.1

ping 192.168.2.1 bunlara ping atabiliriz(local host ve aynı agdaki makineler.)ama google aping atamayız.

Bu demek olur ki;Default Gateway yok.

-ifconfig em 192.168.0.101 netmask 255.255.255.0 (alt ağ maskesi; network admini söyler.)

-Default Gateway ; ifconfig le ayarlanmaz.

-route komutu ile default gateway kullanılır.Yönlendirme bilgisayarın içinde başlar.veri daha bilgisayarın içerisindeyken nereye gideceği ile ilgili yönlenir.

-route deyip enterlarsak halen default gateway imiz yok.

-route add default gw 192.168.0.200

-route -n ile kullanılır.

-destination 0.0.0.0 olarak görülür.BU interneti temsil eder.Yani artık internetteyiz.

-bilgisayara kablo takınca altag maskesi,kendi agımız olarak,belirlenir. ama yetmez.internete çıkmak için agateway e ihtiyacımız var.(bunuda rooterlar saglar.)

şuana kadar:

1.alt ag maskesi(ifconfig)

2.varsayılan ag geçidi(route)

girdik....

3.DNS kaldı.makineler IP ile konuşur.Biz isimlerden anlarız.IP to DNS (bunu dosyaya yazarak ayarlarız.)

-/etc/resolv.conf dosyasına yazarsak ayar kalıcı olur.Yoksa makineyi kapatıp açtığımızda gider.BU dosyada;

nameserver 192.168.0.200 bu bir dns adresidir.

birden fazla nameserver girince bir nameserver a ulaşamazsa ,diğerine gider bu şekilde yukarıdan aşagı doğru gider. Şuana kadar ki network hareketleri tüm linux dagıtımları için gçerliydi bundan sonrası redhat türevlerinde geçerlidir.

-cat /etc/sysconfig/network dersek;

NETWORKING=yes internet var demektir.

-linuxda aynı isime ait birden fazla makine aga çıkabilir.bu yüzden hostname önemsizdir. Ama windows ta önemlidir.

-hostname sertac diyerek değiştirebilirz ama önemsiz.

GATEWAY= ile gateway verebiliriz.buraya yazarsak kalıcı olur.

--bu dosya genel bir dosyadır. Ayrıca her interface e ait dosyalarda vardır.

-cat /etc/sysconfig/network-scripts/ifcfg-em1 gibi.

-less /usr/share/doc/init-scripts-9.30.1/sysconfig.txt----> bütün yapılandırma dosylarıyla ilgili bir dökümandır.

\*GAateway sadece 1 kez tanımlanır.Mesela dosyaya yazıyoruz.1 den fazla yerde gateway tanımlayamayız.

-makine açıldığında bu dosyları okur.ayarları /proc a gömmek için

sysctl -p (p:push)---< kalıcı olarak çekirdek parametrelerini değiştirmenin yoludur.Çekirdek derlenirken bu dosyaya varsayılan degerler yazılmıştır.

-/etc/hosts :windowstada var,trojanlar bu dosyaya bayılır.BU dosya henüz dns mimarisi oluşturulmamışkenden itibaren vardır.

\*/etc/hosts/:yerel networkteki makineleri ayırt ederken dns mimariyle ugraşmamak için kullanılır. TRojanlar bu dosyayı editleyerek aslında gitmek istedğimiz yere değilde başka bir yere gitmemize neden olabilir.

-/etc/nsswithconf :birçok linuxta veritabanı vardır.bunların dosyaları:

shadow:files

group:files

hosts: files dns -->önce dosylara itaat sonra dns hiyerarşisine, bunun sırası değiştirlebilir.

-ifconfig em1; bir fiziksel interface e birden fazla interface verebiliriz.

-ifconfig em1: -->em1 den türetilmiş başka interface tanımlacak demektir.

kalıcı değildir.

-cd /etd/sysconfig/network-scripts/

bir dosya oluşturmalıyız.

cp ifcfg-em1 ifcfg-turemis

NOT:türemiş interface ile otomatik IP alınamaz.Çünkü DHCP mac adresiyle ilişkili olarak adres dagıtır. Mac adresi değişmedikçe İP degişmez.

-şimdi ifcfg-turemis i editliyoruz:

=static

DEVİCE="em1:yedek"

\*\*\*\*service network restart\*\*\*\*\*

-static elle ip girilecekse verilir.

-dhcp : otomatik ip alma ile alakalı protokol

-ONBOOT=makine açılınca bu interface kullanımda mı olacak.

--/etc/sysconfig/network-scripts/ifcfg-em1 : em1 için gateway ve ip adresini buraya yazarsak redhat için ayarlar kalıcı olur.

--peki bu ayarlar çekirdege nasıl ögretiliyor.

.service rsyslog (status|stop|start) bu mimariden hareketle:===> service network ortaya çıkmıştır.

-service network stop

-service ifconfig

-service network start

-service network restart : service network interfaceler için dosylara yazılmış ayarları tek tek okur.ona göre açar.sistem açılışıda bu şekildedir.rsyslog da ayar dosylarını okur. gibi.

not :windows bir makine network e değdiği anda ben burdayım diye bagırır.hiçbir security işi windows üzerinde yapılmaz. Bunun için temiz IP ye ihtiyaç vardır.Bu yüzden ip çakışması çıkar. Linux böyle değildir.

-makinemizi ping e kapatacagız mesela:

-ls /proc/sys/net/ipv4/icmp-echo-ignore-all

-ping paketi gönderdiğimizde poncevabı gekir.icmp aslında ping paketinin teknik adıdır.(burada gelen bütün ping paketlerini aktif et yada pasif et şeklined yazılıdır ve hiçbir metin editoru bu dosyayı editleyemez ama:)

echo 1>/proc/sys/net/ipv4/icmp-echo-ignore-all ile artık pingler ignore edilir.pon cevabı verilmez.

proc dizinin ayrı bir dizin agacı yoktur. Lİnux çekirdeğindeki en ince ayara kadar bu dizinin altından ayarlayabiliriz.;

Peki makineyi kapatıp açarsak; yazdığımız 1 nolacak ,proc dizini ramde çalışır.bU yüzden pc kapanınca gider.Peki kalıcı hale getirmek için ne yapmalıyız.

/etc/sysctl.conf a ayzarak kalıcı hale getirilebilir.

\*\*\*\*\*cat /proc/sys/net/ipv4/ip-forward ın degeri net.ipv4.ip-forward şeklinde degeri yazar / yerine . koyulmuş DİKKAT(aralarındaki ilişki bu)

-nm : (netwrok manager) son kullanıclar içindir

DOsyada NM\_CONTROLLED="no" dersek managerin dosyaları editlemesini engelleyebiliriz.

-ifdown em1:yedek :dogrudan fiziksel interface i kapatır.

-ifup ;interface up tan farklı olarak /etc/sys.conf altındaki dosyayı okuyarak açar.

\*\*\*netstat -punta :tcp ve udp anlamında networkten dinlenen portlarda dahil düzügün bir şekilde döker.(ESTABLISH:makinem şuan kimle konuşuyor.LISTEN:dinliyor.Android makineler de ESTABLISH durumunda bir IP ile konuşmaktadır.)

-ping www.google.com :ping e açık çünkü ping atanların bilgilerini toplar.

(NOT:makineler herşeyi dns server a sorar.)

-host www.google.com ---------------------\

-nslookup www.google.com (detaylı)---------> ikiside IP adresi verir.

-dig www.google.com :DNS Adminleri bundan vazgeçemez.

-whois 160.75.160.110 :bu ip adresi network olarak kime ait görebiliriz.

-traceroute

-ftp:komut satırı ftp istemcisi

-wget:örnegin nette rpm paketi bulduk sunucuya nasıl indirecem. wget kullanarak.

-wget url: bulunduğumuz dizine indirir.

-host url:url in ip sini öğrenebiliriz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*bıtıs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Linux işletim sistemlerinde ağ yapılandırmaları genellikle ilgili dosyalarındaki tanımlamalar ile yapılmaktadır. Hem bu dosyaların içeriği düzenlenerek, hem de grafik arayüzlü (GUI) yönetim araçları kullanılarak ağ ayarları yapılabilir. Dosyaların düzenlenmesi ile yapılan yapılandırmalarda GUI aracılığı ile yapılan yapılandırmalara göre kullanıcının daha teknik bilgiye sahip olmasını gerektirir.

Linux’te ağ ayarlarına ilişkin dosya ve dizinler aşağıdaki gibidir:

• /etc/sysconfig/network Dosyası

• /etc/sysconfig/network-scripts Dizini

• /etc/sysctl.conf Dosyası

• /etc/nsswitch.conf Dosyası

• /etc/hosts Dosyası

• /etc/hosts.conf Dosyası

• /etc/resolv.conf Dosyası

**/etc/sysconfig/network Dosyası**

Bu dosya genel ağ ayarları için kullanılan dosyadır. Bu dosya içinde sistemde ağ desteği verilip verilmeyeceği, ağ geçidi (gateway) tanımı, makine ismi gibi bilgiler bulunur.

Bu dosya aşağıdaki yapıdadır:

NETWORKING=yes

HOSTNAME=ila.itu.edu.tr

GATEWAY=192.168.1.200

GATEWAYDEV="eth0"

Bu dosyada çoğunlukla kullanılan terimler aşağıdaki gibidir:

NETWORKING - [yes|no] Sistemde ağ desteği verilip verilmeyeceğini belirtir.

HOSTNAME - <hostname> Sistemin ismini belirtir.

GATEWAY <gateway ip> Sistemin ağ geçidini (gateway) belirtir. Özellikle birden fazla ağ kartının bulunduğu bilgisayarlarda sistem genelindeki varsayılan ağ geçidinin tanımlanması için kullanılır.

GATEWAYDEV - <gateway\_device> Sistemin ağ geçidine ulaşmak için hangi ağ arayüzünü kullanacağını belirtir.

**/etc/sysconfig/network-scripts/ifcfg-ethN Dosyası**

Sistemde bulunan veya sisteme eklenmek istenilen her bir ağ aygıtına ait yapılandırma dosyasıdır. ifcfg-eth0 ilk arayüz belirtir, eğer ikinci bir arayüz varsa bu ifcfg-eth1 dosyasında bu arayüzlere ait yapılandrımalar bulunur.

/etc/sysconfig/network-scripts/ifcfg-eth0 dosyası aşağıdaki gibidir:

DEVICE=eth0

IPADDR=208.164.186.1

NETMASK=255.255.255.0

NETWORK=208.164.186.0

BROADCAST=208.164.186.255

ONBOOT=yes

BOOTPROTO=none

USERCTL=no

NOT: Yukarıda belirtilen ifcfg-ethN şeklindeki dosyaların isimlerindeki ethN ifadesi sistemdeki ağ arabiriminin de aynı adla tanımlandığının göstergesi değildir. Buradaki asıl tanımlama ifcfg-ethN dosyası içindeki DEVICE tanımının karşılığıdır. Linux çekirdeğinin tanıdığı ağ kartlarını görmek için /proc/net/dev dosyasının içeriğine bakılmalıdır.

# cat /proc/net/dev

Interface |Receive | Transmit

| bytes packets errs drop fifo frame compressed multicast | bytes packets errs drop fifo colls carrier compressed

lo: 208664 818 0 0 0 0 0 0 208664 818 0 0 0 0 0 0

eth0:1656757796 20692193 17 251 0 0 0 0 15641676 113184 0 0 0 0 0 0

sit0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Eğer elle sisteme ait ağ adresinde değişiklik yapmak istenirse, veya yeni bir arayüz üzerinde yeni bir ağ eklenmek istenirse, ifcfg-ethN dosyası düzenlenir veya yeni bir tane oluşturulur, uygun değişiklikler yapılabilir.

Bu dosyada çoğunlukla kullanılan terimler aşağıdaki gibidir :

DEVICE=<devicename> - Ağ aygıtının veya arayüzünün adınıbelirtir.

IPADDR=<ipaddr> - Ağ aygıtına ait IP adresini belirtir.

NETMASK=<netmask> - Ağ maskesi değerini belirtir.

NETWORK=<network> - Bulunulan ağa ait ağ adresini belirtir.

BROADCAST=<broadcast> - Ağa ait broadcast adresini belirtir.

ONBOOT=[yes|no] - Makine boot edip açılırken aygıtın aktif olup olmamasını belirtir. Eğer yes yazılırsa, açılışta aygıt aktif hale gelecektir.

BOOTPROTO=<proto> buraya gelebilecek değerler şunlardır:

none - Herhangi bir boot zamanı protokolü kullanılmayacaktır. static deyimi de kullanılabilir.

bootp - now pump protokolü kullanılacaktır.

dhcp - dhcp protokolü kullanılacaktır

USERCTL=[yes|no] - yes yazılırsa root dışındaki kullanıcılara da bu aygıtı kontrol etme izni verilmiş olur, no sadece super-user root bir aygıtı kontrol edebilir.

**/etc/sysctl.conf Dosyası**

Çekirdek parametrelerinin açılışta okunduğu dosyadır, ağ yapılandırmasına ilişkin bazı değerler burada yer alır.

Yukarıdaki örnekler verilen tanımlamalar çekirdeği direkt olarak etkileyerek istenen veya istenmeyen özelliklerin çekirdek seviyesinde halledilmesi sağlanır. Bu dosyada yapılan değişikliklerin devreye alınması için sistemin yeniden başlatılması veya sysctl -p komutunun çalıştırılması gerekmektedir.

NOT: Linux sistemlerin ağ yapılandırmalarına yönelik çekirdek seviyesinde yapılan yukarıdaki işlemler benzer şekilde aşağıdaki gibi de yapılabilir.

IPV4 yönlendirme açıp kapamak için:

net.ipv4.ip\_forward değeri (0=kapalı; 1=açık) şeklinde ayarlanır.

Sistemin tüm ping isteklerine kapatılması için:

net.ipv4.icmp\_echo\_ignore\_all (0=pinge açık; 1=pinge kapalı) şeklinde ayarlanır.

net.ipv4.ip\_forward değeri /proc/sys/net/ipv4/ip\_forward dosyasına,

net.ipv4.icmp\_echo\_ignore\_all değeri /proc/sys/net/ipv4/icmp\_echo\_ignore\_all dosyasına yazılabilir.

IPV4 yönlendirme açıp kapamak için:

# echo 0 > /proc/sys/net/ipv4/ip\_forward

Sistemin tüm ping isteklerine kapatılması için:

# echo 1 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

**/etc/hosts dosyası**

/etc/hosts dosyası DNS servisi kullanılmadan önce geçerli olan metottur. İsim çözme sunucusu (Name Server) olmadığı durumda, sisteminizdeki uygulamalar makine adına bağlı olarak ip adresi belirlemek için bu dosyaya başvurur.

Bu dosyada sisteme ait makine adı, tam nitelikli alan adı(FQDN) bilgileri bulunur.

/etc/hosts dosyasındaki satırlar aşağıdaki formata sahiptir:

192.168.1.200 ila.itu.edu.tr ila

127.0.0.1 localhost.localdomain localhost

(Soldaki sütunda çözümlenecek ip adresi, yanındaki sütunda ise makine adı bilgisi bulunur.)

"127.0.0.1 localhost.localdomain localhost" kısmı varsayılan (default) olarak makine kurulumunda gelmektedir. Bu ayarla makine loopback arabirim adresine ait IP (127.0.0.1) ile localhost makine adı çözümlemesi yapabilir.

Bu kaydı sildiğimizde makineye kendi üzerinden ping atmaya çalıştığımızda

# ping localhost

ping: unknown host localhost

şeklinde yanıt döner. Buradan da anlaşılacağı üzere makine localhost isim tanımlamasını çözememiştir.

Bunun dışında /etc/hosts dosyasına kullandığımız makinenin veya ağdaki herhangi bir makinenin IP'si, makineye ait FQDN adı ve makinenin adı da girilebilir. Her makine için ayrı bir kayıt olmak zorundadır.

Ağ üzerindeki bilgisayarların artmasıyla birlikte bu dosyanın kullanılması imkansız hale gelmiş ve DNS mimarisi geliştirilmiştir.

**/etc/resolv.conf**

Makine adı için IP adresi tanımlamalarının kararlaştırılmasında kullanılan dosyadır. Bu dosyada isim sunucularına (DNS, nameserver) ait IP adresleri bulunur. Makine bu sunucularla yaptığı sorgulamalar sonucunda ilgili isimdeki makinelerin IP adreslerine ulaşmaktadır.

Bu dosya aşağıdaki yapıdadır:

search itu.edu.tr

nameserver 192.168.1.100

nameserver 192.168.1.101

nameserver 192.168.1.102

search öncelikli olarak hangi etki alanında tamamlama yapıp öncelikli arama yapacağını gösterir. Örneğin yukarıdaki örnekte www şeklinde bir adres ilk olarak www.itu.edu.tr şeklinde isim sunucularına sorulur. Eğer bu şekilde kayıt bulunamaz ise www şeklinde sorulur.

nameserver çözümleme yapacak DNS sunucularının IP adreslerini belirtir. Burada dikkat edilmesi gereken önemli bir husus ilk yazan sunucu birincil (primary) DNS'dir. Onun için hangi DNS sunucusunu birincil (primary) olarak atanmak isteniyorsa o ilk başa yazılmalıdır.

**/etc/nsswitch.conf ve /etc/host.conf Dosyaları**

Bu dosyalar isim-ip çözümlemesinde önceliklendirmelerin belirtilebildiği dosyalardır. Örneğin bir ip’nin veya dns adı’nın hem /etc/hosts dosyasındaki kayıtlardan hem de DNS sorgularından karşılıkları öğrenilebilir. Bu dosyalar çözümlemede dosyanın mı ya da DNS sorgulamasının mı öncelikli olacağını belirler.

Bu iki dosyadan baskın olan nssswitch.conf dosyası içindeki hosts ibaresi isim çözümlemesi için önceliklendirmeleri belirler ve varsayılanda aşağıdaki gibidir.

hosts: files dns

yani öncelikli olarak dosya içeriğine bakacak ve daha sonra dns sorgusu yapacak. Buradaki tanımlamalar değiştirilerek önceliklendirmeler düzenlenebilir. Bu şekilde olan yapılandırmada /etc/host.conf dosyasının içeriği de önemlidir. /etc/host.conf dosyası varsayılanda aşağıdaki gibidir.

# cat /etc/host.conf

order hosts,bind

**ifconfig**

ifconfig komutu makinenin ağ ayarlarının yapılandırmasını öğrenmek veya ağ yapılandırması yapmak için kullanılır.

Ancak bu komutla yapılan yapılandırmalar bilgisayar yeniden açıldığında veya servis tekrar başladığında eski haline dönecektir.

ifconfig <arayüz\_ismi> <IP\_adresi> [ netmask ağ\_maskesi broadcast yayın\_adresi]

# ifconfig

Eth0 Link encap:Ethernet HWaddr 00:0C:29:36:1F:FD

inet addr:192.168.1.200 Bcast:192.168.1.201 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe36:1ffd/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:333112 errors:0 dropped:0 overruns:0 frame:0

TX packets:4266 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:100067509 (95.4 MiB) TX bytes:1204914 (1.1 MiB)

Interrupt:10 Base address:0x1080

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:1786 errors:0 dropped:0 overruns:0 frame:0

TX packets:1786 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:2353842 (2.2 MiB) TX bytes:2353842 (2.2 MiB)

# ifconfig eth0 192.168.1.100 netmask 255.255.255.0

Yeni sistemlerde ağ aygıtlarına ilişkin yapılandırmalar ip komutu ile de yapılmaktadır.

# ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 16436 qdisc noqueue

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast qlen 1000

link/ether 00:0c:29:9c:28:1d brd ff:ff:ff:ff:ff:ff

inet 160.75.5.177/24 brd 160.75.5.255 scope global eth0

inet6 2001:a98:8000:f:20c:29ff:fe9c:281d/64 scope global dynamic

valid\_lft forever preferred\_lft forever

inet6 2001:a98:8000:5:20c:29ff:fe9c:281d/64 scope global dynamic

valid\_lft 2591939sec preferred\_lft 604739sec

inet6 fe80::20c:29ff:fe9c:281d/64 scope link

valid\_lft forever preferred\_lft forever

3: sit0: <NOARP> mtu 1480 qdisc noop

link/sit 0.0.0.0 brd 0.0.0.0

**Sanal Ağ Birimleri**

Sanal ağ birimleri sistem üzerinde gerçekte var olmayan ama sanal olarak bir ağ arabirimi altında çalışabilen alt arabirimlerdir. Örneğin eth1 arabiriminin altındaki sanal arabirimler eth1:N şeklinde belirtilebilir. Bu sanal arabirimler hem bir yapılandırma dosyası aracılığı ile hem de ifconfig komutu kullanılarak sisteme eklenebilir.

# cat /etc/sysconfig/network-scripts/ifcfg-eth0:0

DEVICE=eth0:0

BOOTPROTO=static

IPADDR=192.168.0.1

NETMASK=255.255.255.0

ONBOOT=yes

# ifup eth0:0

# ifconfig eth0:0

inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Interrupt:67 Base address:0x2000

# ifconfig eth0:1 192.168.1.100 netmask 255.255.255.128

# ifconfig eth0:1

eth0:1 Link encap:Ethernet HWaddr 00:0C:29:9C:28:1D

inet addr:192.168.1.100 Bcast:192.168.1.127 Mask:255.255.255.128

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Interrupt:67 Base address:0x2000

**netstat**

netstat komutu ağ bağlantıları, yönlendirme tablosu , arayüz istatistikleri gibi ağ bağlantıları ile ilgili temel bilgileri göstermeye yarayan bir programdır.

netstat [seçenekler]

--numeric, -n - Makine adı, port, kullanıcı adı bilgilerini sayısal olarak gösterir.

--program, -p - Çalışan süreçlerin PID numaralarını gösterir.

--all, -a - Tüm bağlantıları gösterir(listening, non-listening sockets)

--tcp, -t - TCP bağlantılarını gösterir.

--udp, -u - UDP bağlantılarınıgösterir.

--route, -r - Yönlendirme tablosunu gösterir.

# netstat –punta

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 25747/mysqld

tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 2498/portmap

tcp 0 0 127.0.0.1:631 0.0.0.0:\* LISTEN 2823/cupsd

tcp 0 0 0.0.0.0:25 0.0.0.0:\* LISTEN 15564/sendmail: acc

tcp 0 0 0.0.0.0:1020 0.0.0.0:\* LISTEN 2534/rpc.statd

tcp 0 0 160.75.5.177:3306 160.75.5.81:50884 ESTABLISHED 25747/mysqld

tcp 0 0 :::80 :::\* LISTEN 7404/httpd

tcp 0 0 :::22 :::\* LISTEN 24049/sshd

tcp 0 148 ::ffff:160.75.5.177:22 ::ffff:160.75.5.81:64949 ESTABLISHED 3045/0

tcp 0 0 ::ffff:160.75.5.177:22 ::ffff:160.75.5.81:51065 ESTABLISHED 27474/1

udp 0 0 0.0.0.0:514 0.0.0.0:\* 20722/rsyslogd

udp 0 0 0.0.0.0:68 0.0.0.0:\* 2359/dhclient

udp 0 0 0.0.0.0:59980 0.0.0.0:\* 2952/avahi-daemon:

udp 0 0 0.0.0.0:5353 0.0.0.0:\* 2952/avahi-daemon:

udp 0 0 0.0.0.0:111 0.0.0.0:\* 2498/portmap

udp 0 0 0.0.0.0:1014 0.0.0.0:\* 2534/rpc.statd

udp 0 0 0.0.0.0:631 0.0.0.0:\* 2823/cupsd

udp 0 0 0.0.0.0:1017 0.0.0.0:\* 2534/rpc.statd

udp 0 0 160.75.5.177:123 0.0.0.0:\* 2842/ntpd

udp 0 0 127.0.0.1:123 0.0.0.0:\* 2842/ntpd

udp 0 0 0.0.0.0:123 0.0.0.0:\* 2842/ntpd

udp 0 0 :::514 :::\* 20722/rsyslogd

udp 0 0 :::53081 :::\* 2952/avahi-daemon:

udp 0 0 :::5353 :::\* 2952/avahi-daemon:

udp 0 0 fe80::20c:29ff:fe9c:123 :::\* 2842/ntpd

udp 0 0 2001:a98:8000:5:20c:123 :::\* 2842/ntpd

udp 0 0 2001:a98:8000:f:20c:123 :::\* 2842/ntpd

udp 0 0 ::1:123 :::\* 2842/ntpd

udp 0 0 :::123 :::\* 2842/ntpd

**ping**

ping ağdaki bir makineye ulaşılıp ulaşılmadığı kontrol etmek için kullanılır.

# ping 192.168.1.200

PING 160.75.5.234 (192.168.1.200) 56(84) bytes of data.

64 bytes from 192.168.1.200: icmp\_seq=0 ttl=64 time=1.38 ms

64 bytes from 192.168.1.200: icmp\_seq=1 ttl=64 time=0.279 ms

64 bytes from 192.168.1.200: icmp\_seq=2 ttl=64 time=0.275 ms

64 bytes from 192.168.1.200: icmp\_seq=3 ttl=64 time=0.280 ms

64 bytes from 192.168.1.200: icmp\_seq=4 ttl=64 time=0.276 ms

---192.168.1.200 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4000ms

rtt min/avg/max/mdev = 0.275/0.498/1.382/0.442 ms, pipe 2

Sistem üzerinde birden fazla tanımlı ip varsa bu ip’lerden ping atılması –I anahtarı ile sağlanabilir.

# ping -I 192.168.1.100 192.168.0.1

PING 192.168.0.1 (192.168.0.1) from 192.168.1.100 : 56(84) bytes of data.

64 bytes from 192.168.0.1: icmp\_seq=1 ttl=64 time=0.083 ms

64 bytes from 192.168.0.1: icmp\_seq=2 ttl=64 time=0.046 ms

Linux sistemlerde ping komutu sonsuz döngü ile çalışır. Yani ping’lenen bir ip sonsuza durdurulmadığı takdirde sonsuza kadar ping’lenir. Bunun önüne geçmek için –c anahtarı kullanılabilir. Örneğin 3 kez ping paketi göndermek için ping komutu aşağıdaki gibi kullanılır.

# ping -c 3 www.google.com

PING www.l.google.com (74.125.87.99) 56(84) bytes of data.

64 bytes from hb-in-f99.1e100.net (74.125.87.99): icmp\_seq=1 ttl=57 time=24.2 ms

64 bytes from hb-in-f99.1e100.net (74.125.87.99): icmp\_seq=2 ttl=57 time=24.4 ms

64 bytes from hb-in-f99.1e100.net (74.125.87.99): icmp\_seq=3 ttl=57 time=24.0 ms

--- www.l.google.com ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2459ms

rtt min/avg/max/mdev = 24.053/24.262/24.448/0.162 ms

**host**

host komutu sistemde kayıtlı isim çözme sunucularından (DNS sunucuları) basit anlamda sorgu yapmak için kullanılır.

# host www.google.com

www.google.com is an alias for www.l.google.com.

www.l.google.com has address 74.125.87.99

www.l.google.com has address 74.125.87.104

# host 127.0.0.1

1.0.0.127.in-addr.arpa domain name pointer localhost.

**nslookup**

nslookup komutu, isim-ip veya ip-isim çözümlemesi için kullanılır.

**# nslookup**

> ila.itu.edu.tr

Server:192.168.1.100

Address:192.168.1.100#53

Name: ila.itu.edu.tr

Address: 192.168.1.200

> server 192.168.1.101

Default server: 192.168.1.101

Address: 192.168.1.101#53

> ila.itu.edu.tr

Server:192.168.1.101

Address:192.168.1.101#53

Name: ila.itu.edu.tr

Address: 192.168.1.200

**dig**

host ve nslookup komutlarına göre daha karmaşık düzeyde DNS sorguları yapabilen bir komut satırı aracıdır.

dig [@nameserver] [-t query type] [-x ip address] hostname [queryoptions]

# dig @160.75.2.20 -t A www.google.com

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5\_4.2 <<>> @160.75.2.20 -t A www.google.com

; (1 server found)

;; global options: printcmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7073

;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;www.google.com. IN A

;; ANSWER SECTION:

www.google.com. 83859 IN CNAME www.l.google.com.

www.l.google.com. 156 IN A 74.125.87.104

www.l.google.com. 156 IN A 74.125.87.99

;; Query time: 4 msec

;; SERVER: 160.75.2.20#53(160.75.2.20)

;; WHEN: Tue Sep 7 17:24:26 2010

;; MSG SIZE rcvd: 84

# dig -x 160.75.5.81

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5\_4.2 <<>> @160.75.2.20 -x 160.75.5.81

; (1 server found)

;; global options: printcmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63297

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;81.5.75.160.in-addr.arpa. IN PTR

;; ANSWER SECTION:

81.5.75.160.in-addr.arpa. 3600 IN PTR ao081.cc.itu.edu.tr.

;; Query time: 5 msec

;; SERVER: 160.75.2.20#53(160.75.2.20)

;; WHEN: Tue Sep 7 17:25:30 2010

;; MSG SIZE rcvd: 75

**whois**

whois komutu bir IP adresinin hangi ağa dahil olduğunu ve o ağdan sorumlu kişilerin e-posta adresleri, posta adresleri , telefonları gibi bilgileri gösteren bir komuttur. Temelde bir IP bloğu alındığı takdirde, bloğu satın alan ile ilgili bilgiler alınır ve bu bilgiler whois sunucularında tutulurlar. whois komutu ile bu sunucular sorgulanır.

whois <IP\_Adresi> <@whois\_sunucusu>

# whois 18.7.22.83

[Querying whois.arin.net]

[whois.arin.net]

OrgName: Massachusetts Institute of Technology

OrgID: MIT-2

Address: Room W92-190

Address: 77 Massachusetts Avenue

City: Cambridge

StateProv: MA

**traceroute**

traceroute komutu, bir makineye hangi ağ cihazları üzerinden ulaşıldığını gösteren komuttur. Komuta –n seçeneği verilerek isim çözümlemesi yapmaması sağlanabilir.

# traceroute -n www.yahoo.com

traceroute: Warning: www.yahoo.comhas multiple addresses; using 68.142.197.78

traceroute to www.yahoo.akadns.net(68.142.197.78), 30 hops max, 38 byte packets

1 160.75.5.254 0.291 ms 0.202 ms 0.200 ms

2 160.75.126.34 91.534 ms 87.760 ms 89.820 ms

3 193.255.0.33 89.665 ms 88.332 ms 93.416 ms

4 195.175.51.65 93.162 ms 78.439 ms 73.135 ms

5 208.48.239.5 137.630 ms 119.094 ms 105.904 ms

6 67.17.67.57 190.499 ms 192.803 ms 216.150 ms

7 \* 208.50.13.210 206.770 ms 208.51.74.182 326.586 ms

8 216.115.101.145 200.893 ms 216.115.101.147 295.195 ms 216.115.101.145 260.826 ms

9 216.115.104.111 344.088 ms \* 216.115.104.101 254.605 ms

10 68.142.193.29 244.475 ms 68.142.193.25 256.711 ms 263.773ms

11 68.142.197.78 374.899 ms 401.313 ms 236.756 ms

**lftp**

lftp programı ftp sunucusuna bağlanmak için kullanılır. BASH kabuğuna benzer şekilde uzak ftp sunucusunda ([TAB] tuşuyla) dosya adı tamamlama gibi özellikleri vardır. lftp komut istemcisinde ? komutu yürütülerek yardım alınabilir.

# lftp ftp.itu.edu.tr

lftp ftp.itu.edu.tr:~> cd

aes/ cisco/ pub/

lftp ftp.itu.edu.tr:~> cd pub/

linux/ utils/

lftp ftp.itu.edu.tr:~> cd pub/linux/

RedHat/ fedora/

lftp ftp.itu.edu.tr:~> cd pub/linux/fedora/core/

1/ 2/ 3/ 4/

lftp ftp.itu.edu.tr:~> cd pub/linux/fedora/core/4/

Fedora/ images/ isolinux/

lftp ftp.itu.edu.tr:~> cd pub/linux/fedora/core/4/Fedora/

RPMS/ base/

lftp ftp.itu.edu.tr:~> cd pub/linux/fedora/core/4/Fedora/RPMS/

lftp ftp.itu.edu.tr:/pub/linux/fedora/core/4/Fedora/RPMS> get perl-DBI-1.48-4.i386.rpm

584550 bytes transferred

lftp ftp.itu.edu.tr:/pub/linux/fedora/core/4/Fedora/RPMS> exit

**wget**

wget web sunucularındaki dosyaları indirmek için kullanılan bir araçtır. HTTP, HTTPS, FTP protokollerini destekler.

# wget ftp://ftp.itu.edu.tr/pub/linux/fedora/core /4/Fedora/RPMS/webalizer-2.01\_10-28. i386.rpm

--20:55:00--ftp://ftp.itu.edu.tr/pub/linux /fedora/core/4/Fedora/RPMS/ webalizer-2.01\_10-28.i386.rpm

=> `webalizer-2.01\_10-28.i386.rpm'

Resolving ftp.itu.edu.tr... 160.75.2.23

Connecting to ftp.itu.edu.tr|160.75.2.23|:21... connected.

Logging in as anonymous ... Logged in!

==> SYST ... done. ==> PWD ... done.

==> TYPE I ... done. ==> CWD /pub/linux/fedora/core/4/Fedora/RPMS ... done.

==> PASV ... done. ==> RETR webalizer-2.01\_10-28.i386.rpm ... done.

Length: 102,594 (100K) (unauthoritative)

100%[=====================>] 102,594 --.--K/s

20:55:00 (6.99 MB/s) -`webalizer-2.01\_10-28.i386.rpm' saved [102594]

**route**

route komutu ile routing tablosunun düzenlenmesi ve tablonun görüntülenmesi sağlanabilir.

# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.1.0 \* 255.255.255.128 U 0 0 0 eth0

160.75.5.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 \* 255.255.255.0 U 0 0 0 eth0

default 160.75.5.254 0.0.0.0 UG 0 0 0 eth0

**Grafik Ağ Yapılandırma Aracı**

Linux’te grafik ortamda ağ ayarlarının yapılmasıiçin System > Administration > Network menüsünden ya da system-config-network komutu yürütülerek ağ yapılandırma aracına ulaşılabilir.

Eğer masaüstü ortamında değilde konsol üstünde ağ ayarlarını yapmak için system-config-network-tui komutu kullanılabilir.

Devices - Bu sekmede sistemimizde bulunan ağ aygıtları ve bu aygıtlara ait bilgiler gözükür, bu aygıtlara ait yapılandırmalarda düzenlemeler yapılabilir.

Hardware - Ağ aygıtı ile ilgili yapılandırma bilgilerin olduğu, yeni aygıt ekleme veya var olan aygıtta düzenlemelerin yapılabileceği sekmedir.

IPsec - VPN gibi ağ yapılarının kullanıldığı ve bunlar için gerekli ayarlamaların yapıldığı sekmedir.

DNS - Makinenin DNS ile çözümlemelerde başvuracağı sunucuların adreslerinin belirtildiği sekmedir.

Hosts - İsim çözümlemelerinde kullanılmak üzere kayıtların girildiği sekmedir.

**Ağ Servisinin Yönetilmesi**

Linux işletim sisteminde ağ desteği de bir servis olarak çalışmaktadır. RedHat türevi sistemlerde servislerin çalıştırılması ile ilgili scriptler /etc/rc.d/init.d dizini içinde bulunmaktadır. Ağ ayarlarının aktif hale getirilmesi veya ayarların aktif halden çıkarılması için bu dizin içinde bulunan network scripti kullanılmaktadır.

/etc/rc.d/init.d/network {start|stop|restart|reload|status}

Bu script kullanılarak ağ servisi başlatılabilir (start), durdurulabilir (stop), yeniden başlatılabilir (restart), yapılandırma dosyalarını yeniden okuması sağlanabilir (reload) ya da ağ ara yüzlerinin durumu kontrol (status) edilebilir.

Ayrıca bu amaçla, service network {start|stop|restart|reload|status} komutuda kullanılabilir.

# service network stop

Shutting down interface eth0: [ OK ]

Shutting down loopback interface: [ OK ]

# service network start

Bringing up loopback interface: [ OK ]

Bringing up interface eth0: [ OK ]

# service network restart

Shutting down interface eth0: [ OK ]

Shutting down loopback interface: [ OK ]

Bringing up loopback interface: [ OK ]

Bringing up interface eth0: [ OK ]

# service network status

Configured devices:

lo eth0

Currently active devices:

lo eth0

# service network reload

Shutting down interface eth0: [ OK ]

Shutting down loopback interface: [ OK ]

Bringing up loopback interface: [ OK ]

Bringing up interface eth0: [ OK ]

**UYGULAMA 1**

NOT: <sıra\_no>: Bilgisayarınız sıra numarasıdır. IP adresinizin son hanesi(Örn: 192.168.0.10) veya makine ağ adının rakam kısmı (Örn: ila10) sıra numaranızı gösterir.

1. IP adresinizi ve makine isminizi gözlemleyerek sıra numaranızı (<sıra\_no>) öğrenin.
2. Bilgisayarınızın IP adresini anlık olarak 192.168.<sıra\_no>.200 ağ maskesini de 255.255.0.0 olarak değiştirin.
3. 192.168.0.200 IP adresine ping atıp cevap aldığınızı gözlemleyin.
4. Network servisini yeniden başlatarak varsayılan ağ ayarlarınıza geri dönün.
5. www.lpi.org sitesinin IP adresini öğrenin.
6. Kendi IP adresinizin whois sorgusuna dönen sonucunu gözlemleyin.
7. whois sorgusunu tekrarlayın ama bu sefer sonuç kümesi içinde sadece mail: ( büyük küçük harf ayrımı olmaksızın) yazılı satırların gözükmesini sağlayın.
8. Bir üst soruda yaptığınız sorgu sonucunda "OrgAbuseEmail: abuse@iana.org" şekline benzer satırlar döner. Bu dönen sonuçlardan sadece email adresinin yazdığı kısmın dönmesini ve tekrarlayan adreslerin tek olarak gözükmesini sağlayın.

**ÇÖZÜMLER 1**

1-------------------------------

[root@ila2 ~]# hostname

ila2.example.org

[root@ila2 ~]# ifconfig

eth0 Link encap:Ethernet HWaddr 00:23:24:00:3E:B8

inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0

inet6 addr: fe80::223:24ff:fe00:3eb8/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:61743 errors:0 dropped:0 overruns:0 frame:0

TX packets:96623 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:7938559 (7.5 MiB) TX bytes:121699259 (116.0 MiB)

Memory:f0500000-f0520000

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:59095 errors:0 dropped:0 overruns:0 frame:0

TX packets:59095 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:155054071 (147.8 MiB) TX bytes:155054071 (147.8 MiB)

virbr0 Link encap:Ethernet HWaddr BA:14:09:53:40:D3

inet addr:192.168.122.1 Bcast:192.168.122.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:5322 errors:0 dropped:0 overruns:0 frame:0

TX packets:8405 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:746854 (729.3 KiB) TX bytes:3844536 (3.6 MiB)

vnet0 Link encap:Ethernet HWaddr BA:14:09:53:40:D3

inet6 addr: fe80::b814:9ff:fe53:40d3/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:1274 errors:0 dropped:0 overruns:0 frame:0

TX packets:6026 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:500

RX bytes:225600 (220.3 KiB) TX bytes:397877 (388.5 KiB)

[root@ila2 ~]#

2---------------------------------------

[root@ila2 ~]# ifconfig eth0 192.168.2.200 netmask 255.255.0.0

[root@ila2 ~]#

3----------------------

[root@ila2 ~]# ping 192.168.0.200

PING 192.168.0.200 (192.168.0.200) 56(84) bytes of data.

64 bytes from 192.168.0.200: icmp\_seq=1 ttl=64 time=0.199 ms

64 bytes from 192.168.0.200: icmp\_seq=2 ttl=64 time=0.168 ms

^C

--- 192.168.0.200 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1501ms

rtt min/avg/max/mdev = 0.168/0.183/0.199/0.020 ms

[root@ila2 ~]#

4---------------------------

[root@ila2 ~]# service network restart

Shutting down interface eth0: [ OK ]

Shutting down loopback interface: [ OK ]

Bringing up loopback interface: [ OK ]

Bringing up interface eth0:

Determining IP information for eth0... done.

[ OK ]

[root@ila2 ~]#

5----------------

[root@ila2 ~]# host www.lpi.org

www.lpi.org has address 69.90.69.242

[root@ila2 ~]#

6-------------------------

[root@ila2 ~]# whois 192.168.0.2

...

[root@ila2 ~]#

7-------------------------

[root@ila2 ~]# whois 192.168.0.2 | grep -Fi mail:

OrgAbuseEmail: abuse@iana.org

OrgTechEmail: abuse@iana.org

[root@ila2 ~]#

8---------------------------

[root@ila2 ~]# whois 192.168.0.2 | grep -Fi mail: | awk '{print $2}' | sort -u

abuse@iana.org

[root@ila2 ~]#

**ZAMANLAYICI**

**Zamanlanmış Görevler**

NOT:

makinenin başında sistem admini olarak beklemiyeyim amacıyla,ileriye dönük ve zamanlanmış görevler vardır. 3 tanedir.

1.cron:periyodik olarak yapılacak işler örn :2 saatte bir

2.atd:ileriye yönelik herhangi bir zamanda 1 kez yapılacaksa.

3.anacron:yine periyodik ama crondan farkı,cron makine kapalı mı açık mı kontrol etmez.ama anacron buna dikkat eder.bir işimiz 3 saatte bir yapılacak diyelim ama makine 4 saattir kapalı anacron bu durumu dikkate alır.

-cron servisi:service crond,bu servisin çalışır durumda olması gerekir.

-service crond status ile görebiliriz.

yapılandırma dosyası:/etc/crontab dır.

SYNTAX ı; DAKİKA SAAT AYINGUNU AY HAFTANIN GUNU KULLANICI KOMUT

dak saat ayıngunu ay haftanıngunu kullanıcı komut

\* \* \* \* \* - -

-Cron servisi her dakika bu patterni kontrol eder yani saniye mertebesinde birşey yapmak istersek betik yazmak zorundayız.

30 20 \* \* \* - - :hergun saat 20.30 da çalışacak komut

30 20 1 \* \* - - :her ayın ilkgunu 20.30 da çalış

30 20 1 6 \* - - :her yılın 6. ayının ilk gunu 20.30 da çalış

30 8,20 \* \* \* - - : saat 8.30 ve 20.30 da çalış, virgulle isediğimiz kadar ayırma yapabiliriz.

\* 8-20 \* \* \* - - :saat 8-20 arasındaki saatler

\* 30 8-20,23 \* \* \* - -:8-20 arsı ve 23 dahil yarım saatte bir çalış.

20 8-18 \* \* 1-5 \* - -:mesai saatlerinde çalış (cron resmi tatilleri takip edemez bunun için bir plugin yazılabilir.)

\*/2 \* \* \* \* - - :her saatim tam bölünenlerinde iş yap.

cron.hourly :d dosyalar ve bu dizinlerin altında her saat başı çalışacak işler var mesela illa cron ile editlemek zorunda değiliz.

0anacron:bir servis ilk çalışacaksa kendi ismini düzenler, çünkü bu dizinler altındaki betikler alfabetik sırada çalışır.

-/etc/cron.daily:günlük işleri tutar.

-mesela locate hergun kendi veri tabanını günceller demiştik.. burda var o dosya.

-vim /etc/cron.daily/mlocate.cron dersek nasıl yaptıgını görebiliriz.

-renice +19 -p :önceliğini düşürebiliriz.(sistem kasmasın diye yapılır genelde.)

-normalde cron ile yapılcak iş için servisi kapatıp yeniden açmaka gerekir.Ama crontabta böyle değil.

\*\*\*crontab -e dersek:

\* \* \* \* \* date >/dev/pts/2 ==> dakikada bir konsola tasrih bas.

-kullanıcının yarattıgı cronlar /var/spool/cron/root 'a bir dosya olarak atılır. cron komutu dakikada bunun altında dosya olup olmadığının kontrol eder.

-cron çalışın /var/log/cron un altına bilgi atar.

-cron -r ile bütün işleri yok edebiliriz.(tehlikelidir r nin yanında e var çünkü editlemek isterken uçurabiliriz.)

\*\*\*\*\*\*\*\*\*at servisi\*\*\*\*\*\*

-service atd : çalışması lazım . kapatmadığımız sürece çalışır.ileriye dönük bir işi 1 kez yapar.

syntax ı :

at now +2 minutes enterladık;şuandan itibaren 2 dk sonrasına iş yazıyoruz.

at>date/dev/pts/2

at>echo gec oldu > /dev/pts/2

ctrl+d ile çıkış yapabiliriz

--atq komutu ile kuyrukta bekleyen işleri görebiliriz.

-at nin kötü yani iş çalışır,kayıt tutulmaz,loglamaz.

-at now +10 minutes

-at now +2 hours

-at 4:00 5/11/2006 --->nokta atışı tarihte verilebilir.

--kuyruktaki işi silmek için

-at rm işnumarası(ensoldaki sayı)

\*\*\*\*\*\*\*\*\*\*\*\*ANACRON\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Cron,at : makinenin açık yada kapalı olduğunun takip etmez.

-anacron ,gün periyodu ile çalışır.(saat dakika falan yok)

normalde cron a destek için tasarlanmıştır.

-vim /etc/anacrontab : yapılandırma dosyasını açalım.

SHELL = hangi kabukla çalıştıracan.

PATH = PATH de olmayan bir işi tanımlarsak çalışmaz.

Burda komutları tam yol yazmak path den kaçan komutları engellemiş oluruz.

MAILTO=raporu mail atabiliriz.

-period in days deki 2.sutun delaydır.Periyodlar halinde çalışırkenki aralıktır.

-Genelde CRON kullanılır.Anacron u aslında cron dürter. bu cronun attığı dosyalardan da görebiliriz.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Linux işletim sisteminde belirli görevlerin istenildiği zaman otomatik olarak yapılmasını sağlayan zamanlayıcı servisleri bulunur. Bu sayede istenilen işlemler makine başında olmadan da yapılabilir.

Bu servis genellikle otomasyon ve yedekleme amaçlarıyla sunucu yükünün az olduğu zamanlarda ya da periyodik olarak, bazı programların çalışmasını sağlar.

locate komutunun kullandığı slocate veritabanı her gün güncellenir, yedekleme işlemleri genelde sistem kaynaklarının az kullanıldığı saatlerde yapılır.

Linux’te zamanlanmış görevlerin yürütülmesi için cron, at ve anacron servisleri kullanılır. Bu servisler:

crond istenilen işin istenilen zamanda yapılması için,

atd bir seferlik (periyodik olmayan) ileriye yönelik işlem yapılması için,

anacron ise istenilen işin belirli bir zaman diliminde yapılması için kullanılır.

**Cron Servisi**

Cron daemon tarafından kontrol edilen cron servisi, her günün her saatinin her dakikası bir takım dosyalara bakarak çalışacak işleri kontrol eder./var/spool/cron ve /etc/cron.d altındaki dosyalara ve /etc/crontab dosyalarını kontrol ederek yazan komutları çalıştırır. Cron servisi sistemin her an çalıştığını varsayar; sistemin kapalı olduğu zaman diliminde zamanlanmış bir görev varsa, yerine getirilmez.

**Cron Dosya ve Dizinleri**

/etc/crontab - Sistemin cron dosyasıdır, sadece yetkili kullanıcı düzenleme yapabilir.

/etc/cron.d - Cron dosyalarını içeren dizindir.

/etc/cron.hourly - Saat başı gerçekleştirilen işlemlerin tutulduğu dizindir.

/etc/cron.daily - Hergün gerçekleştirilen işlemlerin tutulduğu dizindir.

/etc/cron.weekly - Her hafta gerçekleştirilen işlemlerin tutulduğu dizindir.

/etc/cron.montly - Her ay gerçekleştirilen işlemlerin tutulduğu dizindir.

/etc/cron.yearly - Her yıl gerçekleştirlen işlemlerin tutulduğu dizindir.

/etc/cron.allow - Cron servisini kullanabilen kullanıcıların tanımlanabildiği dosyadır.

/etc/cron.deny - Cron servisini kullanması istenmeyen kullanıcıların tanımlanabildiği dosyadır.

/var/spool/cron - Kullanıcıların cron dosyalarının tutulduğu dizindir.

**Cron Yapılandırması**

Cron yapılandırma dosyalarının özel bir formatı vardır. Dosyada boş satırlar bulunabilir, açıklama satırları # işareti ile başlar. Boş ve açıklama satırları göz önünde bulundurulmaz. Sistemin genel cron yapılandırma dosyası /etc/crontab dosyasıdır.

Cron yapılandırma dosyasında öntanımlı SHELL, PATH, MAILTO ve HOME çevre değişkenleri bulunur. SHELL çevre değişkeni dışındaki değişkenler standart çevre değişkenlerinden farklıdırlar. MAILTO değişkeni cron mesajlarının hangi e-posta adresine gönderileceğini belirtir. PATH değişkeni kabuğun PATH çevre değişkeninden farklı değere sahip olduğu için, bu değişkenin cron'un komutları arayacağı dizinleri içerecek şekilde güncellenmesi gerekir ya da kullanılan her komut için tam yol verilmelidir.

[root@ila ~]# cat /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

HOME=/

Bu değişkenler haricinde yeni çevresel değişkenler tanımlanabilir.

Cron yapılandırma dosyasında yapılacak işlemlerin tanımlandığı satırlar aşağıdaki biçimdedir:

#Dakika Saat Ayın\_Günü Ay Haftanın\_Günü Komut

\* \* \* \* \* <komut>

Dakika 0-59

Saat - 0-23 (24 saat formatında)

Ayın\_Günü - 1-31

Ay - 1-12 ya da jan,feb,mar,...

Haftanın\_Günü - 0-7 (0=pazar, 7=pazar) ya da sun,mon,tue,...

Komut - Çalıştırılmak istenen komut

Herhangi bir değerdeki yıldız karakteri (\*), o alanın alabileceği tüm değerler için komutun yürütüleceği anlamına gelir.

Dakika alanındaki (\*) karakteri komutun belirlenen saatin her dakikasında çalıştırılacağını gösterir.

Virgül (,) ile değerleri ayırma; virgülle ayrılan her bir değeri belirtir. (6,12,18)

Tire (-) kullanımı; belli bir aralık belirtir.(0-30). Örneğin saat kısmındaki 8-20 ifadesi görevin saat 8:00 ile 20:00 arasında çalıştırılmasını sağlar

Kesme işareti (/) kullanımı; bölmeli değer belirtir. Örneğin dakika kısmında \*/15 olması görevin her 15 dakika bir çalıştırılmasını sağlar.

**Cron Yapılandırması - Örnekler**

# Her ayın ilk günü saat 23.12’de etc yapılandırma dizininin yedeğini al.

12 23 1 \* \* cp –aufr /etc /yedek

# Pazartesi, Salı, Çarşamba, Perşembe, Cuma günleri saat 14.30’da sistemi güncelle.

30 14 \* \* 1-5 /usr/bin/yum update >> /var/log/yumUpdate.log

# Her saatin 3, 17, 39 ve 51. dakikalarında bellek kullanımını /tmp/meminfo dosyasına yaz.

3,17,39,51 \* \* \* \* cat /proc/meminfo >> /tmp/meminfo

# 2 dakikada bir otomasyon scriptini çalıştır.

0-59/2 \* \* \* \* /root/bin/otomasyon.pl >> /var/log/otomasyon.log

# Pazartesi günleri saat 22.05’de sistemde tanımlı hiçbir kullanıcıya ait olmayan dosyaları bul.

5 22 \* \* 1 find / -nouser >> /tmp/nouserFiles

**crontab**

crontab, cron işlerini düzenlemek için kullanılan bir komuttur.

crontab [-u <kullanıcı>] <dosya>

crontab [-u <kullanıcı>] [-l | -r | -e][-i]

-u <kullanıcı> - Yetkili kullanıcının diğer kullanıcıların cron dosyalarını düzenlemesi için kullanılır.

-l - Cron dosyalarındaki işleri listelemek için kullanılır.

-r - Varolan cron işlerini silmek için kullanılır.

-e - Cron dosyalarındaki işleri düzenlemek için kullanılır, crontab vi editörünü kullanır.

Komuta bir dosya parametre verilmezse, crontab kullanıcının /var/spool/cron dizinindeki cron dosyası üzerinde işlem yapar.

crontab komutunun bir dosya parametre verilerek çalıştırılması, dosyadaki işlerin cron dosyasının üzerine yazılmasına; var olan cron işlerinin silinmesine neden olur!

**Zamanlanmış Görevlerin Düzenlenmesi**

Sistemde çalışacak zamanlanmış görevler belirli aralıklarla, periyodik olarak çalışması gereken yönetimsel görevler ve belirli zamanlarda çalışması gereken özelleşmiş görevler olarak ikiye ayrılabilir.

Periyodik olarak çalışması gereken yönetimsel işler genelde cron’un ilgili dizinindeki kabuk programları olarak düşünülebilir. Bu amaçla sistemde cron.daily, cron.weekly, cron.mothly ve cron.yearly dizinleri bulunur. Her dizin aynı zamanda çalışacak programlar içerir ve bunlara ilişkin cron yapılandırması /etc/crontab dosyasında mevcuttur. Bu işlemler parametre olarak verilen dizinin altındaki tüm programları çalıştıran /usr/bin/run-parts kabuk programı tarafından gerçekleştirilir.

[root@ila ~]# cat /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

HOME=/

Bu dizinler altında programlar, cron tarafından alfabetik sıra ile çalıştırılırlar. Eğer bir programın diğer programlardan önce çalıştırılması isteniyorsa genellikle program dosyasının adı değiştirilir.

Bu yapıya benzer şekilde aynı anda çalışması istenen programlar için yeni dizinler eklenebilir. Cron’un bu dizinleri işlemesi içinse, dizine ilişkin yapılandırma /etc/crontab dosyasına eklenir.

# Her Cuma saat 13:00’da /etc/cron.belgeler dizini altındaki programları çalıştır.

\* 13 \* \* 5 root run-parts /etc/cron.belgeler

Diğer görevler ise /etc/crontab dosyasında ya da kullanıcının /var/spool/cron dizini altındaki dosyasında tutulabilir.

**Loglar ve Kullanım İzinleri**

Cron’un çalışmasına ilişkin loglar /var/log/cron dosyasında tutulur.

[root@ila ~]# ls -lh /var/log/cron

-rw-------1 root root 101K Mar 17 20:54 /var/log/cron

Kullanıcıların cron kullanımını kontrol altına alabilmek için /etc/cron.allow ve /etc/cron.deny dosyaları kullanılır. Bu dosyaların bulunmaması kullanıma ilişkin bir kısıtlamanın getirilmediğini gösterir.

/etc/cron.allow dosyasında kullanıcı adları belirtilmişse diğer tüm kullanıcılar cron kullanamazlar. Eğer /etc/cron.allow dosyası sistemde yoksa sadece /etc/cron.deny dosyasındaki kullanıcılar cron kullanamazlar.

Dosya formatı her satırda bir kullanıcı adı olacak şekildedir.

[root@ila~]# cat /etc/cron.allow

root

[root@ila~]# su - ila1

[ila1@ila~]$ crontab -l

You (ila1) are not allowed to use this program (crontab)

See crontab(1) for more information

**at Servisi**

Linux’te cron gibi at servisi de zamanlanmış görevlerin yürütülmesini sağlar. at özellikle bir görev belli bir zamanda sadece bir kere çalıştırılmak istendiği zamanlarda kullanılır. At servisi bu görevleri /var/spool/at dizini altında, kuyrukta tutar.

[root@ila ~]# service atd status

atd(pid2207) is running...

**at Kullanımı**

Öncelikle at komutuna görevin ne zaman yerine getirileceği bilgisi parametre olarak verilir. Daha sonra at kabuğunda yürütülmesi istenen komut yazılır ve [CTRL]-[D] tuş kombinasyonu kullanılarak komuttan çıkılır.

at now + 10 minutes - Zamanlanan görev 10 dakika sonra başlatılacak.

at now + 2 hours - Zamanlanan görev 2 saat sonra başlatılacak.

at now + 1 day - Zamanlanan görev 1 gün sonra başlatılacak.

at now + 3 week - Zamanlanan görev 3 hafta sonra başlatılacak.

at teatime - Zamanlanan görev 16:00’da başlatılacak.

at 4:00 5/11/2006 - Zamanlanan görev 11 Mayıs 2006 saat 4:00’da başlatılacak.

[root@ila ~]# at now + 13 minutes

at> echo deneme>/dev/pts/1

at> <EOT>

job 11 at 2006-03-17 21:51

Zamanlanmış görevleri listelemek için atq komutu kullanılır.

[root@ila ~]# atq

11 2006-03-17 21:51 a root

Kuyruktaki bir işi silmek için atrm komutu kullanılır. atq komutunun çıktısındaki iş numarası atrm komutuna parametre olarak verilir.

[root@ila ~]# atrm 11

**atd Kullanımı İzinleri**

Kullanıcıların at kullanımını kontrol altına alabilmek için /etc/at.allow ve /etc/at.deny dosyaları kullanılır. Bu dosyaların bulunmaması kullanıma ilişkin bir kısıtlamanın getirilmediğini gösterir.

/etc/at.allow dosyasında kullanıcı adları belirtilmişse diğer tüm kullanıcılar at kullanamazlar. Eğer /etc/at.allow dosyası sistemde yoksa sadece /etc/at.deny dosyasındaki kullanıcılar at kullanamazlar.

Dosya formatı her satırda bir kullanıcı adı olacak şekildedir.

**Anacron**

Anacron zamanlanmış görevlerin belirli bir zaman diliminde yapılmasını sağlayan bir servistir. Bu servis özellikle sistemin 7/24 çalışmadığı durumlarda cron’un üstlendiği günlük, haftalık ve aylık işlerin yapılıp yapılmadığını kontrol eder.

Anacron çalıştığında iş listesini /etc/anacrontab dosyasından okur. Buradaki her iş için, bu işin belirtilen periyotta (son x günde) yapılıp yapılmadığını kontrol eder. Eğer iş belirtilen zaman diliminde cron tarafından gerçekleştirilemediyse anacron belirli bir gecikme süresince bekledikten sonra işi gerçekleştirir.

**/etc/anacrontab Dosyası**

Bu dosya anacron’un yapılandırma dosyasıdır. Dosyanın formatı aşağıdaki gibidir:

<periyot(gün)> <gecikme(dakika)> <isTanimlayici> <komut>

[root@ila ~]# cat /etc/anacrontab

SHELL=/bin/sh

PATH=/usr/local/sbin:/usr/local/bin :/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

1 65 cron.daily run-parts /etc/cron.daily

7 70 cron.weekly run-parts /etc/cron.weekly

30 75 cron.monthly run-parts /etc/cron.monthly

**UGYULAMA 1**

1. root kullanıcısı olarak her iki dakikada bir /var/log/cron dosyasındaki kayıt sayısını /root/cronlog\_satir dosyasına yazan bir zamanlanmış görev tanımlayın.
2. Şu andan itibaren 10 dakika sonra root kullanıcısının tüm zamanlanmış görevlerini kapatacak ve /root/cronlog\_satir dosyasına kapatma zamanını (date komutu) yazacak bir zamanlanmış görev tanımlayın.

**CEVAPLAR 1**

1----------------

[root@ila ~]# crontab -e

\*/2 \* \* \* \* wc -l /var/log/cron >> /root/cronlog\_satir

[root@ila ~]#

2--------------

[root@ila ~]# at now + 10 minutes

at> crontab -r

at> date >> /root/cronlog\_satir

at> <EOT>

job 1 at 2011-10-07 22:19

[root@ila ~]#

**LOG SİSTEMİ**

**Loglama Sistemi**

NOT:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOGLAMA SİSTEMİ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-Suid de yapılacak herhalbir hatada normal kullanıcıda root gibi çalışabilir.Bu yüzden güvenlik tehdidi oluşturabilir.

-syslog :LOglama servisi(windowsta evt formatında tutulur.)Lİnuxta logların tutuldugu merkezi bir yapı yoktur.Uygulamalaarın kimi gider dosyaya yazar kimi networkten yollar.Kİmi hiç loglamaz.Kimi de syslog kullanır.UYgulamayı yazan kişi syslog a gitsin derse loglar syslog a gider.

-syslog un eksikleri vardır.Geliştirememişlerdir.rsyslog bundan forklandı.

-log üreten ilk uygulama çekirdektir.log larını herhangi bir yere yazmaz.Ram de tutar.

-Redhat E5 e kadar syslog yüklüydü.Redhat E6 dan itibaren rsyslog a geçilmiştir.

-syslog yüklüyken 2 farklı servis çalıştırılır.1.si: kernel syslog(klogd) diğeri syslogd (uygulama servis mesajlarını dinler).

-yapılandırma dosyası /etc/syslog.conf tur.

-rsyslog un ise: /etc/rsyslog.conf tur. -->kuralları aynıdır ama rsyslogda fazlası vardır.

--syslog: logları ayırt etmek için TİP.SEVİYE tutar. (buna uygulamayı yazan kişi karar verir. Bu loga ne olacagına biz karar veririz)

-tipler:auth

authpriv

cron->cron servisi cron tipinde loglar üretip syslog a gönderir.

deamon

Not:daemon kavramı,windowstaki servislerdir.Bir program çalıştırdığımızda 3 işaretçisi vardır.(in,out,err)Daenmonlerın hiçbir işaretçisi yoktur.

-local[0-7]:kullanıcıya bırakılmıştır.örn bir betik yazdık, syslog a bu tipte local1,local2... gibi tip belirleyebiliriz.(Redhat local7 tipinde logda gönderir. Redhat türevi dagıtımlarda local7 yi kullanmamaya dikkat et.)

-/etc/syslog daki herbir satır bir kuraldır.

#:açıklama satırı

\*:özel bir anahtar.(tüm tipler yada tüm seviyeler derken kullanabiliriz.

,:tipleri yada seviyeleri birbirinden ayırır.

örn:mail.notice (notice ve altındaki seviyerlerinde kapsandığı kabul edilir.)

-/var/log/kritik

--uucp,news.crit (news ve altındaki seviyerleri /var/log/kritik e yaz.)

\*:tüm login olan kullanıcıların ekranına yaz.

-cron.\* : cron tipindeki bütün logları /var/log/cron a yaz.

-syslog: veriyi al diske yaz, çekirdekten onay alırsan devam et. - varsa çekirdekten onay beklemedn loglamaya devam eder hızlıdır.Ama güvensizdir. (- /var/log/maillog daki gibi mail trafiği fazla olacağından, syslog şişebilir bunu önler.)

; : birden fazla tip.seviyeyi veya ile baglayarak yazabiliriz.

mail.notice (hiçbir tip demek yani bu seviye hariç deme biçimidir.)

- damemon = err /var/log/ftp deamon tipinin error seviyesi demek (nokta atışı)

- ! :istemiyorum demek

-uucp.none

uucp.!\*

uucp.!debug hepsi aynı anlamdadır.

\*\*\*\*LOglarla neler yapabiliriz\*\*\*

- | ile ben sana logu gönderdiğimde yanındaki programa gönder diyebilriz.

-uid :kullanıcının uid si verilerek belirli bir kullanıcıya gönderebiliriz.

-@host : ile başka bir makineye gönder.(BUrda loglar genelde başka bir makineye gidip yazılır. Amaç ;biri sistemi hacklerse ,logları da gidip silmek istiyecektir, bu şekilde loglara gitmasi zaman alır, bu şekilde deşifre edebiliriz.)

BU şekilde UDP üzerinden 514 ü porttan gönderim yapılır.

-Network daki logları dinlemek için;(syscongfig/syslog)

SYLOGD-OPTIONS = "-m 0 -r" vererek ana makineye loglar gönderilir. Netwrok böyle dinlenir. Ama kötü yanı logun hangi makineden geldiği bilinemez. Bunu rsyslog ayırt edebilir.

-neredeyse tüm ag cihazları rsyslog tipinde mesaj gönderebilir.

--rsyslog : eklenti yazılabilir. (moduler bir yapısı vardır.)

.TCP opsiyonuda sunar. syslog da ise ag cihazları da dahil Tcp gönderemez.

-tip.seviye ile birlikte property (logun tipi) gibi eklemeler yapılmıştır.

-herhangi bir log u uzak makinaya gönderirken;

@adres -> udp ile gönderir.

@@adres -> tcp ile gönderir.

örneklerdeki;

& ->üst kısımdaki kuraların aynısı.

~ ->action kısmına yazılır.Logu imha et demektir.

\*\*\*\*\*template tanımı yapılabilir\*\*\*\*

Bir sistem de birkaç yüz tane ag cihazı olabilir. Hangi makineden ne gelecegini şablonlar yazarak , gruplarız. Bu şekilde 100lerce satır yazmaktan kurtuluruz.

-rsyslog : uzaktan aldığı logları daha iyi işler. zaten localden aldıgı log larda sıkıntı yoktur. (rsyslogdaki r :remote dur.)

örn: netstat -punta ile logların dinlenip dinlenmediğine bakabiliriz.

/etc/rsyslog.conf a ;

local 6.\* @192.168.0.200 ->udp

@@192.168.0.200 ->tcp

-dosyalara yazdık yetmez.servisi dürtmemiz gerekir.

-service rsyslog restart

-logger log yollamak için kullanılır.

-logger "deneme" dersek /var/log/syslog a mesaj mesaj gönderilir.

-logger -p local6.info "deneme mesaj" -> mesajın nereye gittiğine rsyslog.conf a bakarak buna cevap verebiliriz.

-logger -t deyip logu bir etiketle gönderebiliriz.

-loglar merkeze güvenlik için gönerilir. Çünkü sisteme giren bir kişi işini halleder ve en logları siler.sistemdeki birini loglarla izleyebiliriz.

\*\*\*\*\*\*\*\*\*\*\*LOg döndürme\*\*\*\*\*\*\*\*\*\*\*\*

-logları bir dosyaya yazdık yazdık... (/var/log/messages a yada cron a yazdık) Peki bunları kim silecek. Elimizde messages dosyamız var. var olan messages dosyamızı messages.1 e taşır yeniden messages dosyası oluştururuz. Bir dahaki dönmede messages.2 oluşturulur. messages.3 , messages.4 diye gider. (yada .date konular dosylara taşınır.)BUnlara bakarak 4 gün önceki logları görebilirz mesela.Loglar silinse dahi log çevriminden bakabiliriz.

-en gücvenli sistem network elektirk kablosu çıkmış olandır.Kullanılmaz ama kullanılibilir oldukça sistem saldırılabilir olur.

-logları logrotate komutu döndürür.

/etc/cron.daily nin altındadır. Bunu cron çalıştırır.bunun altında logrotate betiği vardır. bunu açıp bakarsak hangi confu editlediğini görebiliriz. Geriye yönelik kaç dosya yazacagımız falan betikte yazar rotate 4 demek geriye dönük 4 günlük tutulabilir demektir.

-Kanuni olarak 5651 e göre logları 180 gün geriye tutulmalıdır.LOgları merkeze çekmek zorunluluğu vardır. Sonra merkezde farklı makinelere çekip orda 180 gün saklarız. Sadece log tutulmaz log ların imzalanmasıda gerekir buda ayrı bir tekniktir.

-dosyadaki create yenisini yaratmasını söyler.

-date extension açılırsa logların sonuna tarih eklenir.( /var/log da vardır.)

-compress log dosyalarını sıkıştır demektir. Genelde %70 verimli sıkışır. içerisinde bolca tarih patterini oldugundan verimli sıkışır.

-include /etc/logrotate.d altındaki dosyalarda logrotate ile ilgili tanımlamalar vardır.Asıl iş burdadır.

-altında dosyaların yapacagı işler

.monthly

0064->hakkı

root->sahibi

minsize 1m -> ayı gelse bile 1m degilse döndürme

rotate 1 -> 1 aylık dön 1 kez yani

-cat /var/log/logrotate.d/syslog

-->bir sürü dosya için çalışacak bir betik açılır.

sharescripts herdosyayı etkiler. 1 kez çalışır her dosyayı etkiler.

-syslog ile log dosyası inode a bakar.[index a ya yazsın mesela]

-logrotate ile inodeB de aynı isimde log oluşur.Bizim servisimiz halen inodeA ya yazar. Yeniden başlatınca conf dosyası yeniden okununca inodeB den devam eder.

-cat /etc/logrotate.d/httpd yi de aynı şekilde inceleyebiliriz.Benzer şekilde script mantığ bunda da vardır.

-delay compress : sıkıştırmayı geciktir.LOg yazarken sıkıştırma yapılırsa dosya bozulur.

lost:son login olan kullanıcının nerden oldugu bilgisi.

örn:lastlog -u root :roottan düşen son logları listeler.(file /var/log/wtmp şeklinde binary bir dosyadır.)

\*\*\*\*\*\*\*LOGANALİZİ\*\*\*\*\*\*

-incelemiyecegimiz logları boşuna tutmayız.

.logwatch

.awstats ... gibi yazılımlar var bunlarında kendi yapılandırma dosyaları avr.bunlardan çıkarabileceğimiz sonuçlarla; sistemim yük altında mı , falan gibi bakarak, ileriye dönük sonuçlar çıkarabiliriz.Bunu yapmazsak kör uçusu yapmış oluruz.BUnları inceleyip istatiskikler çıkrarak devam etmemiz gerekir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITIS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sistemin güvenli bir şekilde çalıştığından emin olmak ve sistemde süregelen işlemleri kontrol etmek sistem yönetiminin önemli bir parçasıdır. Sistemde çalışan süreç ve servislerin durumları sistem logları incelenerek takip edilebilir.

Sistem loglarını incelemek sadece sistemin doğru çalışmasını değil aynı zamanda bir problem çıktığında problemin kaynağını tespit etme ve sorunu gidermede de önemli rol oynar.

Fedora (9 hariç) ve Red Hat türevi sistemler log sitemi olarak syslog’u kullanır. syslog sadece Red Hat tabanlı sistemlerde değil çoğu Linux ve UNIX tabanlı sistemlerde kullanılan standart log yöneticisidir. Ayrıca Windows tabanlı sistemlerde çalışan sürümleri de vardır.

Fedora 9 üstü sürümler ile Ubuntu ise rsyslog isimli başka bir log yöneticisini kullanmaya başlamıştır. Aynı zaman Red Hat işletim sistemlerinde de syslog alternatifi olarak gelmektedir.

**Loglama Sistemi**

Sistemin güvenli bir şekilde çalıştığından emin olmak ve sistemde süregelen işlemleri kontrol etmek sistem yönetiminin önemli bir parçasıdır. Sistemde çalışan süreç ve servislerin durumları sistem logları incelenerek takip edilebilir.

Sistem loglarını incelemek sadece sistemin doğru çalışmasını değil aynı zamanda bir problem çıktığında problemin kaynağını tespit etme ve sorunu gidermede de önemli rol oynar.

Fedora (9 hariç) ve Red Hat türevi sistemler log sitemi olarak syslog’u kullanır. syslog sadece Red Hat tabanlı sistemlerde değil çoğu Linux ve UNIX tabanlı sistemlerde kullanılan standart log yöneticisidir. Ayrıca Windows tabanlı sistemlerde çalışan sürümleri de vardır.

Fedora 9 üstü sürümler ile Ubuntu ise rsyslog isimli başka bir log yöneticisini kullanmaya başlamıştır. Aynı zaman Red Hat işletim sistemlerinde de syslog alternatifi olarak gelmektedir.

**[root@ila ~]# dmesg | head**

Linux version 2.6.18-194.11.1.el5 (mockbuild@hs20-bc2-4.build.redhat.com) (gcc version 4.1.2 20080704 (Red Hat 4.1.2-48)) #1 SMP Tue Jul 27 05:44:43 EDT 2010

BIOS-provided physical RAM map:

BIOS-e820: 0000000000010000 - 000000000009f800 (usable)

BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)

BIOS-e820: 00000000000dc000 - 00000000000e0000 (reserved)

BIOS-e820: 00000000000e4000 - 0000000000100000 (reserved)

BIOS-e820: 0000000000100000 - 000000000fef0000 (usable)

BIOS-e820: 000000000fef0000 - 000000000feff000 (ACPI data)

BIOS-e820: 000000000feff000 - 000000000ff00000 (ACPI NVS)

BIOS-e820: 000000000ff00000 - 0000000010000000 (usable)

**syslog Loglama Servisi**

syslog, sysklogd paketi ile birlikte gelir ve iki farklı süreçten oluşur:

klogd : Çekirdek mesajlarını dinler

syslogd: Uygulama (Servis) mesajlarını dinler.

Her iki süreci de tetikleyen sistem açılış betiği /etc/init.d/syslog betiğidir.

syslog üzerinden loglama yapılırken loglar tip ve seviyelerine göre sınıflandırırlı ve farklı log tip ve seviyelerinin farklı şekilde kayıt altına alınması sağlanabilir. Tüm loglama yapılandırması /etc/syslog.conf dosyasında yazılan tanımlar ile belirtilir.

**/etc/syslog.conf Yapılandırma Dosyası**

/etc/syslog.conf dosyasında logların nasıl tutulacağı tanımlanırken tip ve seviye kavramları ile birlikte tanımlar yazılır.

<tip>.<seviye>[;<tip>.<seviye>[;…]] <işlemi>

Log Tipi(Facility): Log tipi logların genelde hangi amaca hizmet eden servisler üzerinden geldiğinin ayrıştırılması için kullanılır.

Log Seviyelendirmesi(Priority): Log seviyesi ise herhangi bir tip altında gelen mesajın önem derecesine göre ayarlanmıştır. Önem derecesi azaldıkça mesaj sayısı da ters orantılı olarak artacaktır.

Örneğin mail.info tanımı tüm mail servislerinden gelen info seviyesinde ve bu seviyeden daha önemli (notice, warn, err, crit, alert, emerg) mesajlarını kapsar.

İşlem: Log tanımı yapıldıktan sonra ilgili tip ve seviyedeki loglara farklı işlemler uygulanabilir.

Log tanımları yapılırken bazı özel operatörler kullanılabilir. Bunlar:

\* : Bir alandaki tüm seçeneklere karşılık gelir. Örn: mail.\*

= :Tek bir seviye belirtilirken kullanılır. Örn: mail.=info

! : Gözardı edilecek seviyeler belirtilir. Örn: mail.!info

Örnekler:

# Tüm mail logları tek dosyaya yönlendir

mail.\* /var/log/maillog

# warn ve bunun üstündeki tüm mesajları kritk dosyasında tut, auth ve auth priv tipindeki kayıtları tutma.

\*.warn;auth.none;authpriv.none /var/log/kritik

# Tüm kullanıcılar sistem kullanılmaz mesajlarını ilet

\*.emerg \*

# boot.log dosyasına açılış loglarını tut

local7.\* /var/log/boot.log

# Çekirdek mesajlarını konsola yazdır.

kern.\* /dev/console

#Aşağıdaki iki satır aynı anlama gelir

news,mail.\* /var/log/mail\_and\_news.log

news.\*;mail.\* /var/log/mail\_and\_news.log

# err seviyedindeki ftp loglarını /var/log/ftp’ye yönlendir

ftp.=err /var/log/ftp

# ! İşareti warn ve üstündeki seviyeler devre dışı #bırakcaktır ve debug,info ve notice seviyesindeki kayıtlar tutulacaktır.

kern.debug;kern.!warn /var/log/kernel.log

# herhangi seviyedeki sistem güvenlik loglarını "logsunucusu" makinesine yönlendir

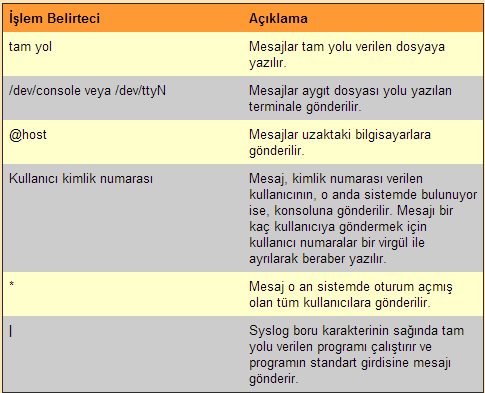
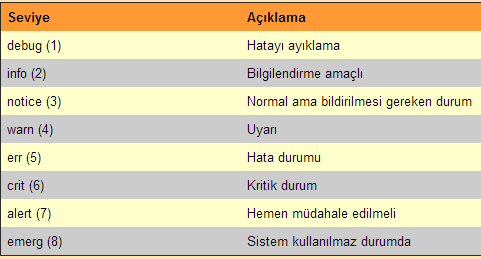
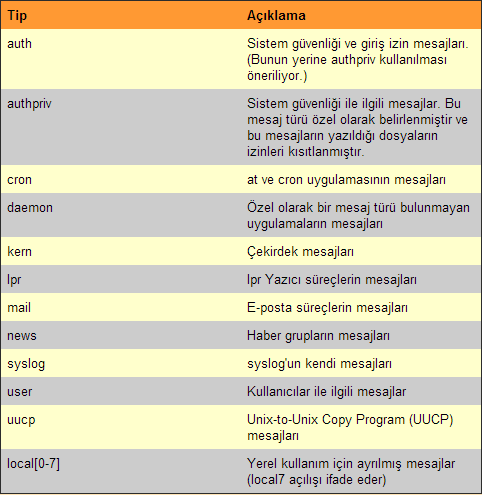
authpriv.\* @logsunucusu

#Aşağıdaki üç satır aynı anlama gelir

uucp.none

uucp.!\*

uucp.!debug



**Log Dosyasının Yapısı**

Log dosyalarında her satırda bir kayıt bulunur ve her satırın standart bir yapısı vardır. Farklı servislerin farklı şekilde kayıt tutması yerine bütün servislerin aynı standartlar altında kayıt tutması logların incelenmesinde önemli kolaylıklar sağlar. Her satırın yapısı 5 alandan oluşur.

zaman konak uygulama [pid] mesaj

Zaman - Mesajın kaydedildiği zaman ve tarihi belirtir.

Konak - Mesajın hangi konaktan geldiğini belirtir.

Uygulama - Hangi program veya servisin mesajı yolladığını belirtir.

Pid - Köşeli parantez içinde mesajı yollayan uygulamanın süreç numarasını belirtir.

Mesaj - İlgili mesajı belirtir.

Mar 18 15:43:52 ila sshd(pam\_unix)[29816]: session opened for user root by root(uid=0)

Mar 18 15:45:48 ila ntpd[29810]: ntpd exiting on signal 15

Mar 18 15:46:07 ila sshd(pam\_unix)[11372]: session opened for user ila by (uid=0)

Mar 18 15:46:11 ila ntpd\_initres[29812]: parent died before we finished, exiting

Mar 18 15:47:29 ila kernel: eth0: link up, 100Mbps, full-duplex, lpa 0x41E1

**rsyslog Loglama Servisi**

rsyslog servisi rsyslog paketi ile birlikte gelir ve iki farklı süreçten oluşur.

rklogd : Çekirdek mesajlarını dinler

rsyslogd: Uygulama (Servis) mesajlarını dinler.

Her iki süreci de tetikleyen sistem açılış betiği /etc/init.d/rsyslog betiğidir.

rsyslog servisi birçok temel özellikleri itibari ile syslog servisine benzemektedir. Son sürümlerine bakıldığında rsyslog servisinin syslog servisinden ayrıldığı temel noktalar çevresel değişkenlerden yararlanması, modüler yapıda çalışması, logların ayıklanabiliyor ve hem logların uzak sunucuya hem de veritabanı sunucusuna gönderilirken şifrelenebiliyor olmasıdır. Tüm loglama yapılandırması /etc/rsyslog.conf dosyasında yazılan tanımlar ile belirtilir.

**/etc/rsyslog.conf Yapılandırma Dosyası**

Genel yapısı itibari ile /etc/syslog.conf dosyası ile benzer bir yapı gösteren rsyslog.conf dosyası üzerinde template yapısı ve logların ayıklanması gibi ek özellikler de yapılandırılabilir. Detaylı bilgi için man rsyslog.conf.

Logların Ayıklanması: Rsyslog servisinde logların çevresel değişkenlere göre ayıklanması mümkündür. Bunun için /etc/rsyslog.conf dosyasında aşağıdaki gibi bir yapılandırma girilmelidir.

:property, [!]compare-operation, "value" logfile

Property: Çevresel değişkenler bu kısımda belirtilmektedir.

compare-operation : 4 çeşit değer içerir. Bunlar “contains”, “isequal”, “startswith”, “regex”

value : değeri

Sunucunun kendi içinde döndürdüğü logları /var/log/local/messages dosyasına yazılmasını sağlayan ayıklama aşağıdaki gibi belirlenebilir.

:HOSTNAME, isequal, "localhost" /var/log/local/messages

Template Yapısı: Template’ler logların hem farklı dosyalara hem de dosya içinde çeşitli formatlarda yazılabilmesini sağlar. Temel olarak template yapısı aşağıdaki gibidir.

Örneğin merkezi bir log sunucusuna uzak sunuculardan aynı log tipinde gelen logların farklı dosyalara yazılabilmesi için aşağıdaki gibi bir template hazırlanabilir.

$template webaccess,"/var/log/httpd/access\_log\_%FROMHOST%"

local6.info ?webaccess

Bu şekilde local6.info tipinde gelen web logları “webaccess” template i içinde işlenir ve geldiği sunucu ismine göre farklı log dosyalarına yazılır. Yukarıdaki örnekte web1 ve web2 şeklinde iki farklı makinadan log gelmesi durumunda web1 sunucusundan gelen loglar /var/log/httpd/access\_log\_web1 ve web2 makinasından gelen loglar /var/log/httpd/access\_log\_web2 dosyasına yazılır.

Örnekler:

:HOSTNAME, isequal, "server1" /var/log/server1.log

& ~

:HOSTNAME, isequal, "server2" /var/log/server2.log

& ~

$template webserver,"/var/log/web/webserver\_%HOSTNAME%"

local1.\* -?webserver

& ~

**Log Sunucusu Mantığı**

Sunucuların kendi üzerlerinde log depolamaları bazı durumlarda yüksek maliyet getirir ve de güvensiz bir yoldur. Bir saldırı anında saldırgan sunucu üzerindeki logları da silebilir. Aynı zamanda logların analiz işlemleri için gerekli programların da hizmet veren sunucularda olması performans açısından dezavantajları olan bir durumdur. Bu ve benzeri sebeplerden ötürü genellikle logların tek bir merkezde toplanması ve loglar üzerindeki tüm işlemlerin merkezde yapılması amaçlanır.

Aynı zamanda servis sağlayıcıların uymak zorunda olduğu 5651 numaralı kanuna istinaden gerekli işlemlerde bu merkezi log sunucusunda yapılır.

Syslog ve Rsyslog logları uzak bir sunucunun UDP/514. portuna gönderme yapabilir. Red Hat türevi sistemlerde /etc/sysconfig altındaki syslog veya rsyslog dosyasında, ubuntu sistemlerde ise /etc/defaults/rsyslog dosyasında servisin açılış parametrelerine -r anahtarı eklenmesi ve ardından servisin yeniden başlatılması sonrasında syslog veya rsyslog servisi uzak makinelerden gelen logları dinlemeye başlar.

[root@ila ~]# grep SYSLOGD\_OPTIONS /etc/sysconfig/syslog

SYSLOGD\_OPTIONS="-m 0 -r"

Bundan sonraki aşamada farklı sunuculardan gelen loglar üzerine yapılacak işlemler servislerin yapılandırma dosyalarında tanımlanarak yapılır fakat bu durum 1. kur seviyesinin ötesindedir.

**Komut Satırından Log Gönderme (logger komutu)**

Kabuk programlar yazarken ilgili betiğinde tıpkı diğer servisler gibi syslog üzerinden loglama yapılması istenebilir. Bu işi yapmak için logger komutu kullanılabilir.

logger [seçenekler] <mesaj>

Seçenekler:

-p Loga yazılacak mesajın log tipi ve seviyesi belirtilir.

-t Loga yazılacak süreç ismi ya da etiket.

**Log Çevrimi (logrotate)**

Sistemin ürettiği logların dosyalarda tutulması zamanla log dosyalarının boyutunun yükselmesine neden olacaktır. Yüksek boyutlu log dosyaları da inceleme ve analizi yavaş olmakla beraber disklerin dolmasına sebep olacaktır.

Bu soruna çözüm olarak logların döndürülmesi gerekir. Log döndürme işleminde aktif olarak kullanılan log dosyasının isminin sonuna genelde .1 eklenerek farklı bir isim altında kayıt edilmesi ve servisin loglarını yeni boş dosyaya yazmasıdır. Bir sonraki döndürmede .1 uzantılı olan dosya .2 olarak ismi değiştirilir ve bu işlem yapılandırmada tanımlanan sınırlar süresince tekrarlanır. Yien yapılandırmaya bağlı olarak döndürme işlemi sonucunda dosya isminin sonuna .1 eklenmesi yerine döndürülme tarihi de yazılabilir.

Linux sistemlerde log döndürme işlemleri genellikler logrotate komutu kullanılarak yapılır. logrotate sistem tarafından günlük olarak zamanlayıcı servisi ile /etc/cron.daily/logrotate betiği üzerinden tetiklenmektedir.

logrotate komutu kullanılarak, logların çevrimi dışında logların sıkıştırılması veya logların herhangi bir kullanıcıya mail ile gönderilmesi işlemleri de yapılabilir.

Logrotate komutunun varsayılan yapılandırma dosyası /etc/logrotate.conf dosyasıdır.

logrotate yapılandırma dosyasında belirtilebilecek seçenekler şunlardır:

Logların hangi aralıklarla çevrilebileceği: daily, weekly, monthly veya boyuta göre

Geriye dönük ne kadar logun saklanacağı

Sıkıştıma işlemi

Hangi log dosyalarının çevrileceği

/etc/logrotate.d/ dizini içindeki dosyalar yapılandırmaya dahil edilerek tanımların farklı dosyalarda tutulması sağlanmıştır.

Logların mail ile yollanması

Yapılandırma dosyalarında yapılan değişikliklerin düzgün çalışıp çalışmayacağı

logrotate -dv /etc/logrotate.conf komutu ile kontrol edilebilir. Bu komut logları döndürmeden yapacağı işlemleri ekrana yazar.

**last ve lastlog Komutları**

last komutu ile sisteme son login olan kullanıcıların listesine ulaşılabilir. Komut bilgileri /var/log/wtmp dosyasından alır. Eğer komuta kullanıcı adı bilgisi verilirse sadece o kullanıcının oturum açma zamanlarını listeler.

[root@ila ~]# last

ila pts/7 10.0.0.3Sat Mar 18 14:05 still logged in

root pts/6 10.0.0.3Sat Mar 18 10:22 still logged in

root pts/5 10.0.0.3Sat Mar 18 09:44 still logged in

lastlog komutu kullanıcıların son login olma tarihlerini listeler. Komut bilgileri /var/log/lastlog veri dosyasından alır. -u anahtarı ile kullanıcı adı bilgisi verilirse sadece o kullanıcının son oturum açma zamanını gösterir. [root@ila ~]# lastlog

Username Port From Latest

root pts/6 10.0.0.3 Sat Mar 18 10:22:08 +0200 2006

bin \*\*Never logged in\*\*

daemon \*\*Never logged in\*\*

adm \*\*Never logged in\*\*

lp \*\*Never logged in\*\*

**Log Analizi**

Sistem loglarını inceleyip bu loglardan istatistikler çıkaran bir çok açık kaynak yazılım bulunmaktadır. Bu yazılımlar görsel olarak logların takibini kolaylaştırdıkları gibi log analizlerinden web sayfaları oluşturularak internet üstünden servislerin takibinin yapılmasınıda sağlamaktadır. Önemli bazı log analiz yazılımları şunlardır:

logwatch - Red Hat dağıtımları ile birlikte gelmektedir.

awstats - Fedora dağıtımları ile birlikte gelmektedir.

Swatch - Sisteme bağlı olan kullanıcıların yaptıkları işlemleri takip edip gerekli işlemler yapmayı sağlar.

Freq - Hangi kullanıcının ne kadar login olduğunu gösterir.

SMA - Sendmail loglarını analiz edip, istatistikler çıkarır.

Log analizinde kullanılabilinecek bir çok yazılım vardır, aşağıdaki adreslerden ihtiyaçlar doğrultusunda istenilen yazılma ulaşılabilir.

**UYGULAMA 1**

1. Bilgisayarınızın log sistemini yapılandırarak, local6.info tipindeki logları 192.168.0.200 IP adresine göndereceği şekilde yapılandırınız.
2. logger komutunu kullanarak local6.info tipinde ve mesaj olarak makine adınızı içeren bir log üretin.

adımdan sonra eğitmen ile irtibata geçerek logun sunucuya ulaştığını kontrol ettirin.

**ÇÖZÜMLER 1**

**1-------------**

[root@ila ~]# vim /etc/rsyslog.conf

...

local6.info @192.168.0.200

[root@ila ~]#

[root@ila ~]# service rsyslog reload

Reloading system logger... [ OK ]

[root@ila ~]#

2------------------------

[root@ila ~]# logger -p local6.info $HOSTNAME

[root@ila ~]#

**SİSTEM AÇILIŞI**

**Sistem Açılışı**

NOT:

1-Makinede ilk çalışan şey BIOS (Basic input output system) dir.Görevi;makinede baglı olan donanımları algılamaktır. Ardından sistemin nerden boot edilecegine bakılır.2 mekanizması vardır.

a-MBR b-EFI BIOS:amaç işletim sistemini açacak olan tetiklemiyi yapmaktır.(bootloader olarak anılan bir yazılım vardır.)efi bios ta bootloader dan farklı bir mekanizma vardır.

-Bootloader olarak anılan linuxta bir çok yazılım vardır.

1. Lilo (Lİnux Loader)

-.Diğeri GRUB ; GRUB2 de vardır.Ubuntu uzun zamandır GRUB2 kullanmaktadır.Linux ta Lilo çoktan terk edilmiştir.GRUB un EFIBIOS destegi yok Grub2 destekliyor şeklinde farklılıkları vardır.

2-GRUB çalıştı. işletim sistemini çalıştırdı.(Yani çekirdegi vmlinuz u diskten bulup ram e taşıdı., bellege yüklemek çalışması anlamına gelmektedir.işletim sisteminin çalışması budur.)

-Çekirdek çalıştığında , bütün donanımalr Bıos tan alınır, ilgili aygıtları kullanacak çekirdek modulleri(windows daki driver mantıgı),devreye alınır.Daha grafik ekran falan yok.Çekirdek sadece 1 program başlatır. Bu da init tir.(PID si 1 olan süreç)

init in görevi (3 ayrı proje vardır.):

1.traditional init:çok sade, basit ve işe yarardır.

2.upstart:ubuntu başlattı.

3.systemd :redhat başlattı ---->2. ve 3. de son kullanıcıya yaklaşmak adına yönetimini zorlaştırmışlardır.

Görevi,sistemi inşaa etmek,diskleri mount etmek, seviyerleri başlatmak, İP ayarları... gibi.çekirdek bunları sallamaz.init çalıştırır.init herşeyi hazırlar.Ardından Grafik ekran dürtülür.init grafik ekrana ait bileşenleri hazırlar, Kde,Gnome neyse ona verir.

Login ekranına kadar gelinir.Ardından mingeti(redhat) diye bir uygulama başlatılır. fedora da adı(agety dir.)KUllanıcı adı şifre sonra uygulamadır.

Bu aşama farklılaşır.Wİndowds derki; kullanıcı login oldugunda açılış süreci tamamlanır.Lİnux da login ekranına gelindiğinde sistem açılmış olur.YAni;linux sadece donanımları algılama seviyesidir.Linux dediğimiz şey aslında çekirdektir.

\*\*init : /sbin/init/

Grub a kadar geldik, bu aşamada parametre girerek çalıştırabiliriz.E ye bastık; aslında /etc/grub.conf da konfigure edebileceğimiz bir satır açılır.BUrada title her yeni menu elemanının tanımın gösterir..

0

1

2 diye gider.0 defaulttur.

-timeout = ne kadar bekleneceğini belirtir.

---root() Grub olarak daha işletim sistemi yokken,çekirdeği bulup çalıştırmak lazım.Dogrudan diske erişmesi gerekir. Benim yükleyeceğim çekirdek dosyasının dizin kökü nerede?hd0 ilk disk, hd1 2.disk

(hd0,0) :hd0 daki 0.bölüm

(hd1,1) :1.bölüm.

Bunu cat /boot/grub/device.map diyerek görebiliriz.

-kernel ile: (hd0,0) da yani 1. diskin 1. alanında /boot/vm-linuz yanında yazan parametrelerle çalışır.(ro=kökü çekirdeğe read-only bagla.init yazılabilir olarak fstab ı okursa;bir sonrak aşamada bağlar.)

Label1= bunu ; blkid ile öğrenebilriz.(uid si falanda verebilriz.)

Grubun dosya sisteminin kökünü öğrenemesi ayrıdır.çekirdeğin öğrenemeisi farklıdır.Çekirdeğe bu bölümde söylenir.

Grubu diskten alıp çekirdeği ram den aldırarak live cd yapabilriz.

-çekirdek modullarinin yüklenmesi için;

ll /lib/modules/2.6.18.3 ; göründüğü gibi çekirdeğin çekirdeğin modullari yüklemesi için gereken modul harddisktedir.Burda bir çıkmaz var.

Bunun için çekirdek açılıncaya kadarki moduller bir dizin altında ram e koyulur.Çekirdek açılana kadar moduller burdan yüklenir.

not:zcat Ç cat gibi davranmak üzere sıkışmış dosya üzerinden çalışır.Sıkışmış dosyaları açmakla uğraşmaz.

-zcat /boot/initrd-2.6.....img diye bir dosya var, karşımıza cpio arşivi çıkartır. -t anahatrını zcat e vererek çekirdek modullerini görebilriz.

-çekirdeğin quite gibi anhtarları vardır.Çekirdek anlamadığı anahtarı init e baslar. Yani çekirdege rhgb anahtarını verdik diyelim bu çekirdeğin ahtarı değildir, init in ahatarıdır. Çekirdege bu anahatı versek inite paslanır.

-init /bin/bash dersek default initten farklı bir init çalıştırabiliriz.Linus torvalsında çekirdeği derlerken yaptıgı buydu.

-Lİnux-init-çekirdek: herkes kendi işini yapacak şekilde hazırlanmış moduler yapıdaır.Çekirdek parametreleri bu yapı şeklinde verilebilir.Lİnux un bu kadar yaygın olmas sebebi budur..Çekirdeğin üstünde istediğin şekle büründürebilrisin.

------GRUB2-------

Şuana kadar GRUB gördük;

-vim /etc/grub.d altında bir sürü dosya vardır.Farklı farklı dosyaların inşaa ettiği betikler vardır.

vim /boot/grub/grub.conf/ sistem açılışındaki yapılandırma dosyası burda bir betik çalıştırır.Ama editlenmez.grub.mkconfig i de çalıştırıdığımzda , bu dosya yeniden üreitilir.BU yüzden etilememinin bir mantıgı yoktur.

-traditional init:hala kullanımdadır.Ubuntu Fedora falan bulamaz ama.

vim /etc/inittab bunda çalışma seviyesi diye bir kavaram vardır.

System.d de bu terk edilmiştir.Bu seviyeler mesela:

0:makineyei kapat

1:single user mode : şifre kırmak için falan kullanılır.

2

3 5 ten farklı olarak grafik ekran yok

4

5:grafik ekran

6:reboot

id:3:initdefault (3 hangi seviyede çalışacagıdır.)

sysinit:initten sonra çalışacak ilk şeydir.

vim /etc/inittab dan init in çalışmasıyla oynayabiliriz.

vim /etc/rc.d/rc :bilgisayarı sysV mimarisine göre açar.

-init 3 dersek ilgili çalışma seviyesine geçeriz.Grafik ekran yoktur bu seviyede.

-runlevel dersek;

5 3 çıktı (azönce 5 dedik 3 tersinirdir.)

-init 0 dersek makine kapanır.

\*\*\*upstart\*\*

vim /etc/init conf dosyaları var.BUnlarla olaya mudahale edebilriz.ne zaman açılıp kapanacağını kendi kontrol eder.SystemV gibi bir servis mekanizmasına odaklanılmamıştır.

/etc/systemd/system/default.target :sistemin hangi mimariye göre açılacağı

-chkconfig:servislerin hangi çalışma seviyesinde nasıl çalışacagını görebilriz.

-chkconfig network off dersek makine açılırken network kapalı açılır.

-chkconfig --level 25 network off 2. ve 5. seviyede network kapalı.

AÇılışta açılacak servisi böyle dürtebilriz.

makineyi kapatmak:

-init0

-poweroff

-shutdown -h now(zaman dilimi)

açmak:

-init6

-reboot

-shutdown -r now

--------

Login olunurken neler olur?

LOgin olurken çalışan betiker vardır.

-vim /etc/profile betiği çalışır.Adam login oldugunda olacak herhangi birşey varsa bu dosyaya yazılır.

-vim /etc/bashrc :login olunurken bu dosyalarla çevresel değişkenler ortamlar falan hazırlanır.

-vim .bash-profile :kullanıcının ev dizininde bulunur.editleyebiliriz.

-en önemli olan ayar: umask tır: benim umask degerimin ne olacagı aliaslar tanımlıyorduk kalıcı olmasını istiyorsak buraya yazarız.

------------------

/etc/issue :mingetty programıyla login olununca ekrana ne yazılacagı burda yazar.

-cat /etc/motd : userler login oldugunda yazılacaklar burdadır.

\*\*\*shutdown -r +6 "sistem kapanıyor.." : bunu login olunan bütün terminaller e gönderir.

-shutdown her dk da bir ekrana makina kapanacak bilgisi gönderir.5. dk da n sonra shutdown komutu sisteme login olunmasını engeller.

/etc/nologin dosyas yaratırsak, normal kullanıcılar ssh falan hiçbir şekilde baglanamaz.shutdownda 5 dk ya geldiğinde bu dosyayı yaratır.

-shutdown -c dersek ;root shutdown sürecini iptal edebilir.(Kapanma süreci iptal edilir.)/etc/nologin yaratırsak sisteme root dışındaki kullanıcıların girişi engellenir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*bıtıs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Linux yüklü bir işletim sisteminin nasıl açıldığını bilmek önemlidir. Böylece açılışta yaşanabilecek bir problemin sebebi daha rahat anlaşılabilir.

Linux yüklü bir sistemin açılış sırası şudur:

1. Bilgisayarın elektrik düğmesine basılması ile birlikte, bilgisayara gömülü olarak gelen BIOS (Basic Input Output System) yazılımı çalışır ve fiziksel donanımları algılar ve kendi kontrollerini yapar.
2. BIOS donanım algılama işlemini tamamladıktan sonra kendisine tanımlanan boot aygıtı üzerinden sistemi çalıştırır. Bu aygıt CD/DVD ROM veya USB disk olabileceği gibi genellikle sistemdeki sabit disklerden biridir. BIOS ilgili sabit diskin MBR olarak adlandırılan ilk 512byte’lık kısmındaki önyükleyici programı belleğe yükler ve çalıştırır.
3. Önyükleyici kendisine belirtilen çekirdek imajını bulur, hafızaya yükler ve çekirdeği başlatır
4. Çekirdek, aygıtları ve aygıt sürücülerini tanımlar, (/) dizinine ait disk bölümünü (genellikle) salt okunur olarak bağlar ve çalışmaya başlar.
5. Çekirdek aksi bir durum belirtilmedikçe /sbin/init programını başlatır.
6. init, geriye kalan tüm sistem açılışından sorumludur. Sistemin ayarlanması, /etc/fstab'ta yer alan dosya sistemlerinin bağlanması, servislerin başlatılması, vb... her işlemi yapar.
7. Sistem açılışında son başlatılan süreç oturum açmamızı sağlayan mingetty/agetty programıdır.

**Önyükleyici (Bootloader)**

Önyükleyici, işletim sisteminin ilk başlangıcının yapılmasından sorumlu programdır. İşletim sistemi çekirdeğinin bulunduğu yeri ve başlangç seçeneklerini ayarlayarak ilk başlangıcı yapar ve bilgisayarın kontrolünü çekirdeğe verir.

Her işletim sisteminin bir önyükleyicisi vardır ve sistemin ilk başlatılmasını tetikler. Linux üzerinde farklı önyükleyici projeleri bulunmaktadır. En yaygın bilinenler lilo, grub ve grub2 projeleridir.

**LiLo (Linux Loader):**

Çok eski bir önyükleyicidir. Günümüz modern dağıtımların çoğu kullanmamaktadır. Tek yapılandırma dosyası /etc/lilo.conf dosyasıdır. /etc/lilo.conf dosyasında bir değişiklik yapıldıktan sonra ilgili değişikliğin MBR alanına yazılması için lilo komutunun işletilmesi gerekir. Lilo Linux dışında Windows işletim sistemlerini de destekler.

Daha fazla bilgi için: http://freshmeat.net/projects/lilo/

**GRUB (Grand Unified Boot Loader):**

GRUB en sık kullanılmaya başlanan önyükleyici yazılımdır. Lilo projesinin üzerine bir çok yenilik geliştirilmiştir. GRUB üzerinde bir değişiklik yapıldığında tekrardan MBR yüklemeye gerek yoktur. Red Hat ve türevi dağıtımlar da GRUB kullanır.

GRUB yazılımı gelişmişliği neticesinde MBR üzerinde ayrılan 466 byte alanına sığamaz. Bu sebeple iki fazlı bir mantığı vardır. Birinci faz MBR alanı üzerinden çalışır ve /boot/grub dizini altından ikinci fazı çalıştırır. Bu sebeple /boot dizininin ext3 veya ext4 olması gerekir.

GRUB yazılımının temel yapılandırma dosyası /boot/grub/menu.lst veya /boot/grub/grub.conf dosyasıdır. Red Hat türevi sistemlerde /etc/grub.conf dosyası da kolay kullanım için bu dosyalardan ilgili olanına sembolik link ile bağlanmıştır.

default tabiri hangi işletim sisteminin varsayılan olarak açılacağını belirtir. 0 ilke seçenek, 1 ikinci seçenek, …

timeout seçeneği de menü ekranında ne kadar bekleneceğini saniye cinsinden gösterir. Bu süre içerisinde bir işlem olmazsa, süre dolduğunda varsayılan işletim sistemi açılır.

[root@ila ~]# ll /etc/grub.conf

lrwxrwxrwx 1 root root 22 Sep 24 21:52 /etc/grub.conf -> ../boot/grub/grub.conf

**GRUB2**

GRUB2, GRUB Legacy projesinin geliştirilmiş yeni sürümüdür. GRUB2 daha iyi taşınabilirlik ve uyarlanabilirlik, daha geniş karakter desteği, dinamik modül yönetimi ve bellek yönetimi gibi GRUB Legacy'ye göre daha gelişmiş özelliklere sahiptir. GRUB2'nin en büyük farkı yapılandırma dosyalarının kabuk betikleri olarak hazırlanmış olmasıdır. Böylece kolayca yönetilebilmektedir.

Temel yapılandırma dosyası /boot/grub2/grub.cfg dosyasıdır. Bu dosya sistem açılışında GRUB2'nin ayarları okuduğu asıl dosyadır, fakat doğrudan bu dosyanın değiştirilmesi tavsiye edilmemektedir.

GRUB2 için varsayılan ayarlar /etc/default/grub dosyasında bulunmaktadır. Bu dosyadan yapılandırma betiklerin kullanılacak varsayılan değerler atanabilir yada değiştirilebilir. Varsayılan işletim sistemi, bekleme süresi gibi değerler buradan ayarlanır. Asıl yapılandırma dosyası yerine /etc/grub.d/ dizini altındaki yapılandırma dosyaları üzerinden işlem yapılmalıdır. Bu dosyalar kabuk betikleri olup kolayca değişirilebilir yada yenisi yaratılabilir. Bu dosyaların isim sentaksı XX\_\* şeklindedir. Alt çizgiden önceki iki basamaklı sayı o betiğin hangi sırada çalıştırılacağını belirler.

/etc/grub.d dizini altında öntanımlı olarak gelen yapılandırma betikleri açıklanmıştır:

00\_header : Temel fonksiyonları ve varsayılan ayarları yükler. Bu betik ilk önce çalıştırılmalıdır. Bu nedenle ismi "00\_" sıra etiketiyle başlamaktadır.

10\_linux : Önyükleyici kurulurken varolan varsayılan linux dağıtımına ait önyükleme kayıtlarını içerir.

20\_linux\_xen : Fedora dağıtımında gelen bu seçenek, XEN sanallaştırma ortamıda kullanılan özel çekirdek için gerekli önyükleme ayarlarını içerir.

20\_memtest86+ : Bazı dağıtımlarda gelen bu seçenek Bellek Test Aracını içerir.

30\_os-prober : Varsayılan işletim sistemi dışında sistemde yüklü olan diğer işletim sistemlerini depolama cihazlarında arayıp bulma ve önyükleme menüsüne ekleme işini yapar. Windows sürümlerini de destekler.

40\_custom : Kullanıcıların kendilerine özel yapılandırmalarını yapabilmesi için hazırlanmış taslak yapılandırmadır.

Bu dosyaların dışında dağıtımlar farklı isimlerde dosyalar yaratabilir. Sistem yöneticileri de kendilerine özel dosyalar yaratabilirler. Bu yapılandırma dosyalarının asıl yaılandırma dosyasını oluştururken kullanılabilmesi için çalıştırma izni (x = execute) olmalıdır. Çalıştırma izni olmayan dosyalar göz ardı edilir.

Değişikliklerin GRUB2'nin okuduğu /boot/grub2/grub.cfg dosyasına yazılması için grub2-mkconfig komutunun çalıştırılması gerekmektedir. Bazı dağıtımlarda bu komut doğrudan dosyayı güncellerken, Fedora gibi bazı dağıtımlarda ise bu komuta -o parametresi ile /boot/grub2/grub.cfg dosyasını belirtmek gerekmektedir.

[root@ila ~]# cat /etc/default/grub

GRUB\_TIMEOUT=5

GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"

GRUB\_DEFAULT=saved

GRUB\_DISABLE\_SUBMENU=true

GRUB\_TERMINAL\_OUTPUT="console"

GRUB\_CMDLINE\_LINUX="rd.md=0 rd.lvm=0 rd.dm=0 $([ -x /usr/sbin/rhcrashkernel-param ] && /usr/sbin/rhcrashkernel-param || :) vconsole.keymap=trq rd.luks=0 vconsole.font=latarcyrheb-sun16 rhgb quiet"

GRUB\_DISABLE\_RECOVERY="true"

[root@ila ~]#

[root@ila ~]# ls -1 /etc/grub.d/

00\_header

10\_linux

20\_linux\_xen

20\_ppc\_terminfo

30\_os-prober

40\_custom

41\_custom

README

[root@ila ~]# grub2-mkconfig

Generating grub.cfg ...

#

# DO NOT EDIT THIS FILE

#

# It is automatically generated by grub2-mkconfig using templates

# from /etc/grub.d and settings from /etc/default/grub

#

### BEGIN /etc/grub.d/00\_header ###

...

done

[root@ila ~]#

**Sys-V-init**

Çekirdek yüklendikten sonra sistemin geri kalanını başlatacak asıl süreç çekirdek tarafından başlatılan /sbin/init sürecidir. Bu init süreci sistemin geri kalan açılışını yapar. Sistemin açılışı için birden fazla mekanizma mevcuttur. Bunlardan en çok kullanılanı Sys-V-init yapısıdır

**init Yapısı ve Çalışma Seviyesi Kavramı**

İnit, sistemde çalışan tüm süreçlerin ana süreci konumundadır. Sistem açılışı sürecinde bir numaralı süreç (PID=1) olarak çalışmaya başlar ve sistemin kapanışına kadar çalışmasını sürdürür. Temel görevi, /etc/inittab yapılandırma dosyasını okuyup, diğer programları belirli bir sırada başlatmak ya da durdurmaktır.

Günümüzde Ubuntu tarafından desteklenen upstart projesi ile yeni bir init süreci geliştirilmektedir. Ubuntu, Fedora 9 ve sonrası dağıtımlarında upstart projesi kullanılmaktadır, fakat üst seviye işletim sistemi Red Hat Enterprise Linux ailesinde henüz kullanıma geçilmemiştir.

Çalışma Seviyesi Kavramı: Sistem başlatılırken grafik ekrana ait uygulamalar açılmayabilir, ağ üzerinden servis verilmeyebilir, sadece root kullanıcısının oturum açabilmesi gibi farklı durumlar oluşabilir. Bu tür farklılıkların karşılığı init sürecinin başlatacağı programlara bağlıdır. Bu farklılıklara çalışma seviyesi denir. Her dağıtımın çalışma seviyeleri veya varsayılan çalışma seviyeleri farklıdır.

Sistemin herhangi bir anda hangi çalışma seviyesinde olduğunu runlevel komutunu yürüterek öğrenebiliriz. Komut çıktı olarak bir önceki çalışma seviyesini daha sonra ise güncel çalışma seviyesini verecektir. Eğer bir önceki çalışma seviyesi yoksa bunun yerine N harfi görülecektir.

[root@ila ~]# runlevel

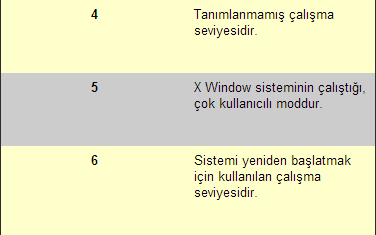
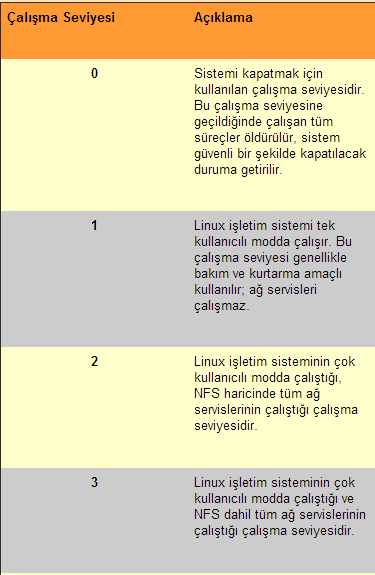
N 3

Sistemin çalışma seviyesi, init ya da telinit komutuna geçilmek istenilen çalışma seviyesinin parametre olarak verilmesi ile değiştirilebilir.

[root@ila ~]# telinit 5

[root@ila ~]# runlevel

3 5



**/etc/inittab Yapılandırma Dosyası**

Çalışma seviyelerinin kontrol ve idare etmenin en kolay yolu, init yapılandırma dosyası, /etc/inittab dosyasını incelemek ve yapısını kavramaktır. Bu yapılandırma dosyası temelde hangi çalışma seviyesinde hangi programların çalışıp, hangilerinin çalışmayacağı gibi bilgileri içerir.

id:5:initdefault:

satırı sistemin öntanımlı çalışma seviyesinin 5 olduğunu belirtmektedir. Dosya içindeki tüm satırlar bu biçimdedir ve iki nokta üst üste işareti ( : ) ile ayrılmış 4 alandan oluşur.

kimlik:çalışma seviyeleri:işlem:komut

* Kimlik - İlgili işi belirten eşsiz tanıtıcıdır.
* Çalışma Seviyeleri - Bir sonraki alanda belirtilen işlemin uygulanacağı çalışma seviyesini (seviyelerini) belirtir.
* İşlem - Yapılacak işlemin türünü (karakteristiğini) belirtir.
* sysinit - Çalışma seviyesi göz önünde bulundurulmaksızın sistem açılırken ilk olarak çalıştır.
* boot - Çalışma seviyesi göz önünde bulundurulmaksızın sistem açılırken çalıştır.
* bootwait - Çalışma seviyesi göz önünde bulundurulmaksızın sistem açılırken çalıştır fakat boot deyiminden farklı olarak işlem sonlanana kadar bekle
* once - Süreci sadece bir kez başlat
* wait - Süreci başlat ve sonlanana kadar bekle
* respawn - Çalışır durumda değil ise süreci başlat, durursa yeniden başlat
* initdefault - Öntanımlı sistem çalışma seviyesinin belirtir.
* ctrlaltdel - [CTRL]-[ALT]-[DEL] tuş kombinasyonu kullanıldığında gerçekleştirilecek işlemi belirtir.
* Komut - Çalıştırılacak komutu belirtir (seçmeli)

init programının öntanımlı çalışma seviyesi öğrenmek için baktığı id:X:initdefault satırında çalıştırmak için bir komuta gerek yoktur. Ayrıca burada belirtilen değer 0 veya 6 olmamalıdır.

**Sistem Açılış Betikleri**

Red Hat işletim sisteminde SysV mantığına uygun bir şekilde açılış betikleri bulunmaktadır. Bu betiklerin hepsi /etc/rc.d/ dizini altında bulunur.

/etc/rc.d/rc.sysinit: Bu betik init devreye girdiğinde ilk çalıştırılan betik olup sistemi ilk kullanıma hazırlar. Bazı görevleri

Sistemin isminin tanımlanması

Çekirdek parametrelerinin ayarlanması

Klavye ve sistem fontunun ayarlanması

Dosya sistemlerinin bağlanması ve kontrol edilmesi

LVM ve RAID cihazlarının aktive edilmesi

Takas alanlarının aktive edilmesi

Kullanıcı kotalarının aktive edilmesi

/etc/rc.d/rc: Sistemin ilk açılışı (initdefault) ya da sistem çalışırken çalışma seviyesinin değiştirilmesi için gerekli birkaç işlem vardır. Bu işlemleri gerçekleştirmek için /etc/rc.d/rc betiği geçilecek olan çalışma seviyesini parametre alarak çalıştırılır. Bu script init programı tarafından çağırılır ve sonlanması beklenir.

[root@ila ~]# grep '/etc/rc.d/rc ' /etc/inittab

l0:0:wait:/etc/rc.d/rc 0

l1:1:wait:/etc/rc.d/rc 1

l2:2:wait:/etc/rc.d/rc 2

l3:3:wait:/etc/rc.d/rc 3

l4:4:wait:/etc/rc.d/rc 4

l5:5:wait:/etc/rc.d/rc 5

l6:6:wait:/etc/rc.d/rc 6

rc betiğinin temel görevi geçilmek istenen çalışma seviyesine göre, önceden belirlenmiş, sistem ve ağ servislerini belirli bir sırada kapatmak veya açmaktır. Çalışma seviyesine göre başlatılacak ya da sonlandırılacak servislerin betiklerinin bulunduğu 7 farklı dizin (/etc/rc.d/rcX.d/, X=0...6 ) vardır.

[root@ila ~]# ll -d /etc/rc.d/rc?.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc0.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc1.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc2.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc3.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc4.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc5.d

drwxr-xr-x 2 root root 4096 Sep 19 23:20 /etc/rc.d/rc6.d

Bu dizinlerde adı S ile başlayan betikler bu çalışma düzeyine geçişte, K ile başlayan betikler ise bu çalışma seviyesinden bir başka çalışma seviyesine geçerken çalıştırılırlar. Bu betiklerdeki K ve S harflerinden sonra gelen, 00 dan 99 kadar sayıları betiklerin çalışma sırasını belirlemektedir. Daha küçük değerde sayı içeren betik daha önce çalıştırılır. Linux dağıtımlarında servislerin başlatılıp, sonlandırılmasını sağlayan scriptler /etc/rc.d/init.d/ dizini

altında bulunmaktadır. Çalışma seviyelerinin dizinlerinde bulunan scriptler aslında /etc/rc.d/init.d/ dizini altındaki scriptlere sembolik linklerdir.

[root@ila root]# ls -l /etc/rc.d/rc3.d/

total 272

lrwxrwxrwx 1 root root 24 Oct 10 19:04 K02NetworkManager -> ../init.d/NetworkManager

lrwxrwxrwx 1 root root 19 Oct 10 19:03 K05saslauthd -> ../init.d/saslauthd

lrwxrwxrwx 1 root root 19 Oct 10 19:04 K10dc\_server -> ../init.d/dc\_server

lrwxrwxrwx 1 root root 16 Oct 10 19:03 K10psacct -> ../init.d/psacct

...

lrwxrwxrwx 1 root root 15 Oct 10 19:04 S97rhnsd -> ../init.d/rhnsd

lrwxrwxrwx 1 root root 28 Oct 10 19:05 S98cups-config-daemon -> ../init.d/cups-config-daemon

lrwxrwxrwx 1 root root 19 Oct 10 19:03 S98haldaemon -> ../init.d/haldaemon

lrwxrwxrwx 1 root root 11 Oct 10 19:03 S99local -> ../rc.local

/etc/rc.d/rc.local: Bu dosya diğer tüm başlangıç betiklerinde sonra çalıştırılır. Sistem başladıktan sonra yapılması istenen işler bu dosya içerisine eklenir.

[root@ila ~]# grep -Ev '^#|^$' /etc/rc.local

touch /var/lock/subsys/local

/sbin/route add -net 192.168.0.0 netmask 255.255.0.0 dev eth0

NOT: Masaüstü için rc.local benzeri işlem /etc/X11/xinitrc-common dosyasından yapılır.

**Çalışma Seviyelerinin Özelleştirilmesi**

Red Hat dağıtımlarında en sık kullanılan çalışma seviyeleri çoklu kullanıcı ve ağ servislerinin olduğu 3. çalışma seviyesi ile buna ek olarak grafik arayüzünde bulunduğu 5. çalışma seviyeleridir. Bu çalışma seviyelerinde çalışacak servisler özelleştirilebilir. Bu işlem için en sık kullanılan chkconfig ve ntsysv komutlarıdır.

chkconfig Komutu:

Komutun temel kullanım şekli aşağıdaki gibidir:

chkconfig [--list|--add|--del] servis

chkconfig --level <çalışma-seviyeleri> servis [on|off|reset]

Çalışma seviyelerinde hangi servislerin çalışıp hangilerinin çalışmayacağını görmek için chkconfig komutuna --list parametresi verilir. Servis bazında sorgulama yapmak için komuta servis parametresi de eklenebilir.

[root@localhost ~]# chkconfig --list sendmail

sendmail 0:off 1:off 2:on 3:on 4:on 5:on 6:off

chkconfig tarafından yönetilmek üzere yeni bir servis eklemek için:

chkconfig --add <servis>

Belirtilen servisi chkconfig yönetiminden çıkarmak için:

chkconfig --del <servis>

Not: Bu işlem sonrasında /etc/rc.d/rc[0-6].d/ dizinlerindeki servise ait sembolik linkler chkconfig tarafından düzenlenir.

Belirtilen servisi istenilen çalışma seviyelerinde açık/kapalı konuma getirmek:

[root@ila ~]# chkconfig --level 2345 sendmail off

[root@ila ~]# chkconfig --list sendmail

sendmail 0:off 1:off 2:off 3:off 4:off 5:off 6:off

[root@ila ~]# chkconfig --level 35 sendmail on

[root@ila ~]# chkconfig --list sendmail

sendmail 0:off 1:off 2:off 3:on4:off 5:on6:off

Servisin belirtilen çalışma seviyeleri için öntanımlı duruma getirilmesi:

chkconfig komutu öntanımlı değerler için /etc/rc.d/init.d/ dizininde bulunan servis scriptlerine başvurur.

[root@ila ~]# more /etc/rc.d/init.d/acpid

#!/bin/bash

#

# /etc/rc.d/init.d/acpid

#

# Starts the acpi daemon

#

# chkconfig: 345 44 56

# description: Listen and dispatch ACPI events from the kernel

# processname: acpid

[root@ila ~]# chkconfig acpid reset

/etc/rc.d/init.d dizini altında bulunan bu servis dosyasının başında servisle ilgili yorum satırları bulunmaktadır. chkconfig: ile başlayan satırdaki 345 sayıları servisin başlatılacağı öntanımlı çalışma seviyelerini belirtir. 44 sayısı servisin başlatılma önceliğini, 56 sayısıda servisin durdurulma önceliğini göstermektedir. Eğer servis, öntanımlı olarak hiçbir çalışma seviyesinde başlatılmayacak ise ilgili satırdaki çalışma seviyesileri kısmında - işareti kullanılır.

NOT: chkconfig servislerin sistem açılışında çalışıp çalışmayacaklarını belirler. O anda çalışan sevisi kapatmaz veya başlatmaz.

ntsysv Komutu:

Servislerin yönetilmesi için ntsysv komutu da kullanılabilir. Komut çalıştırıldığında konsolda grafik bir arayüz açılır. Bu arayüzde servisler arasında geçiş ok(yön) tuşlarıyla ya da [PageUp], [PageDown] tuşları ile yapılır. Bir servisin belirtilen çalışma seviyesinde otomatik olarak çalışmasını sağlamak için [SPACE] tuşuna basılarak \* işareti konulur.

Komutun genel kullanımı aşağıdaki gibidir.

ntsysv --back --level <çalışma\_seviyeleri>

ntsysv komutuna hiçbir seçenek verilmezse, o anda bulunulan çalışma seviyesi için işlem yapar.

**Servislerin Açılıp Kapatılması**

/etc/rc.d/init.d/ dizini altındaki servis betikleri herhangi bir zamanda servisi durdurmak, başlatmak, durumunu öğrenmek veya yapılandırmayı tazelemek için kullanılabilirler.

Bu scriptlerin temel kullanımı aşağıdaki gibidir:

/etc/rc.d/init.d/<servis> {start|stop|restart|status|reload}

veya

/etc/init.d/<servis> {start|stop|restart|status|reload}

Ayrıca servisler service komutuyla da yönetilebilirler.

service <servis> {start|stop|restart|status|reload}

[root@ila ~]# /etc/rc.d/init.d/sendmail reload

reloading sendmail:[ OK ]

reloading sm-client:[ OK]

[root@ila ~]# service sendmail stop

Shutting down sendmail: [ OK]

Shutting down sm-client: [ OK]

NOT: ubuntu üzerinde upstart projesi kullanılmasından dolayı /etc/init.d altındaki betikler istenen işlemleri yapamayabilir. Bu sebeple ubuntu üzerinde servislerde işlem yaparken service komutu kullanılabilir. Buna ek olarak service komutuna işlem olarak girilen start, stop, restart ve status tabirleri de birer komut olarak kullanılabilir.

<start|stop|restart|status> <servis\_adı>

root@ubuntu:~# service ssh restart

ssh start/running, process 3465

root@ubuntu:~# restart ssh

ssh start/running, process 3470

root@ubuntu:~#

root@ubuntu:~# service ssh status

ssh start/running, process 3470

root@ubuntu:~# status ssh

ssh start/running, process 3470

**Upstart**

Upstart projesi, Ubuntu tarafından klasik init yapısına alternatif olarak başlatılan bir sistem açılış ve servis yönetim sistemidir. Upstart projesi olay tabanlı bir çalışma mantığına sahiptir ve servisler arası iletişimi D-BUS denilen mesajlaşma kanalı üzerinden gerçekleştirir.

Sys-V-init yapısı sunucu sistemlerine uygun olsa da özellikle masaüstü sistemlere yöneldiğinde Servisler sıra ile başladığı için açılış süresinin uzun olması

Sistem çalışırken eklenen yada çıkarılan donanımlara tepki verememesi

Servis başlama ve durma kontrollerinin servisin kendisinin yapma zorunluluğu gibi eksiklikleri ortaya çıkmaktadır.

Upstart projesi temel olarak bu sorunlar üzerine geliştirilmiş ve özellikle dinamik sistemler için tasarlanmıştır. Olay tabanlı çalışması sayesinde sisteme eklenip çıkarılan donanımlara göre kendiliğinden işlemler yapabilir ve servisleri buna göre düzenleyebilir. Sistem açılışında servisleri sırayla çalıştırmak yerine, birbirlerine bağımlı olmayan sistemleri paralel olarak başlatarak sistem açılışını hızlandırmaktadır.

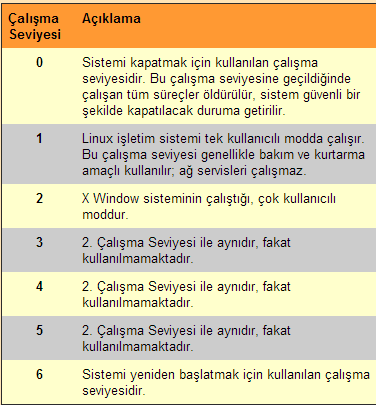
Upstart, sistemde olayları algılamak ve yönetmek için kullanıcı uzayında çalışan bazı yardımcı servisler kullanmaktadır. Böylece bu parçalardan birinin hata vermesi durumunda çekirdek hatasının kısmen önüne geçilmiş olmaktadır.

**Init ve Çalışma Seviyeleri**

Sys-V-init yapısında olduğu gibi Upstart yapısında da sistemi başlatan ve yöneten süreç /sbin/init programıdır. Sys-V-init yapısında sistem başlangıç seçeneklerini içeren /etc/inittab dosyası artık kullanılmamaktadır. Ancak geriye uyumluluk nedeniyle Upstart, eğer /etc/inittab var ise bu dosya içerisinden sistemin hangi çalışma seviyesinde başlayacağını öğrenebilir. Bunun dışındaki diğer yapılandırmalar /etc/init/ dizini içerisindeki iş\_adı.conf şeklinde adlandırılan dosyalardan öğrenilir.

Sistemin hangi çalışma seviyesinde olduğu runlevel komutu ile öğrenilebilir. Çalışma seviyesinin değiştirilmesi de init ya da telinit komutu ile yapılabilir.

Redhat/Fedora sistemler çalışma seviyelerini Sys-V-init yapısı ile birebir aynı şekilde yorumlar. Fakat Debian/Ubuntu sistemlerde çalışma seviyeleri yandaki gibi tanımlanmıştır. Bu tanımlamalar yapılandırma dosyaları ile kolayca değiştirilebilir.



**/etc/init/ Dizini**

Init süreci başladıktan sonra sistem açılışını gerçekleştirmek için ihtiyaç duyduğu yapılandırmalar /etc/init/ dizininde bulunur. Bu dizin altında her bir iş için .conf uzantılı bir dosya bulunmaktadır ve bu dosyalar içinde işlerin başlama ve sonlanma koşulları, birbirlerine gereksinimleri ve durumlar arasında geçiş işlemleri tanımlanmaktadır.

Upstart kapsamında sistemdeki tüm süreçler İş (Job) ve Olay (Event) kapsamında tanımlanır. İş, /etc/init/ dizini altında yapılandırma dosyası olan tüm servis ve süreçler olarak tanımlanır. Bu dosyalar ile bir işin başlama ve bitiş şartları ve gerekli kontrolleri yapılabilir.

Sistemde kendiliğinden yada kullanıcı tarafından gerçekleştirilecek herhangi bir değişiklik Olay olarak adlandırılır. Sistemde gerçekleşecek bir olay, upstart tarafından başlatılan servisler ile algılanıp ilgili işlerin tetiklenmesini sağlar. Böylece sistemde gerçekleşecek olaylar anında tepki verilerek yönetilebilir.

Upstart yapılandırma dosyaları, Sys-V-init'den farklı olarak kendi yapılandırma sentaksına sahiptir ve doğrudan bir kabuk betiği değildir. Başlıca yapılandırma koşulları şu şekilde tanımlanabilir:

description : Yapılandırma dosyasına iş için bir tanım girmeyi sağlar

start on : İş'in hangi durumlarda başlayacağını belirtir. Burada sistem başlangıç seviyesi, bir olay yada başka bir iş belirtilebilir. Ayrıca mantıksal (ve / veya) ifadeler de yazmak mümkündür. Başka bir iş için "starting", "started", "stopping" ve "stopped" önekleri kullanılabilir.

stop on : İş'in hangi durumlarda sonlanacağını belirtir. Başlangıç seçenekleri ile aynı şekilde ifadeler kullanılabilir.

emits : İş'in tetikleyeceği olayları tanımlar. Olay ismi belirlenirken yıldız ("\*"), soru işareti ("?") ve köşeli parantezler ("[" "]") kullanılabilir.

script / end script : İş'in durumlarına göre belirli kabuk betikleri çalıştırmayı sağlar. Betiklerin hangi durumda çalıştırılacağı başlangıç etiketine göre atanır. Başlangıç etiketleri pre-start script , post-start script, pre-stop script ve post-start script şeklinde durum ön ekleri alabilir. Bu etiketler arasında yazılanlar doğrudan kabuğa aktarılır. Bu nedenle kabuk değişkenleri ve özel karakterleri kullanılabilir.

env : Yapılandırmada ve script etiketleri arasında kullanılacak değişkenleri tanımlar.

exec : Script etiketi ile aynı şekilde çalışır fakat tek bir komut için kullanılabilir. pre-start exec , post-start exec, pre-stop exec ve post-start exec şeklinde durum önekleri alabilir.

expect : İşin kaç kez fork() sistem çağrısını yapacağını tanımlamayı sağlar. Bu sayede PID'si değişse bile işi takip etmeyi sağlar. Bu değerin yanlış girilmesi durumunda işin süreçleri takip edilemez ya da iş başlayamaz.

nice : İşin süreçlerinin öncelik değerini belirler

respawn : Düzgün bir biçimde sonlanmayan işlerin kendiliğinden tekrar başlamasını sağlar.

task : Kısa süreli çalışacak işler için kullanılır. İş sonlanana kadar kendisini başlatan işin bekletilmesini sağlar.

[root@ila ~]# cat /etc/init/rsyslog.conf

# rsyslog - system logging daemon

#

# rsyslog is an enhanced multi-threaded replacement for the traditional

# syslog daemon, logging messages from applications

description "system logging daemon"

start on filesystem

stop on runlevel [06]

expect fork

respawn

pre-start script

/lib/init/apparmor-profile-load usr.sbin.rsyslogd

end script

script

. /etc/default/rsyslog

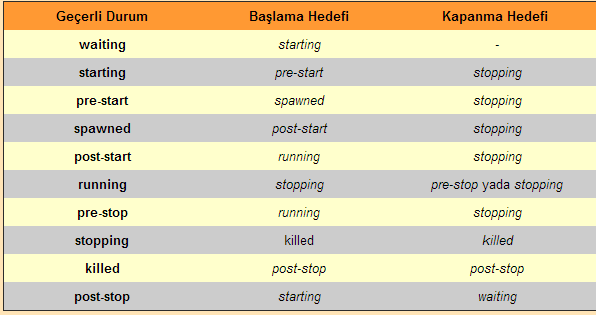
exec rsyslogd $RSYSLOGD\_OPTIONS

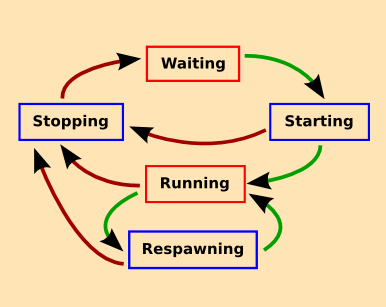
end script

**İş Durumları ve Sistem Açılış Betikleri**

Upstart mekanizmasında her iş bir durumda bulunmaktadır. Bu durumlar üzerinden işin başlama ve sonlanma süreci daha kolay yönetilebilmekte ve işlerin birbirine bağımlılıkları bu durumlar ile belirlenebilmektedir. İşin bulunduğu durum ve bu durumlardan geçebileceği durumlar yandaki tabloda gösterilmiştir. Yapılandırma dosyalarında bu durumlar için ayrı ayrı komutlar çalıştırılabilir ve böylece bir işin başlamadan önce, başladıktan sonra yada diğer durumları için özel yapılandırmalar girilebilir.

Upstart mekanizmasında servisleri başlatan kısım yapılandırma dosyalarındaki exec yada script kısımlarıdır. Genellikle birkaç satır komut yeterli olmaktadr. Fakat çoğu servisin upstart yapısına henüz tamamen geçirilmemesi nedeniyle Sys-V-init yapısındaki başlangıç betikleri de kullanılmaktadır. Sys-V-init betikleri /etc/init.d/ dizini altında bulunurken, Upstart tarafından yönetilen servisler için bu dizin altındaki sembolik linkler /lib/init/upstart-job isimli upstart başlangıç betiğine yönlenmiştir. Böylece Sys-V-init yapısında olan servisler hala eski betik yapısını kullanabiliyor iken, Upstart yapısında olan servisler de yönetilebilmektedir





**initctl Komutu ve İşlerin Yönetimi**

Upstart yapısında çalışan servisler "initctl" komutu ile yönetilebilir. Servislerin başlatılması, sonlandırılması, başlangıç şartlarının kontrol edilmesi ve bilgi alınması bu komut ile yapılabilir. Servis yönetiminin dışında olay tetiklemek için de kullanılmaktadır.

initctl start iş\_adı : Servisi elle başlatır

initctl status iş\_adı : Servisin durumunu ve PID'sini görüntüler

initctl stop iş\_adı : Servisi elle durdurur

initctl restart iş\_adı : Servisi yeniden başlatır

initctl reload iş\_adı : Servise SIGHUP sinyai göndererek yapılandırmasını tekrar okumasını sağlar

initctl emit olay\_adı : Bir sistem olayını tetikler. Olay çağrısı asenkron yada senkron olabilir

initctl list : Servislerin listesini, durumlarını ve PID'in ekrana yazdırır

initctl log-priority log\_seviyesi : Upstart init sürecinin log seviyesini değiştirir

initctl show-config iş\_adı : Bir işin yapılandırma dosyasına göre hangi şartlarda çalışacağını gösterir

initctl help : Yazılabilecek komutları gösterir

[root@ila ~]# initctl start rsyslog

rsyslog start/running, process 2550

[root@ila ~]# initctl status rsyslog

rsyslog start/running, process 2550

[root@ila ~]# initctl stop rsyslog

rsyslog stop/waiting

[root@ila ~]# initctl start rsyslog

rsyslog start/running, process 2565

[root@ila ~]# initctl restart rsyslog

rsyslog start/running, process 2577

[root@ila ~]# initctl show-config rsyslog

rsyslog

start on filesystem

stop on runlevel [06]

[root@ila ~]#

**systemd**

Redhat tarafından geliştirilen systemd projesi, sys-V-init ve Upstart'a alternatif olmakla beraber aslında daha genel anlamda sistemin yönetilmesi için yaratılmıştır. Sys-V-init genel anlamda sistemin açılışı ve kapanışında servisleri açılıp kapanmasını kapsar. Upstart biraz daha ileri giderek olay tabanlı bir mimari ile servislerin kendiliğinden başlaması yada durdurulması gibi sistem açıkken olacak değişikliklere göre düzenlenebilir bir yapı kazandırmıştır. systemd yapısında ise yalnızca sistemin yada servislerin açılıp kapanması değil, genel anlamda sistemde çalışacak servislerin ve süreçlerin yönetilmesi amaçlanmaktadır.

Systemd ile servis açılışlarının paralellenmesi, soketler ve DBUS üzerinden tetiklenerek servis başlatılması, süreçlerin cgroups ile takip edilmesi, sistemin anlık görüntüsü (snapshot) alınarak istendiğinde bu duruma dönülebilmesi gibi yetenekleri vardır. Bunların dışında systemd kendi loglama yapısına sahiptir. Bu yapı systemd üzerinden çalışan servislerin standart girdi, standart çıktı ve hata çıktılarını da syslog gibi bir loglama servisine aktarmaya olanak sağlamaktadır.

Upstart'ın iş ve olay bakış açısı yerine systemd sistemi Unit (Birim) olarak adlandırdığı parçalara ayırır. Sistemde bulunan servis, soket, aygıt vb. herşey bu birimlerden birine dahil edilir ve o şekilde yönetilir. Birimler aşağıdaki şekilde tanımlanmıştır:

service : Sistemde çalışan servisleri içerir ve yönetir.

socket : Soket dosyalarını yönetir. Her soket yapılandırmasına karşılık bir servis yapılandırması olması gerekmektedir.

device : Sistemde bulunan aygıtları yönetir.

mount : Sistem dosya sistemi bağlama olaylarını yöneten birimdir.

automount : Dosya sistemi bağlama işin erişim anında yapmayı sağlayan birimdir. Her "automount" yapılandırmasın karşılık bir "mount" yapılandırması olması gerekmektedir.

target : Tek başına bir işlevi yoktur, mantıksal olarak diğer birimleri yönetmeyi sağlar. Çalışma seviyeleri bu şekilde tanımlanır.

snapshot : Sistemde anlık olarak birimlerin durumunu kaydedip daha sonra bu durumu dönmeyi sağlayan birimdir.

**Init ve Çalışma Seviyeleri**

Sistem açılırken ilk başlayan süreç /usr/lib/systemd/systemd sürecidir. Varsayılanda linux çekirdeği /sbin/init sürecini başlatacağı için genelde /sbin/init yolu systemd sürecine bir sembolik linktir. Systemd birimlerinin yapılandırmaları /lib/systemd/system/ dizin altındaki dosyalar ile yapılır. Sys-V-init yapısındaki ana yapılandırma dosyası olan /etc/inittab dosyası systemd tarafından kullanılmaz. /lib/systemd/system/ dizinindeki yapılandırmalar sistem güncelenmesi gibi operasyonlarda değişebilir. Bu nedenle yapılacak değişiklikler /etc/systemd/system dizini altında yapılmalıdır.

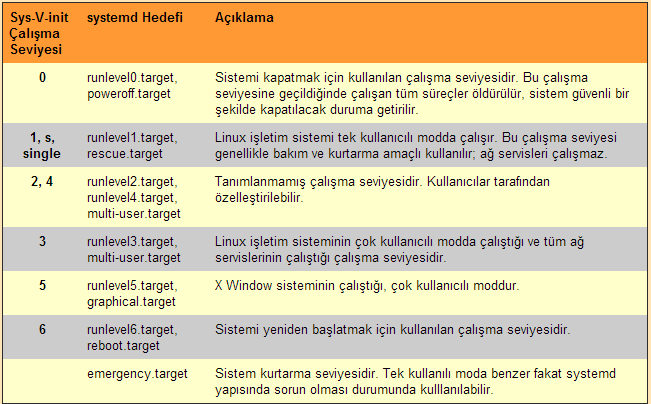
Çalışma seviyesi kavramı systemd ile beraber yerini target kavramına bırakmıştır. Bu kavram aslında diğer birim yapılandırma dosyalarını gruplamayı sağlar. isim.target ana yapılandırma dosyası iken, isim.arget.wants/ dizini de içerdiği diğer birimlerin yapılandırma dosyalarını hedef gösteren birer sembolik link içerir. Böylece hangi çalışma seviyesinde hangi servisin çalışacağı belirlenebilir. Bu dizinin içerisinde bir başka target dosyasına da sembolik oluşturulabilir. Böylece servis grupların birbirlerini başlatması sağlanabilir. Sys-V-init yapısındaki çalışma seviyeleri ve systemd karşılıkları tabloda verilmiştir.

Varsayılan açılış seviyesi default.target tarafından belirlenir.

[root@ila ~]# ll /etc/systemd/system/default.target

lrwxrwxrwx. 1 root root 36 Aug 26 22:53 /etc/systemd/system/default.target -> /lib/systemd/system/graphical.target

Açılış sırasında çekirdeğe verilecek parametreler ile de systemd açılış seviyesi değiştirilebilir. Bu systemd.unit=multi-user.target şeklinde belirtilebilir.



**/etc/systemd/system ve /lib/sytemd/system/ Dizinleri**

Birim yapılandırma dosyalarının bulunduğu asıl dizin /lib/systemd/system/ dizinidir. Burada ilgili servislerin kurulumu ile gelen yapılandırma dosyaları vardır. Buradaki yapılandırmalarda değişiklik yapılması önerilmemektedir. Bunun yerine /etc/systemd/system/ dizini altında sistem yöneticilerinin kendi yapılandırmalarını yapması gerekmektedir. /etc/systemd/system/ dizininde yapılan yapılandırmalar diğer dizine baskın gelecektir.

Her bir birim için bu dizin altında bir birim yapılandırma dosyası bulunmaktadır. Her bir dosya isim.birim\_ismi şeklindedir. Dosyalar, bloklara ayrılmıştır. Her birim dosyasında ortak olan Unit ve Install bloklarının yanı sıra, birim tipine özel bloklar da bulunmaktadır. Bu bloklar içerisinde ilgili birimin bağımlılıkları ve seçenekleri belirtilmektedir.

[Unit] Bloğu

Tüm birim dosyalarında ortak bloktur. Birim ile ilgili genel ayarları ve tanımları içerir. Başlıca anahtarları şunlardır:

Description : Birim tanımlamak için kullanılır.

Requires : Birimin çalışması için gerekli olan diğer birimleri belirtir. Birim aktifleştirildiğinde bu satırda yazan diğer birimler de aktifleştirilir. Eğer buradaki birimlerden herhangi biri aktifleştirilemezse asıl birim de aktifleştirilemez.İstenen birimler buraya yazılabileceği gibi, aynı zamanda birim yapılandırma dosyası ile aynı dizindeki isim.birim\_ismi.requires/ dizini altında ilgili birimlere sembolik link yaratılarak da belirtilebilir.

Wants : Requires ile aynı işlevi görür. Fakat buradaki birimlerin aktifleştirilmesinde bir sorun oluşması durumunda asıl birim durdurulmaz. İstenen birimler buraya yazılabileceği gibi, aynı zamanda birim yapılandırma dosyası ile aynı dizindeki isim.birim\_ismi.wants/ dizini altında ilgili birimlere sembolik link yaratılarak da belirtilebilir.

Conflicts : Birimin çakışma yarattığı birimler tanımlamak için kullanılır. Buraya yazılan birimlerden herhangi biri aktif ise bu birim aktifleştirilemez.

Before ve After : Aynı anda tetiklenen birimlerin açılış ve kapanış sırasını belirler. Genellikle requires yada want ile beraber kullanılır.

OnFailure : Hata durumunda aktifleştirilecek birimleri belirtir.

AllowIsolate : Kullanıcı tarafından verilecek system isolate komutuna verilecek tepkiyi belirler. Genellikle sistem durumunu değiştiren target birimlerinde izin verilir. Varsayılanda bu komuta izin verilmez.

[Install] Bloğu

Tüm birim dosyalarında ortak kullanılan bloktur. Birimlerin yapıladırma ayarlarını içerir. Başlıca anahtarları şunlardır :

Alias : Birime alternatif isimler tanımlamak için kullanılır. Birim bu isimler ile de aktifleştirilebilir.

WantedBy ve RequiredBy : Bu birimin hangi birimler tarafından bağımlılık olarak belirtileceğini tanımlar. Bu sayede birim etkinleştirilirken ilgili servislerin .wants ve .requires dizinlerine sembolik yaratılmasını sağlar.

Bu iki bloğun dışında birim tipine özel bloklar da bulunmaktadır. Service birim tipi için ilgili [Service] bloğu aşağıdaki anahtarları içerir :

Type : Servisin süreç tipini belirler. simple , forking , oneshot , dbus , notify ve idle değerlerini alabilir.

PIDFile : Servisin pid numarasını tutan dosya yolunu belirtir.

ExecStart : Servisi başlatmak için gerekli olan komutu tanımlar.

ExecStartPre : Servis başlamadan hemen önce çalıştırılacak komutları tanımlar.

ExecStartPost : Servis başladıktan hemen sonra çalıştırılacak komutları tanımlar

ExecStop : Servisi durdurmak için gerekli olan komutu tanımlar.

Restart : Servisin hangi durumlarda kapandıktan sonra tekrar başlayacağını belirtir. no , on-success, on-failure, on-abort ve always değerlerini alabilir.

**systemctl Komutu ve Birimlerin Yönetilmesi**

Sys-V-init yapısında servisleri başlatmak ve yapılandırmk için kullanılan service ve chkconfig komutları systemd yapısında da desteklenmekdir ve arka planda systemctl komutuna yönlendirilmiştir. Bu komut systemd yapısındaki birimler hakkında bilgi almak ve yönetmek için kullanılır

systemctl list-units : Aktif birimleri listeler

systemctl start birim : Birimleri başlatır

systemctl stop birim : Birimleri durdurur

systemctl reload birim : Birimlerin yapılandırmasını tekrar okumasını sağlar

systemctl restart birim : Birimler yeniden başlatır

systemctl isolate birim : Tek bir birimi başlatır diğer tüm birimleri sonlandırır.

systemctl enable birim : Birimin başlangıçta etkinleştirilmesini sağlar

systemctl disable birim : Birimin başlangıçta pasif olmasını sağlar

systemctl daemon-reload : systemd yapılandırma dosyalarında yapılan değişiklikleri algılamasını sağlar

systemctl rescue : Sistemin kurtarma moduna girmesi sağlar

systemctl poweroff : Sistemin kapanmasını sağlar

systemctl reboot : Sistemin yeniden başlatılmasın sağlar

systemctl switch-root : İşletim sisteminin kökü ve init'in değiştirilmesini sağlar

systemctl --help : Komutun kullanım seçeneklerin gösterir

[root@ila ~]# systemctl status rsyslog

rsyslog.service - System Logging Service

Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled)

Active: active (running) since Wed 2013-09-11 16:16:59 EEST; 2s ago

Main PID: 12740 (rsyslogd)

CGroup: name=systemd:/system/rsyslog.service

└─12740 /sbin/rsyslogd -n

Sep 11 16:16:59 ila.itu.edu.tr systemd[1]: Starting System Logging Service...

Sep 11 16:16:59 ila.itu.edu.tr systemd[1]: Started System Logging Service.

[root@ila ~]# systemctl disable rsyslog

rm '/etc/systemd/system/syslog.service'

rm '/etc/systemd/system/multi-user.target.wants/rsyslog.service'

[root@ila ~]# systemctl stop rsyslog

[root@ila ~]# systemctl start rsyslog

[root@ila ~]# systemctl enable rsyslog

ln -s '/usr/lib/systemd/system/rsyslog.service' '/etc/systemd/system/syslog.service'

ln -s '/usr/lib/systemd/system/rsyslog.service' '/etc/systemd/system/multi-user.target.wants/rsyslog.service'

**Mesaj Dosyaları**

/etc/issue - Login prompt’undan önce yazılacak yazı. Bu dosyada işletim sistemi, çekirdek ve mimari gibi bilgiler bulunur.

[root@ila~]# cat /etc/issue

Fedora Core release 4 (Stentz)

Kernel \r on an \m

/etc/motd - Kullanıcı sisteme giriş yaptığında yazılacak yazıdır (Message Of The Day). Genellikle ileriye yönelik bakım/onarım çalışmaları duyurma ya da kullanıcıları bilgilendirmeye yönelik yazılar bulunur.

root@ubuntu:~# cat /etc/motd

Linux ubuntu 2.6.32-24-generic-pae #43-Ubuntu SMP Thu Sep 16 15:30:27 UTC 2010 i686 GNU/Linux

Ubuntu 10.04.1 LTS

Welcome to Ubuntu!

\* Documentation: https://help.ubuntu.com/

System information as of Sun Sep 26 12:50:24 EEST 2010

System load: 0.0 Processes: 82

Usage of /: 10.9% of 8.01GB Users logged in: 1

Memory usage: 43% IP address for lo: 127.0.0.1

Swap usage: 0% IP address for eth0: 160.75.5.183

Graph this data and manage this system at <https://landscape.canonical.com/>

**Sistemin Kapatılması**

Linux bir sistemi kapatma veya yeniden başlatma işlemleri için farklı komutlar kullanılabilir. En sık kullanılan komut shutdown komutudur.

Sistemin hemen kapatmak için shutdown -h now komutu, yeniden başlatmak için ise shutdown –r now kullanılabilir.

Kapatma işlemi ileri bir zamanda yapılmak isteniyorsa, komuta +n parametresi verilerek n dakika sonra kapanması sağlanabilir ve bir açıklama girilebilir ya da kapatılmak istenilen tarih tam olarak yazılabilir.

[root@ila ~]# shutdown –r 17:50 “Sunucu bakim amaciyla yeniden baslatiliyor”

shutdown komutuna now dışında bir zaman parametresi verildiğinde, shutdown komutu /etc/nologin isimli dosyayı oluşturur. Bu dosya var oldukça root kullanıcısı hariç, hiç kimse sisteme oturum açamaz.

Başlatılan bir shutdown sürecini sonlandırmak için shutdown komutu -c anahtarı ile yürütülür.

shutdown komutu dışında sistemi kapatmak için poweroff; yeniden başlatmak içinde reboot komutları da kullanılabilir.

Sistemin kapanma zamanı geldiğinde, shutdown, init programına yeniden başlama için 6. çalışma seviyesine geçmesini, sistemin tamamen kapatılması için ise 0. çalışma seviyesine geçmesini söyler.

**Kapanış Adımları:**

init öldürebildiği tüm süreçleri öldürür.

Sistem dosyaları kilitlenir, kapanışa hazırlık yapılır.

Bellek tamponundaki veriler sync programı ile dosya sistemine yazılır.

/ haricindeki tüm dosya sistemleri unmount edilir.

/ dosya sistemi salt-okunur olarak tekrar bağlanır.

rc0.d/rc6.d komutları çekirdeğe reboot, halt ya da poweroff komutlarını kullanarak sistemi yeniden başlatmasını ya da durdurmasını/kapatmasını söyler.